Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 1, February 2024, pp. 263-272

DOI: https://doi.org/10.52436/1.jutif.2024.5.1.1792
p-ISSN: 2723-3863
e-ISSN: 2723-3871

PREDICTING FANTASY PREMIER LEAGUE POINTS USING CONVOLUTIONAL NEURAL NETWORK AND LONG SHORT TERM MEMORY

Anas Satria Lombu[*1], Irving Vitra Paputungan[2], Chandra Kusuma Dewa[3]

[1,2,3]Informatics, Faculty of Industrial Engineering, Universitas Islam Indonesia, Indonesia
Email: [1]20917037@students.uii.ac.id, [2]irving@uii.ac.id, [3]chandra.kusuma@uii.ac.id

(Article received: January 26, 2024; Revision: February 13, 2024; published: February 16, 2024)

Abstract

Fantasy Premier League is a fantasy sports-based game focused on football, particularly the English Premier League. each manager in this game is given the opportunity to build a virtual team for one season. A virtual team consists of various player positions that will earn FPL points based on their real-word performance. This research aims to implement a deep learning algorithm to predict FPL points generated by players based on their performance in the last 5 matches using a dataset collected from August 14, 2021, to May 21, 2022. The prediction model is designed using a Convolutional Neural Network algorithm consisting of one-dimensional Convolution layers, Max Pooling, and Dense layers. Additionally, a Long Short Term Memory algorithm with LSTM layers and Dense layers totaling 64 units is added as a comparison model to determine the best performing deep learning model in this study. In the first scenario, with a 70:30 data ratio, the average Mean Squared Error values obtained for 4 player positions using CNN are 0.0052 and 0.0027 for LSTM. Meanwhile, in the second scenario with an 80:20 data ratio, the evaluation results are 0.0027 for CNN and 0.0022 for LSTM. the model evaluation results indicate that the LSTM algorithm, utilizing three gates in the model architecture, is superior in recognizing historical data sequences compared to the CNN algorithm.

Keywords: *Convolutional Neural Network, Fantasy Fremier League, Long Short Term Memory.*

1. INTRODUCTION

Fantasy Premier League is a virtual-based football team management game that allows English Premier League (EPL) fans to become a team manager to assemble their dream team. At the end of the 2021/2022 season, it experienced an 11 percent increase in active users (known as manager) from the previous season, with approximately 9.1 million registered managers [1]. The combination of brilliant tactics from the coach and the high-level individual skills displayed by EPL players has become one of the key factors in the popularity of this game. FPL also provides easy access, as managers can play directly through the internet, both via the app and the website. The gaming ecosystem can be created privately (in your own league) or publicly, with 20 clubs and 38 matches during a season. In technical terms of the game, a manager will receive virtual money worth £100. They will select players to meet the dream team quota, considering the player price list for adjustment from the result of dynamically moving supply and demand among fellow managers. The dream team consists of 15 players (11 main players and 4 substitutes) [2]. Players are divided into 4 positions based on field area, with a minimum of 2 goalkeepers, 5 defenders, 5 midfielders, and 3 forwards in the player lineup. The selected players are expected to consistently deliver their best performance in every match to avoid being placed on the transfer list. Players on the transfer list will be evaluated alongside other players who are deemed capable of performing well in every match. Player performance aligns with a manager's final achievement in this game. Managers will receive FPL points on player performance in EPL (referred to as 'game weeks') that take place [3]. The captain of the team will receive double points, and extra bonus will be given to the player top three players in a match [4].

In recent years, the utilization of machine learning has been widely adopted in handling Fantasy Premier League (FPL) case studies. The applications of the Multi-Linear Regression algorithm have the capability to process varied performance statistical data (features), such as passing, duels, tackles, goals, yellow cards, and red cards [5]. The statistical data sourced through the FPL API has generated a total 2314 points and an average of 36 points per game week. This performance appears to be superior when compared to the betting odds data obtained from the football API, which only resulted in 1994 points and an average of 52 points for the season [6]. The dataset is one of the crucial factors in building the model. with other methods such as Support Vector Machine, Multi-Layer Perceptron, and Long Short Term Memory (LSTM) performing regression and classification based on the previous five matches to present a comparative analysis of these three different methods in predicting performance in the upcoming matches [7]. Similar to Lindberg's research, a

comparison was made with other machine learning algorithms such as Linear Regression, Decision Tree, and Random Forest [8]. Before diving into the modeling phase, feature importance is used for the selection of features that will be utilized. Columns containing categorical data are converted into numerical data to make them recognizable by the computer. This process is called encoding [9]. Support Vector Machine (SVM) is used to predict ELO ratings for two teams that are currently competing. SVM has the highest accuracy in predicting ELO rating, which are based on the actual performance and productivity of goals generated by the two teams in the match [10].

Classification models are also utilized to predict the probability of a team winning or not. The division of data into training and test sets has different scenarios, and some of the dataset division scenarios include 75:25, 80:20, 90:10 [11]. The scenarios that have been created will be implemented using KNN (K-Nearest Neighbors) and Naïve Bayes algorithms. The implementation of the hyperplane serves as the separating line between two different classes [12]. Even the CNN algorithm, which is typically used to address issues related to object recognition in images and video [13]–[15]. Modifications were made to enable time series forecasting of future FPL player points. The model utilizes a 1D convolutional layer architecture, which requires data transformation to fit the model input [16]. The workflow of the kernel is different, as it typically moves from left to right, but in this case, it starts sequentially from the previous matches and ends with the match to be predicted, moving from top to bottom.

The collection of statistical data related to player performance results in varying perceptions when determining which data managers should use to select players who will perform well in the upcoming matches. Traditionally, many managers have relied on their personal feelings and sentiments to address this issue [17]. Typically, CNN are associated with data in the form of images and videos, but they can be adapted and applied to time-related tasks as well [18]. Nevertheless, the implementation of the CNN algorithm is quite intriguing for further execution in this paper to predict the player performance in the next game week, considering both the last 5 and 10 previous matches. Additionally, two scenarios differentiate the amount of training and testing data used: 80:20 and 70:30.

