Master Thesis

Predicting Young Soccer Players Peak Potential with Optimal Age

AbdulRehman Tahir

March 2018

Predicting Young Soccer Players Peak Potential with Optimal Age

Master Thesis

By

Abdul Rehman Tahir

1770314

MSc Computer Science
Data Science and Smart Services

Enschede, March 2018

Prof. Dr. Ir. Raymond Veldhuis
Dr. Ir. L.J. Spreeuwers
Dr. Jan Van Haren

UNIVERSITY OF TWENTE.

Acknowledgments

I am grateful to many people for their contribution, support, and confidence on myself, which made my masters journey rousing and an experience in itself. Notably, I would like to express heartfelt gratitude to my advisors Prof. Dr. Raymond Veldhuis, and Dr. Luuk Spreewers for their continuous guidance and evaluative feedback throughout the thesis. They always welcomed me whenever I had a question about my research. Their support and trust not just educate me to have a deep understanding of applied Machine Learning but it also encouraged me to develop a research attitude. These are the best mentors I have ever worked with.

I pay special thanks to my friends Amir Khan, Aqib Saeed and Ali Imam for all their support and motivation.

Finally, I would like to express my deepest and very sincere gratitude to my parents and siblings, for providing me unflagging support, encouragement, and unfailing prays at every step. This achievement would not have been possible without them. Thank you.

Abstract

Potential is the individual ability to perform. To measure the potential, experts systematically use many intelligence, ability, and competence based tests and makes a composite score from their results, which is considered to be an individual Potential. Most of the psychologists agree on the fact that potential increases and decreases with age. In this research, we developed an algorithm that can be used to predict the peak potential of young soccer players with optimal age. We used different machine learning techniques from traditional methods to deep learning methods to develop an algorithm that can predict the peak potential of young soccer players using their playing data between the age of 15 till 19. We used Lasso regression and FeedForward neural networks as our baseline models. We considered this problem as a time-series forecasting problem or sequence prediction problem. Our proposed model is a variant of recurrent neural networks– LSTMs. We have found that LSTMs outperformed baseline models and performed with zero prediction error on the test set when used with player-specific models.

Contents

List of Figures

List of Tables

List of Acronyms

FFN	Feed Forward Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
SGD	Stochastic Gradient Descent
SRNN	Simple Recurrent Neural Network
WFV	Walk Forward Validation
ReLU	Rectified Linear Unit
CV	Cross-Validation
LOOCV	Leave One Out Cross Validation
Pid	Player ID
DS	Defensive Skill
OS	Offensive Skill
SSN	SciSkill Now
RTCE	Resistance
OSTM	Offensive share player this match
Mid	Match ID
MD	Match Date
MP	Minutes Played
FPid	Field Position ID
Tid	Team ID
Cid	Competition ID
SS	SciSkill Index
OSP	Offensive Share Player
DNN	Deep Neural Network
RFE	Recursive Feature Elimination
MPPE	Minimum potential prediction error

CHAPTER 1

Introduction

Estimating or predicting the performance of players and results of matches related to different sports is now a part of most team related sports. Because of this, data-driven analysis is widely used in sports analysis. Soccer is one of the most popular sports in the world. In sports, most of the analytics is related to the players' future performance, which helps the management in team selection. Some of the analysis are based on the physical characteristics (height, matureness, etc.) and other based on skills, For example, soccer skills include (dribbling, offensive skills, defensive skill, etc.). Predicting future performance of soccer players is paramount, as it is directly related to the coaches and managers to make right decisions, which leads to team formation and help in players trading for leagues.

From a long time in past since 1920, Sports and psychology are interlinked in the field of research. Mostly sports psychology is used to determine the player performance related to mental approach. It is also a part of studies that are related to talent identification [24]. From the psychology point of view, many studies have been done in finding the relation between personality traits and sports behavior.

Almost from last two decades, a study under research is Perceptual Cognitive Skills where the focus remains on how quickly and efficiently players respond in the complex situations; these studies are mostly done in the ball sports [24]. In team sports, like soccer or football an intelligent team player is considered to be the one who performs well, but besides that, he should possess other important features such as good memory, multi-tasking, etc. Intelligent players should have an adaptable personality with the situations. Most of such abilities are part of game intelligence in sports [24]. In neuropsychology, they are called Executive Functions [24]. Executive functions are more commonly called Cognitive Process. The cognitive process stimulates the thought process and action whenever there is a non-routine situation occurs. Examples include problem-solving, multitasking and many more.

The development of cognitive process takes place throughout the childhood till adolescence. According, to most of the researchers it is until the age of 19 [24], but others do argue and relate it to practicing and hard work. Cognitive functions have a really important role in sports like soccer where the player has to decide in every new moment under time constraints. They must have to take quick actions and plan decisions according to the situation [24]. Cognitive functions, intelligence abilities, psychological factors, physical capacities together come under one umbrella, and it is Potential. Potential is humans' ability. Potential depends on many factors. To measure human potential, psychologists and experts use person-oriented approach also presented by (Bergman and Magnusson, 1997; Bergman et al., 2003). In person-oriented approach, we consider every single individual and make a number of measurements related to different traits and then similar individuals grouped for further analysis [42].

In the next sections, we will discuss briefly potential and how it gets measured especially in soccer players.

1.1 Potential

The term potential is widely used in different fields from physics to the social sciences. In humans, we relate potential to one's ability, and as mentioned earlier it depends on many factors like intelligence factors, psychological factors, cognitive functions, etc. One of the sports psychologist and mental training expert Dr. Patrick Cohn defined the athlete potential, as "It's *your capacity to perform at the uppermost range of your ability."* He further said that potential is something or its ability which individual posses but they have yet to achieve. One of the psychologists explained the potential as [32]

Performance = Potential – interference

Where interference includes coaching, exercise, practice, etc. potential is ability and performance is how effectively someone uses its abilities.

Following model shows the dependence of potential on different factors

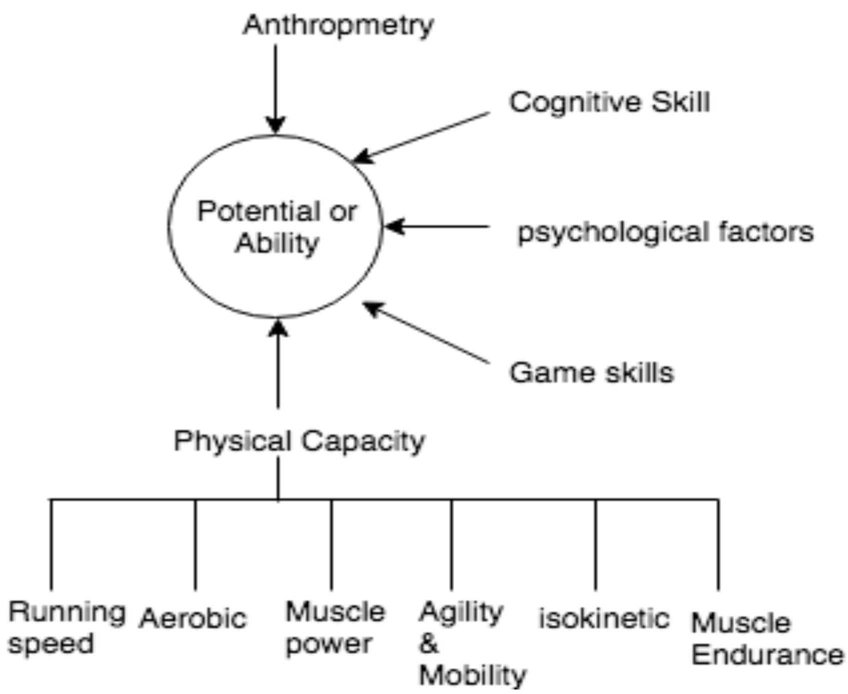

Fig 1.1: Model showing the dependence of players potential on different factors[46].

We are not going into detail of each factor; we are just defining each of them.

Anthropometry: It means scientific study of human body measurements (height, weight, organs, etc.) and proportions.

Game Skills: In this research, we are dealing with soccer players. So we are going to mention some of the soccer skills which include 1) shooting under attacking skills, 2) defensive skills, 3) resistance shown by each player, 4) offensive skills, 5) Each player offensive share, 6) Per match player's offensive, etc. these are all the skills which vary from player to player.

Psychological Factors: There are many psychological factors, which involves pressure, confidence, motivation, anxiety, mental preparation and many more [27]. In this research, we are not considering the effect of these factors on potential.

Cognitive skills: These are the skills, which relates how a player responds to the situation. It involves how a player tackles or perceives a certain situation. It involves decision-making and multitasking tasks as well. Influence of these skills on potential is not considered in this study.

Physical Capacity: It involves player running speed (sprint), muscle power, body build, and agility– it's the capability to response the stimulus that involves rapid whole body movement. In this research, we are not considering the effect of these factors on potential.

1.2 Measuring Potential

Different methods and tests would carry out to measure potential. Some of them are intelligence and ability tests. Both of these tests are domain specific. Few other tests used for measuring potential include competency and development assessment [41].

Measurement of Potential in Soccer Players

Measuring player potential is an extensive process, which involves a battery of tests and observations. The widely used tests for measuring the potential of soccer players include judgment of coaches on five basic skills: passing, catching, kicking, running, and defense. Other tests which are used for measuring soccer player potential includes coaches rating, personal history index –considers players age, weight, height, etc., physical index – considers players ten-yard start, fifty- yard dash, pull-ups and several other. Psychological index –it includes several factors like intelligence, assertiveness, and self-sufficiency [38]. All of these and many other tests are carried out in measuring player potential. Experts systematically, use all the test results and assign a composite score, which is a player potential. The scale of this score mostly starts from zero and goes upwards and gets cutoff under 100 or 200 in most of the FIFA rankings. Some use the scale of 0-100 and others use it from 0-200. Players' potential assessment is a continuous process, which mostly carried out on a yearly basis and also depends on when it is required.

So far we have explained the general idea related to potential, factors affecting it and how to measure it. All these tests and assessments are time-consuming and also require many other resources like experts rating and opinion, etc. In this research, we are interested in using machine-learning approaches to predict player future potential based on their previous year's data. By doing that we can make effective decisions which are mostly related to players selection and players trading.

Most of the previous work or early researches remained focused on the performance of players, match outcomes, player performances in coming matches using machine learning approaches but when it comes to the potential prediction of players especially in soccer sport it does not get that much attention. Moreover, significant work has done in the potential prediction of players related to sports like Basketball. Potential is players' ability to do something while performance is how he performs in that specific match using those abilities.

1.3 Statement of the Problem

The research question is "How precisely we can predict the peak potential of young soccer players with optimal age, using machine learning techniques."

1.4 Overview

As an overview, we are interested in developing a model that can predict the peak potential of young soccer players' with the help of machine learning methods. To train and test the model, we used those specific players from the dataset whose potential is already measured from the age of 15 till 26. By using those players' data, we have ground truth-values of potential at all ages of a specific player. The data we have considered for this research has 16 features for each player; all of the features have a numerical value. We started off with the traditional machine learning approaches where we have employed Ridge/Lasso models and Decision trees. Later, we employed feedforward neural networks with multiple configurations. We considered Feedforward neural network and Lasso regression as our baseline models. We applied two different approaches 1) Multiplayer model approach and 2) Player specific approach. We used Mean-squared-error (MSE) as an evaluation metrics as it is the most widely used evaluation metrics for regression problems. We considered this problem as a sequence prediction problem or time-series forecasting problem and proposed the use of deep learning method to solve this problem.

The precise solution to this problem could help many soccer clubs. Especially, small soccer clubs can make huge profits by having information about player future potential, small clubs can recruit those young players and can make huge profits from their trading when they reach their peak. The trading of players between clubs is called Sports Trade and players' are a primary asset in it.

Our main contribution involves the use of deep learning technique– especially a variant of the recurrent neural network (LSTMs) with only an auto-regressed target as a feature for the peak potential prediction of soccer players with optimal age.

Rest of the thesis is organized as follows: Chapter 2 describes the background study and related work. Chapter 3 contains all about the data and pre-processing applied. Chapter 4 is related to the model architectures and general explanation related to model working. Chapter 5 depicts the results and discussion of results, obtained from the performed experiments. Chapter 6 takes the thesis towards a conclusion and Future Work.

CHAPTER 2

Background Study and Related Work

In this section, we begin with the background study and related work specifically where machine-learning methods are used for predicting the players potential (especially in soccer players and some of the relevant work from other sports) and at what age most of the players reach their peak? And what are the important variables considered for potential prediction?

After that, we mentioned some of the related work regarding deep learning and the growing interest of its implication in different fields.

2.1 Literature Review

Brefeld et al. (2016) [4] presented work on soccer player's performance in the upcoming match. They used five seasons data of German football league named Bundesliga. They used support vector regression and multitask ridge regression for the performance prediction. For the relevant feature identification, they employed recursive feature elimination technique. They proposed player specific models and general models. They concluded that SVR models are performing better than ridge regression, however; SVR-player specific models suffer from sparse data if we don't use a window size of 8 matches. Player specific models with multitask ridge regression and multitask support vector regression outperformed the traditional or single task SVR and ridge regression. They used as many as 20 features as predictors for the prediction of next match performance.

Malina et al. (2007) [21] performed an analysis of soccer skills prediction on 69 young players mainly on passing, shooting and dribbling skills using multiple linear regression. Their study focused on quantifying skill in soccer based on age, experience, growth, and functional capacities. They performed six soccer related tests (ball control with the body, ball control with head, passing, shooting accuracy, dribbling with pass and dribbling speed) to quantify the skill and assign a composite score to each player. Then they used the player characteristics like age, training, height, body mass, etc. to find the relative contribution of each characteristic in composite skill. They found that maturity status was connected with higher composite skill. It also mentioned that composite skill score (which belongs Game skills–part of potential) was not well explained by these predictors.

Post et al. (2009) [39] proposed a detailed analysis of tactical skills and their improvement amid 191 youth soccer players from ages of 14 through 18 from Dutch clubs. All players get self-assessed using Tactical Skills Inventory for Sports Test where players from three different field positions (attackers or forward, midfielders, defenders) are informed to rate themselves on a 6 point Liker scale; each one has to compare themselves with the elite soccer player in the same age category. The multilevel regression model is used to carry out this study. The study is used to analyze how tactical skills get changed with age in each field position. The result shows that tactical skills get improve with age more in forwards or attacker as compared to midfielders and defenders.

A research work presented by Kevin Wheeler [40], where he tried to predict the NBA player performance using SVM, Naïve Bayes, and Linear regression. The results mentioned after the research shed light upon the inability to predict the player's performance accurately using these methods and mentioned some of the basic challenges.

Sharda et al. (2009) [2] propounded research related to performance prediction of cricketer's using neural networks. They trained the model for each player where for training any specific player they used its historical data. The data they have considered is from 1985 until 2007. After training the model, they used it for the near term player performance prediction. They proposed player selections in their national teams for World cup 2007 based on the results of their model. Model evaluated using the actual performance of the players in the World cup. The results showed that neural networks performed outstandingly in decision support mechanism for team selection. In some of the experiments, neural networks performed with an accuracy of 87%.

Seife Dendir (2016) [3] presented an analysis what the optimal age or peak age of soccer players is. The data they considered is from the top four European soccer leagues and is come from WhoScored.com. The analysis also takes into consideration the players position in the field. The analysis had been made using simple age distribution and bivariate approaches (simple plotting of data). The results showed that forward reach their peaks or peak potential at the age of 25 while defenders peak potential or optimal age is 27 and mid-fielders lie in between 25-27age band. It is also mentioned that peak age relies on the potential or ability.

A detailed work presented by Visscher,C et al. (2009) [22] where they used the Tactical Skills Inventory for Sports for youth players to find out which players will be professionals and which will be armature in their future. The study involved 105 top youth soccer players from an age group of 17-18. A logistic regression technique was employed to single out the tactical skill that leads the player to professional performance in adulthood. The results found that if a player at the age of 17-18 performs well in positioning and deciding tactical skill, then the player has an 80% chance to become a professional in adulthood.

Yi Li et al. (2014) [1] presented an analysis of the performance prediction of NBA players. They have used the entire history data. They proposed a new metric for the evaluation of NBA player performance, which based on factorial analysis and currently existing player performance metrics. Training set created by using clustering algorithm based on player's age. To predict performance for a specific player, all the rest of the players in that cluster would be considered as a training set. For prediction, they used SVM as a learning algorithm. They presented the new evaluation metric is efficient.

2.2 Limitations of Existing Approaches

Most of the previous work done remained focus on the match outcome prediction; players next match performance prediction and player's future performance prediction. In most of the studies, they relied more on the tactical skills and also psychological factors. We did not find such a relevant work where machine learning algorithms were employed to predict the soccer players potential precisely or a player specific model which accurately predicts or forecast the player peak potential and the optimal age (at what age they reach that potential).

After the insight to the related work, on the performance or potential prediction [2,4,21,40] using machine-learning algorithms, we can easily categorize the algorithms used. In First Category, we can consider algorithms like linear models such as linear regression, multiple linear regression, and ridge regression. These models have the limitations as they rely on a set of features, which are extracted using different feature extraction techniques. However, they are not always fully capable of explaining the target. The Second Category includes algorithms like Support Vector Regressors, we have seen from the related work that their performance is much better than linear models in most of the cases but was not satisfactory, and they are quite computationally expensive as mentioned in [8]. All these models need ample amount of historical data to get trained on.

To prevail over all the limitations in the existing approaches, we proposed the deployment of deep neural networks and deep sequence learning methods where feature extraction is handled automatically, which usually require domain knowledge when used with traditional methods. In particular, we are looking for a model that uses least data to get train on and made an accurate prediction. We are interested in the prediction of young players peak potential with optimal age (at what age they reach that potential).

2.3 Sequence Prediction using Deep Learning

This section gives an overview of deep learning implication in diverse fields. In many of the domains especially related to time series problems, it outperformed many other learning algorithms. Domains like speech recognition, machine translation, economic time series and text prediction its performance is considered as standard. We shed light on some of its projects.

