Athens University of Economics and Business



School of Information Sciences and Technology
Department of Informatics
Athens, Greece

Master Thesis
in
Computer Science

Evaluating Actions in Sports Analytics using Deep Learning

Dimitrios Klagkos

Committee: Professor Vana Kalogeraki (Supervisor)

Professor Androutsopoulos Ioannis

Professor Dimitrios Gunopulos

November 2020



Dimitrios Klagkos

Evaluating Actions in Sports Analytics using Deep Learning

November 2020

Supervisor: Prof. Vana Kalogeraki

Athens University of Economics and Business

School of Information Sciences and Technology

Department of Informatics

Mobile Multimedia Laboratory

Athens, Greece



Abstract

Over the recent decades, Sports Analytics has developed into an intriguing and challenging task in the research community. Sports is a way for people to pass their time pleasantly and relieve their stress and anxiety, often caused by their everyday routines. One important component that all sports events have in common is the uncertainty that the sports match provide and cause excitement and curiosity to those who observe it. From a scientific point of view, the researchers want to quantify this uncertainty through models and probabilities in order to predict and explain events. So, Sports Analytics is a field with many interesting aspects to investigate such as injuries, evaluation of actions and predictions of outcomes of a sport match. In this paper, we address the problem of the evaluation of actions of players in a football match. We propose models based on Deep Learning, including Fully Convolutional Neural Networks, Long Short Term Models and the combination of them using preprocessing steps and regularization techniques to make more robust predictions for the unseen data. Our experimental evaluation illustrates the working and benefits of our approach.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Περίληψη

Κατά τη διάρκεια των δεκαετιών το Sports Analytics υπήρξε ένα ενδιαφέρον και δύσκολο έργο μεταξύ των ερευνητών. Φυσικά, ο αθλητισμός είναι ένας τρόπος για τους ανθρώπους να περνούν ευχάριστα το χρόνο τους και να ανακουφίζουν το στρες και το άγχος τους που προκαλούνται από την καθημερινή τους ρουτίνα. Ένα σημαντικό συστατικό, που έχουν όλοι κοινό είναι η αβεβαιότητα που παρέχει ένας αθλητικός αγώνας και τους προκαλεί τον ενθουσιασμό και την περιέργεια για όσους το παρακολουθούν. Από επιστημονική άποψη, οι ερευνητές θέλουν να ποσοτικοποιήσουν αυτήν την αβεβαιότητα μέσω μοντέλων και πιθανοτήτων προκειμένου να προβλέψουν και να εξηγήσουν γεγονότα. Έτσι, το Sports Analytics είναι ένα πεδίο με πολλά ενδιαφέροντα πράγματα για έρευνα όπως τραυματισμούς, αξιολόγηση δράσεων και προβλέψεις για τα αποτελέσματα των έναν αθλητικό αγώνα. Σε αυτό το άρθρο, αντιμετωπίζουμε το πρόβλημα της αξιολόγησης των ενεργειών των παικτών στο ποδόσφαιρο. Προτείνουμε μοντέλα βασισμένα στη Βαθιά Μάθηση, όπως Πλήρως Convolutional Neural Networks, Long Short Term Models και ο συνδυασμός τους χρησιμοποιώντας βήματα προεπεξεργασίας και τεχνικές κανονικοποίησης για να κάνουμε πιο ισχυρές προβλέψεις για τα καινούργια δεδομένα.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Acknowledgements

I would like to thank my parents and my brother for the support they have shown me during the whole procedure of my master in Computer Science in Athens university of Economic and Business. In addition, i would like to express my gratitude to my supervisor professor Vana Kalogeraki who guided during my research and supported me through her suggestions, motivation and enthusiasm for the subject of the research. I would like to thank all the professors who helped me to learn and use new techniques and knowledge for the processing of data. I would also like to thank all my classmates and especially Euthimis and Alexandros for their support and companion during our time in the master.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Contents

Abstract v

Acknowledgements vii

1 Introduction 1
- 1.1 Research Context . . . 1
- 1.2 Thesis Structure . . . 4

2 Background and Related Work 5
- 2.1 Optical Tracking Data . . . 5
- 2.2 On-Ball Stream Event Data . . . 6

3 System Design 9
- 3.1 Data Description . . . 9
- 3.2 Challenges . . . 11
  - 3.2.1 Missing and wrong values . . . 11
  - 3.2.2 Imbalance Data . . . 13
- 3.3 Cleaning Data . . . 14
- 3.4 Variables Encoding . . . 14
  - 3.4.1 One-Hot-Encoding . . . 14
  - 3.4.2 Entity Embedding Layer . . . 15
- 3.5 Scaling . . . 15
- 3.6 Splitting . . . 16
  - 3.6.1 Sliding Window Method . . . 16
- 3.7 Target Variable . . . 18

4 Problem Description and Metrics 19
- 4.1 Our Problem . . . 19
- 4.2 Construction of the Labels . . . 21
- 4.3 Classification Problem . . . 21
- 4.4 Activation Functions . . . 22
  - 4.4.1 Rectified Linear Unit(RELU) . . . 22
  - 4.4.2 Sigmoid . . . 22
  - 4.4.3 Tanh . . . 23
- 4.5 Objective Function . . . 23

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

4.5.1 Binary Cross Entropy . . . 23
4.6 Evaluation Metrics . . . 23
4.6.1 Brier Score . . . 23
4.6.2 Accuracy . . . 24
4.6.3 Precision . . . 25
4.6.4 Recall . . . 25
4.6.5 F1 Score . . . 25
4.6.6 Area Under the Curve(AUC) . . . 26

5 Method **27**
5.1 Long Short Term Model . . . 27
5.1.1 Attention . . . 28
5.2 Convolutional Neural Network . . . 29
5.2.1 Convolutional layer . . . 30
5.2.2 Pooling Layer . . . 31
5.2.3 Residual Connections . . . 31
5.2.4 Squeeze and Excitation . . . 32
5.3 Temporal Convolutional Neural Network . . . 32
5.3.1 Dilated Convolution . . . 33
5.4 Long Short Term Model- Fully Convolutional Neural Network . . . 33
5.5 Temporal Convolutional Neural Network- Long Short Term Model . . . 34
5.6 Regularization Techniques . . . 34
5.6.1 Early Stopping . . . 35
5.6.2 Dropout . . . 35
5.6.3 Recurrent Dropout . . . 35
5.7 Hyperparameters . . . 36
5.8 Hyperparameter Tuning . . . 36
5.8.1 Bayesian Optimization . . . 36
5.9 Reduction Technique . . . 37
5.9.1 Principal Component Analysis . . . 37

6 Evaluation **39**

7 Conclusion **49**

Bibliography **51**

List of Figures **53**

List of Tables **55**

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

1 Introduction

1.1 Research Context

One important factor that effects a team's performance and the possible outcome of a sport game is the actions that the players execute through the game. So, the evaluations of those actions could be used to assess individual players or even teams in order to measure the contribution of each player's actions and how they benefit their team. This is a very intriguing task as it could be used to analyse and scout players or even used from media to show which players are the best in accordance to certain metrics. However, on the other hand, it is also a difficult task, as with the convectional boxscore statistics such as number of passes, shots, etc. which could be used to make metrics, can not fully catch the potential or the contribution of an individual action.