The aim of this study is to compare deep learning models using Convolutional Neural Network and Long Short Term Memory algorithms to predict Fantasy Premier League points in the upcoming matches based on their performance in their last five matches as a timesteps. the application of 50 epochs is employed accross the LSTM, Conv1D, and dense layers, with each layer containing 64 units. The goal of the Lasso regression model is to conduct feature selection to decrease data dimensions. It is anticipating that the adoption of deep learning will aid managers in tackling data processing challenges and achieving the highest possible points in every Fantasy Premier League match.

2. RESEARCH METHODS

In this section, it explains the stages of work the beginning to the end of the research. The research workflow consists of five stages, starting from data collection, data preprocessing, data train, data test, model and evaluation. As shown in Figure 1.

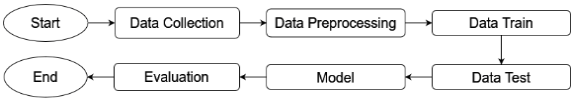

Figure 1. Proposed Methodology

2.1. Data Collection

Historical data is recorded in the system and is publicly available for research purposes. To facilitate the data, Anand has made Fantasy Premier League player performance statistics on his GitHub account, which can be access through the page https://github.com/vaastav/Fantasy-Premier-League/tree/master/data. The available data is from the 2016/2017 to 2023/2024 season. In this research, only the data from the 2021/2022 season is used because at the end of the season, players can move to different clubs, causing the registered players to change. first match of that season began on august 14, 2021, and the last match was played simultaneously on May 22, 2022. The collected data until the end of the season consists of 24.565 rows and 36 columns. Sample of data is presented in Table 1.

Table 1. Fantasy Premier League Dataset

name	position	xP	bps	Total_points
Aaron Ramsdale	GK	2.5	12	9
Aaron Creswell	DEF	2.1	11	1
Aaron Moy	MID	0.0	0	0
Aaron Connolly	FWD	0.5	-3	1

2.2. Data Preprocessing

The data provided in the data collection process does not guarantee that the data is clean. The data will be divided into four parts based on positions, namely goalkeeper, defender, midfielder, and forward. First, Missing values and duplicated check focus on player performance statistics and the number of matches played in a single season. Each player is expected to play 38 matches (home & away) in a season. However, some players have played fewer matches. This can be due to factors as injuries, non-participation in matches, or new players joining the team in the middle of the competition [19]. The author assumes that players with fewer than 30 matches will affect data quality and therefore removed. Meanwhile, players with more than 30 matches but less than or equal to 38 require data imputation to complete their records. The imputation

of a value of 0 is done for data completeness purposes [20]. The handling missing values can be accomplished by either removing or imputing them to ensure the data used is not biased [21].

Feature selection contributes to minimizing the impact of high dimensionality on the dataset by identifying the subset of features that can accurately describe the data. a new data basis can save computational time and improve accuracy [22]. Least Absolute Shrinkage and Selection Operator (LASSO) regression is one of the models capable of selecting the most relevant variabels from a variety of available variables. Dimensionality reduction is achieved by setting the value to 0 for irrelevant variabels, resulting in a simpler model [23].

The next step in data preprocessing is normalization, where each chosen feature exhibits varying value ranges in comparison to other features. These variations in ranges can impact the performance of the modeling process. The purpose of normalization is to standardize the data range to a consistent value scale. In this study, the value scale will confined within the 0 to 1 range, a technique commonly referred to as a Min-Max Scaler. According [24], the formula for Min-Max scaler is as follows (1).

$$X_{new} = \frac{X - min(X)}{max(X) - min(X)} \quad (1)$$

Where, X_{new} is the new value from the normalized result, X is old value, max(X) is maximum value in the dataset, min(X) is minimum value in the dataset.

The last stage of data preprocessing involves transformation, to prepare the dataset will be used as input for a model. Time data is very important in using time series forecasting models to present specific historical data, therefore, input data needs to adjusted for modeling in this research. The dataset will be transformed into an n x m matrix, n signifies gameweek (time), while m represents features or variabel [16].

2.3. Data Training

In the data collection phase, the dataset will be divided into two parts, and one of them is the training data. the performance of a model can be influenced by the quantity and quality of the training data it has [25]. Previously, the data has gone through a series of steps to ensure data quality of integrity. On the other hand, the amount of training data used can vary significantly. Machine learning research that deals with predicting basketball match outcomes often discusses the ratio of comparison between training data and testing data is 80:20 [26]. The data split ratio represented by Eetvelde is 70:30 [27]. Where the larger ratio represents training data, and the other ratio represents testing data. the goal of having a larger portion dedicated to training data is to allow the model to learn from more data, thereby creating an optimal prediction model in this research, both data splitting ratios are used.

2.4. Data Testing

One of the components of data splitting is the testing data [28]. as the name suggests, the benefit of testing data is easily recognizable, it plays a significant role in the evaluation phase. Testing data is used to assess the performance of the model built on new data. therefore, the model is expected to not only predict data that it has already encountered in the training data but also be able to recognize new data, ultimately resulting in an optimal model [29]. In practice, the amount of testing data is typically equal to or less than the training data. in this research, the testing data accounts for around 10% to 30% of the dataset. This proportion is commonly used to assess the model's generalization and predictive capabilities on unseen data.

2.5. Model

2.5.1. Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning method with the ability to directly extract features from input data, especially image data. in general, CNN uses a convolution process to apply a specific-sized convolution kernel (filter) to an image [15]. This allows the computer to acquire new information that represents the image through the process of multiplying different parts of the image with the convolution kernel. Real-world case studies that can be addressed using this algorithm include object detection, image recognition, and image classification [30]–[32].