Kale. C. D et al. (2017) [28] presented work on medical data analysis. The data is related to the patients in ICU (intensive care unit). It was a multivariate time series observations data for each patient, which is based on sensor data and lab tests. They used LSTMs for pattern recognition in clinical measurements. They trained the model to classify 128 diagnoses, using 13 clinical measurements. They compared its performance with multi-layer perceptron. They found that LSTM performance outperformed the multilayer perceptron.

Nichols, P. E et al. (2016) [29] used LSTM for the prediction of volatility in stock market related to top 500 companies in the U.S. It called S&P 500. They compared the LSTM performance with linear Lasso/Ridge and autoregressive GARCH. The mean absolute error 24.2% is what they got from LSTM, and this outperformed the others by at least 31%. They also mentioned that deep learning models could show promising predicting results in stock behavior problems.

Li, R et al. (2017) [30] proposed work related to household load forecasting using pooling based recurrent neural network. They aim to learn the uncertainties like load aggregation and spectral analysis, which are avoided consistently with the use of traditional approaches. They also explained the limitation of feedforward neural network, which is over-fitting that we could have when we try to make it deep by adding a stack of hidden layers. Data collected from 920 smart-metered customers from Ireland. The proposed method outperformed the ARIMA by 19.5%, SVR by 13.1 % and simple RNN by 6.5%. All the evaluations are made using RMSE (root mean squared error).

Sick, B et al. (2016) [31] presented a work related to the power forecasting of solar power plants. They made energy output forecasting for 21 solar power plants with the help of different deep learning algorithms, which includes Deep Belief Networks, AutoEncoder and LSTM. The results indicated that deep learning algorithms forecasting performance is much superior to the Artificial Neural Networks.

Only a few of the domains or few of the work have been mentioned in the above deep learning related work, and there is a lot more, these are just a glimpse of having an idea how the interest is growing in this area. Deep learning models are capable of extracting features from the data directly means there is a minimal need for data pre-processing and in some of the cases no feature engineering is required. Optimal deep architecture design varies with the problem on hand and is mostly done by trial and error.

Next sections shed light on the data and preprocessing, working of algorithms in detail followed by the results section.

Chapter 3

Data and Pre-processing

In this chapter, we described the data in detail. We also mentioned the source of data and what features does the data have? Besides that, we also mentioned pre-processing steps that we have applied.

3.1 Data Collection and Analysis

The data we have considered for this research is collected by SciSports, and SciSports has extracted the data from Scoresway.com. SciSports is one of the fastest growing sports analytics companies in Netherland. They mainly use data intelligence to understand soccer. SciSports has transformed the data according to their business needs. In data transformations, they made these specific features and their values are get measured by Scisports custom built in-house algorithm. The data has around 224K distinct players from more than 75 soccer leagues. Each player in the dataset has 16 features. Following table 01, mentions feature names, their acronyms, meanings, and range.

Feature name	***Acronym***	***Meaning***	***Range min-max***
Player_id	Pid	Assigned unique id to players by Scoresway.com	Player_Number
Sciskill Index	SS	SS index computed after the match. Scisports in-house created index	0-200
Sciskill_now	SSN	Scisports in-house created a feature	0-200
Potential	Potential	Each player potential computed after the match	0-200
Resistance	Resistance	Opposition factor	0-100
Defensice_Skill	DS	Players defensive skill	0-10
Offensive Skill	OS	Players offensive skill	0-10
Match date	MD	Date of match used by Scoresway.com	dd/mm/yy
Offensive_share _player	OSP	The weighted avg. offensive share of the player	0-1
Match_id	MID	The unique id of match used by Scoresway	Match id
Position_id	Fpid	The unique id of the position of the player	Int 0-65
Minutes Played	MP	The number of minutes played by player in the match	0-90

Competition_id	CID	The unique id of the competition that match played in	id
Offensice_share_this_match	OSTM	Offensive share of player in the specific match	0-1
Team id	Tid	The unique id of team player played for	id
Age	Age	Player age	Mm/yy

Table 3.1: Data Features with acronyms, meaning and range

3.2 Data Pre-processing:

In data pre-processing step, we have discarded some of the players' data that played only for a year and players that have only played under the age of 15. We discarded those records because if we want to predict those players future potential, we remain unable to verify our results, as we do not have ground truth-values for them. Furthermore, we also discarded some of the players data that started playing later than the age of 26, because most of the players reach their peak potential between the age of 25-27 [3] and we are interested in the prediction of players peak potential.

Chapter 4

Model Architectures

In this chapter, we have explained the approaches, which we have used to predict the peak potential of soccer players. There are different approaches, which have been applied for potential prediction. It ranges from regression-baaed approaches to time series, towards neural networks and deep learning methods. Following is the overview of algorithms, which we have used for the prediction of peak potential of soccer players with optimal age. Before moving towards model's architecture, we first explain the research problem.

Problem Definition

To solve the research problem "How precisely we can predict the peak potential of young soccer players with optimal age, using machine learning techniques," we used those specific players from the dataset that started playing between the age of 15 and 18 and have played subsequently till the age of 26. We trained the model on the players' data from the age of 15 till 19 and predicted same players potential at the age of 19 till 26. The reason behind using those specific players is to have a ground truth-value for players' potential at all ages. So, we can evaluate our model performance by comparing predicted potential at specific age with the ground truth-value at that age.

We propose two categories of models 1) multiplayer models and 2) player specific models.

1) Multiplayer model– In the multiplayer model, we have used multiple players data for model training and testing. Using this approach, we used multiple players data that started playing between the age of 15 till 18 in training set and same players data from the age of 19 till 26 in the test set. Using multiplayer model, we considered that model parameters are not player dependent and we have used same model parameters to predict multiple players future potential.

2) Player specific model– In the player specific model, we have used only a player data for model training and testing. More formally, we used player's data from the age of 15 till 19 in training set and player's data from the age of 19 till 26 in the test set. We trained the model using training set and made players potential prediction on the test set. In the player specific model, we build a separate model for each player to predict their future potential, considering that model parameters will vary from player to player.

In this thesis, we first employed traditional machine learning models like linear models and feedforward neural networks for potential prediction and will consider their best performance as our baseline. We propose a deep learning method especially a variant of the recurrent neural network–LSTMs. We considered this problem as a time-series forecasting problem–that considers sequential data –where every current value has some dependency on the recent past. LSTMs have the intrinsic property of learning long dependencies in sequential data.

4.1 Multivariate Linear Regression

The linear regression model is most commonly used regression technique in forecasting. Its implementation and performance evaluation is comparatively easy [8]. Regression models are used where we have to examine the relationship between one or more independent variables let's say x_1, x_2, x_3, ..., x_k and a continuous dependent variable lets say Y [9]. In regression analysis, we try to predict the value of the dependent variable as accurate as possible using independent variable values [9]. In our case, the potential is dependent variable and features such as defensive skills, offensive skills are all independent variables or predictors. The model can be written in the form.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k \qquad (4.1)$$

Where Y is the target (potential), $\mathbf{x_i}$ is the feature input and β_i are called the regression coefficients (weights or parameters). To calculate the cost, we have used squared loss objective function, which can be expressed as

$$Cost = \frac{1}{2}\sum_{i=1}^{m}(\hat{Y} - Y)^2 \qquad (4.2)$$

Where $\hat{Y}$ is the predicted value, and Y is the actual value. The difference between them is the cost or Loss. The model training is explained in detail in section 4.6.1.

4.2 Decision Tree

A decision tree has a tree structure or graphical representation, which is easy to understand as it resembles human reasoning and is mostly used to represent tree models. Their illustration is one of the advantages they have over other learning algorithms [14].

In tree architecture the top node is called the root node– root node contains all the data and no edges entering it, and there is only one root node in a directed or rooted tree. As trees have graphical representation, so Graph G=(V, E) where V stands for nodes or vertices and E is for edges. A node with no child or descendent is called leaf node or terminal. Rest of the nodes (except root node) is called internal nodes or child nodes [16].

Decision tree models are non-parametric. They made decision rules for prediction by using greedy top-down recursively splitting of data [12]. In Regression trees, predictors or explanatory variables (continuous or categorical) are used to predict continuous or numeric dependent variable. The splitting of predictors into nodes is based on conditions like if-then-else. All the predictor variables are considered as the candidates for child node partition and the one, which minimizes the variance, would be considered as the best candidate and get selected for further splitting [13]. Each split partitioned the data into contradictory (cannot be true both at a time) child nodes– which can give maximum homogeneity at that split. Homogeneity depends on the type of dependent variable. Node homogeneity is explained by impurity. If the nodes are completely homogeneous, then impurity is zero. As homogeneity decreases impurity increases. In classification problems, we most commonly use information index (entropy), Gini index and Twoing index as measures of impurity (splitting standard).

For regression tress we mostly use Variance and Sum of absolute deviation on median [15]. Variance gives average distances from mean [17].

$$Var(x) = 1/n\ \Sigma_i{}^n{}_{=1}(x_i\text{-}x_i)^2 \qquad (4.3)$$

When we use variance with decision trees, we still do greedy minimization but use variance for partition into child nodes [17]. In the case where we have numeric explanatory variables or predictors for splitting we use greater than or less than values, for some selected value. The splitting is applied to each node further. Splitting continues till large grown tree and then gets pruned to the appropriate or proper size [15]. Modeling has two goals 1) description– simply explain the structure of data 2) prediction– how accurate it predicts the unobserved data. Decision trees are exceptionally good in both. In our case, each of the leaf nodes contains a partition of a players potential. Root node and child nodes have the features (DS, SS, OS, etc.), which are selected using binary tree split with the variance reduction method. The player potential prediction value is the one in which leaf's partition it falls. To find out to which partition players potential belongs to or to predict the players potential we need to traverse the tree using a sequence of questions at each feature level based on the answers of that questions we assign it a partition at leaf node which is the predicted potential as well.

4.3 Time Series

Time series approaches are one of the oldest methods used for forecasting. Time series approaches include a range of models from simple ones like Auto-Regressive (AR) model, Moving Average (MA), Exponential Smoothing to the complex ones like ARMA (Auto-regressive moving average), ARIMA (auto-regressive integrated moving average), etc. [8].
Time series methods are constructed on the supposition that there are internal structures in the data such as trend, seasonal and auto-correlation, which could be, exploited by using these models. ARMA models are mostly used for stationary process whereas ARIMA is the modification of ARMA to cater the problems of non-stationary processes [19].

Auto-regressive (AR) models use past values of the dependent variable; we use linear combinations of those past values to predict the dependent variable. The term *auto-regression means self-regression of dependent variable* [18].
The mathematical form or function of the auto-regressive model can be written as

$$y_t = c + \vartheta_1 Y_{t-1} + \vartheta_2 Y_{t-2} + \ldots\ldots + \vartheta_p Y_{t-p} + e_t \quad (4.4)$$

Where c is a constant and e_t is white noise [18]. Equation 4.4 resembles multivariate regression, but in multivariate regression, we use a linear combination of multiple predictors to predict the dependent variable, and in AR model we use lagged values of Y_t as predictors, Y_t is a target itself. AR models can be written as AR(p) model, where p is an integer represents a number of lag values to use in a model and is responsible for different time series patterns[18]. We have used AR(1) model, we have auto-regressed the potential (target), and created only 1-time lag feature of potential which is last year potential and used that as a predictor with linear regression to predict players future potential.

4.4 Feed Forward Neural Networks

Most commonly used neural networks are Multilayer perceptron also cited as MLPs. Its simplest architecture consists of at least three layers. 1) Input layer 2) hidden layer 3) output layer. Each layer has a number of neurons in it, and all are fully interconnected in a feed-forward way. They also referred as Feed Forward due to the architecture they posses for information flow. Neurons in the input layer are connected to the neurons in the first hidden layer (if we have multiple hidden layers than neurons in the first hidden layer get connected with the next hidden layer neurons) and hidden layer neurons are connected with the output layer neurons (in a single hidden layer architecture model otherwise last hidden layer neurons are connected to the output neuron) [2].

The classic architecture of the neural network is composed of different nested functions that's why we call them networks. [10]. The input layer neurons take input the predictors or independent variables. The output layer has the target or dependent variable (in our case it will be the player potential). At the first hidden layer, we apply the non-linear transformation with the help of activation function *f*. The most commonly used activations functions are (Rectified linear Unit (relu), Sigmoid and Tanh). The hidden layer output is in the form of $f\,(\Sigma(x;\,\Theta)+\beta)$ where Θ is the randomly assigned small weight to all the incoming links of each neuron (neurons in hidden layer and output layer),x is the input variables, and β is the bias.

In most of our experiments, we have used Relu or Rectified linear Unit as an activation function as it is recommended for modern feed-forward neural networks [10]. This function is quite close to linear function and is considered as a piecewise linear function with two linear pieces.

This type of models does not have feedback connections like in recurrent neural networks [10]. If there are no hidden layers in the network, it will be called a linear network, and it uses linear regression for learning [2].

At the input layer, we fed the network with all the features (SSN, DS, OS, OSTM, resistance, OSP, SS, Fpid), small weights get initialized between input layer and hidden layer, hidden layer and output layer. At the hidden layer we have used Relu as an activation function, and at output layer, we have used a linear function. The predicted potential we get from the output layer, after each forward pass, we compute the difference between predicted potential and actual potential with the help of squared loss objective function.

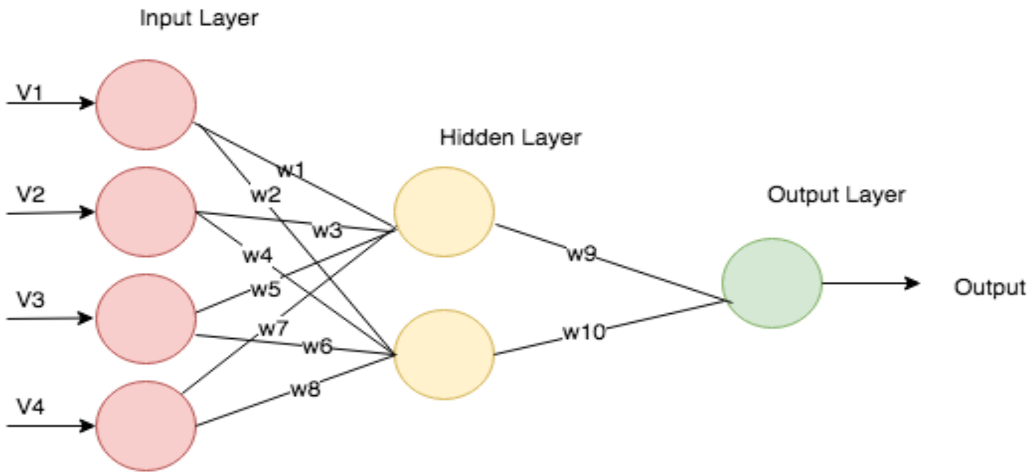

Fig.4.1: Baseline Feed forward Neural Network General Architecture. In all experiments at the output we have used linear activation, and for hidden layer, we have used relu, tanh and sigmoid activation functions with different experiments. It's a general model with one hidden layer and two neurons whereas we have used multiple layers with a different number of neurons.

4.5 Long Short-Term Memory (LSTM)

The backbone of recurrent neural networks is their cyclic connections, which makes them robust and widely used tool especially to model sequence problems. Tasks related to sequence labeling and prediction like language modeling or handwriting recognition, Simple RNN achieved unprecedented success. As a disparity with DNNs (Deep Neural Networks), SRNNs have a dynamically changing window concerning the input sequence as compared to Feed-forward networks where we have static or fixed size window. [34].

SRNNs (Simple Recurrent Neural Network) have a cyclic architecture that caters for consistently providing the previous time-step network activations as input to the current time-step. SRNNs keep track of activations for each time-step which in return makes them quite deep networks, as their depth increases it becomes hard to train such models using backpropagation through time (BPTT) because of the exploding and vanishing gradient problems–which means that when gradient is backpropagated through time it either explodes (grow exponentially) or decays (vanishes)[35].

Exploding gradient problems can be tackled by gradient clipping. The vanishing gradient is more taxing, and there has been a lot of research done to resolve the issue of vanishing gradient problem. One of the noticeable works includes the use of second-order optimization algorithms by (Martens, 2010; Martens & Sutskever, 2011) but it increases the computational cost significantly [36]. One of the other techniques to resolve the vanishing gradient problem is by the use of RNNs weights regularization proposed by (Pascanu et al., 2012). The standard used for dealing with vanishing gradient problem is LSTM (Long and short-term memory) developed and addressed by Hochreiter & Schmidhuber (1997)[35].

The LSTM (Long short-term memory) is a type of RNN and is easy to train as compared to Simple RNN [35]. LSTMs have an intrinsic property to capture or store long-term temporal dependences or information [37]. The optimization problems, which Simple RNNs get, do not affect LSTMs. In the hidden layer of LSTM architecture, it contains units called memory blocks.

Memory block consists of many components namely memory cells and gates. The memory cell is used to store the network temporal state with the help of gates. These gates (input, output, forget) with the help of activation function control information flow. Memory cells have self-connections like a carousel. Below is the detailed diagram of LSTM Memory block with all its components.