In recent years, the groundbreaking rise of big data and the development of new algorithms has allowed us to gather new data like tracking data which are spatio-temporal data and which we can analyse an action at a certain time and space. This kind of data give us the chance to use information of the players with time and space aspects with machine learning techniques to make estimates of probabilities or expected real values depending on the algorithm we choose to use to make those estimates and evaluate the actions.

In our work, we will use tracking data. In NBA the sportvu data is an optical tracking dataset acquired from cameras, which have been used to evaluate actions [SPG19]. This is of course important as it may not be used to predict a match result but it contributes to the improvement of the plays of the team and the we can see the contribution of the individual players in the team. However, it is difficult to obtain this kind of data since it may be proprietary by some company or the other way to obtain them is to use Computer Vision algorithms for Object Detection, Object Recognition and Action Recognition. On the other hand, there are some public tracking data that may be used for football data analysis [Pap+19], which of course differ from sportvu data of NBA but they can be used the same way for another sport.

One thing we should mention is the difference between the optical tracking data and the tracking that we will use. The optical tracking data like the ones of sportvu that StatsPerform provide for NBA has the positions for each player in a certain time and the ball's position as well. On the other hand, the tracking data that we use, they provide us

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

only the position of the ball and the event that takes place with it. This on-ball stream event data needs to be processed with care like the optical tracking data in order to achieve our goal. With this type of data, there have been some attempts to utilize them with the use of machine learning models in order to make predictions and reach conclusions.

In our thesis, our goal is to evaluate the actions of the players in a game. This kind of problem has gained a great interest in recent years. One critical question here is how do we evaluate an action. In the current literature, researchers have proposed different ways to evaluate actions through machine learning models and the use of pre-processing techniques. For instance, optical tracking data has been used by Cervone et. al for evaluating actions in a basketball game through a markov-chain approach with stochastic processes where they try to estimate the Expected Possession Value for each action a team makes in a possession. On the other hand, in football, efforts have been made to evaluate the substitutions of the players in the game and how they effect the performance indicators. which are more valuable. They first use Random forest to find which variables are important for the outcome in an offensive and defending manner. Then, they use those variables as dependent variables for a regression model, in which the measure the importance of the substitution with a dummy variable which indicates if the substitution was offensive or defensive. A really interesting scientific problem is the evaluation of passes which are the main event in most sports as they greatly affect the goal which is to score. In [BV18] they use possession subsequences and reward passes with Expected Goal Score and they use clustering techniques with K-Nearest-Neighbor and Dynamic Time Warping. In football where there are many kinds of events in a game De Croos et. al [Dec+19] use probability classifiers in order to evaluate the actions of players through subsequences and they reward each action if it contributed to reach a goal in the next actions of the team or punish an action if it leaded the team in possession at that certain time to concede a goal in the next actions, So, they then take the difference of the two probabilities of two consecutive action of the team to evaluate the action that happened in the next time step. The [SPG19] utilized an approach of estimating the Expected Point Value of an action through subsequences in basketball with a Deep Learning method of Stacked-Long Short Term Model and the use of the embedding layer to represent the players in the court.

In our Thesis we will utilize the above approaches and especially [Dec+19] with the metrics they recommend to evaluate actions. We propose Deep Learning models which have been used in Sequential problems like Long Short Term Model [HS97] with Attention, Fully Concolutional Neural Networks [Mat+92] and the combination of them [Kar+19] and a modified InceptionTime model the use of Embedding Layer Encoding which helps to reduce the dimensions and avoid overfitting. For the training of our models we will use the public wyscout data of [Pap+19], which is on-ball stream event data of Premier League of 2017-2018 season.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