In recent years, this algorithm has also been used for time series forecasting. Delano Ramdas attempted to implement time series forecasting using 1D convolution [16]. The tabular data used as input for the model will be transformed into an n x m matrix, where n represents gameweek (time) and m represents the statistical performance values of players (features). The prediction model will be generated based on the specified input window size. Figure 2 shows the movement of the kernel starting from the top and moving downward, following the size of the input window. 1D convolution has a kernel size of n x m, where m will have the same length as the features. The architecture concept of CNN like this is not significantly different from 2D convolution. The key distinction lies in the direction of kernel movement, which changes from left to right 2D convolution to top to bottom in 1D convolution.

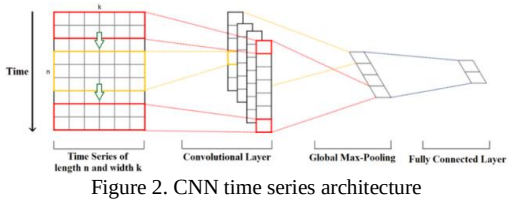

Figure 2. CNN time series architecture

2.5.2. Long Short Term Memory

Another prediction model used for comparison in this case study is Long-Short Term Memory (LSTM). time series forecasting is often associated with neural network mechanism, such as Recurrent Neural Network (RNN) and LSTM. the ability to retain previous information is a critical factor in handling time series forecasting. RNN has limitations in memory to store information over a long period (vanishing gradient problem). In detail, LSTM is an advancement if the RNN algorithm, capable of retaining information for longer durations, making it a valuable tool for time series forecasting [33]. The gated structured supports LSTM in retaining information and updating each memory cell. Figure 3 is an illustration of the LSTM model architecture.

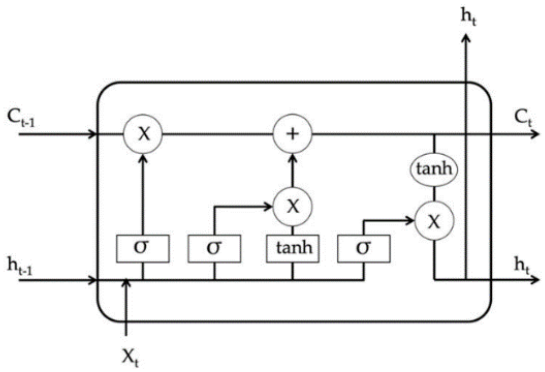

Figure 3. LSTM model architecture

The forget gate is the initial gate that aims to filter out information that should be discarded using a sigmoid function. The input to forget gate is obtained from the previous hidden state (h_{t-1}) and the input vector (x_t) results in an output in the form of a vector ranging from 0 to 1 due to the sigmoid function used in the forget gate [34]. The output vector with a value of 0 will be automatically dropped, while the vector with a value of 1 will be stored in memory. This is how the forget gate in LSTM helps in retaining or forgetting information from the previous hidden state based on its relevance.

After successfully storing information in memory, the next step is to determine the information that will be added to the cell state (C_t). The input gate plays a dual role in this process. Together with the sigmoid function, it determines the update value, while, in conjunction with the tanh function, it generates a vector that serves as the new candidate value. The results from the sigmoid and tanh functions are added together to update the cell state. The final result is determined by the output gate, which is obtained from the tanh function applied to the cell state, multiplied by the output of the forget gate. This combination of gates and functions is what makes LSTM capable of retaining or forgetting information and updating the cell state in a controlled manner.

2.5. Evaluation

The evaluation in this research is conducted using Mean Squared Error (MSE) to determine the performance of the time series prediction model. The MSE metric will calculate the square of the difference between each predicted value and the true value, then take the average of all the squared errors. MSE measurement is formulated given (2).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2 \quad (2)$$

Where, MSE is result of the mean squared error value, n is number of samples in the data, y represents the true values of sample, and $\hat{y}$ represents the predicted values of the sample.

3. RESULT

3.1. Data Collection

The collection of data accessible through Anand's Github repository has made data publicly available at no cost. The utilization of data is specifically centered on the 2021/2022 season. Throughout one season, a total 24.565 data points were acquired from various player positions. Figure 4 indicates that the highest data usage is observed in the midfielder position, amounting to 10.155, followed by defenders with 8.326, attackers with 3.275, and goalkeepers with 2.809.

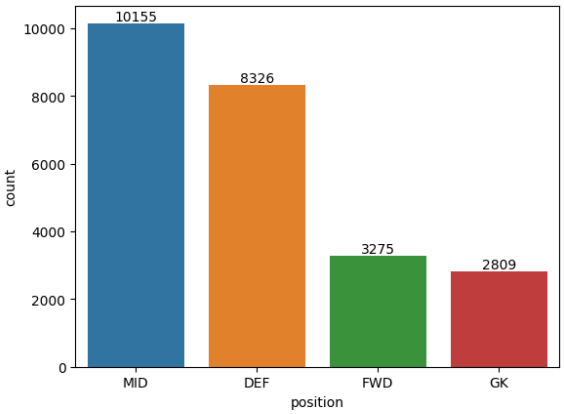

Figure 4. Data collection by position

3.2. Data Preprocessing

The processing of data occurs after successful data collection. At this stage, it is crucial to produce optimal results in a model. No missing values or data duplications were identified in the collected data. however, some players lack complete data, potentially impacting the time series model. Consequently, need to remove and imputation are performed to maintain data quality. The majority of Arsenal players have only played in 37 matches, indicating that 1 match went undetected by the system. Imputation data is implemented to ensure each player has an equal number of matches. During one season, a player names Ben Davies is registered

with two different teams, requiring the removal of data from his previous team. Figure 5 illustrates the quantity of data that will be utilizied in the subsequent process for each position.