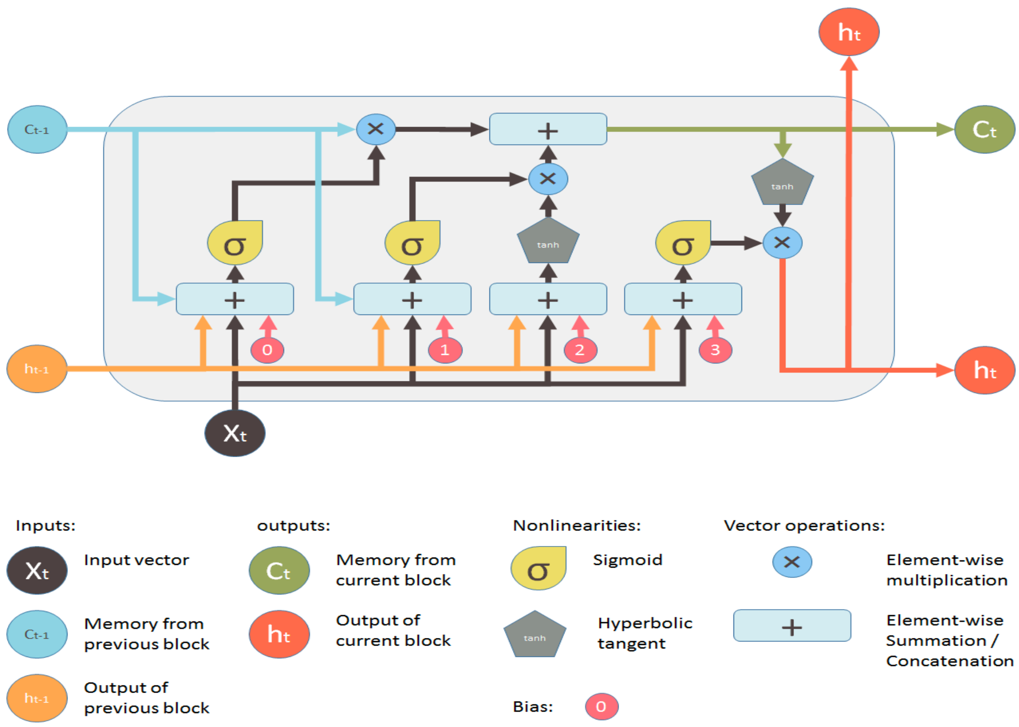

Fig.4.2: LSTM unit with all details [43].The figure depicts the inputs to each gate, activation functions applied at each step and memory carousal.

The diagram above is an LSTM Memory block. It has three inputs C_{t-1}, $H_{t-1,}$ and X_t. C_{t-1} is the memory from the previous LSTM block and is called the memory cell. H_{t-1} is the output of the previous LSTM block, and X_t is the input at current time-step. The old memory passes through the bitwise multiplication operation where it gets multiplied by a value from forget gate, which decides which information to pass through and which one to forget. If the value is close to zero, then it means forget most of the memory, and if it is close to 1, it means to pass through all the old memory.

Forget Gate

The forget gate is a one layer neural network having inputs X_t–input for the current block, previous block memory C_{t-1}, previous block output H_{t-1} and bias. This one layer neural network has a sigmoid activation function, and its output is the forget value. The weights calculated and outputted through forget gate are as follows.

$$f_t = \sigma\ (\ W_f . [h_{t-1}, x_t] + b_f) \qquad (4.5)$$

Input Gate

The second gate is input gate. Input gate has two simple one layer neural networks. Both have similar inputs (bias, the output of the previous cell, current input) like forget gate and both outputs a value one with sigmoid activation function and the other with tanh activation function. The output from a neural network with sigmoid is i_t, and the output of the other neural network with tanh is $\check{c}_t$. The value from these neural networks gets multiplied and added to old memory cell C_{t-1} and forms the new memory C_t. The weights calculated and outputted through input gate are as following [43].

$$i_t = \sigma (W_i . [h_{t-1}, x_t] + b_i) \qquad (4.6)$$
$$\check{c}_t = tanh (W_c . [h_{t-1}, x_t] + b_c) \qquad (4.7)$$
$$C_t = f_t * C_{t-1} + i_t * \check{c}_t) \qquad (4.8)$$

After the creation of new memory, LSTM block generates its output with the help of output gate.

Output Gate

Output gate takes input bias, previous block output H_{t-1}, current blocks input X_t and also the new memory. Output gate decides how much new memory should output to the next LSTM block or unit with the help of sigmoid activation function. The output of the current unit is H_t, which is formed by a bitwise multiplication operation between output gate and current unit squashed new memory C_t. Following are the weights calculated and outputted through the output gate [43].

$$o_t = \sigma (W_o . [h_{t-1}, x_t] + b_o) \qquad (4.9)$$
$$h_t = o_t * tanh(C_t) \qquad (4.10)$$

We have used LSTMs with M - 1 sequence prediction model. As predictors, we have used only a time lag feature of potential, which is last year potential. By using M to 1 model, we give many time sequential inputs (one at a time) to the model. For example, to predict the player potential at the age of 21, we sequentially give inputs from the age of 17 till 21 and then make a potential prediction at the age of 21.

4.6 Training Methods

So far, we have explained the general architecture of models. Now we will explain how the different models will get train.

4.6.1 Multivariate Linear Regression Training

Using the equation 4.1 as stated below again.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k$$

The input of the model is a vector of samples and features (DS, OS, resistance, etc.) and the target is potential. The model training starts by randomly assigning small weights to the features and model get trained by iteratively optimizing these weights by using loss function and optimization algorithm. We have used the squared loss as an objective function as mentioned in equation 4.2 and later explained in section 5.4. The cost computed through the entire training set and get minimized using optimization algorithms. More formally, the weights (β) of the model get learned or optimized using iterative optimization algorithms like gradient decent method or its variants like SGD. The general formula of gradient descent is as follows.

$$\beta_j := \beta_j - \propto \frac{\partial}{\partial \beta_j} \text{Cost}(\beta_0, \beta_1 \dots \dots \beta_n) \quad j=0,1,\dots\dots n \qquad (4.11)$$

The optimization algorithm we have used in experiments is SGD. SGD updates the weight (β) similarly (4.11) but using small batches as compared to gradient descent. For weight optimization using SGD we have used different learning rates ($\propto$) (2.0, 0.5, 3.0). We have also used the regularization parameters to reduce the over-fitting problem [10]. We have used L1 or Lasso regularization and L2 or Ridge regularization. In L1 or Lasso regularization the unimportant features are considered near to zero weight and is also used for feature selection. With regularization we modify our cost function by adding a penalty term ($\lambda \parallel \beta \parallel_1$). It can be written as follows

$$Lasso = \ minimize_{\beta \in \mathbb{R}^p} \frac{1}{2} \parallel y - X\beta \parallel_2^2 + \lambda \parallel \beta \parallel_1 \qquad (4.12)$$

Where y belongs to response and $X \in \mathbb{R}^{n*p}$ be a matrix of predictors and $\lambda \geq 0$ is a tuning parameter. Similarly, In Ridge or L2 regularization, we penalize the coefficients with large values more [11]. We can write L2 regularization as follows

$$Ridge = \ minimize_{\beta \in \mathbb{R}^p} \frac{1}{2} \parallel y - X\beta \parallel_2^2 + \lambda \parallel \beta \parallel_2^2 \quad (4.13)$$

Where y belongs to response or actual value and $X \in \mathbb{R}^{n*p}$ be a matrix of predictors and $\lambda \geq 0$ is a tuning parameter. When $\lambda = 0$, it will be a linear regression, $\lambda = \infty$, we get ridge coefficient (or penalty ($\lambda \parallel \beta \parallel_2^2$)) = 0 and for λ in between, is the case of fitting a linear model y on X and shrinking the coefficients. The evaluation metric used for model performance is mean squared error and is explained in section 5.2.

4.6.2 FFN Training

In FFN, the input of the model is a vector of samples and features (DS, OS, OSM, resistance, etc.). The model randomly assigns small weights to the input features. The learning of these weights and biases is called model training. In Feed Forward Neural Network, the gradient is computed using Backpropagation algorithm. The error gets computed after each forward pass through the network throughout the model training. The computed error is then passed backwards to update the model weights. Precisely, partial derivatives of the loss function are computed w.r.t each layers weight using chain rule mechanism. Once we have the gradient or (partial derivative w.r.t weight), we have used optimization algorithm SGD to update the model weights using equation 4.16. For Example, if x^i is the i^{th} input value to neuron, w_i is the weight given to x^i and m is the number of inputs to the neuron, then the output of neuron using non-linear activation function can be expressed as :

$$n = \sum_i^m w_i \, x^i \qquad (4.14)$$
$$z = \sigma(n)$$

Partial derivative of loss function w.r.t weight w_i using chain rule can be written as:

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial z} \frac{\partial z}{\partial n} \frac{\partial n}{\partial wi} \qquad (4.15)$$

Where L represents the loss, to compute the loss we have used squared loss objective function as explained in section 5.4. Due to large parameter space in deep learning models, there are many local minima beside global minimum means that the error surface is highly non-convex. To overcome the issue of getting stuck at the local minima, we have used iterative optimization algorithm Stochastic Gradient Descent (SGD) to minimize the Loss The evaluation metric used for model performance is mean squared error and is explained in section 5.2.

4.6.3 LSTMs Training

Forward Pass

The input of the model is a 3d tensor (samples, timestamps, data dimensions). The model gets trained using the equations 4.5 – 4.10 in a feed-forward way from time-step t-1 to T using only 1 time-lag feature of potential which is last year's potential. We must have to mention here time-steps because LSTMs consider dependency between the inputs over time [47].

Backward Pass

To understand the backward pass we have to first consider the following RNN diagram:

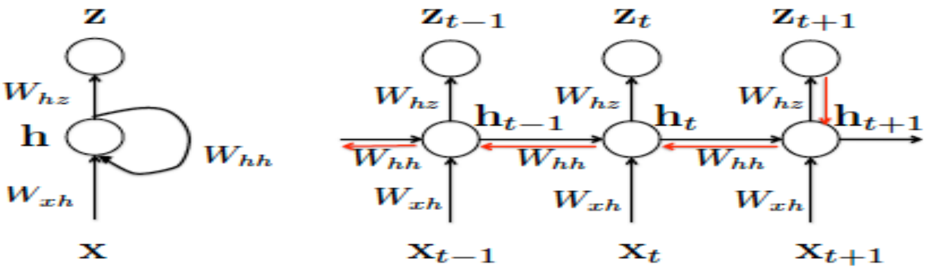

Fig. 4.3. Its an SRNN. The left is the recursive description and the right is the unfold form of SRNN model with time [47].

We are just mentioning the summary of backward pass, how error gets backpropagated and how weights get updated only at Time t. In simple RNN the loss computed at the output layer is backpropagated through time. As shown in the diagram (red cycle), we first take partial derivative of loss at the output layer w.r.t the weight W_{hz}. After that, we take the partial derivative of loss w.r.t weight W_{hh} (feedback loop) at the hidden layer, and then we take the partial derivative of loss w.r.t to the weight W_{xh} . To understand the backpropagation with RNN, we have considered extremely simple RNN architecture with 1 neuron in the hidden layer and explained the backpropagation for a single time-step t. In this case, the model parameters θ are { W_{xh} ,W_{hh}, W_{hz}}. These are the weights except the biases that's get updated using SGD at single time-step t [47]. SGD updates the weight using the following equation.

$$\theta = \theta - \eta\partial\theta \qquad (4.16)$$

η is the learning rate.

When we use LSTMs as compared to SRNN, the extra model parameters come in are at the hidden layer. The weight W_{hh} at a hidden layer in SRNN is further gets split into the input gate weight, output gate weight and forget gate weight in LSTMs. Using LSTMs, in the backward pass, we compute partial derivative of loss

(loss -computed using Squared loss objective function at the model's output) w.r.t all the weights associated with these gates beside the output and input layer weights. In LSTMs the model parameters θ are {W_{hz},W_{xo}, W_{xi}, W_{xf}, W_{ho}, W_{hi}, W_{hf}, W_{hc}} except biases. For each time-step t during backpropagation, we take partial derivative of loss w.r.t all these weights (θ). After taking the derivative we update the weights using SGD optimizer as shown in equation 4.16 [47]. The evaluation metric used for model performance is mean squared error and is explained in section 5.2.

The weights can be explained as:
$W_{hz=}$ weight hidden to output layer at time t
$W_{xo=}$ Weight output gate get from current input $X_{t,}$ at time t
$W_{xi=}$ Weight input gate get from current input $X_{t,}$ at time t
$W_{xf=}$ Weight forget gate get from current input $X_{t,}$ at time t
$W_{ho=}$ Weight output gate get from hidden layer (feedback loop), at time t
$W_{hi=}$ Weight input gate get from hidden layer (feedback loop), at time t
$W_{hf=}$ Weight forget gate get from hidden layer (feedback loop), at time t
$W_{hc=}$ Weight memory cell get from hidden layer (feedback loop), at time t

We have discussed the core idea of backpropagation through time (BPTT), just to develop the basic understanding of parameters learning in RNNs–LSTMs. We strongly recommend the reader to consult [47] for further details on computing partial derivatives and for an in-depth understanding of backpropagation through Time (BPTT) with LSTMs.

Chapter 5

Empirical Analysis and Discussion

In this chapter, we will discuss the performance measure, evaluation strategy and the results from all the experiments carried out using different datasets.

5.1 Evaluation Procedure

There are many different methods, which can be employed to evaluate machine-learning models; it also depends on the problem at hand. Some commonly used approaches include K-fold Cross Validation, Leave One Out Cross Validation (LOOCV, suitable for small and midsize datasets as it is computationally expensive with large datasets [45]) and splitting data randomly into the test, train and validation set. In our case, we used stratified K-fold cross-validation with k= 10, for some of the experiments where we considered a multiplayer model with traditional machine learning methods. In K-fold cross-validation, training data split into K smaller blocks or sets, where k-1 folds are used for training and 1 left out block or set is used for validation. To ensure that model is generalized we used holdout test set. We also used Walk forward validation method. It is a model evaluation method used mostly with time series or sequential dataset [44]. In training set, it keeps consecutive previous years or months data (for example in our case players data from the age of 15 till 18 in training set), and in the test set, it contains the following years or months data (players data from the age of 19-25). It helps with sequential data because by this we train the model sequentially with time. For example, by using this technique, it is not possible that we train the model on data from years 17 or 18 and validate it on the data from the age 16, which would be possible with k-fold cross-validation. It works as follows:

1) Select the sequential or temporal data for some specific period in training set. Let's say in our training set; we used the player's data from the age of 15 till 19.
2) Fit the model using a training set
3) Make a prediction on the test set, which contains data from the following year (For example in our case player's data from the age of 19. For every next iteration it increases by 1 year)

4) Save prediction results, which can be used later with performance metrics.
5) Increase the window size of the training set (For example, consider the player's data from the age of 15 till 20 in training set).
6) Repeat steps (2) to (5) until age 26 in our case.

5.2 Performance Metrics

We have used mean-squared-error metric (MSE) to assess the model performance on the test set, as it is one of the most widely used metrics for regression problems. It is considered as a standard metric for some of the research studies. In model comparison among different models, a metric that can discriminate more among model results is desirable, and MSE is better in it as it gives high values to adverse conditions [33]. For Example, studies related to air quality, meteorology, etc. [33]. It gets calculated for the dataset as

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i^2 \qquad (5.1)$$

Where ε_i^2 is the square of the difference between model prediction and actual value. In our case it is the difference between the predicted value of potential and ground-truth value of the potential.

5.3 Baseline

We have proposed two baseline models. First– Player specific model with Lasso regression using walk forward validation and Second–Player specific model with Feedforward neural network using walk forward validation. The performance of our baseline models prompted us to use recurrent neural networks because baseline models consider each input independently, these models do not consider the sequential nature of data, whereas our problem is of time series forecasting (predicting young player potential with optimal age) where every next value has some dependency on the recent data. To solve this problem we must have to cater the sequential nature of data, and that's what recurrent neural networks LSTMs do. For the implementation of traditional machine learning methods we have used Scikitlearn [48] and for the implementation of feedforward neural network and deep learning models–LSTMs we have used TensorFlow because for implementing complex architectures, TensorFlow provides efficiency and flexibility [49].

5.4 Dataset Preparation

These are the steps, which remain same for all the experiments. For all the experiments we have used players' data from the age of 15 till 26 because of valid ground truth-values for players at all ages. For all the experiments, we have used z-score normalization to rescale the data features or dimensions; it is used when the feature values are on the different scale. It can be expressed as:

$$z = \frac{x-u}{\sigma} \qquad (5.2)$$

Where u is the mean and σ is the Standard deviation. Same-scaled data dimensions helps gradient-based optimization algorithms to give equal importance to each feature and also helps in convergence. If we do not rescale the data dimensions, then some of the model parameters can update faster. To standardize or normalize the dataset, we have used z-score normalization. It rescales the data points to the origin (or the features are centered around zero with standard deviation of 1). We have used different splitting techniques for training and test set, which get explained later. For all experiments, we have used Squared Loss as objective (loss or cost) function as it is the most widely used loss function with regression problems. Loss function evaluates the difference between model's output and expected value. It can be written as:

$$L = \frac{1}{2}\sum_{i=1}^{m}(\hat{Y} - Y)^2 \qquad (5.3)$$

Where $\hat{Y}$ is the predicted value (model output), and Y is the expected value. M resembles the number of samples.

5.4.1 Modified Dataset

For most of the experiments, we have used the Mean value dataset. We modified the raw dataset because in our opinion it has some basic problems. For Example One of the problems we have found in the raw dataset is an irregular pattern in players' data– means players are not playing monthly, quarterly or yearly (players' data do not have specific intervals). We named this new dataset as Mean Value Dataset.

1) We have transformed the players' age to years only, which is in months and years in the raw dataset. By converting the age to years only, we managed to make regular intervals in players' data. For Example: if a player has the age of 23.1, 23.2,23.3 we downcast it to 23 (as it is less than 23.5 otherwise 24). One other reason, we changed the players' age from

months and years to years only because we are interested in predicting players' future potential per age year. For example potential at age 20 or potential at age 21.

2) We selected only important features to predict potential from the raw dataset by using recursive feature elimination technique. RFE is a *stepwise backward feature elimination* method [7]. It's an elementary approach to figure out important features and get rid of irrelevant features [4]. It ranks features using some sort of their importance. In RFE, we initially consider all the features to predict potential and recursively removes one by one based on their ranking and also on the significant improvement in model performance. To statistically verify the significant improvement in model performance we used the paired t-test with a p-value < 0.05. The important features we have considered are (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP, Age, Pid).