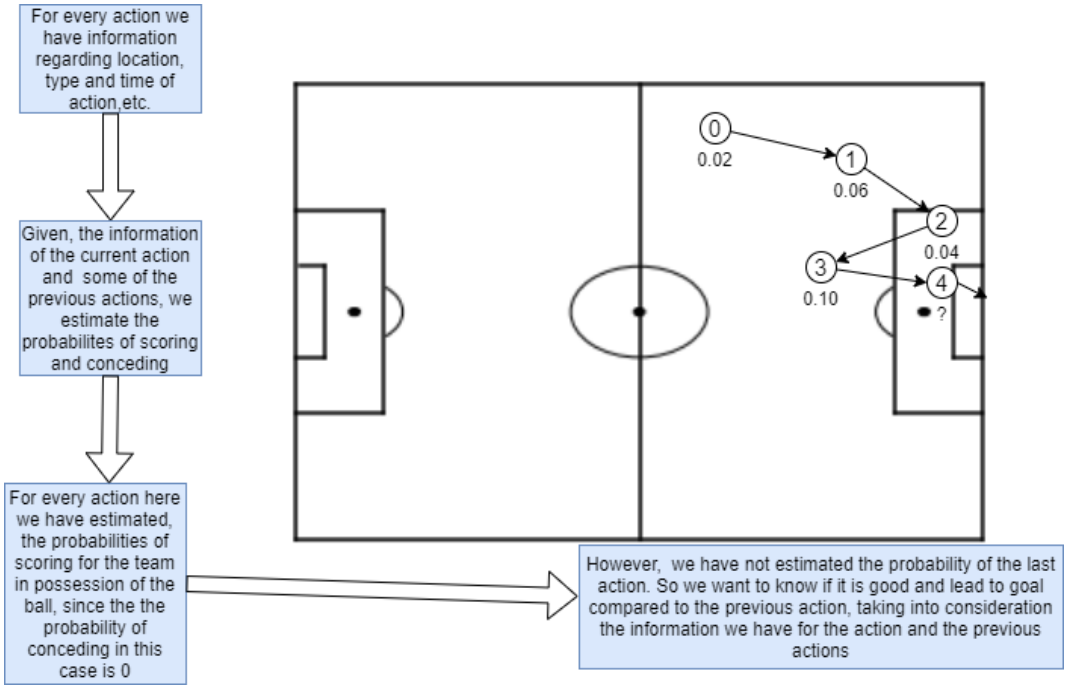


Fig. 1.1: The evaluation of actions in a attack phase of a team

For our problem, the evaluation of the actions of the players will be based on probabilistic metrics, inspired from [Dec+19] of scoring and conceding a goal in the next actions, given the information we have in the current action and in the previous ones. So as we see in figure 1.1 we have subsequence of actions. For every action we want to estimate the probability of scoring and conceding and see if our action has the potential to lead to a goal in the future actions. Here, as we are in a attacking phase of the team in possession of the ball the probability of conceding a goal for that team is 0. So, when we compute the probabilities for the first 4 actions, given the information we have for every action and the previous actions of every action, we compare the probability with the probability of the previous action to see if it contributed more for the team. The probabilities of the first actions are low as we are away from the opponent's area, but as the team reach the area of the opponent's goalkeeper we see that the actions are more important and the action before the last action has the biggest value. This indicates that the action has the potential to lead to goal for the team in possession with higher chances of achieving form the previous action. The last action is shot, but we do not know the probability, as we have to compute to see if this shot can lead to a goal with high probability. That help us to decide what action we could take, given our previous actions' information and the current one's, like spatial information, type of action and the time of the action.

1.2 Thesis Structure

In section 2 we review and analyze papers and thesis related to our work. In section 3 we describe our data, our problems' challenges and our preprocessing steps. In section 4 we define our problem and the metrics. In section 5 we describe our methods. In section 6 we evaluate our models and see the results through the metrics. In section 7 we reach a conclusion about our work.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

2 Background and Related Work

As we have mentioned there are a few works in the literature which focus on tracking data and specifically to the evaluation of actions. We will divide them depending on the type of the data.

2.1 Optical Tracking Data

A good way to evaluate actions in sports was introduced by Cervone et al. [Cer+16]. The actions developed a Multi-Resolution Stochastic Process by utilizing the optical tracking data for computing the Expected Possession Value(EPV). The full model comprises three models: a mircrotransition model, a macrotransition entry model and a macrotransition exit model, which are utilized to describe the prediction of movements, the occurrence of an event and the outcome of the macrotransition (shot,rebound,turnover,etc) accordingly. These processes which describe the movement and the state of the possession at a certain time depend on each other and so the models themselves depend on each other. The final model which has all the information has a likelihood of all the models and the estimations are made through semi-Bayesian approach with prior probabilities. The authors mention that the likelihoods for the macrotransition models are those of Poisson Regression and Logistic Regression, as they use log-linear hazard model for modeling the time of the next event and the type of event given the location of the players and the ball. In the Logistic Regression every prediction depends on what kind of Coarsed state and end state event we have. In a Coarsed state of a Possession,a player has the ball and we have a transition from possession state to transition state, which means that an event took place. An end event state can be shot which ended the possession. The Authors performed a complex model which can catch the continuous and discrete changes within a game. In football Lucey et al. [Luc+14] proposed an Expected Goal Value (EGV) metric which is computed with an ordinal Logistic Regression. They use as input phases of the game before a shot (pass,corner,cross,dribbling) which are for the offensive part of the team but also the strategic features such as defender proximity, interaction of surrounding players and speed of play coupled with shot location.

Apart from conventional statistical and machine learning models, deep learning has been used in recent years. Sicilia et al. [SPG19] proposed a new framework named Deephoops to evaluate actions. First they try to predict the next action based on the previous actions

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

and the location of all the players in the court. Then they use empirically scoring expected points which are the points for each terminal action per possession. The Deephoops framework consists of a stacked Long Short Term model with input the positions of all ten players in the court, the position of the ball and the time of an event took place. They also use an embedding layer to represent which players are in the court. Like we said,the goal is to predict the next action of the team in possession. Then, after they have obtained the estimated probabilities for each terminal action, they compute the expected point for each terminal action. For every terminal action we compute the sum of scored points for possessions which conclude with the terminal action divided by the total number of the possessions which conclude with the terminal action. The difference between two actions in different time steps gives the expected value gain for the last action of the two. They also focus on the imbalance with a specific scheme and in each window they try to reduce the imbalance of the distribution of the labels. The calibration of the predictions of the actions is also mentioned as they want to have realistic predictions of the probabilities and not only the discrimination of the actions.

2.2 On-Ball Stream Event Data

In Stream Event Data we have only the On-Ball Events and not the positions of the other players, in order to predict the next action of the team and evaluate them empirically. [BV18] proposed a way to evaluate passes. They used possession sequences and trained an Expected Goal Score model with Extreme Gradient Boosting Model to evaluate shots and then use them as labels that reward an possession sequence. Then, they split those possession sequences into possession subsequences where each contain a pass in it and the next subesequence has the pass from the previous subsequence and can have another action like dribbling as well apart from the next pass that the new subsequence will contain. They utilize the Dynamic Time Warping technique along with the K-Nearest-Neighbor technique in order to form clusters of the positions of the players and generally of the football pitch and assign the rewards to each cluster. The clusters that are away from the opponent's area can be used with K-Nearest-Neighbor technique and compute the mean of the two closer clusters for our cluster in interest and then compute the reward of the subsequence. They also used a rating metric to compare the players based on the predictions they made. Another work evaluated substitutions that happen in a match. [Lee20] proposed a machine learning method to evaluate substitutions. They used various variables to predict the outcome of a match with dummy variable which indicate if the substitution was offensive or defensive and the measured the importance of the variables with a Random Forest. They found which variables were the best for the offensive and which were the best for the defensive performance of the team. They use regression models to predict the previous best found variables and evaluate through statistical hypothesis

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

testing whether the dummy variable of offensive or defensive substitution effects the variables.