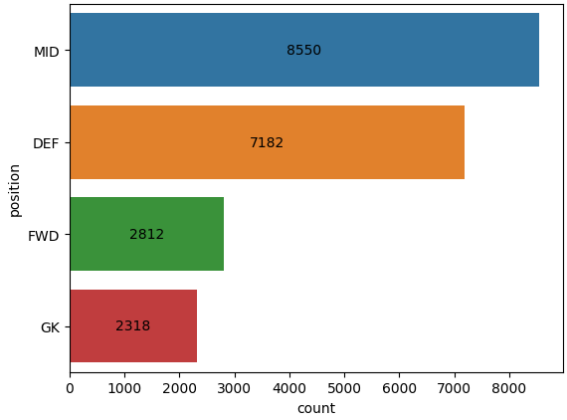

Figure 5. Data preprocessing by position

The data contains numerous predictor variables, not all of which will be utilized in the modeling process. Only variables with significant relationships to the target should be chosen. Feature selection aids in simplying the data dimensions, employing Lasso Regression in this study.

Lasso regression ulitizes a model to select relevant variables as output, with a minimum importance value set at 0.01 for variable selection. Variables that successfully undergo the feature selection process and consistenly appear in every player position include assists, bonus, ict_index, own_goals, yellow_cards and red_cards.

Goalkeeper exhibit a strong association with the variables penalties_saved, indicating a goalkeeper's success in making saves from penalty kicks. Conversely, in more attacking-dominant positions, penalties_missed can significantly impact performance, as failure results in point deductions for the player. Figure 6 displays the output for the goalkeeper position.

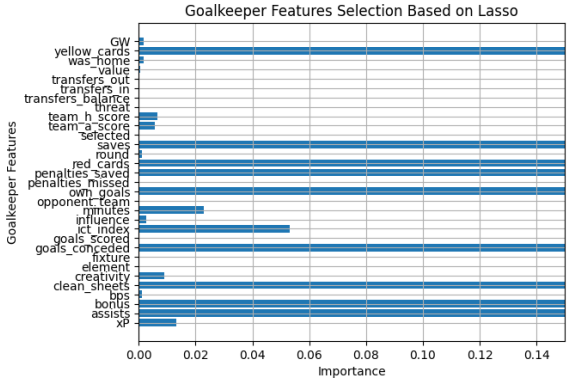

Figure 6. Goalkeeper feature selection

The significant variables generated by the lasso model in the defender position, include xP, assists, bonus, bps, clean_sheets, goals_conceded, goals_scored, minutes, own_goals, yellow_cards, and red_cards, as seen in Figure 7.

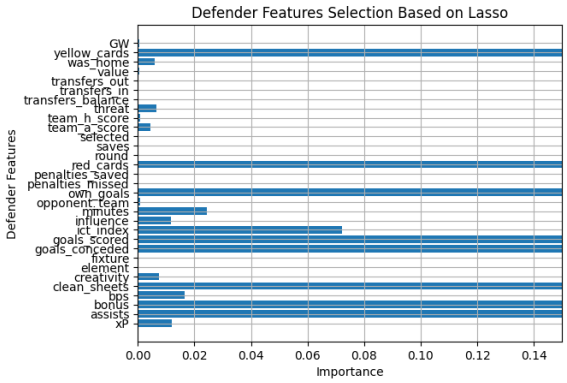

Figure 7. Defender feature selection

The similarity of selected variables with the defender position is not significantly different, own_golas can be changed with penalties_missed on midfielder. Figure 8 illustrates feature selection for the midfielder position.

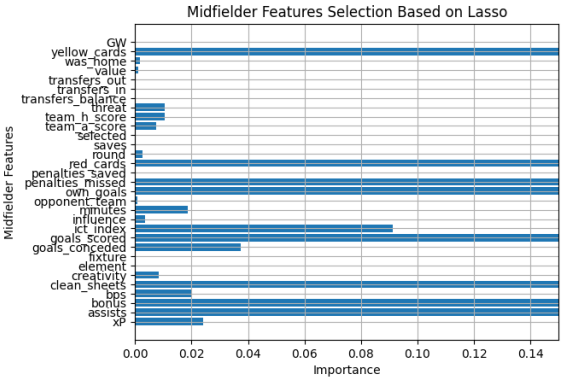

Figure 8. Midfielder feature selection

In the forward position, the number of selected variables is the lowest compared to other positions. Figure 9 illustrates the feature selection result using lasso regression model.

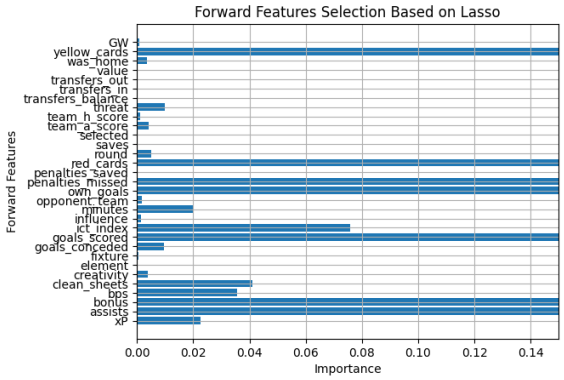

Figure 9. Forward feature selection

The result of the feature selection needs to be supplemented with additional features such as name, gw, and total_points. The benefits of adding these variabels include facilitating data transformation and serving as the research target.

The feature selection process results in several selected variables with the potential to have different value ranges from one variable to another. Significant differences in values can distrupt the model's ability to recognize patterns in the given data. to address this, a normalization of the value ranges on the same scales is applied to all selected numerical variables using Min-Max scaler. In this way, the values of the variables will be transformed into a range from 0 to 1. Table 2 shows the normalization results for the goalkeeper sample data.