3) We calculated the mean of potential and 7other features (DS, OS, SSN, Resistance, OSTM, SS, OSP) with respect to specific age for all players. For example, if a player has played three matches at the age of 23.1, 23.2,23.3 and have the values for all the features, First we rounded off age to 23 (as it is less than 23.5 otherwise 24) after that we took the mean of each eight features. By taking the mean, we have a single value of features at a specific age for each player. Following are the samples from both datasets that would help the reader in understanding point 3.

Pid	Age	Ss	Ssn	Potential	Resistance	DS	OS	OSTM	Fpid	OSP
4	23.074	5.9	3.0	5.9	5.7	1.26	1.46	0.0	1	0.49
4 . .	23.115 . .	6.2 . .	3.1 . .	6.2 . .	5.8 . .	1.32 . .	1.64 . .	0.0 . .	1 . .	0.48 . .

Player's data values in the Raw dataset

Pid	Age	Ss	Ssn	Potential	Resistance	DS	OS	OSTM	Fpid	OSP
4	23.0	6.83	3.43	11.09	5.91	0.99	1.65	0.0	1	0.46

Player's data values in Mean Value dataset

5.5 Results

5.5.1 <u>Results from the experiments using raw dataset</u>

<u>Result 1</u>: Table 1 depicts the results of players' future potential prediction error using Ridge/Lasso regression with the raw dataset.

For this experiment, we have used a raw dataset with all features (Pid, DS, OS, SSN, Resistance, OSTM, Mid, MD, MP, FPid, Tid, Cid, SS, Age, OSP, Potential). We selected those specific players from the dataset that started playing between the age of 15 and 25 and played subsequently till the age of 26. After normalizing the data, we split it into test and training set. In training set, we used the players' data from the age group of 15 - 25, and in the test set, we used the same players' (as in training set) data from the age of 26. We have considered all the 15 features as predictors or explanatory variables (Pid, DS, OS, SSN, Resistance, OSTM, Mid, MD, MP, FPid, Tid, Cid, SS, Age, OSP) and the target or dependent variable is potential. We used this experiment as one-step forecasting. This experiment is an attempt to test a multiplayer model. For this experiment, we have used multivariate Ridge and Lasso regression. The model training is explained in detail in section 4.6.1. We tested the model using 10- fold cross-validation. We configured the model using different learning rates (2.0,0.5,3.0). The optimal learning rate we got is 3.0 using multivariate Lasso regression. We get a mean potential prediction error of 37.07, using multivariate lasso regression with learning rate of 3.0. Multivariate Lasso regression models outperformed the Ridge, regression model. In table1, we have only mentioned the best performance from the models obtained. Rest of the model configurations that we have tried and results can be found in Appendix A1.

Dataset	Training set	Test set	Regularization	Learning rate	Test Set (MPPE).Players age 26
All features (raw dataset)	Same players age=15-25	Same players 26	Lasso	3.0	N=36460p n=36460p
					37.07
			Ridge	3.0	69.815

Table 1: Multiplayer Model. Using Ridge/Lasso Regression. Same players in training and test set. Raw dataset used
N = represents the number of players in training set
n= represents the number of players in test set

Result 2: Table 2 depicts the results of players' future potential prediction error using decision tree regressor with the raw dataset.

For this experiment, we have used a raw dataset with all features (Pid, DS, OS, SSN, resistance, OSTM, Mid, MD, MP, FPid, Tid, Cid, SS, Age, OSP, Potential). We selected those specific players from the dataset that started playing between the age of 15 and 25 and played subsequently till the age of 26. After normalizing the data, we split it into test and training set. In training set, we used the players' data from the age group of 15 - 25, and in the test set, we used the same players' (as in training set) data from the age of 26. We considered all the 15 features as predictors or explanatory variables (Pid, DS, OS, SSN, RTCE, OSTM, Mid, MD, MP, FPid, Tid, Cid, SS, Age, OSP) and the target or dependent variable is potential. This experiment is an attempt to test a multiplayer model. For this experiment, we have used decision tree regressor. We have used Variance reduction to measure the quality of a split. We tested the model using 10- fold cross-validation. We configured the model with max-depth=2, splitting criteria or feature selection used is variance reduction and minimum samples required at leaf is set to 1. The evaluation metric used for the model performance is mean-squared-error. The mean potential prediction error using decision tree is 76.22 with the raw dataset.

Dataset	**Training set**	**Test set**	**Test set (** mean p.prediction error **)**
			1-yr **26yr (MSE)**
All features-raw dataset	Same players 15 -25	Same players 26	N=36460 n=36460 76.228

Table 2: Multiplayer Model. Results using Decision tree regressor with raw dataset

The reason behind the use of multiple players' data in training set with a minimum starting age of 15 till 25 and do a step forecasting or one year ahead potential prediction is just to have enough data for model training and evaluate model performance. We have seen from the results that models performances are not satisfactory and have enormous mean potential prediction errors. In our opinion, Better performance of Lasso regression than Ridge regression, also indicates that all the features are not important.

5.5.2 Results using mean value dataset

Result 3: Table 3 depicts the results of players' future potential prediction error using Lasso regression with mean value dataset.

For this experiment, we have used mean value dataset with 11 features (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP, Potential, Age, Pid). We selected those specific players from the dataset that started playing between the age of 15 and 25 and played subsequently till the age of 26. After normalizing the data, we split it into test and training set. In training set, we used the players' data from the age group of 15 - 25, and in the test set, we used the same players' (as in training set) data from the age of 26. We have considered eight features as predictors or explanatory variables (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP) and the target or dependent variable is potential. We used this experiment as one-step forecasting. This experiment is an attempt to test a multiplayer model. For this experiment, we have used multivariate Lasso regression. The model training is explained in section 4.6.1. We tested the model using 10- fold cross-validation. We configured the model using different learning rates (0.5, 0.4, 3.0). The optimal learning rate we got is 0.5 using multivariate Lasso regression. We get a mean potential prediction error of 8.50, using optimal learning rate of 0.5 with multivariate Lasso regression.

Dataset	**Training set**	**Test set**	**Learning rate**	**Test set (** mean p.prediction error **)**
Mean values best features (8)	Same players age=15-25	Same players		**N=40565 n=40565**
			0.5-Lasso	8.502

Table 3: Multiplayer Model using Lasso Regression. Same players in training and test set using mean value dataset.

<u>Result 4</u>: Table 4 depicts the results of players' future potential prediction error using Decision Tree regressor with mean value dataset.

For this experiment, we have used a mean value dataset with 11 features (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP, Age, Pid, Potential). We selected those specific players from the dataset that started playing between the age of 15 and 25 and played subsequently till the age of 26. After normalizing the data, we split it into test and training set. In training set, we used the players' data from the age of 15 till 25, and in the test set, we used the same players' (as in training set) data from the age of 26. We considered eight features as predictors or independent variables (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP) and the target or dependent variable is potential. This experiment is an attempt to test a multiplayer model. For this experiment, we have used Decision tree regressor. We have used Variance reduction to measure the quality of a split. We tested the model using 10- fold cross-validation. The model configuration is same as we used for experiment 2. The evaluation metric used for the model performance is mean-squared-error. The mean potential prediction error using decision tree with mean value dataset is 55.509.

Dataset	Training set	Test set	Test set (mean p. prediction error)
			1-yr 26 yr MSE
8-Best features (Mean Values)	Same players 15- 25	Same players 15-25	N=40565 n=40565 55.509

Table 4: Multiplayer Model. Results using Decision Tree Regressor with mean value dataset

Result 3 and Result 4 showed improvement in the model performance when employed with mean value dataset. We have used the same longer training window of 10 years as in experiment 1 and 2, is just to highlight the model performance using different datasets.

Moreover, we created a new dataset where we used time lagged features of potential as predictors and ignored all others and named the dataset as Time lag dataset. New features created by auto-regressing the potential. We created this dataset because after converting the players' age from years and months to years only, we managed to create a sequence in players' data with respect to age. With having sequential data, we can analyze the patterns evolving over time, which is also called Time Series analysis. We have created two auto-regressed features, which are last year potential and the difference between a current year and last year potential.

Result 5: Table 5 depicts the results of players' future potential prediction error using Decision Tree regressor with time series dataset.

For this experiment, we have used a time lag dataset with five features (Age, Pid, Potential, last year potential, difference b/w this year potential and last year potential). We selected those specific players from the dataset that started playing between the age of 15 and 25 and played subsequently till the age of 26. After normalizing the data, we split it into test and training set. In training set, we used the players' data from the age of 15 till 25, and in the test set, we used the same players' (as in training set) data from the age of 26. We used 2 time-lag features as predictors or independent variables (last year potential, difference b/w this year potential and last year potential) and the target or dependent variable is potential. This experiment is an attempt to test a multiplayer model. For this experiment, we have used Decision Tree regressor. We tested the model using 10- fold cross-validation. The model configuration is same as we used for experiment 2. The evaluation metric used for the model performance is mean-squared-error. The mean potential prediction error using decision tree with time lag dataset is 135.30.

Dataset	Training set	Test set	Test set (mean P.prediction error)
			1-yr **26 yr** **MSE**
Mean values(2-time lag features)	Same players age=15-25	Same players **26 yrs**	N=36348 n=36348 **135.308**

Table 5: Multiplayer Model. Results using Decision Tree regressor with time lag dataset

Result 6: Table 6 depicts the results of players' future potential prediction error using Lasso regression with time lag dataset.

For this experiment, we have used time lag dataset with five features (Age, Pid, Potential, last year potential, difference b/w this year potential and last year potential). We selected those specific players from the dataset that started playing between the age of 15 and 25 and played subsequently till the age of 26. After normalizing the data, we split it into test and training set. In training set, we used the players' data from the age of 15 till 25, and in the test set, we used the same players' (as in training set) data from the age of 26. We have used 2 time-lag features as predictors or independent variables (last year potential, difference b/w this year potential and last year potential) and the target or dependent variable is potential. This experiment is an attempt to test a multiplayer model with time lag dataset. For this experiment, we have used multivariate Lasso regression. The model training is explained in section 4.6.1. We tested the model using 10- fold cross-validation. The optimal learning rate we got is 2.5 using multivariate Lasso regression. The evaluation metric used for model performance is mean-squared-error. We get a mean potential prediction error of 55.450, using learning rate of 2.5 with multivariate Lasso regression.

Dataset	Training set	Test set	Learning rate	Test set (MPPE) 26 yrs. N=36348 n= 36348
Mean values (2 lag values)	Same players	Same players 26 yrs	**2.5-Lasso**	55.450

Table 6: Multiplayer Model Lasso regression with time lag dataset

From the experiments 5 and 6, we have seen that models performance get reduced when employed with time lag dataset. We think that the features we made by auto-regressing the potential are not capable enough to explain the dependent variable, which is potential. Other than that, in time series analysis every next input is dependent on the recent past, but these models considered each input independently.

In the next experiments, In order to develop a model that can be used to predict the young players peak potential, we used the players' data between the age of

15 and 18 in training set and same players' (as in training set) data from the age of 19 till 26 in the test set. We have preferred the age interval 15- 19 to train the model because we want to develop a model that can predict the peak potential of young soccer players. And we are interested in predicting the players potential till the age of 26 because most of the players peak potential reach by the age of 26 [3]. For next experiments, we also used Player specific model beside multiplayer model. For next experiments, we used mean value dataset and time lag dataset.

5.5.3 <u>Results using multivariate Lasso Regression:</u>

<u>Result 7:</u> Table 7 depicts the results of players' future potential prediction error using Lasso regression with mean value dataset.

For this experiment, we have used mean value dataset with 11 features (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP, Potential, Age, Pid). We selected those specific players from the dataset that started playing between the age of 15 and 18 and played subsequently till the age of 26. After normalizing the data, we split it into test and training set. In training set, we used the players' data from the age of 15 till 19, and in the test set, we used the same players (as in training set) data from the age of 19 till 22. We have considered eight features as predictors or independent variables (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP) and the target or dependent variable is potential. This is an attempt made to come up with a multiplayer model to analyze how far we can predict with accuracy the players potential using fixed size window. In this experiment, we have not used walk forward validation. We trained the model using training set and predicted the test set age intervals one by one. In this experiment, we used the fixed window size for training (15 till 19). We firstly predicted the players' future potential for the age 19, and then keeping the training set same we predicted the players future potential at the age of 20 and so on till the age of 22. For this experiment, we have used multivariate Lasso regression. The loss (cost or objective) function we have used is Squared loss. We have used the SGD (stochastic gradient descent) for weight optimization. We tested the model using 10- fold cross-validation with a learning rate of 0.5. The evaluation metric used for the model performance is mean-squared-error. The result shows that as we move forward, our potential prediction error starts growing. It shows that we cannot use fixed size training window to predict multistep ahead players potential precisely.

Model	Training	Test set players age (mean P.prediction error)		
		19	20	21
LR-Lasso(0.5)	All players (15-18)	520.910	589.106	607.926

Table 7: Multiplayer model using mean value dataset with fixed sized window and Lasso regression

<u>Result 8:</u> Table 8 shows the results of players' future potential prediction error from the experiment where we used Lasso regression with Walk Forward validation method as mentioned in section 5.1.

For this experiment, we have used mean value dataset with 11 features (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP, Potential, Age, Pid). We selected those specific players from the dataset that started playing between the age of 15 and 18 and played subsequently till the age of 26. After normalizing the data, we split it into test and training set. In training set, we used the players' data from the age of 15 till 19, and in the test set, we used the same players (as in training set) data from the age of 19 till 26. We have considered eight features as predictors or independent variables (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP) and the target or dependent variable is potential. This is an attempt made to come up with a multiplayer model using walk forward validation. All steps related to WFV explained in detail in section 5.1. We have used multivariate Lasso regression. The model training is explained in section 4.6.1. We trained the model with a learning rate of 0.5. From the results, we can see that if we use a window length of 8 years than we can best predict the potential of all players using multivariate lasso regression as it gives minimum mean potential prediction error with the multiplayer model.

Model	Training	Test set age (mean P.prediction error)						
		19	20	21	22	23	24	25
Lr – (0.5) Lasso	All players (15-18)	520.910	269.293	137.870	60.608	18.027	6.784	7.341

Table 8: Multiplayer model using Lasso Regression and walk forward validation with mean value dataset

<u>Result 9:</u> Table 9 shows the results of specific player potential prediction error with age where we used Lasso regression with Walk Forward validation method. It is our baseline model 1.

For this experiment, we have used mean value dataset with 11 features (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP, Potential, Age, Pid). We have used only a specific player. After normalizing the data, we split the data into test and training set. In training set, we used the player's data from the age of 15 till 19, and in the test set, we used the player's data from the age of 19 till 26. We have considered eight features as predictors or independent variables (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP) and the target or dependent variable is potential. This is an attempt to test a player specific model using walk forward validation. All steps related to WFV explained in detail in section 5.1. We have used multivariate Lasso regression. The model training is explained in section 4.6.1. From the results, we can see that to predict each player's future potential with minimum potential prediction error we need to use different window size for model training. We found that model parameters vary player to player or are player specific. As compared to the results of the multiplayer model from experiment 8 we found that multiplayer model results are not reliable. The results of this experiment proved that model parameters are player specific and also to predict each player future potential with MPPE we need a different window size for model training. We also found that we need a different learning rate to train the model for each player. For example, for player id 1964 we get the minimum potential prediction error by using a window size of 4-years for model training whereas for player id 5206, we cannot predict future potential precesily even by using a window size of 9-years for model training. It is our baseline model 1.

	Model	Training specific player (15-18)	Testing players (potential prediction error)						
1			19	20	21	22	23	24	25
	0.5L	P-194	13763.920	15363.023	10297.565	141.276	1.872	4859.571	1428.410
2	7.0 L	P-1964	2523.025	102.406	6.706	53.625	137.234	61.330	25.720
3	7.0 L	P-5206	1759.055	699.083	1033.902	2175.555	2926.033	3066.091	1877.914

Table 9: Baseline Model 1. Player Specific model with walk forward Validation

Following graphs showing specific player potential w.r.t age. The graph shows the predicted value of potential with an orange curve and actual value of players potential in blue curve.