The work that is most close to our approach is that of De Croos et al. [Dec+19] that proposed metrics to evaluate actions in a football game based on probabilities of scoring and conceding a goal. They use predictions of probabilities and that we mentioned before and they subtract with the probabilities of the previous actions of team. This way they can measure the contribution of the current action by making a comparison of the probabilities of the current action and that of the previous past action of the team. They create a metric with the sum of the subtracted probabilities of scoring and the subtracted probabilities of conceding a goal. Those actions must be consecutive in respect to the time they occurred. One main difference from our approach is that they use machine learning techniques like Catboost, Extreme Gradient Boosting, Logistic regression, Support Vector Machine, whereas we use deep learning models and we utilize a different encoding technique. This technique is called Embedding Layer, which can be used to see which players are in the event on the ball. That encoding technique helps us to decrease the cardinality of the categorical variable and use new variables which come from the weight matrix of the neural network like we will see in the next section 3. The authors also use the ratings of the players based on the metrics we mentioned before.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

3 System Design

3.1 Data Description

In our approach we utilize Spatio-Temporal data. Since we can not use the whole game in one time window we will use a subsequence of actions from a possession of a team. This way we can catch the information of events or location of the players that happen through a possession. In particular, in basketball games like in NBA the team that is on the offensive usually do their attacks and plays in mere seconds and not all 24 seconds of the possession, so they may have changed their attack and have made a good micro-action like a pass or a screen. In football, on the other hand, there may be more events since there is not a time constraint here, regarding the possession and we have a lot of more passes, fouls and a bigger stadium. So, the small changes that happen in a possession of a a team may determine the the course of actions that the team will make in the its next attempt

With each of the players we associate the following set of attributes. These will be utilized to compute the player's rating value based on the prediction of the probabilities of the actions for each player and event.

- Playerid: The first variable is the unique id for each player so we can distinguish the player that makes the action.
- Player Name: The name of the player
- Teamid: The id of the team of the player
- Minutes: The minutes that a player played in match.

The data that we will utilize for our research in the part of the construction of the model consists of nine attributes. In other words, these variables will be the input of our models. We have four categorical variables and nine numeric variables. Of course, for the representation of the categorical variables we will use some encoding techniques like embedding layer and one-hot encoding depending on how many categories each variable has. Therefore, the events of the football data which are the the inputs of the model, have the following characteristics:

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

- Playerid: The first variable is the unique id we have for every player so we can distinguish the player that make the action.

- Half Time: We have the half time which the event takes place and as we know in football we have two Half Times of 45 minutes.

- Time: The time that the event takes place on the pitch. It is measured in minutes.

- Actions: Here we have the type of event that happens and the player involved tries for his team's success. In this characteristic we have 23 categories which we will show and describe in the matrix 1 below:

- Xstart: This variable indicates one of the two cordinates which show the position of the event that takes place when it first happens and it is measured in percentiles for the team which is in the possession of the ball and the goal of 100% is the opponent's goalkeeper's area.

- Ystart: This variable indicates one of the coordinates which show the position of the event that takes place when it first happens and it is measured in percentiles for the team which is in the possession of the ball and the goal of 100% is the right corner of the opponent's goalkeeper's area.

- Xend: This variable indicates one of the two coordinates which show the position of the event that takes place when the ball reach its target and it is measured in percentiles for the team which is in the possession of the ball and the goal of 100% is the opponent's goalkeeper's area.

- Yend:This variable indicates one of the coordinates which show the position of the event that takes place when the ball reaches its target and is measured in percentiles for the team which is in possession of the ball and the goal of 100% is the right corner of the opponent's goalkeeper's area.

- Result: Whether the action of the player which tries to be in the possession of the ball is successful or not.

- Speed of play: The distance of the end position from the start position of the action.

- Directionx: If the player gave the ball forward or moved the ball back from the start location of the action.

- Directiony: If the ball is directed left or right from the start location of the action.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

- Distance from opponent post: The distance of the action from the opponent's goalkepper post.

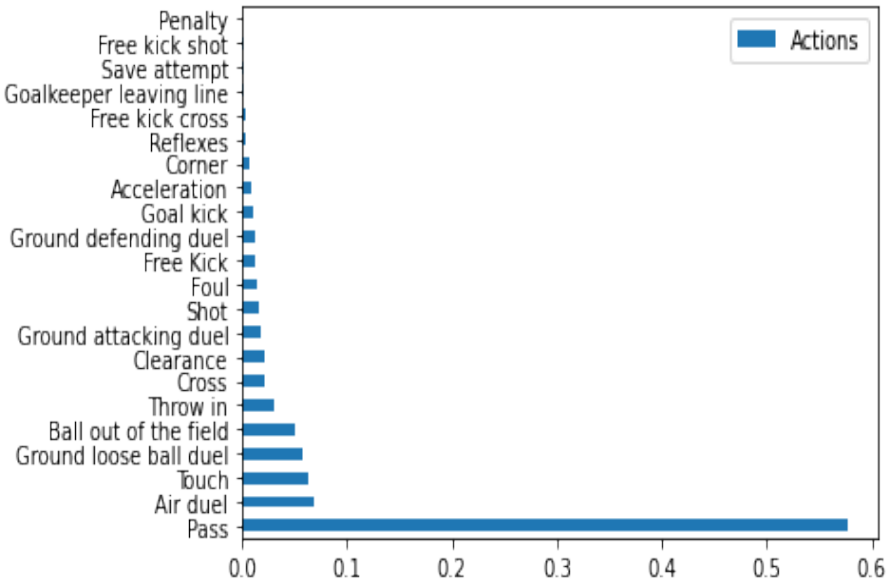


Fig. 3.1: Barplot of the Actions.

In the above figure we plot the distribution of these events. As the figure demonstrates, the most frequent event is the pass; this is expected as it is the game's key event in terms of the teams' tactics to reach their opponent or maintain the ball in their possession. One of the most important actions is the shot, as it its result which may be a goal or not is what determines a team success at the end of the game. However, if we take into account its frequency, we can see that it is not easy to create many chances of shooting. Apart from that,the chances of scoring a goal are even fewer as the shots can not always hit the target and be successful, since they may not win the goalkeeper or are inaccurate.