Table 2. Normalization of goalkeeper data sample

assists	bonus	goals_conceded	ict_index	saves
0.0	0.0	0.285714	0.148148	0.1
0.0	0.0	0.285714	0.296296	0.3
0.0	0.0	0.714286	0.530864	0.5
0.0	0.0	0.0	0.0	0.0

The data has undergone several prior processes will then be used as input for a model. the data will be transformed according to the input format of the model, where in this study, a time series model is employed. The model recognizes patterns in the data based on time, where each match (gameweek) represents a unit of time. Meanwhile, features or variables are arranged from left to right. Player performance statistical data is segmented based on timesteps.

Figure 10 displays data transformation result of player data containing performance information based on their respective timesteps. Timesteps are set at 5, meaning data from the previous 5 matches is taken to predict the outcome of the next match.

```
array([[[0.13265306, 0.        , 0.        , ..., 0.        ,
         0.        , 0.        ],
        [0.08673469, 0.        , 0.        , ..., 0.        ,
         0.        , 0.        ],
        [0.08673469, 0.        , 0.        , ..., 0.        ,
         0.        , 0.        ],
        [0.31632653, 0.        , 1.        , ..., 0.        ,
         0.        , 0.        ],
        [0.29081633, 0.        , 0.        , ..., 0.        ,
         0.        , 0.        ]],

       [[0.19387755, 0.        , 0.        , ..., 0.        ,
         0.        , 0.        ],
        [0.12755102, 0.        , 0.        , ..., 0.        ,
         0.        , 0.        ],
        [0.08673469, 0.        , 0.        , ..., 0.        ,
         0.        , 0.        ],
        [0.14285714, 0.        , 0.        , ..., 0.        ,
         0.        , 0.        ],
        [0.1377551 , 0.        , 0.        , ..., 0.        ,
         0.        , 0.        ]],

       [[0.12244898, 0.        , 0.        , ..., 0.        ,
         0.        , 0.        ],
        [0.07653061, 0.        , 0.        , ..., 0.        ,
...
         0.        , 0.        ],
        [0.18367347, 0.        , 1.        , ..., 0.        ,
         0.        , 0.        ],
        [0.33163265, 0.        , 0.        , ..., 0.        ,
         0.        , 0.        ]]])
```

Figure 10. Result of data transformation.

3.3. Train and Test Data

In this study, will implement two scenarios for data partitioning. Intially, the split between training and testing data will be executed with a ratio 70:30. secondly, the data partitioning will be conducted with ann 80:20 ratio. In both scenarios, the number of training data is more dominant compared to the testing data. Table 3 shows the quantity of training and testing data for each position in the first scenarios.

Table 3. Train test scenario 1

position	train	test
GK	1596	722
DEF	5016	2166
MID	5966	2584
FWD	1938	874

In the second scenario, the number of data in the training set is larger than the number of training data in the first scenario. Table 4 outlines the quantity of data to be used in modeling based on position.

Table 4. Train test scenario 2

position	train	test
GK	1824	494
DEF	5738	1444
MID	6840	1710
FWD	2242	570

3.4. Model

Convolutional Neural Network and Long Short Term Memory model will be constructed to predict the total points in the next match based on the player's performance in several previous matches. This approach is known as time series forecasting, where the model will identify patterns based on the sequence of time.

Convolutional Neural Network model consists of a Conv1D layer with 64 units, a kernel size of 4, and ReLu (Rectified Linear Unit) activation function as the input of the model. Subsequently, the model will reduce the dimension of the data by selecting the maximum values obtained from the multiplication between the input model and filters, a process known as MaxPooling1D. The outcomes of all convolutions will be flattened using the Flatten Layer. The convolution results will then be passed to a dense layer with 64 units and ultimately end at the output layer. The architecture of the constructed CNN models is depicted in Figure 11.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 2, 64)             2880

 max_pooling1d (MaxPooling1  (None, 1, 64)             0
 D)

 flatten (Flatten)           (None, 64)                0

 dense (Dense)               (None, 64)                4160

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 7105 (27.75 KB)
Trainable params: 7105 (27.75 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

Figure 11. CNN model architecture

Another constructed model is the Long Short Term Memory, as illustrated in Figure 12. the preprocessing data, set as input model, is processed within the LSTM layer with 64 units, accompanied by the ReLu activation function. Before to the output layer, dense layer in LSTM achitecture has 64 units to combine with activation function as well.

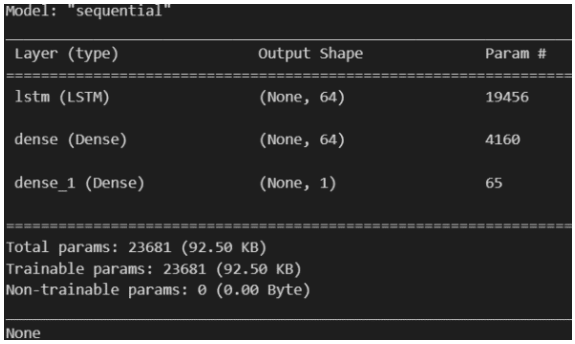

Figure 12. LSTM model architecture.

3.5. Evaluation

Mean Squared Error (MSE) serve as a metric to evaluate the model's performance when dealing with the FPL case study. Acccording [38], dataset is separated into 2 scenarios. In the first scenario, the data divided into 70:30 ratio between training and testing data. in the next scenario, data is distributed using an 80:20 ratio. This study represents four different models were developed by position to predict total point in the upcoming matches based time series data.

3.5.1. Comparison Model

LSTM model produces a smaller average value of 0.0027 compared to the CNN algorithm which is 0.0053. each model distinguished by position, the LSTM algorithm consistently outperforms CNN with optimal error values starting from the goalkeeper position to the forward position, namely 0.0081, 0.0008, 0.0002, and 0.0020 in Table 5. when compared to the CNN model, the error values in ascending order are 0.0098, 0.0035, 0.0012, and 0.0063. The difference is average values between the models appears to be significant, indicating that the LSTM performs very well in the first scenario.