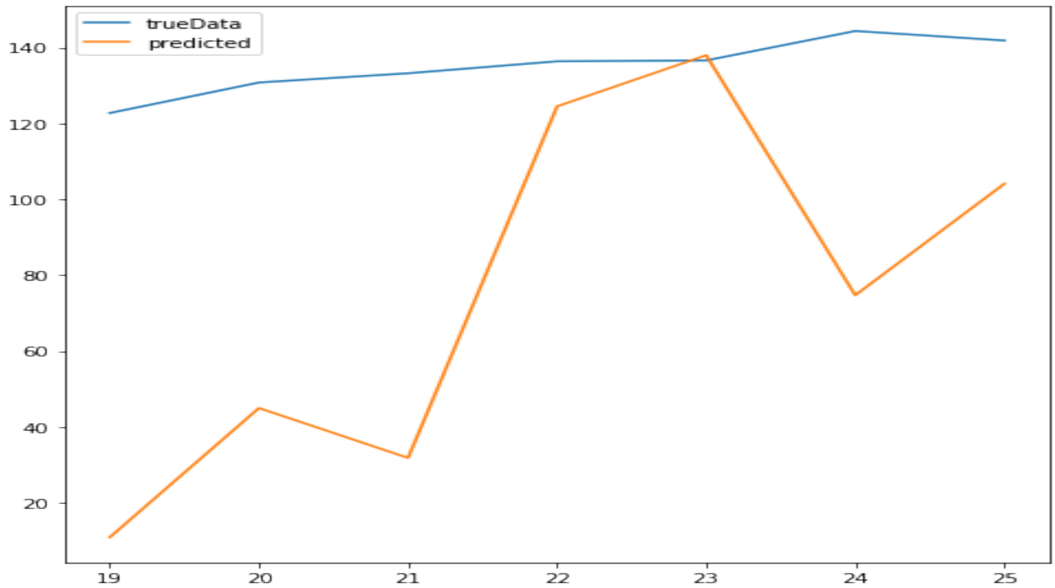

Fig. 5.1: player specific model potential w.r.t age. The blue line shows the actual potential, and orange-line shows the predicted potential. The graph is for player-id 194

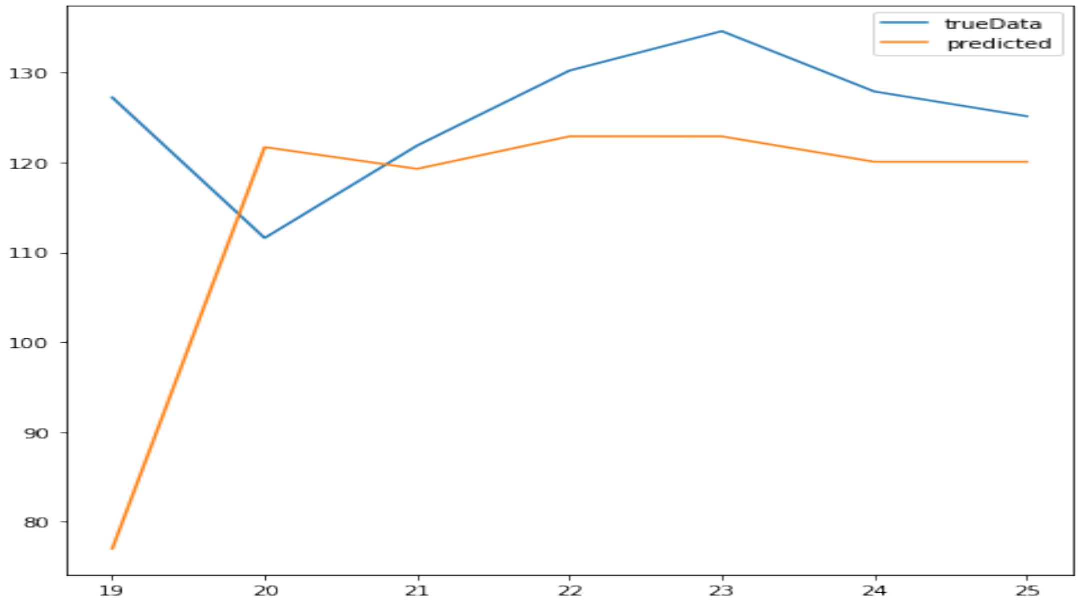

Fig. 5.2: player specific model. potential w.r.t age. The blue line shows the actual potential, and orange-line shows the predicted potential. The graph is for player-id 1964

5.5.4 Results From Feed Forward Neural Network (FFN):

Using FeedForward Neural Network we tried every experiment for 8 times and taken the average of potential prediction error at a specific age. We used numpy. seed (7) as well for reproducibility.

Result 10 & 11: Table 10 and 11 shows the results of players' potential prediction error from the experiment where we used Feed Forward Neural Network with

Walk Forward validation method. The only difference between table 10 and table 11 is that the former one is with SGD optimizer and the latter one is with Adam optimizer. The results showed that model with SGD optimizer outperformed the other one.

For this experiment, we have used mean value dataset with 11 features (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP, Potential, Age, Pid). We selected those specific players from the dataset that started playing between the age of 15 and 18 and played subsequently till the age of 26. After normalizing the data, we split it into test and training set. In training set, we used the players' data from the age of 15 till 19, and in the test set, we used the same players (as in training set) data from the age of 19 till 26. We have used eight features as predictors or independent variables (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP) and the target or dependent variable is potential. This is an attempt to test a multiplayer model using walk forward validation. All steps related to WFV explained in detail in section 5.1. In this experiment, we have used Feedforward neural network. The model training is explained in section 4.6.2. We trained the model with different configurations (using different activation functions, optimizers, learning rates and with multiple layers). We have shown the best results in table 10 and 11, all the other results with different configurations can be found in Appendix A2 and A3. We trained the model with different learning rates (0.0001,0.001,0.003,0.0003), but we get the best model performance with SGD (lr =0.001) with six hidden layers and relu as an activation function using nine years window size for model training. From the results, we can see that if we use a window size of 9 years for model training than we can predict the players' future potential with minimum prediction error using Feedforward neural network with the multiplayer model.

Model	Optimizers	Layers	Activation	Test player ages (mean P.prediction error)						
				19	20	21	22	23	24	25
FFN (Training all players (15-18)-	SGD 0.0001	6	relu	603.72	240.09	111.85	50.52	15.55	9.49	13.02
		2	relu	513.186	202.205	128.264	44.461	17.55	10.65	13.87
		4	relu	221.20	129.85	131.64	138.72	18.00	21.86	9.75
	Sgd 0.001	**6**	**relu**	**504.90**	**266.154**	**107.58**	**49.57**	**15.05**	**8.73**	**8.42**

Table 10 Multiplayer Model using Sgd optimizer and walk forward validation

Model	Optimizers	Layers	Activation	Test player ages (mean P.prediction error)						
				19	20	21	22	23	24	25
FFN (Training all players (15-18)-	adam 0.0001	6	Relu	3823.51	500.99	307.27	197.25	112.31	81.37	63.15
	0.001	6	relu	522.32	181.59	94.46	50.05	28.48	21.05	24.70

Table 11: Multiplayer Model using Adam optimizer with walk forward validation

Result 12: Table 12 results are from the experiment where we tested a player specific model with Feed Forward Neural Network using Walk Forward Validation.

For this experiment, we have used mean value dataset with 11 features (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP, Potential, Age, Pid). After normalizing the data, we split it into test and training set. In training set, we used the player's data from the age of 15 till 19, and in the test set, we used the player's data from the age of 19 till 26. We have considered eight features as predictors or independent variables (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP) and the target or dependent variable is potential. We have used Feedforward neural network to train the model. The model training is explained in section 4.6.2. We trained the model with different configurations (using activation functions (relu, sigmoid, tanh) with different combinations of them, optimizers (sgd, adam), learning rates (0.001,0.5, 1,3,7) and with multiple layers (1 till 8)). We have shown best results in table 12. In this experiment, we have used Adam as an optimizer to minimize the loss. We have used five hidden layers. For each hidden layer, we alternatively used sigmoid and tanh activation function. In table 12, we have mentioned the number of neurons in each hidden layer. The player's selection is entirely random. We can see from the results that to predict future potential precisely, for player id 1964, we need 8 years of window size to train the model. For player id 194 and 5206 we need a window size of 4 years and 3 years respectively, to train the model. In our opinion, the difference in window size is because of the potential behavior of players. The former one (id -1964) has no trend in potential with age – means its potential is not surging or plunging with age, but for the latter ones with player id – 194 and 5206, potential is increasing steadily with age. From the results, it is clear that we need a smaller window size to predict player's future potential with minimum potential prediction error that have some trend in potential with age. We considered this model as our second baseline model. By the comparison of results 10, 11 and 12 it again proved that we cannot use the multiplayer model to predict players future potential because model parameters are player specific. To predict each player future potential with MPPE we need a different window size for model training.

Player	Optimizers	Layers	Activation	Test player ages (potential prediction error)						
				19	20	21	22	23	24	25
1964	Adam 5.0 L.Rate	5	Sig,Tanh, Sig 1,1,1,1,1	133.32	509.06	1.08	70.37	178.27	11.29	0.39
194	Adam 7.0 l.r	5	Sig,Tanh, Sig	5488.20	201.40	1.35	17.65	69.50	834.57	1174.81
5206	Adam 7.0 lr	5	Sig,Tanh, Sig	7.72	1.06	175.78	1761.58	2659.62	2562.64	4084.35

Table 12: Second Baseline model. Player specific model using Feed Forward Neural Network with walk forward validation. Mean Value dataset used.

Following graphs showing the specific player potential w.r.t age using Feedforward neural network with mean value dataset and walk forward validation. The graphs showed the predicted value of potential in the orange and actual value of player potential in blue.

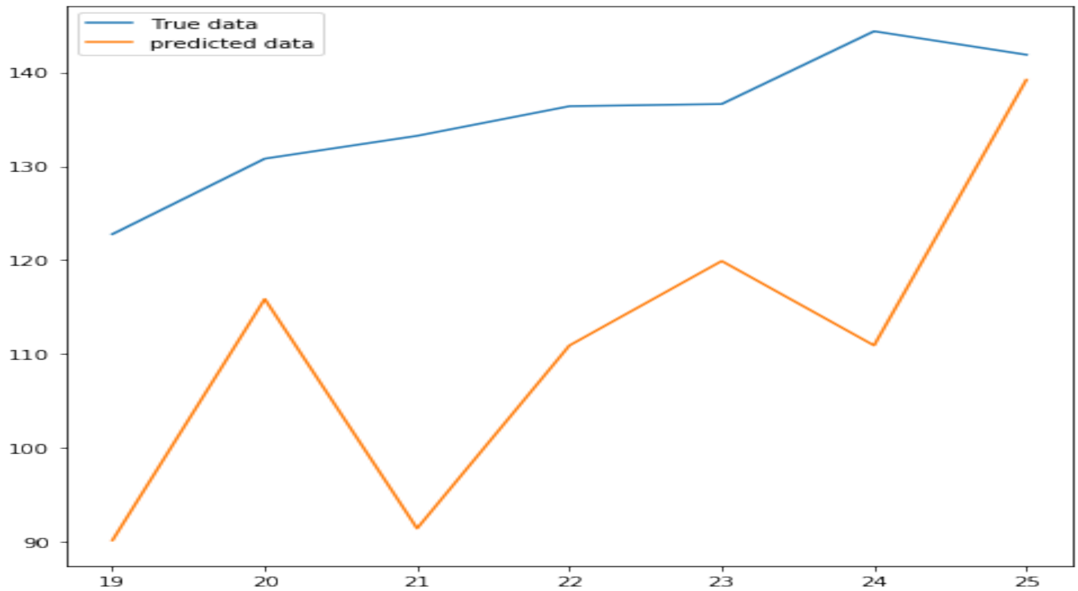

Fig. 5.3: player specific model. potential w.r.t age. The blue line shows the actual potential and orange- line shows the predicted potential. The graph is for player-id 194

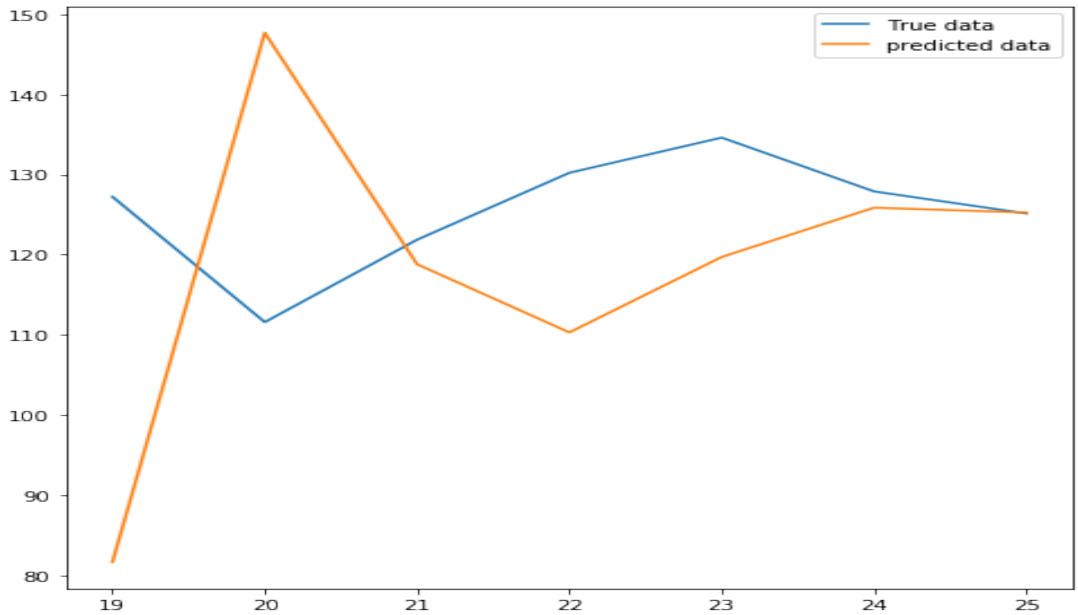

Fig. 5.4: player specific model. potential w.r.t age. The blue line shows the actual potential, and orange-line shows the predicted potential. The graph is for player-id 1964

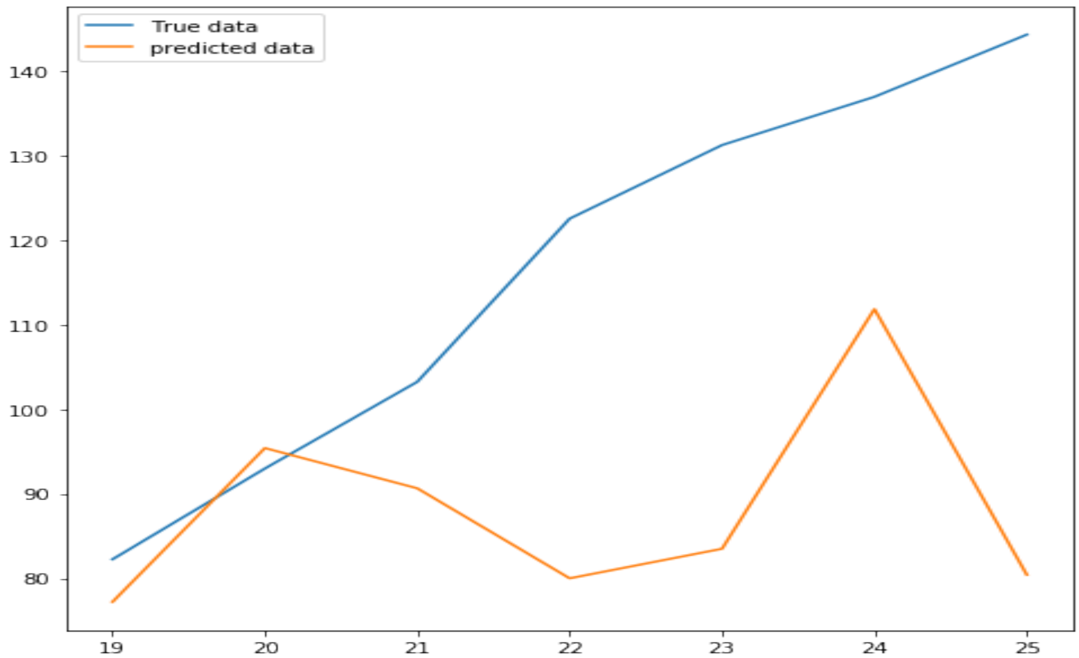

Fig. 5.5: player specific model. potential w.r.t age. The blue line shows the actual potential, and orange-line shows the predicted potential. The graph is for player-id 5206

Result 13: Table 13 presents the results of the experiment where we recursively added back the player's predicted potential in the training set before next prediction made. In this experiment, all the model configurations remained same as in previous experiment (no. 12). This experiment is an attempt to test player's specific model. In this experiment, we recursively added back the player's predicted potential at the specific age to the training set before we made next potential prediction. As we can see from the results, potential prediction error increases with age. It is because potential prediction error for every specific age gets accumulated with the next.

Player	Optimizers	Layers	Activation	Test player ages (potential prediction error)						
				19	20	21	22	23	24	25
1964	Adam 5.0	5	Sig,Tanh ,Sig 1,1,1,1,1	171.78	53.81	98.96	8.49	104.08	2.13	2.89
194	Adam 7.0	5	Sig,Tanh ,Sig	3410.13	2625.52	3598.94	3643.02	4349.47	4823.84	4642.35
5206	Adam 7.0	5	Sig,Tanh ,Sig	0.834	2.303	104.985	1951.126	4680.30	5853.16	5782.34

Table 13: Player specific model using Feed Forward Neural Network with adding back prediction.

Following graphs showing the specific player potential w.r.t age using Feedforward neural network with mean value dataset and where we recursively added back the predicted potential to the training set. The graph shows the predicted value of potential in orange and actual value (G.T) of player potential in blue.