3.2 Challenges

3.2.1 Missing and wrong values

The data that we want to process may contain some actions that do not have the information we need, as some actions like the Ground attacking duels and Ground defending duels have some of-the ball actions which will confuse our model later. So, we drop those actions and we keep the on-ball actions for the player that is attacking and the player that is defending. The shots of the teams due to their format will not help us, as their end locations are not right for our problem. We, change their values to have as end locations to the target which is the opponent's goalkeeper post if the shot is accurate or near it if it is inaccurate. If actions contain missing values, we can drop the action or give a value based

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Actions	
Action	Description
Pass	The player gives the ball to another player through many ways, long or short passes, etc.
Ground Attacking Duel	When the player has the possession of the ball and tries to pass through the defenders.
Ground Defending Duel	When the defender is playing against the attacking player and he is trying to stop his attacking run or dribbling.
Air Duel	When the player jumps and tries to gain the possession of the ball in the air.
Touch	When the player touch the ball, even if it is to deflect it.
Ground Loose Ball Duel	When no one is the possession of the ball and the players are fighting to gain the possession for their team.
Ball Out Of The Field	When the ball goes out of the pith.
Throw In	When the player throws the ball in for their team.
Cross	When the player tries a cross to find his teammates in the opponent area from the side.
Clearance	When the player tries to play safe and throw the ball away so they avoid any kind of mistake.
Shot	When the player makes a attempt to shot and score.
Foul	When a player commits a foul.
Free kick	When a player tries to pass or cross after the foul his team won.
Goal kick	When the goalkeeper tries to pass or cross after a foul or a possession gained after a out of field ball.
Acceleration	When the player tries to increase his agility and speed to move the ball or pass a player.
Corner	When a player do a cross or a pass from the right or left corner.
Reflexes	When the goalkeeper tries to react to short shot or something unexpected shot.
Free kick shot	When a player attempts to shot from a foul the team gained from the opposing team.
Save attempt	When the goalkeeper tries to save the shot.
Goalkeeper leaving the line	When the goalkeeper tries to catch the ball or deflect it.
Penalty	When a player tries to shot a penalty.

Tab. 3.1: The actions of a player

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

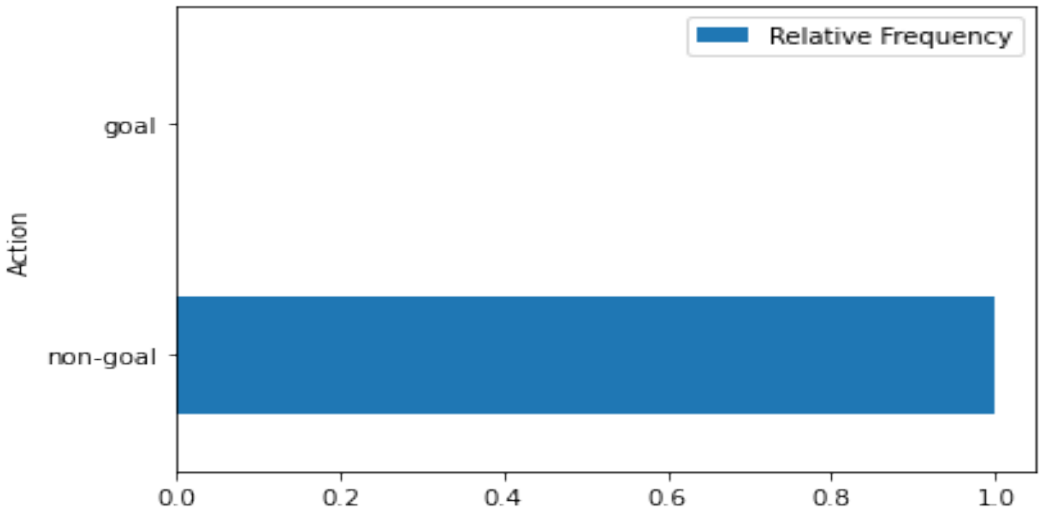


Fig. 3.2: The barplot of our goals

on the information from previous actions we can take the median or if it is a location do interpolation.

3.2.2 Imbalance Data

When dealing with large datasets, there are many problems with data imbalances regarding the distribution of their labels and their input variables. This may cause problems, as the model we choose will be trained and predict the majority class in most cases, so it will most likely overfit. For instance, if we have 20% positive labels and 80% negative labels, then the model will learn to predict the negative ones more often. In real-life problems, like the sports data that we consider in this work and specifically spatio-temporal football data, this is something very common. As we can see in the the barplot in Figure 2 below the relative frequency of the goals is extremely low with the 0.17% of the goals and 99.83% of non-goals. The difference of course is so great that we understand that every model we try to make, must be build with great care so it does not overfit. So, even if we have later actions which we reward for leading to a goal the label as we will see in section 4, will still be really low.

So, we reach the conclusion that our data set is extremely imbalanced and the probabilities to find the right action which will result in a goal is very low. In Section 4 we will present our problem and we will construct our labels later, we may increase the number of positives as we will reward or punish the actions that happened before a goal. However, this increase in frequency will be again low as we will have to choose an appropriate window size for our sub-sequence of events which will be the input of our model.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

There are techniques which could help to limit the problem of our imbalanced data or even eliminate this imbalance in the distribution of the labels but the nature of our data do not allow the use of most of them. So, oversampling and undersampling[ZL05] are out of the question, since they will cause our data distribution to be altered and the time and spatial aspect of our data will not be the same. One of the techniques which could be utilized is the penalized form of our loss function through the use of classification weights. These weights will give the class we want more importance. On the other hand, this depends greatly on the window size we will choose as it is an important hyperparameter which effects the model performance and the final predictions.

3.3 Cleaning Data

We check the data if they have some missing values in our variables and we drop some data samples.Some actions are recorded two times as we have some actions like the duels which may have more players who are not in the ball and that is some additional information which may confuse our model and like we have mentioned it may not lead to good results. Every action is on-ball event so some actions which contain more players than one or two, can cause the model to regard actions in which players who do not have the ball, as different on-ball actions. That would be misleading.

3.4 Variables Encoding

One of the challenges of our dataset is that there are 4 categorical and 9 numeric variables. So, we have numeric variables with nine dimensions which has a space of real values and we denote as: $X^T \in R^5$. As for the 4 categorical variables we have to decide a way to represent them, as their space is not determined before we utilize some encoding techniques, in order to concatenate them with the numeric ones. So we can then proceed to preprocess each of them and implement our model with all the information we need to train it and validate it and then to evaluate it.