Table 5. Comparison of model evaluation in the first scenario

position	CNN	LSTM
GK	0.0098	0.0081
DEF	0.0035	0.0008
MID	0.0012	0.0002
FWD	0.0063	0.0020
Average	0.0052	0.0027

In the second scenario, LSTM is not very dominant. CNN model can outperform the LSTM model in goalkeeper position, with an evaluation value comparison of 0.0040 for CNN and 0.0060 for LSTM. However, this advantage does not last long because the CNN model cannot maintain the minimum error values in the other three models in defender, midfielder, and forward. Whereas the LSTM model produces smaller error values, thereby improving model performance. The best model evaluation results in the other three positions are, in ascending order 0.0003 for defender and midfielder, and 0.0022 for forward. The LSTM model excels in 3 positions while CNN only achieves one best result in the goalkeeper position. The average values produced between these two algorithms are 0.0027 for the CNN and 0.0022 for the LSTM model as seen in Table 6.

Table 6. Comparison of model evaluation in the second scenario

position	CNN	LSTM
GK	0.0040	0.0060
DEF	0.0020	0.0003
MID	0.0020	0.0003
FWD	0.0030	0.0022
Average	0.0027	0.0022

3.5.2. Predicted Value on The Test Data

Implementating the best model is LSTM in second scenario on the test data to assess the model's performance with new data. the x-axis on the graph will represents the number of players in the test dataset, and the y-axis represents the total points. Predicted values will be presented in a graph starting from the goalkeeper's poisition.

In the model prediction graph as shown in Figure 13, it is observed that two out of three actual values that are relatively high, specially 8, the model has not yet been able to predict close to the actual value. The predicted model values only fall within the range of 4-5. However, in some other predictions, the predicted values are higher than the actual ones.

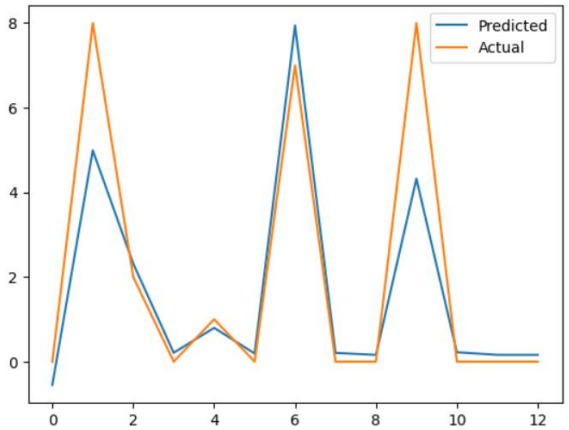

Figure 13. Goalkeeper predicted values on the test data

The prediction results of the defender model applied to the test data are illustrated in Figure 14. The blue line, representing the model's predictions, shows closer proximity to the orange line, which represents the actual values, compared to the goalkeeper prediction model.

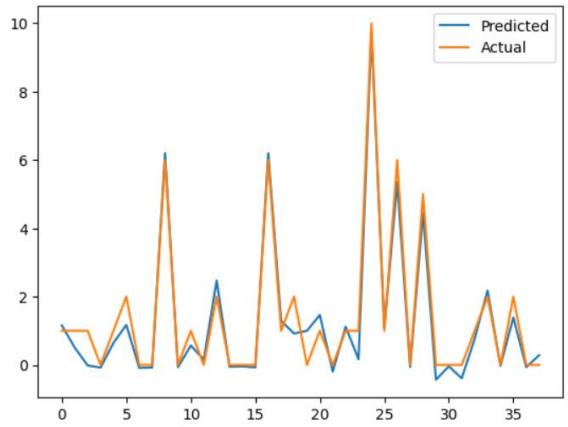

Figure 14. Defender predicted values on the test data

In Figure 15 shows the predictions results for total points applied to midfielders, totalling 45 players. Several predicted value align precisely with the actual value along the same line.

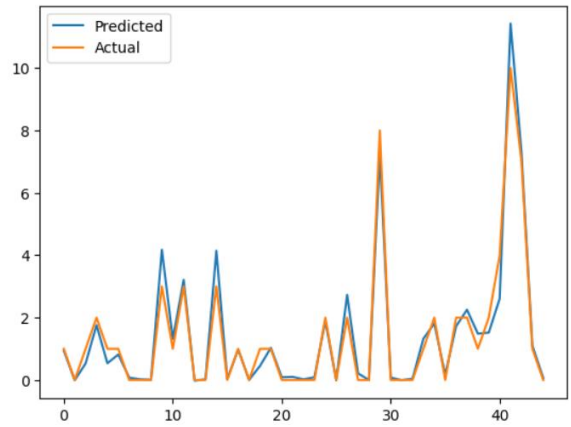

Figure 15. Midfielder predicted values on the test data

The model prediction results for 15 forward players in the test data are depicted in Figure 16. The predicted values tend to be higher than the actual values in the range of 0 to 2. This significant difference contributes to larger model evaluation values as well. However, for higher actual values, the model's predicted values slightly approach the actual values.

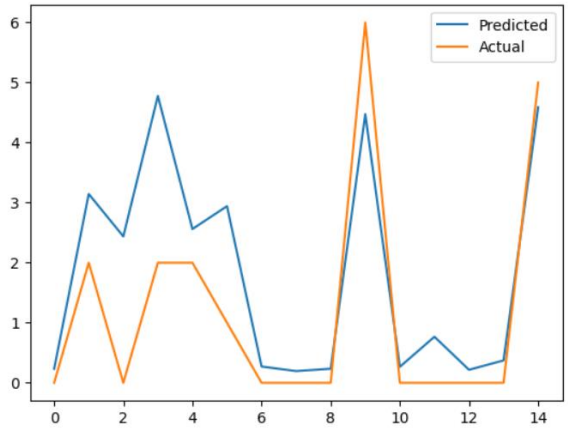

Figure 16. Forward predicted values on the test data.