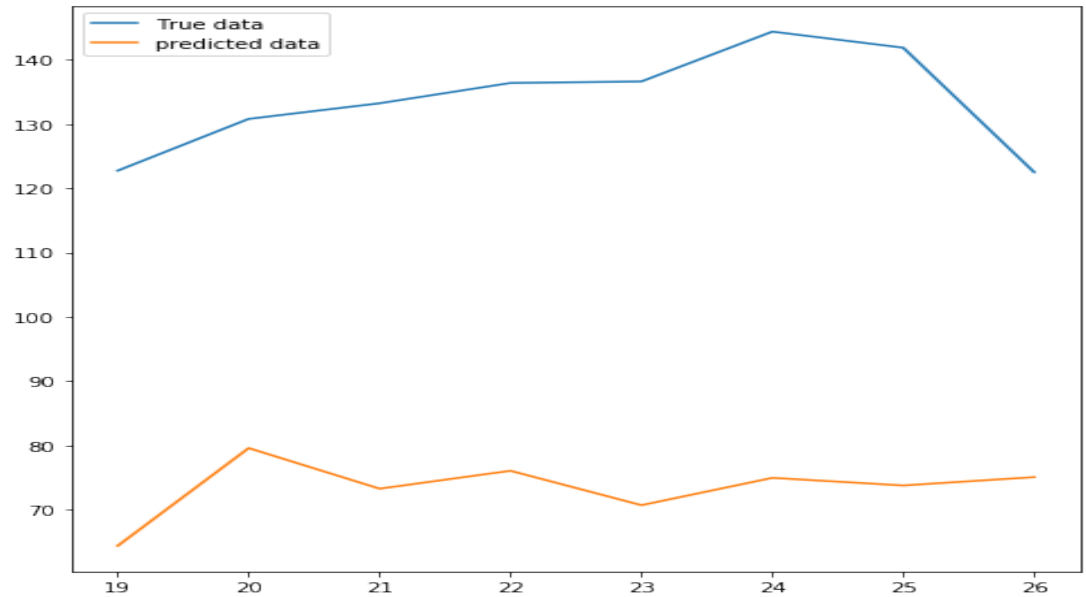

Fig. 5.6: player specific model. potential w.r.t age. The blue line shows the actual potential, and orange-line shows the predicted potential. The graph is for player-id 194 using FFN with adding back prediction

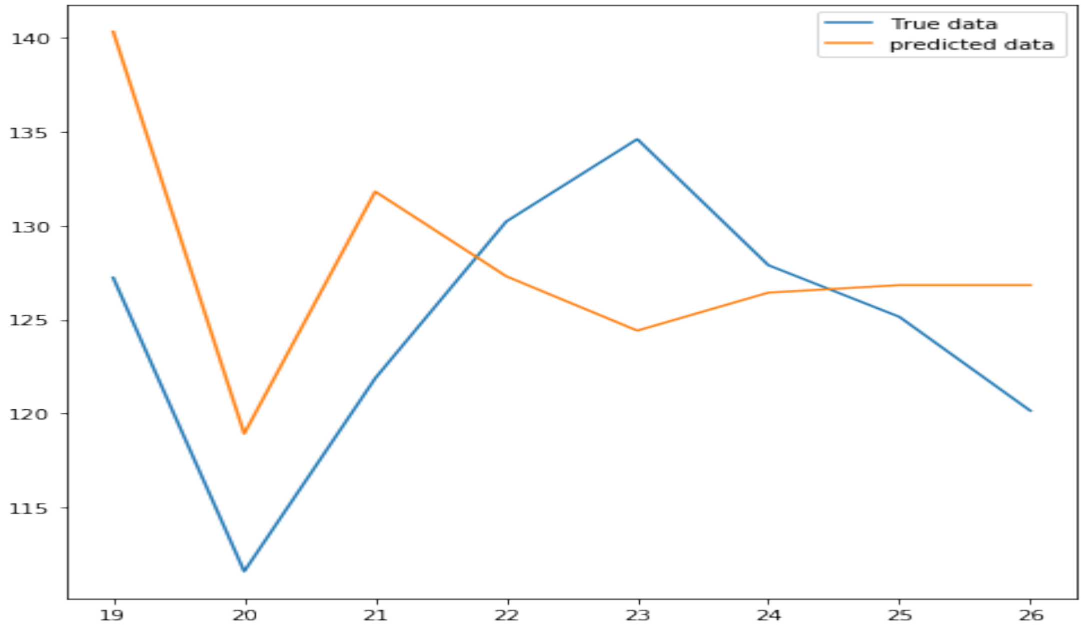

Fig. 5.7: player specific model. potential w.r.t age. The blue line shows the actual potential, and orange-line shows the predicted potential. The graph is for player-id 1964 using FFN with adding back prediction

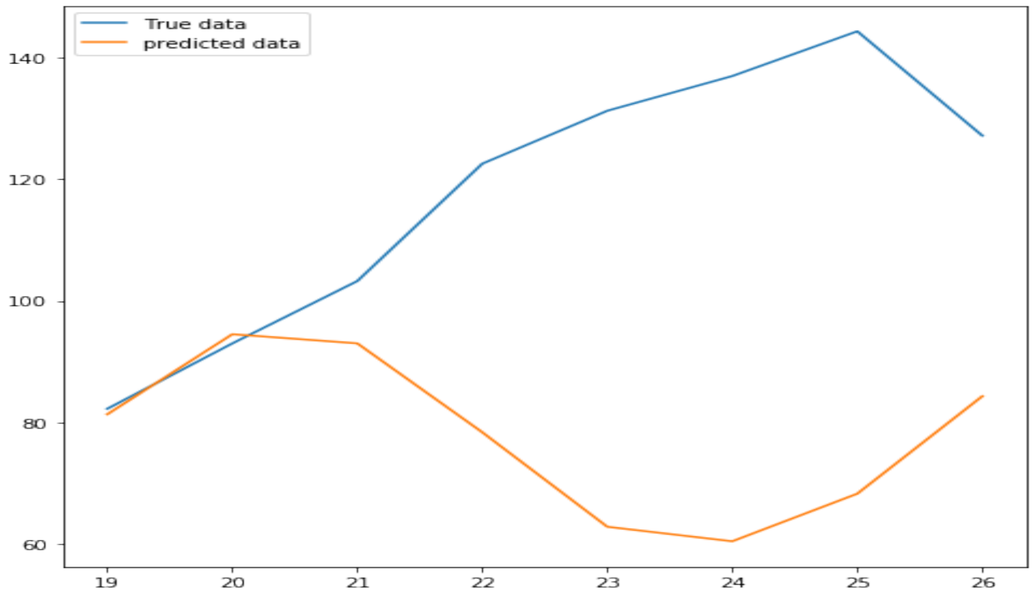

Fig. 5.8: player specific model. potential w.r.t age. The blue line shows the actual potential, and orange-line shows the predicted potential. The graph is for player-id 5206 using FFN with adding back prediction

Result 14: Table 14 depicts the results of player potential prediction error with age from the experiment where we have used Feed Forward Neural Network with time lag dataset.

The dataset we used for this experiment has four features (Pid, Age, potential, last year potential (time lag feature)). For this experiment, we created only 1-

time lag feature (last year potential) by auto-regressing the target, which is potential. After normalizing the data, we split it into test and training set. In training set, we used only the player's data from the age of 15 till 19, and in the test set, we used the same player's data from the age of 19 till 26. We have used 1 time lag feature as a predictor or independent variable, which is last year potential and dropped all other features. The target or dependent variable is potential. We have used a player specific model with walk forward validation. For this experiment, the model architecture of feed-forward neural network consists of 2 hidden layers with sigmoid and tanh as an activation function respectively. We also mentioned the number of neurons in each hidden layer in table14. The model training is explained in section 4.6.2. We found that results obtained using time lag dataset (table 14) for some players have less potential prediction error at specific age as compared to the (table 12) results where we used multiple independent variables as predictors. It means that time lag features of potential are more capable of predicting potential than other features (DS, OS, SSN, Resistance, OSTM, FPid, SS, OSP) using FFN. We also noticed that for each player, the model needs a different learning rate for optimal results. It also shows that model parameters are player dependent.

Player	Optimizers	Layers	Activation	Test player ages (potential prediction error)						
				19	20	21	22	23	24	25
1964	Adam 3.0	2	Sig 4N, Tanh1N	714.28	24.29	0.07	9.65	162.08	21.23	16.9
194	Adam 3.0	2	Sig4N, Tanh1N	104.24	112.79	313.83	167.91	150.92	316.98	166.92
5206	Adam 7.0	2	Sig4N, Tanh1N	2.67	197.78	313.15	1147.40	840.99	1621.56	1092.77

Table 14: A Player specific model with Feed Forward Neural Network using time lag dataset and walk forward validation.

Following graphs showing the specific player potential w.r.t age using Feed Forward Neural Network with time lag dataset and walk forward validation. The graph shows the predicted value of potential in the orange and actual value of player potential in blue.

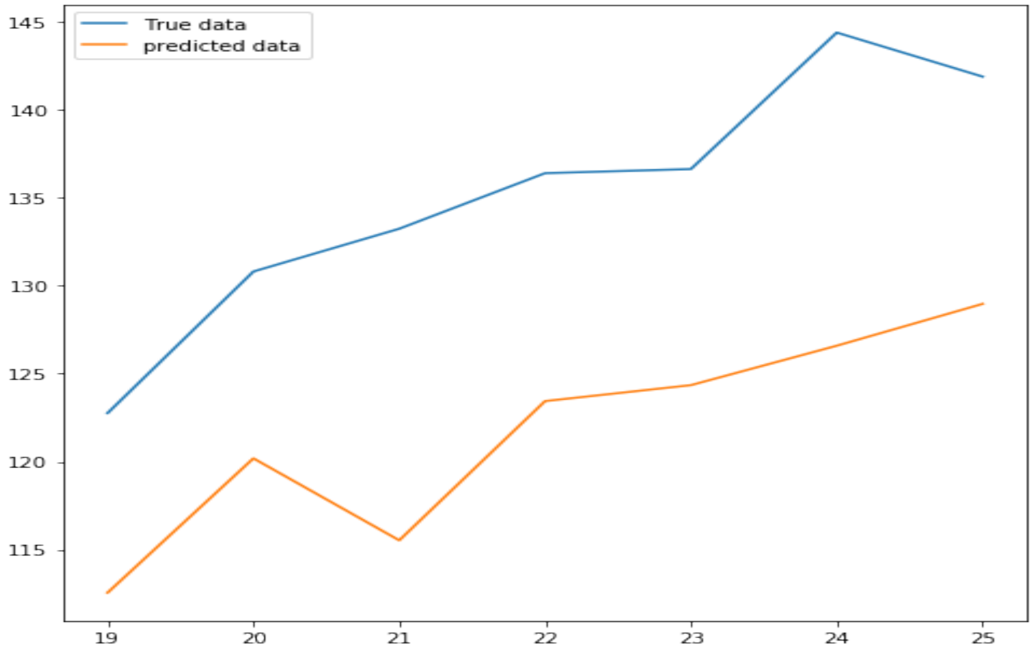

Fig. 5.9: player specific model. potential w.r.t age. The blue line shows the actual potential, and orange-line shows the predicted potential. The graph is for player-id 194.FFN with time lag dataset

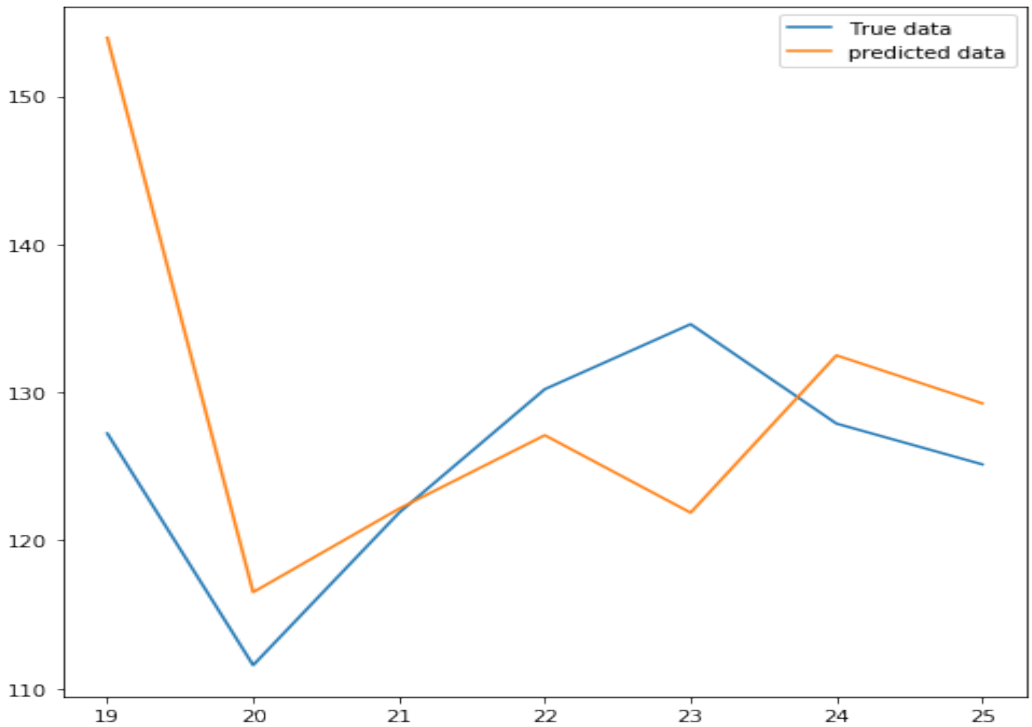

Fig. 5.10: player specific model with FFN. potential w.r.t age. The blue line shows the actual potential, and orange-line shows the predicted potential. The graph is for player-id 1964.

5.5.5 Results using LSTM

With LSTMs we only used time lag dataset. As LSTMs are capable of extracting features automatically, which are effective as compared to the manual feature extraction process, which needs domain knowledge and is cumbersome. LSTMs work on sequential data. So, to use a dataset with LSTMs, we used two approaches.

1) Window size approach– where we made sequences using window size, window size tells that how much previous year data should be included in one sequence. It made small sequences (equal to the size of the window) of data and used that sequences for training. There are some architectural drawbacks with this approach due to considering multiple players for a multiplayer model. When used with multiple players it remains unable to distinguish where the new player data is going to start. So the last sequence of player x gets merged with the first sequence of player x+1.

2) The second approach we used is with time lag dataset– we created a single time lag feature (last year potential) from the given potential. We used this approach for all of our experiments.

Result 15: Likewise, previous experiments with other learning algorithms, we trained a multiplayer model using LSTM (Long short-term memory). Table 15 results show the players potential prediction error using LSTMs.

The dataset we considered for this experiment has four features (Pid, Age, potential, last year potential (time lag feature)). For this experiment, we created only 1time-lag feature (last year potential) by auto-regressing the target, which is potential. After normalizing the data, we split it into test and training set. In training set, we used players' data from the age of 15 till 19, and in the test set, we considered the same players (as in training set) data from the age of 19 till 26. We have used 1 time-lag feature as a predictor or independent variable, which is last year potential and dropped all other features. The target or dependent variable is potential. We made this attempt to test a multiplayer model using LSTMs. The network architecture we have used for this experiment has one hidden layer with 32 LSTMs cells or units. The training of the model is explained in detail in section 4.6.3. The loss (cost or objective) function we have used is Squared loss. In this experiment, we have used Adam as an optimizer with a learning rate of 0.001 to minimize the loss as it outperformed SGD. From the results, we can see that LSTMs remained unable to make a sequence from the data and that's why their mean potential prediction error is high.

Model	Optimizers	Layers	Test player ages (mean p. prediction error)						
			19	20	21	22	23	24	25
LSTM (Training all player	adam 0.001	1	163.88	111.56	91.53	67.55	54.55	55.52	57.33

Table 15: Multiplayer Model.walk forward technique using LSTMs

Result 16: Table 16 depicts the results of player future potential prediction error from the experiment where we trained a player specific model and used the walk forward validation with LSTMs.

The dataset we used for this experiment has four features (Pid, Age, potential, last year potential (time lag feature)). For this experiment, we created only 1 time-lag feature (last year potential) by auto-regressing the target, which is potential. After normalizing the data, we split it into test and training set. In training set, we used only single player's data from the age of 15 till 19, and in the test set, we used same player's data from the age of 19 till 26. We have used 1 time-lag feature as a predictor or independent variable, which is last year potential and the target or dependent variable is potential. We have used a player specific model with walk forward validation. For this experiment, we used recurrent neural network variant LSTMs. The network architecture we have used for this experiment has one hidden layer with 10 LSTMs cells or units. We also tried it with 1,32 and 64 units. We used multiple optimizers (SGD, adam) with different learning rates (0.001, 0.01) to train the model as shown in the table. The model training is explained in section 4.6.3. The results showed that LSTMs learns the sequence when applied to the player specific model. We get zero prediction error with this model, so we can use this model to find out players peak potential with optimal age.

Player	Optimizers	Layers	LSTM Block or units	Test player ages (potential prediction error)						
				19	20	21	22	23	24	25
1964	SGD (0.01)	1	1,10,32,64	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Adam (0.001)									
194	SGD (0.01)	1	1,10,32,64	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Adam (0.001)									

Table 16: LSTM Player specific model with walk forward validation.

Result 17: Table 17 results are from the experiment where we recursively added the player's predicted potential in training set before we made next potential prediction.

The dataset we used for this experiment has four features (Pid, Age, potential, last year potential (time lag feature). For this experiment, we created only 1 time-lag feature (last year potential) by auto-regressing the target, which is potential. After normalizing the data, we split it into test and training set. In training set, we considered only single player data from the age of 15 till 19, and in the test set, we considered same player data from the age of 19 till 26. We have considered 1 time-lag feature as a predictor or independent variable, which is last year potential and the target or dependent variable is potential. The model training is explained in section 4.6.3. In this experiment we first predicted the player potential at the age of 19 then, for the potential prediction at age 20, we used the previously predicted potential value for 19 and added it in our training set and then made the potential prediction for the age 20 and we continued it till the age of 26. We tested 25 different players some of the results are shown in table 20, rest of the results and player potential graphs are in Appendix A4. Players we tested mostly have one year of training data as they started playing at the age of 18 (according to our dataset), some of them have two years of training data, and only a player with id 154236 has a data starting from 16 and last till 22. We used all these different scenarios to verify the model performance.

Player	Optimizers	LSTM Units or Blocks	Layers	Test player ages (potential prediction error)							
				19	20	21	22	23	24	25	26
1964	SGD (0.01)	1,10,64	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
194	SGD (0.01)	1, 10, 64	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2885	SGD (0.01)	64	1	0.0	0.0	0.0	0.0	0.0	0.0	0.962	0.0
5206	SGD (0.01)	64	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6408	SGD (0.01)	64	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
33340	SGD (0.01)	64	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 17: player specific mode using LSTM. With recursively adding predicted values back in training set.

Following graphs showing specific player potential w.r.t age. The graph has two curves, blue is for actual potential at a specific age (ground truth-value), and the orange line is the predicted value of potential using LSTMs model. The graph shows one single curve, which means that both values of potential (actual and predicted) are overlapping and have 0 prediction error.