3.4.1 One-Hot-Encoding

One way to represent our data is to make a binary variable for each category with 0 or 1 as values. The value 1 indicates that the category is used as the input and the value 0 that it is not used. It is a very good strategy if we want our model to have interpretation and know which category is used every time. However, if we have many categories, then we will have to create many binary variables, which will cause our model to overfit and not have a good predictive ability for unseen data. Another issue is that the categories in this form

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

are independent from each other and thus we can not see any correlation between them. Nevertheless, it is a good way to implement the model when it has a few dimensions.

3.4.2 Entity Embedding Layer

Another technique that we choose to use for encoding our categorical variables is the Entity Embedding Layer[GB16]. Although the term Embedding Layer is frequently utilized in the field of Natural Language Processing with great success, in recent years it can be also used for categorical encoding as well. Well, it is a technique that can be implemented through Neural Networks. We can utilize the integer transforming of each class and create a One-Hot-Encoding multidimentional variable $X \in 0, 1$ and then use a Wight Matrix whose each row represent one category of our categorical variable. That problem can be notated as:

$$X -> WX$$

where $W \in R^{dx|categorical|}$ is a Weight Matrix, d its row dimension and |categorical| the number of categories of a categorical variable and $X \in R^{|categorical|x1}$ the One-Hot-Encoding Vector.

However, the dimensions of the W Matrix are the length of the categories as rows but the columns are a hyperparameter which needs configuration. So, we have as input each row of the Weight Matrix for each category. This way we have succeeded to map the categories of the categorical variable into Euclidean Space Vector with less dimensions. As a result, we may avoid overfitting, since we will have less variables to represent each category. In addition, we can see how the variables are correlated in this space.

3.5 Scaling

When, we have all our variables representations decided, we can then use preprocessing techniques in our data, in order to to have a smooth training of our model and better results. The range of values of our numeric data can cause the weights to increase tremendously and our model would not be able to to train smoothly and make robust predictions. That is known as the Exploding Gradient problem. One way to tackle this kind of problem is to scale the data. We achieve that through Standardization [ACL00]. It is a technique often used in statistics and machine learning. In this technique we subtract the mean from the variable and divide by its standard deviation and this way we have for every numeric variable a mean of 0 and standard deviation of 1.:

$$\frac{X - M}{S}$$

Distribution of Goals	
Set	Relative Frequency
Training Set	0.17%
Validation Set	0.18%
Test Set	0.18%

Tab. 3.2: Distribution of the Sets

$$\text{where } M = \frac{\sum_{i=1}^{N} x_i}{N} \text{ and } S = \sqrt{\frac{\sum_{i=1}^{N} (xi - M)^2}{N}}$$

3.6 Splitting

Of course, before we apply the above technique we have to split the data into training, validation, and test sets. This splitting will allow us to train our model, validate it as we have to configure the hyperparamters like window size, embedding size and others that we will see later and evaluate it in an independent test set, so we do not induce bias in model's predictions. The data is usually split randomly. However, in this case that would cause the distribution of our data to be altered in a great degree. That happens as we have Spatio-Temporal data, in other words the time and space attributes matter. So, for our model training we take 70% of the original training data which are the first 266 games of the Premier League, then the 15% as validation set which are the next 57 games and the last 15% which are the last games as test set. That way, we will not disturb our data temporal relationship and we will predict the unseen data with real-time future games. However, this approach creates a big question here, and that is whether the distribution of our labels is the same across all sets. That is a very important aspect in machine learning as id the distribution differ greatly, as we may have misleading results for the predictions of our unseen data, since the model will have learned to predict well on a training dataset which is different from the independent validation and test sets. So, in order to avoid this kind of problem we will check for every set the relative frequency of the positive and negative labels. In the table 3.2 above, we can observe that the relative frequencies are about the same. Now, we have the data prepared we must also transform them into subsequencies o that we can capture all the events and put them as input in the model to train.

3.6.1 Sliding Window Method

Apart from that, even in this sport it is important to see the sequence of the events like passes to evaluate the attack in short time. Then, if we want to do that we have to define a slide temporal window, in other words a temporal window with overlapping values.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

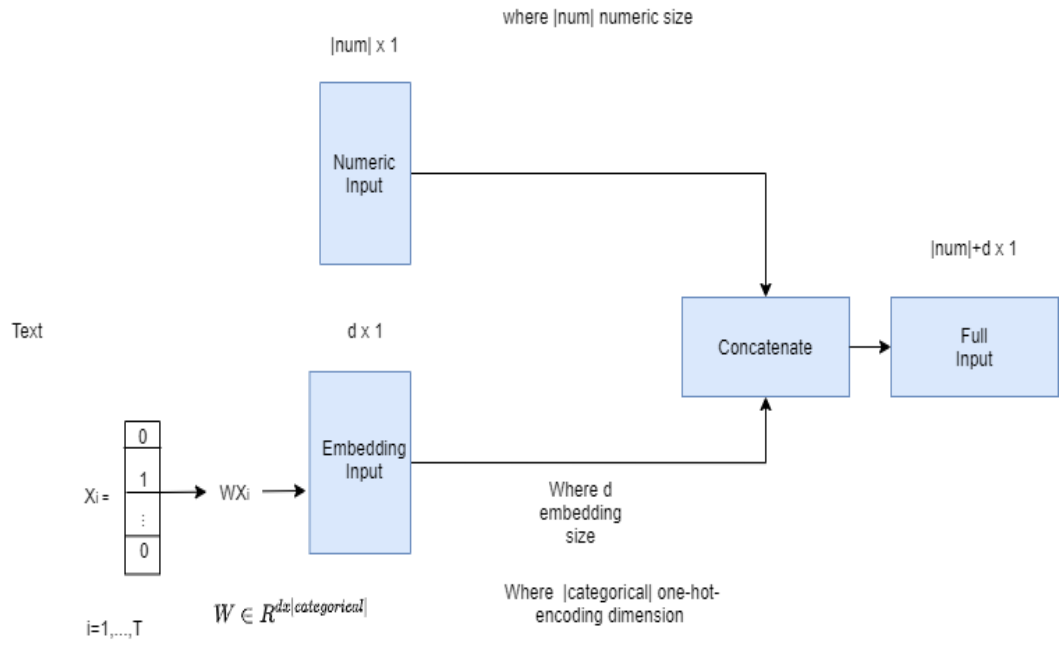


Fig. 3.3: The data's concatenation

Those values may differ depending on the sport, since in basketball we can use only the tracking data of the players and the ball to classify events and then assign expected sore values in each event and in football we may use these events as exploratory variables and as a target variable a binary variable which indicates if the attacking team scored or not in this subsequence of actions. We denote as a sliding window: $X^t = (x_{t-k}, ..., x_{t-1}, x_t)^T$, where each element it has is action with dimensions $x_{t-k}, .., x_{t-1}, x_t \in R^{|features|}$, as every action has the input features we mentioned in section 3.1.