4. DISCUSSION

The objectivity in fantasy premier league games focuses on the total point productivity of a virtual team designed by a manager. Points are generated through the real performance evaluated in every match. Managers can utilize various aspects in assessing player performance to select players to their virtual team to achieve targets in this game.

Several previous studies have utilize player statistics data from fantasy premier league to build a predictive model. various modeling algorithms that have been employed include Support Vector Machine, Random Forest, Naive Bayes, K-Nearest Neighbors, Decision Tree, Convolutional Neural Network and Long Short Term Memory. The selection of FPL seasons utilizing only single season aims to avoid bias and the use of time series models in deep learning is conducted to design prediction models based on historical data.

In this study, the Convolutional Neural Network and Long Short Term Memory algorithms will go through several stages such as data collection, data preprocessing, data train, data test, modeling, and model evaluation to compare the lowest error values of the difference algorithms using two scenarios data splitting scenarios.

Data quality significantly influences the outcome of a modeling process. Raw data needs to be processed before being used as input for a model. the data preprocessing stage has consumed a considerable amount of time in this study. The dataset obtained from data collection has a large number of variables. Feature selection is generally performed by examining the correlation between variables. A different feature selection approach to reduce unimportant variables is conducted using a regression model called Lasso. Consistent variables produced by Lasso include assist, bonus, ict_index, own_goals, yellow_cards, and red_cards.

At the end of the process, the best model from the two scenarios is obtained by calculating the average value from four FPL player positions. The best MSE metric used to calculate the squared difference between predicted and actual values obtained a values of 0.0022 for LSTM. Meanwhile, the CNN algorithm has a metric value of 0.0027. the utilization of the second dataset splitting scenario is better than the first scenario with a larger amount of training data, enabling the model to produce lower error values.

5. CONCLUSION

Implementating Long Short Term Memory algorithm on the dataset collected from August, 14 2021 to May, 21 2022 to predict the total points in the 6th gameweek using data from the previous 5 matches resulted in an MSE value of 0.0022. LSTM algorithm produced a lower error value than CNN through the data splitting scenario of 80% data

training and 20% data test. In another data splitting scenario, the LSTM layer and a dense layer with 64 units in the model still dominate the average error values generated from the four player positions. The MSE metrics generated based on player positions, sorted from goalkeeper to forward, are 0.0060, 0.0003, 0.0003 and 0.0022 respectively. Testing the prediction model against test data shows that the prediction line closely approximates the actual line.

The research hopes to implement to combinations of models such as CNN-LSTM in the future, with the ability to reduce and recognize suitable data sequences in line with the objectivity of this study. Hyperparameter tuning can be conducted to find the best parameters in a model to improve its performance and achieve optimal results.

ACKNOWLEDGE

Special thanks to Ministry of Education, Culture, Research, and Technology who has support for this research.

REFERENCES

[1] R. Masters, "Annual Report 2021/22," 2022. [Online]. Available: https://www.premierleague.com/about/publications.

[2] A. T. Nugroho, "Gamifikasi, Pemasaran di Era Digital: Studi pada Pengguna Game Fantasy Premier League di Indonesia," *J. Ris. Komun.*, vol. 4, no. 2, pp. 261–274, 2021, doi: 10.38194/jurkom.v4i2.376.

[3] M. M. Khamsan and R. Maskat, "Handling Highly Imbalanced Output Class Label: a Case Study on Fantasy Premier League (Fpl) Virtual Player Price Changes Prediction Using Machine Learning," *Malaysian J. Comput.*, vol. 4, no. 2, pp. 304–316, 2019, [Online]. Available: https://www.calculator.net/standard-deviation-calculator.html.

[4] J. D. O'Brien, J. P. Gleeson, and D. J. P. O'Sullivan, "Identification of skill in an online game: The case of Fantasy Premier League," *PLoS One*, vol. 16, no. 3, p. e0246698, Mar. 2021, [Online]. Available: https://doi.org/10.1371/journal.pone.0246698.

[5] M. A. Al-Asadi and S. Tasdemir, "Predict the Value of Football Players Using FIFA Video Game Data and Machine Learning Techniques," *IEEE Access*, vol. 10, pp. 22631–22645, 2022, doi: 10.1109/ACCESS.2022.3154767.

[6] N. Bonello, J. Beel, S. Lawless, and J. Debattista, "Multi-stream data analytics for enhanced performance prediction in fantasy football," *CEUR Workshop Proc.*, vol. 2563, pp. 284–292, 2019.

[7] A. Lindberg and D. Söderberg, "Comparison of Machine Learning Approaches Applied to Predicting Football Players Performance," Chalmers University of Technology, 2020.

[8] M. Bangdiwala, R. Choudhari, A. Hegde, and A. Salunke, "Using ML Models to Predict Points in Fantasy Premier League," *2022 2nd Asian Conf. Innov. Technol. ASIANCON 2022*, pp. 1–6, 2022, doi: 10.1109/ASIANCON55314.2022.9909447.

[9] R. Baboota and H. Kaur, "Predictive analysis and modelling football results using machine learning approach for English Premier League," *Int. J. Forecast.*, vol. 35, no. 2, pp. 741–755, 2019, doi: 10.1016/j.ijforecast.2018.01.003.

[10] C. Herbinet, "Predicting Football Results Using Machine Learning Techniques," *2011 Proc. 34th Int. Conv. MIPRO*, vol. 48, pp. 1623–1627, 2018, [Online]. Available: https://www.imperial.ac.uk/media/imperial-college/faculty-of-engineering/computing/public/1718-ug-projects/Corentin-Herbinet-Using-Machine-Learning-techniques-to-predict-the-outcome-of-profressional-football-matches.pdf.

[11] D. . Prabowo, "Prediksi hasil pertandingan sepakbola english premier league dengan menggunakan algoritma k-nearest neighbors dan naive bayes classifer," Universitas Islam Indonesia, 2020.