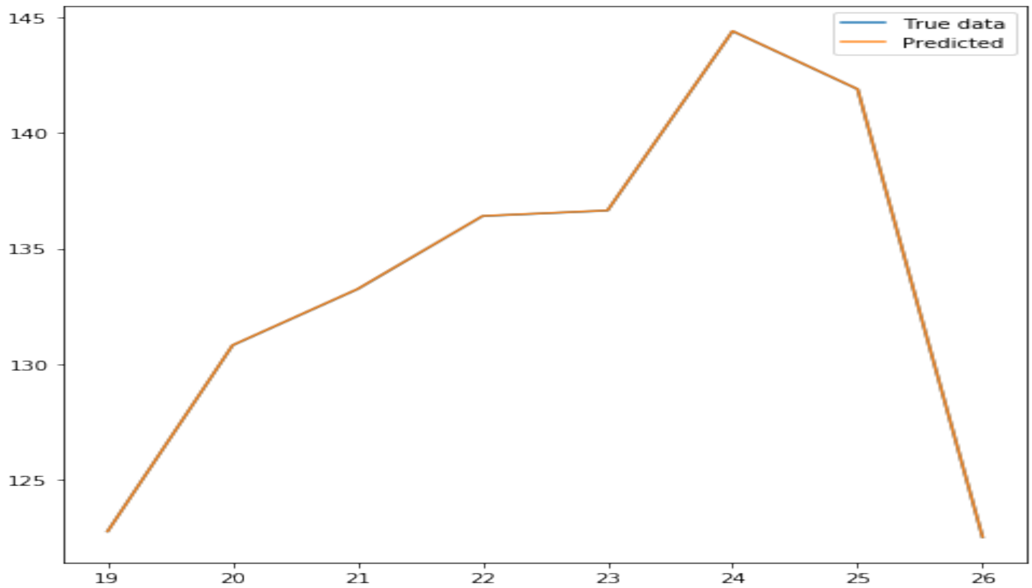

Fig. 5.11: player specific model. Shows the overlapping of predicted and actual Values of potential w.r.t age using LSTMs. player-id 194

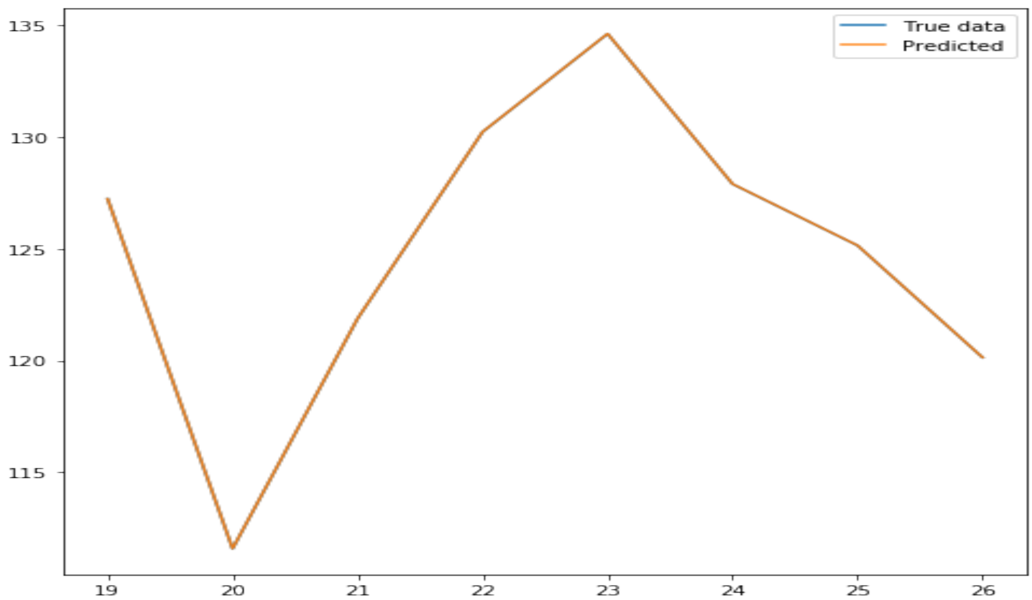

Fig. 5.12: player specific model. Shows the overlapping of predicted and actual Values of potential w.r.t age using LSTMs. player-id 1964

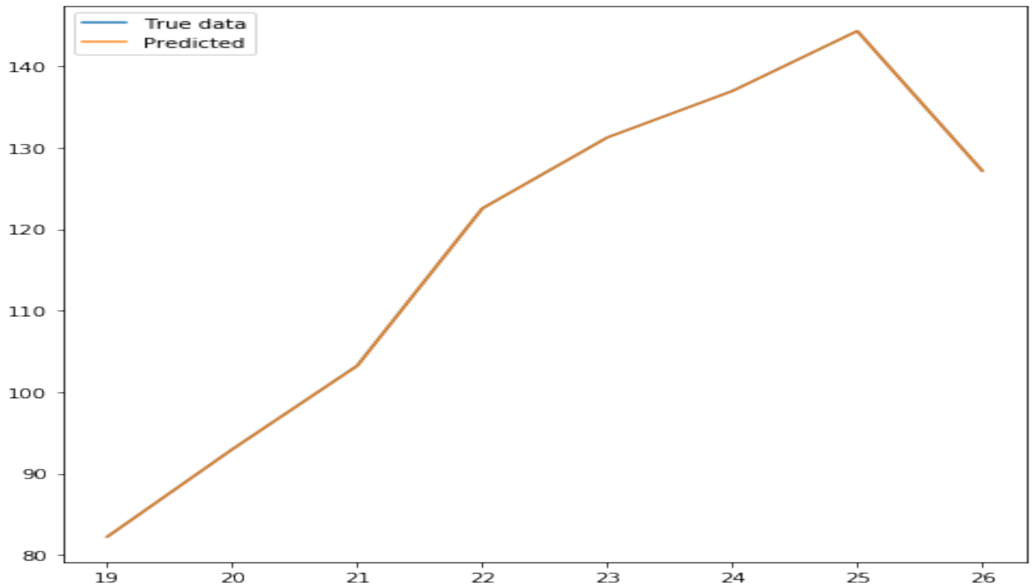

Fig. 5.13: player specific model. Shows the overlapping of predicted and actual Values of potential w.r.t age using LSTMs. player-id 5206

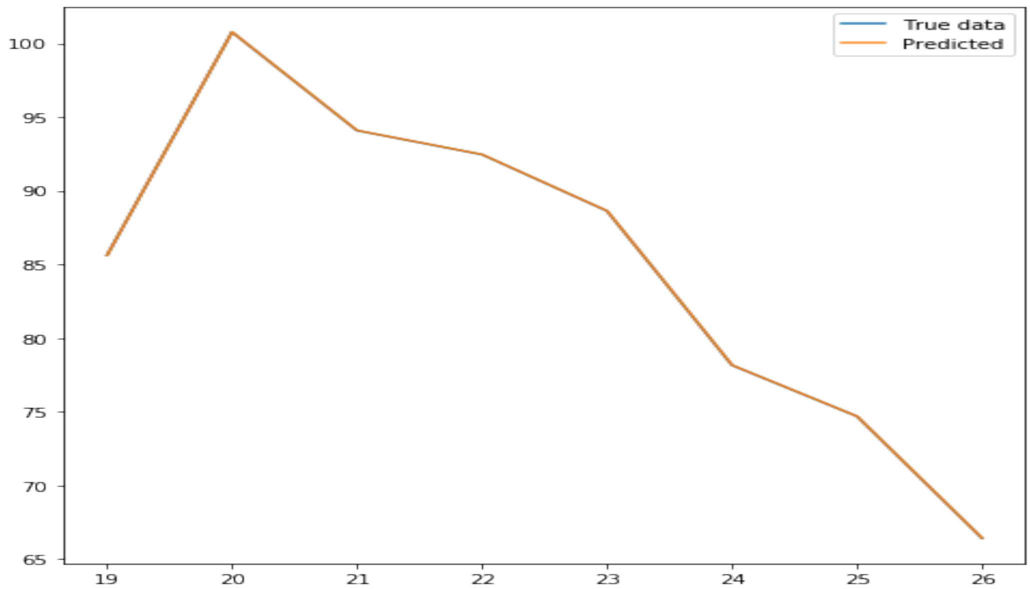

Fig. 5.14: player specific model. Shows the overlapping of predicted and actual Values of potential w.r.t age using LSTMs. player-id 2885

5.6 Discussion

In this thesis, we intensively assessed the use of deep neural networks for time series forecasting problem or the sequence prediction problem, which is young soccer player's peak potential prediction with optimal age. This study exhibits that recurrent neural network LSTMs performed exceptionally well, with the sequence prediction problems even with the small training set.

In data-preprocessing, we discarded some of the players data. We found that by using mean value dataset, our model performance improved as compared to model performance with the raw dataset. We also noticed that z-score normalization has the great impact on the model performance, especially on deep learning methods.

From the results, we found that multiplayer model results using lasso regression, FFN and LSTM are not dependable as they show that we can predict all players future potential with MPPE by using same window size and with same model parameters. It proved wrong when we have used a player specific model with lasso regression FFN and LSTMs. We found that to predict the future potential of every specific player; we need a different learning rate and different model parameters. Furthermore, we found that to predict each player future potential from the age of 19 till 26 with MPPE we need a different window size for training the model.

From the results, we found that we cannot use Lasso regression even with the player specific model for precisely predicting young players future potential because it does not get train on small window size specifically by using only the player's data between the age of 15 till 19 and thus needs longer window size. Moreover, we have seen that we remained unable to predict future potential precisely for player id 5206, even after using a window size of 9 years for model training. Likewise, Using FFN with the player-specific model, we remained unable to made precise future potential predictions of young soccer players by using only the player's data between the age of 15 till 19 for model training. However, FFN performed much better than the baseline model 1. As compared to baseline model 1, for some players, it predicted the future potential with MPPE by using small training window and also players potential prediction error at specific age are much lesser than baseline model1.

We have noticed that we can predict player's future potential with less potential prediction error at specific age by using only auto-regressed time lag features of potential as predictors with FFN. In our opinion, auto-regressed time lag features are more capable of explaining the potential as compared to other extracted features.

From the experiments, we have seen that LSTMs performed accurately and had given zero potential prediction error. LSTMs performed in such a way because they catered the sequential nature in data and have considered some dependency between the recent inputs with time. They have the property of dynamically changing window size means hidden layer does not only depends on the current input but also depends on the previous hidden state. By this technique, they learned the entire sequence. We have used M-1 sequence prediction model with LSTMs. In many to one model, we train the model by giving multiple inputs with consecutive time-steps and then made a prediction. If we gathered all the model predictions, they become a sequence, and we can call it sequence prediction.

Chapter 6

Conclusion and Future Work

The aim of this thesis is to use deep learning method for the sequence prediction. In our case, it's about the prediction of young soccer players peak potential with optimal age and also to find out which variables or predictors are important to predict the potential precisely. After experiments and results we come up with a conclusion that feature extraction and feature engineering has an important role in training traditional machine learning algorithms, but when it comes to deep learning methods, they don't rely much on handcrafted features, they are capable of finding hidden patterns in the data with time. Feature extraction and feature engineering is a time-consuming process which also depends on researchers ability and requires domain knowledge. After all this, it is not guaranteed that extracted features are fully capable of explaining the target. We also conclude that baseline models performance improved significantly when we have used them with regular interval or mean value dataset which we have created using the mean values of features at a specific age.

From the results, we concluded that it is not possible to predict young players future potential precisely, using only their initial years of data specifically from the age of 15 till19 with the multiplayer model.

We can also conclude that it is not possible to predict young players future potential precisely, using only their initial years of data specifically from the age of 15 till 19 with lasso regression and FFN even with player-specific model because these model considers every input independently and do not cater the sequential nature of data and also requires more training data.

From the results, the performance of recurrent neural networks LSTMs is obvious. It prevailed over all the limitations which baselines models get tangled with. LSTMs used only one auto-regressed time lag feature and got trained on very small data. We think that the outstanding performance of LSTMs is due to their intrinsic ability of memory, by using that they consider some dependency between recent inputs with time. We concluded that we could use LSTMs for precisely predicting the young players peak potential with optimal age.

One of the major limitations of our proposed model is that we have to train a separate model for each player, which is time-consuming as well. As a future work, we suggest splitting the data by field positions and then applying the traditional machine learning methods, we think that by splitting dataset in such a way can improve the performance of multiplayer model as well. Another important future work would be the use of clustering by players' age; each cluster contains players of same age or cluster can be made on the field positions as well. As we converted the dataset into regular interval by using mean of features at specific age for each player, related to this, one of the future work could be done with data generation, up-sampling or downsampling method, where we generate the data for the missing periods and by doing that we can come up with a dataset that have regular intervals. One of the future works would be the use of decision tree and Arima methods with the player-specific model.

In our opinion, forecasting problems like this one should not rely on the set of features other than time lag features because in forecasting problems we predict future, and we do not have the values for those variables or predictors in future. For such problems (where we consider set of features) we first have to predict the features and then use them for further prediction or forecasting. We think such an approach can lead to more unreliable prediction as compared to the one where we used time lag features.

Appendix

A1 Hyper-parameters

Dataset	Training set	Test set	Learning rate	Test Set (MSE)
				26yr MSE
All features	Same players age=15-25	Same players		**N=36460p n=36460p**
			3.0-L	37.069
			2.0L	39.637
			3.0-R	69.815
			0.05-R	69.815
Mean values best features (8)	Same players age=15-25	Same players		**N=40565 n=40565**
			0.5-L	8.502
			0.4-L	8.667
			3.0-L	15.941

Table A1: Hyperparameter Configuration using Lssso Regression.

Model	Optimizers	Layers	Activation	Test player ages (MSE)						
				19	20	21	22	23	24	25
FFN (Training all players (15-18)-	SGD 0.0001	2	relu	616.59	287.78	149.16	62.61	20.96	10.16	12.84
		4	relu	529.81	268.80	131.33	56.16	16.35	9.11	12.40
		6	**relu**	**603.72**	**240.09**	**111.85**	**50.52**	**15.55**	**9.49**	**13.02**
	Sgd 0.001	2	Tanh	651.47	262.91	114.01	66.02	30.16	17.37	17.34
			relu	**513.186**	**202.205**	**128.264**	**44.461**	**17.55**	**10.65**	**13.87**
			Sigmoid	640.80	326.01	172.12	77.29	33.28	18.85	19.07
		4	Tanh	526.83	232.48	65.16	29.78	23.02	15.62	43.01
			Sigmoid	746.68	509.11	411.20	366.26	339.31	278.19	332.72
			relu	**221.20**	**129.85**	**131.64**	**138.72**	**18.00**	**21.86**	**9.75**
		6	**relu**	**504.90**	**266.154**	**107.58**	**49.57**	**15.05**	**8.73**	**8.42**

Table A2. Hyperparameter configuration (using different learning rates, hidden layers, activation functions) using Feed Froward Neural Network using walk forward Validation. Training players(15 till 19) and Test set same players .

Referred from table 13 using FFN

Model	Optimizers	Layers	Activation	Test player ages						
				19	20	21	22	23	24	25
FFN (Training all players (15-18)-	adam 0.0001	2	relu	4149.51	3426.23	2594.30	1828.66	958.77	453.66	293.11
		4	relu	3895.39	1714.19	423.91	263.40	148.86	121.15	91.70
		6	**Relu**	**3823.51**	**500.99**	**307.27**	**197.25**	**112.31**	**81.37**	**63.15**
	adam 0.001	2	relu	707.09	359.67	170.23	60.74	25.02	22.90	26.57
		4	relu	590.85	240.91	105.77	53.40	21.88	19.61	25.56
		6	**relu**	**522.32**	**181.59**	**94.46**	**50.05**	**28.48**	**21.05**	**24.70**

Table A3. Hyperparameter configuration (using different hidden layers and learning rates) using Feed Froward Neural Network using walk forward Validation. Training players(15 till 19) and Test set same players

Player	Optimizers	LSTM Units or Blocks	Layers	Test player ages (MSE)							
				19	20	21	22	23	24	25	26
1964	SGD (0.01)	1,10,64	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
194	SGD (0.01)	1, 10, 64	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2885	SGD (0.01)	64	1	0.0	0.0	0.0	0.0	0.0	0.0	0.962	0.0
5206	SGD (0.01)	64	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6408	SGD (0.01)	64	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
33340	SGD (0.01)	64	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4719	SGD (0.01)	10	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9883	SGD (0.01)	1	1	0	0	0	0	0	0	0	0
17459	SGD (0.01)	10	1	0	0	0	0	0	0	0	0
20020	SGD (0.01)	1	1	0	0	0	0	0	0	0	0
154236	SGD (0.01)	10	1	0	0	0	0	N/a	n/a	n/a	N/a
31061	SGD (0.01)	10	1	0	0	0	0	0	0	0	0
15482	SGD (0.01)	10	1	0	0	0	0	0	0	0	0
18029	SGD (0.01)	10	1	0	0	0	0	0	0	0	0
18418	SGD (0.01)	10	1	0	0	0	0	0	0	0	0
18985	SGD (0.01)	10	1	0	0	0	0	0	0	0	0
36596	SGD (0.01)	1	1	0	0	0	0	0	0	0	0
33337	SGD (0.01)	1	1	0	0	0	0	0	0	0	0
33193	SGD (0.01)	1	1	0	0	0	0	0	0	0	0
22855	SGD (0.01)	1	1	0	0	0	0	0	0	0	0

Table A4 Hyperparameter Configuration using LSTMs with recursively adding predicted values back to training set

Following graphs showing specific player potential w.r.t age. The graph has 2 curves, blue is for actual potential at specific age (ground truth-value) and the orange line is for the predicted value of potential using LSTMs model. The graph shows 1 single curve, which means that both values of potential (actual and predicted) are overlapping and have 0 prediction error.