Where k is the step defined from the window length. Every $x_t$is the row of the input matrix X^t,it also has a multidimensional input at time step t and columns the features for each action. The length of the window will be computed as a hyperparameter when we train the model and we also validate to find the best length in order to have better predictions. Then, after we have done this, we can split the data into training, validation and test data in order to fit the model and also to make good predictions with unseen data, like we have mentioned the target may differ but the model will be the same.

Now for every time step we can concatenate our data variables which are numeric and the embedding representation into the final form of our input feature before they are used for the training of the model. This procedure can be better explained with the figure 3.3 above. All the procedure is described in the figure 3.4 above.

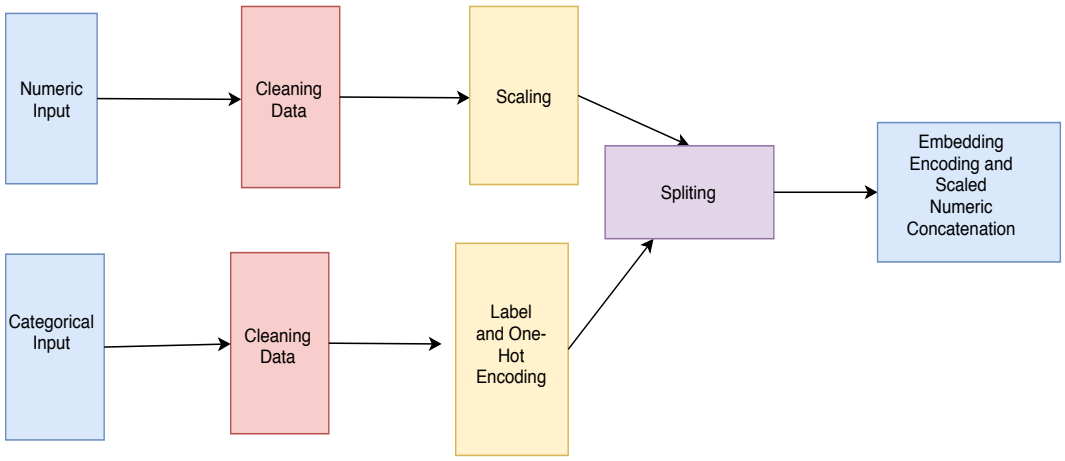


Fig. 3.4: The Preprocessing Steps Procedure

3.7 Target Variable

The preprocessing steps that we mentioned earlier are for our input variables before we train a model. However, every problem, apart from input has an output. In our case, we have the probability of scoring and conceding a goal in the next actions, given our current action's information along with the information of k previous actions. So, we want those probabilities using deep learning models like Long Short Term Model, Convolutional Neural Network and their combinations. These models are described in section 5, along with their hyperparameters and techniques which make the models to avoid overfitting and gain better results for unseen data predictions. When we have the results for every model, we will evaluate them in a test set, based on the metrics we will see in the next section 4. After that we will use the best model to compute the evaluation metric for every action, which we will see in the next section 4.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

4 Problem Description and Metrics

4.1 Our Problem

In this section we describe the objective of our approach. The goal of our work is to evaluate the actions of the players in a football match. So, the question that arises here is how we do that? There are cases where we could take actions as output variable and predict an event that will happen based only on tracking data and then assign an expected value of scoring based on the frequency of scoring after that event in the possession of the team who haw the ball. While this approach applies well in a basketball game where we have the positions of all the players on the court and the time of the possession of the ball is constrained, the shots that lead the team to score points are much more than the corresponding shots in football in which the expected goals per match may be 3 goals. Therefore, in our football data we have only on ball events, there is not a time constraint in the possession of a team and we have more events than basketball. So, the implementation of this procedure seems infeasible in our case. The approach that we will follow is set up based on the work of [Dec+19] that made a new metric of evaluations of actions for the players in a football game based on probabilities. This way is different than other conventional approaches in which they only predicted the outcome of a shot, whether it was a goal or not. In this case we want to reward the actions in a specific length of sequence of actions like we mentioned before that can lead to score. Apart from that, we do not only measure the chances of scoring a goal given the previous actions, but also the chances of conceding a goal. If we have those types probabilities, we can observe each event or action that can alter the chances of scoring or conceding a goal for each team in possession of the ball in the next k actions given the current action in time step t. As we said earlier, we have a subsequence X^t of events with all the attributes that represent them. We denote as $P_{score}(X^t, H)$ the probability of a home team H scoring in the next time step at a certain time step of the game, and as $P_{concede}(X^t, H)$ the probability of conceding a goal. The same notation can be applied for the visitor team V as with the probability of scoring in the future time steps $P_{score}(X^t, V)$ and the probability of conceding $P_{concede}(X^t, V)$.

Given that we have both probabilities we can assess a action of a player for home or visitor team with the teamid for every action and given our input matrix from t-1 step to a future t. So, we can compute the impact of each action compared to the previous action by subtracting the previous past probability from the future probability of the two consecutive events.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

$$\Delta P_{scores}(x_t, teamid) = P_{score}(X^t, teamid) - P_{score}(X^{t-1}, teamid)$$

So, if the next action has a higher chance of scoring, the metric will be positive, while if the the chance of scoring is lower then then the metric will be negative. The same metric can be computed for the probabilities of conceding a goal:

$$\Delta P_{concede}(x_t, teamid) = P_{concede}(X^t, teamid) - P_{concede}(X^{t-1}, teamid)$$

where x_t is the last value in our time window and so the current action in time step t. In this metric we have similar interpretation as it will be positive if the next action has a higher probability of the team conceding a goal or it will be negative if the probability of the next action has a lower probability of conceding a goal.