[12] F. Melky, S. Sendari, and I. A. Elbaith, "Optimization of Heavy Point Position Measurement on Vehicles Using Support Vector Machine," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 9, no. 3, pp. 673–683, 2023, doi: 10.26555/jiteki.v9i3.26261.

[13] S. An, M. Lee, S. Park, H. Yang, and J. So, "An Ensemble of Simple Convolutional Neural Network Models for MNIST Digit Recognition," 2020, [Online]. Available: http://arxiv.org/abs/2008.10400.

[14] D. Bhatt *et al.*, "Cnn variants for computer vision: History, architecture, application, challenges and future scope," *Electron.*, vol. 10, no. 20, pp. 1–28, 2021, doi: 10.3390/electronics10202470.

[15] S. Ilahiyah and A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network," *JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones.*, vol. 3, no. 2, pp. 49–56, 2018.

[16] D. Ramdas, "Using Convolution Neural Networks to Predict the Performance of

Footballers in the Fantasy Premier League," University of South Africa, 2022.

[17] S. Bhatt, K. Chen, V. L. Shalin, A. P. Sheth, and B. Minnery, "Who should be the captain this week? leveraging inferred diversity-enhanced crowd wisdom for a fantasy premier league captain prediction," *Proc. 13th Int. Conf. Web Soc. Media, ICWSM 2019*, no. Icwsm, pp. 103–113, 2019, doi: 10.1609/icwsm.v13i01.3213.

[18] N. J. L. Marfu'ah and A. Kurniawardhani, "Comparison of CNN and SVM for Ship Detection in Satellite Imagery," *Automata*, vol. 1, no. 1, 2020, [Online]. Available: https://journal.uii.ac.id/AUTOMATA/article/view/13973.

[19] P. Pokharel, A. Timalsina, S. Panday, and B. Acharya, "Proceedings of 12 th IOE Graduate Conference Fantasy Premier League-Performance Prediction," vol. 8914, pp. 1862–1869, 2022.

[20] A. E. Karrar, "The Effect of Using Data Pre-Processing by Imputations in Handling Missing Values," *Indones. J. Electr. Eng. Informatics*, vol. 10, no. 2, pp. 375–384, 2022, doi: 10.52549/ijeei.v10i2.3730.

[21] Y. Zhang and P. J. Thorburn, "Handling missing data in near real-time environmental monitoring: A system and a review of selected methods," *Futur. Gener. Comput. Syst.*, vol. 128, pp. 63–72, 2022, doi: 10.1016/j.future.2021.09.033.

[22] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1060–1073, 2022, doi: 10.1016/j.jksuci.2019.06.012.

[23] A. Iparragirre, T. Lumley, I. Barrio, and I. Arostegui, "Variable selection with LASSO regression for complex survey data," *Stat*, vol. 12, no. 1, 2023, doi: 10.1002/sta4.578.

[24] H. Henderi, "Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer," *IJIIS Int. J. Informatics Inf. Syst.*, vol. 4, no. 1, pp. 13–20, 2021, doi: 10.47738/ijiis.v4i1.73.

[25] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?," *Adv. Neural Inf. Process. Syst.*, vol. 35, no. NeurIPS, 2022.

[26] R. P. Alonso and M. B. Babac, "Machine learning approach to predicting a basketball game outcome," *Int. J. Data Sci.*, vol. 7, no. 1, p. 60, 2022, doi: 10.1504/ijds.2022.124356.

[27] H. Van Eetvelde, L. D. Mendonça, C. Ley, R. Seil, and T. Tischer, "Machine learning methods in sport injury prediction and prevention: a systematic review," *J. Exp. Orthop.*, vol. 8, no. 1, 2021, doi: 10.1186/s40634-021-00346-x.

[28] D. Mualfah, W. Fadila, and R. Firdaus, "Teknik SMOTE untuk Mengatasi Imbalance Data pada Deteksi Penyakit Stroke Menggunakan Algoritma Random Forest," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 3, no. 2, pp. 107–113, 2022, doi: 10.37859/coscitech.v3i2.3912.

[29] I. Walsh *et al.*, "DOME: recommendations for supervised machine learning validation in biology," *Nat. Methods*, vol. 18, no. 10, pp. 1122–1127, 2021, doi: 10.1038/s41592-021-01205-4.

[30] Q. Li, Y. Chen, and Y. Zeng, "Transformer with Transfer CNN for Remote-Sensing-Image Object Detection," *Remote Sens.*, vol. 14, no. 4, 2022, doi: 10.3390/rs14040984.

[31] E. Oktafanda, "Klasifikasi Citra Kualitas Bibit dalam Meningkatkan Produksi Kelapa Sawit Menggunakan Metode Convolutional Neural Network (CNN)," *J. Inform. Ekon. Bisnis*, vol. 4, no. 3, pp. 72–77, 2022, doi: 10.37034/infeb.v4i3.143.

[32] F. Isdaryani, N. Cholis Basjaruddin, and A. Lugina, "Masked Face Recognition and Temperature Monitoring Systems for Airplane Passenger Using Sensor Fusion," *J. Nas. Tek. Elektro dan Teknol. Inf.* |, vol. 11, no. 2, pp. 140–147, 2022.

[33] S. Mahjoub, L. Chrifi-Alaoui, B. Marhic, and L. Delahoche, "Predicting Energy Consumption Using LSTM, Multi-Layer GRU and Drop-GRU Neural Networks," *Sensors*, vol. 22, no. 11, pp. 1–20, 2022, doi: 10.3390/s22114062.

[34] W. T. Handoko and A. N. Handayani, "Forecasting Solar Irradiation on Solar Tubes Using the LSTM Method and Exponential Smoothing," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 9, no. 3, pp. 649–660, 2023, doi: 10.26555/jiteki.v9i3.26395.