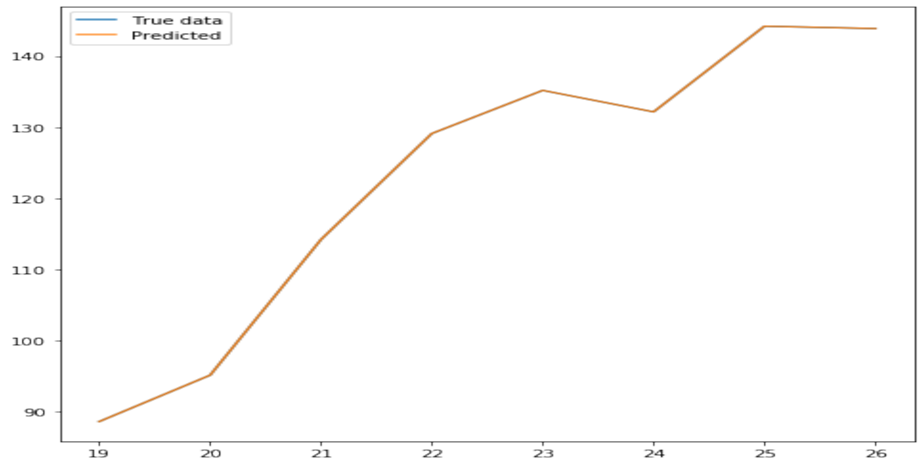

Fig. 18: player specific model. Shows the overlapping of predicted and actual Values of potential w.r.t age using LSTMs. player-id 15482

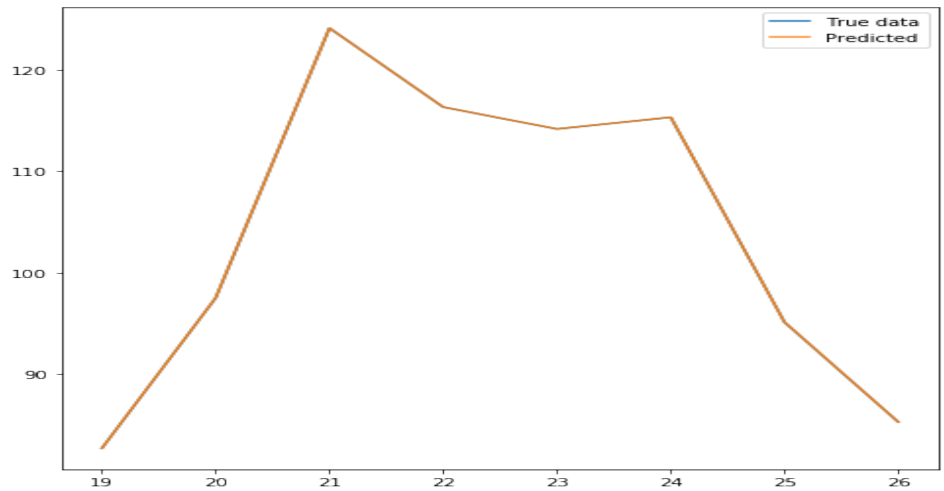

Fig. 19: player specific model. Shows the overlapping of predicted and actual Values of potential w.r.t age using LSTMs. player-id 18029

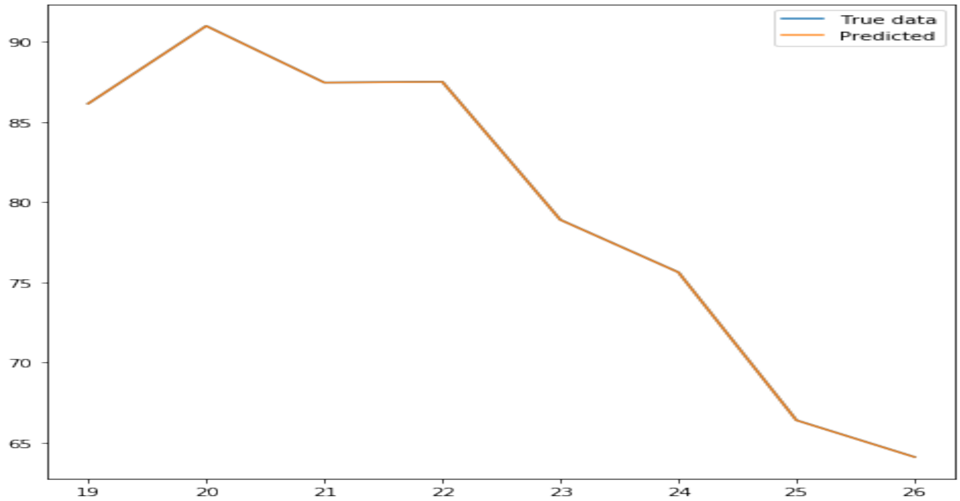

Fig. 20: player specific model. Shows the overlapping of predicted and actual Values of potential w.r.t age using LSTMs.player-id 4719

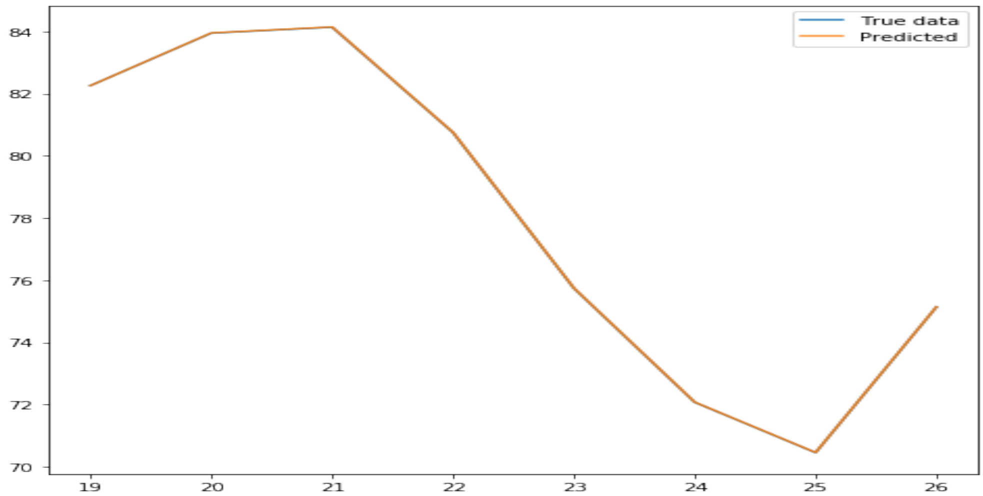

Fig. 21: player specific model. Shows the overlapping of predicted and actual Values of potential w.r.t age using LSTMs.player-id 33337

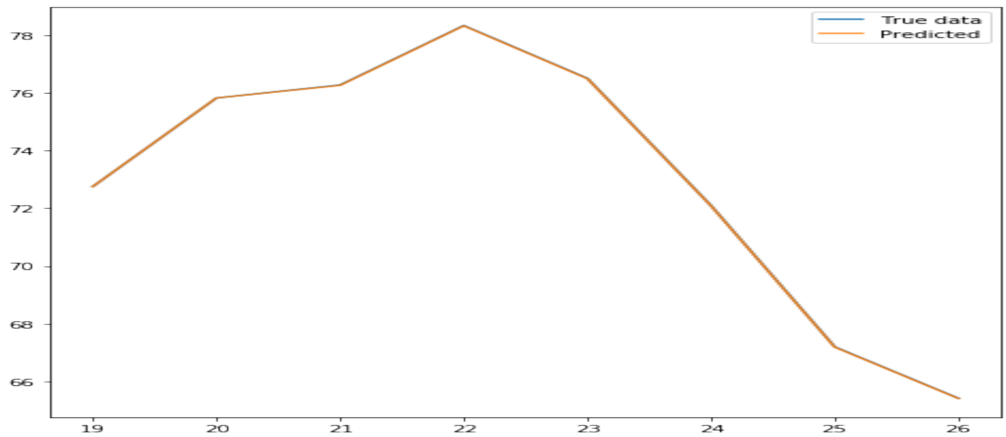

Fig. 22: player specific model. Shows the overlapping of predicted and actual Values of potential w.r.t age using LSTMs.player-id 36596

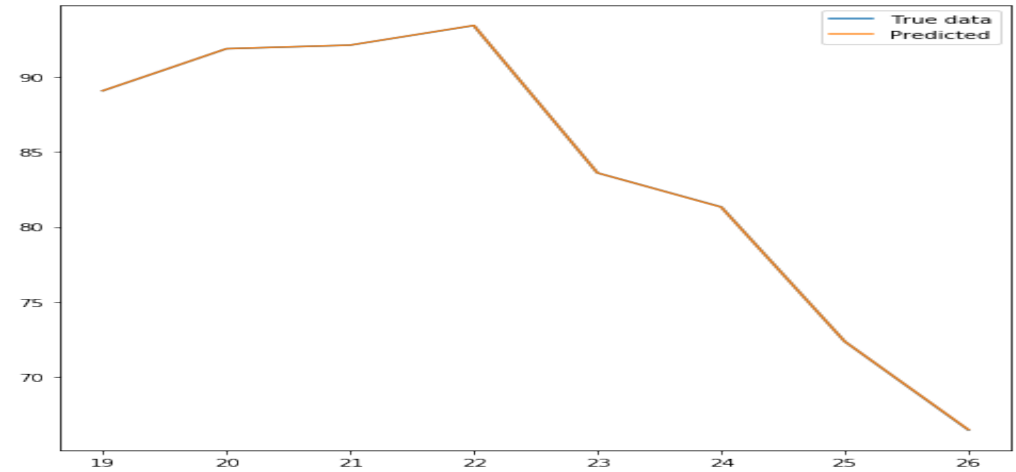

Fig. 23: player specific model. Shows the overlapping of predicted and actual Values of potential w.r.t age using LSTMs.player-id 17459

References

[1] Li, Yi & Ma, Lingxiao. (2015). Predict NBA players' career performance based on SVM. 157-161.

[2] Rama Iyer, Subramanian & Sharda, Ramesh. (2009). Prediction of athletes performance using neural networks: An application in cricket team selection. Expert Syst. Appl.. 36. 5510-5522.

[3] Dendir and Seife.(2016), When do soccer players peak? A note. Journal of Sports Analytics. 2(2), 89-105.

[4] Arndt, C. and Brefeld, U. (2016), Predicting the future performance of soccer players. Statistical Analysis and Data Mining: The ASA Data Science Journal, 9: 373–382.

[5] Toering TT, Elferink-Gemser MT, Jordet G, Visscher C.(2009). Self regulation and performance level of elite and non elite youth soccer players. J Sports Sci, 27(14),1509-17

[6] Figueiredo AJ, Coelho e Silva MJ, Malina RM. (2011). Predictors of functional capacity and skill in youth soccer players. J Med Sci Sports, 21(3), 446-54.

[7] Yang, K., Yoon, H. and Shahabi, C. (2005).A supervised feature subset selection technique for multivariate time series. Proceedings of the Workshop on Feature Selection for Data Mining: Interfacing Machine Learning with Statistics, 92-101.

[8] Hahn,H., Meyer-Neiberg ,S., Stefan, P.(2009) Electric load forecasting methods: Tools for decision making. European Journal of Operational Research. 199(3), 902-907.

[9] Amral,N., Ozveren ,CS., King,D. (2007). Short term load forecasting using multiple linear regression.Universities power engineering Conference.1192-1198.

[10] Youshua Bengio, Ian Goodfellow and Aaron Courville. (2016). Deep Learning. MIT Press.

[11] Corral Bobadilla M., Fernandez Martinez R., Lostado Lorza R., Somovilla Gomez F., Vergara Gonzalez E.P. (2017) Cyclone Performance Prediction Using Linear Regression Techniques. In: Graña M., López-Guede J., Etxaniz O., Herrero Á., Quintián H., Corchado E. (eds) International Joint Conference SOCO'16-CISIS'16-ICEUTE'16. ICEUTE 2016, SOCO 2016, CISIS 2016. vol 527.

[12] Tso, Geoffrey & K. W. Yau, Kelvin. (2007). Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. Energy. 32. 1761-1768.

[13] Muhammad A. Razi and Kuriakose Athappilly. 2005. A comparative predictive analysis of neural networks (NNs), nonlinear regression and classification and regression tree (CART) models. Expert Syst. Appl. 29(1), 65-74.

[14] R.C.Barros, M.P. Basgalupp and Alex A. Freitas. (2012). A survey of evolutionary algorithms for decision tree induction. IEEE Transactions on Systems, Man, and Cybernetics. 42(3). 291-312.

[15] Glenn De'ath and Katharina E. Fabricus. (2000). Classification and Regression Trees: A powerful yet simple technique for ecological data analysis. Ecological Society of America. 81(11). 3278-3192.

[16] S. Rasoul Safavian and David Landgrebe. (1991). A Survey of Decision Tree Classifier Methodology. IEEE Transactions on Systems, Man, and Cybernetics. 21(3). 660-674.

[17] Brendan Kitts. Regression Trees.

[18] Rob J Hyndman, George Athanasopoulos. Forecasting: principles and practice.

[19] Feinberg E.A., Genethliou D. (2005) Load Forecasting. In: Chow J.H., Wu F.F., Momoh J. (eds) Applied Mathematics for Restructured Electric Power Systems. Power Electronics and Power Systems. 269-285.

[20] Ogutu, J.O., Piepho, H., & Schulz-Streeck, T. (2011). A comparison of random forests, boosting and support vector machines for genomic selection. BMC proceedings.

[21] Malina, R. M., Ribeiro, B., Aroso, J., & Cumming, S. P. (2007). Characteristics of youth soccer players aged 13–15 years classified by skill level. *British Journal of Sports Medicine*, *41*(5), 290–295.

[22] Kannekens R, Elferink-Gemser MT, Visscher C. (2011). Positioning and deciding: key factors for talent development in soccer. J Med Sci Sports, 21(6), 846-52.

[23] Figueiredo AJ, Coelho e Silva MJ, Malina RM. (2011). Predictors of functional capacity and skill in youth soccer players. J Med Sci Sports, 21(3), 446-54.

[24] Vestberg T, Gustafson R, Maurex L, Ingvar M, Petrovic P. (2012). Executive functions predict the success of top soccer players.

[25] Zuber, C., Zibung, M., & Conzelmann, A. (2016). Holistic Patterns as an Instrument for Predicting the Performance of Promising Young Soccer Players – A 3-Years Longitudinal Study. *Frontiers in Psychology*, *7*, 1088.

[26] website link (http://www.peaksports.com/sports-psychology-blog/8-tips-to-unlock-potential-in-sports/)

[27] Abdullah, Mohamad Razali; Musa, Rabiu Muazu; Maliki, Ahmad Bisyri Husin Musawi B.; Kosni, Norlaila Azura; Suppiah, Pathmanathan K. (2016). Role of psychological factors on the performance of elite soccer players. Journal of Physical Education and Sport. 16(1).

[28] Lipton, C.Z, Kale,C. D ,Elkan C, Wetzel, R. (2015). Learning to diagnose with LSTM recurrent neural network. eprint arXiv:1511.03677.

[29] Xiong, R, Nicols, P.E, Shen,Y. (2015).Deep learning Stock Volatility with Google Domestic trends. eprint arXiv:1512.04916

[30] Shi, H., Xu, Minghao and Li, R. (2017). Deep learning for household load Forecasting. A novel pooling deep RNN. IEEE Transactions on Smart Grid. PP(99). 1-1.

[31] Gensler, A., Henze, J., Sick, B. Raabe,N. (2016).Deep learning for solar power forecasting. An approach- using autoencoder and LSTM Neural Networks. Systems, Man, and Cybernetics (SMC).

[32] Simon Hartley. (2011). Peak performance every time.

[33] T.Chai, R.R.Draxler. (2014).Root mean square error or mean absolute error?Arguments against avoiding RMSE in the literature. GeoScientific Model Development. 1247-1250.

[34] Sak,H., Andrew,S., Beaufays,F. (2014). Long Short-Term Memory Based Recurrent Neural Network Architectures For Large Scale Acoustic Modeling. Fifteenth Annual Conference of the International Speech Communication Association.

[35] Jozefowicz,R., Zaremba, W., and Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'15), Francis Bach and David Blei (Eds.), Vol. 37. JMLR.org 2342-2350.

[36] Martin, S., M., Schlüter,R., and Ney, H. (2012). LSTM Neural Networks For Language Modeling. Interspeech.

[37] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28 (10), pp. 2222-2232.

[38] Larry, D. M., A study of the measurement of potential Football ability in high school players.

[39] Kannekens R, Elferink-Gemser MT, Visscher C.,Post .J .W(2009). Self-Assessed Tactical Skills in Elite Youth Soccer Players: A Longitudinal Study. 109(2), 459-472.

[40] Kevin Wheeler. Predicting NBA player performance.

[41] Freifeld, L. (2012). Assessing Human Potential To Predict Future Success. TrainingMag. https://trainingmag.com/content/assessing-human-potential-predict-future-success

[42] Bergman, R.L., Wangby, M. (2014). The person-oriented approach: A short theoretical and practical guide. 2(1), pp. 29-49.

[43] Olah, C. (2015). Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[44] Stein, M. R. (2007). Benchmarking default prediction models: pitfalls and remedies in model validation. Journal of Risk Model Validation. 1(1). Pp. 77-113.

[45] Cheng, H., Garrick, D. J., & Fernando, R. L. (2017). Efficient strategies for leave-one-out cross validation for genomic best linear unbiased prediction. *Journal of Animal Science and Biotechnology*. *8*(38)

[46] Bramley, J.W. (2006). The relationship between strength, power and speed measures and playing ability in premier level competition rugby forwards. MasterThesis. Queensland University of Technology, Australia.

[47] Chen, G. (2018). A gentle tutorial of Recurrent Neural Network with Error Backpropagation.

[48] Pedregosa *et al.* (2011). Scikit-learn: Machine Learning in Python. JMLR 12,pp.2825-2830

[49]Abadi,M., Barham,P., Chen,J., Chen,Z., Davis,A., Dean,J.,Devin,M., Ghemawat,S.,Irving,. G., Isard,M., Kudlur,M., Levenberg, J., Monga,R., Moore,S., Murray,G.D., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X. (2016). TensorFlow: A system for large scale machine learning.