Therefore, by taking the difference of the two metrics we have the final metric which takes into account the chances of scoring and conceding a goal of a team. We subtract the metric of conceding a goal from the metric of scoring a goal. We denote the new metric as:

$$V(x_t, team) = \Delta P_{scores}(X^t, teamid) - \Delta P_{concede}(X^t, teamid)$$

Now, that we have seen the way to define our problem in terms of probabilities we can use another metric like De Croos et al. [Dec+19] thought of rating each player. The aggregation of all the probabilities of all actions and the value metric $V(x_t, team)$ for each player can be used to define a rating metric for each player:

$$Rating(player) = \frac{90}{m} \sum V(Xactionsi)$$

where $Xactionsi$ denotes the actions that a player tried on a season and m the minutes that the player played with his team the whole season.

We have to mention here, that the probability of home team scoring and the probability of a visitor team conceding a goal is the same and of course the same can be said for the other way around:

$$P_{score}(X^t, H) = P_{concede}(X^t, V)$$
$$P_{score}(X^t, V) = P_{concede}(X^t, H)$$

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

In other words, we can take into consideration only two probabilities of the team in possession whether it scored or conceded a goal.

4.2 Construction of the Labels

The labels in the majority of the existing solutions would be if the team in possession scored a goal or not. Unfortunately, such approaches will not be able to catch the potential of other actions likes passes, which also contribute to the shot. So, to identify how much each action contributes to the problem we defined in the previous subsection 4.1, we denote as $Y_1 = 1$ the possibility of scoring a goal in the next 10 actions, otherwise it is 0 and $Y_2 = 1$ the possibility of conceding a goal in the next k actions, otherwise it is 0. For our experiments we will use k=10, since we will compare our results with the results of [Dec+19]. So, this way we will consider actions that happened before a shot and contributed to the team's efforts to score. On the other hand, we punish actions that leaded for the team in possession to concede a goal.

4.3 Classification Problem

The next step of our approach is to estimate those probabilities. Since, we want to estimate a probability of rewarding or punishing actions that leaded to a goal in the next 10 time steps, we have a double classification problem.

A classification problem in statistics and machine learning is the task of identifying a label class given an input data X^t. So, we can define a function $P(y = 1/X^t) = f(X^t)$, where X^t is our multidimensional input data, f is the classifier we choose and y a binary variable with $y \in 0, 1$, $y = 1$ is to reward or punish an action and $y = 0$ is that this action leads to no goal.

There are several discrimination problems that could be used, but since we will use only deep learning algorithms to estimate the probabilities, we will focus on the neural network classifier which is the base for every method we will consider in the next chapter. A neural network is a machine learning algorithm, which consists of units named neurons and its origin comes from the biological neural networks of the human brain. These artificial neural networks act like a brain that have connections between the units. There are three types of layers, the input layer where we have the input variables, the hidden layer where each unit is a function and the output layer which may be a function as well. Every connection has a weight where we can multiply the 'signal' from the previous layer and give it as input to the next. The network tries to learn through data, but in order to train it, it must have a specific function named loss function which we want to minimize it as it

is the function which show us how much father our predictions are from the real output values. This type of error can be minimized through optimization techniques where we want to estimate the weights and have a small loss to train a model but try not to overfit it. The optimization technique that is used in neural networks is Back-Propagation with Gradient Descent which we adjust the weights to reduce the local error with the help of the gradients during learning. That procedure is happening backwards through the neural network. Though, in recent decades because the data magnitude has increased we use to train the model with batches of data, so it can converge quickly. There are some gradient descent with batches optimization techniques, that tune the learning through training and that approach we will follow for our experiments in section 6 and the algorithm is Adam[KB14].

4.4 Activation Functions

As we said earlier, the hidden units use functions of the weight matrix multiplied by the input of the previous and with that way we try to catch the relationship, usually non-linear in order to approximate the true function. Now, we will see the activation functions that we will use later.

4.4.1 Rectified Linear Unit(RELU)

Rectified Linear Unit(RELU) is a famous activation function for neural networks architectures with good properties. It is a sparse activation, in other words, usually 50% of the data are activated. It tackles in some extent the Vanishing Gradient Problem compared to other functions, it is fast and efficient in computation. Of course, it is not zero-centered and it is unbounded and some neurons may be stacked.

$$f(x) = x^{+} = max(0, x)$$

where x is the input.

4.4.2 Sigmoid

The Sigmoid function is useful function which has values that range between the values of 0 and 1. So, it is used in the output layer usually when we want to predict the probability of an outcome like in our case. However, the Sigmoid is prone to Vanishing Gradient for very high or low values of input and it is expensive regarding the computation time.

$$f(x) = \frac{1}{1 + \exp -x}$$

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

4.4.3 Tanh

The Hyperbolic Tan is like Sigmoid regarding most properties but it has values in the range of (-1,1) and it is zero-centered, so it is easier to model inputs that have strong positive or negative values. On the other hand, like Sigmoid function it has the same disadvantages of Vanishing Gradient and inefficient computation.

$$f(x) = Tanh(x) = \frac{\exp 2x - 1}{\exp 2x + 1} = \frac{\exp x - \exp -x}{\exp x + \exp -x}$$

4.5 Objective Function

The model has to be trained and the weights need to be updated based on the minimization of the objective function through the gradients. One especially good loss for a binary classification problem is Binary Cross Entropy.

4.5.1 Binary Cross Entropy

When we have a classification problem, where we have to predict probabilities and the output is discrete variable, which takes 0 or 1 for values we use Binary Cross Entropy for our model. The Binary Cross Entropy is also known by the name of negative log Likelihood of the Bernouli or Binomial distribution where we have a success or failure in a problem.

$$L(y_i, y_i^-) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log y_i^- + (1 - y_i) \log 1 - y_i^-$$

The y_i is the output label and y_i^- is the predicted probability. This loss indicates the difference between our predicted probabilities and the true distribution of probabilities of the labels.

4.6 Evaluation Metrics

4.6.1 Brier Score

In our work we chose as evaluation metric for the validation the Brier Score. This metric measures the accuracy of our probabilistic predictions, in other words, how much our predicted probabilities deviate from the groundtruth outcome. So, if we have a low Brier

Score for some prediction, we then have better calibrated predicted probabilities. The Brier Score is computed as follows:

$$BS = \frac{1}{N}\sum_{i=1}^{N}(y_i - o_i)^2$$

Where y_i is the true label outcome and o_i is the predicted probability, which takes values from 0 to 1 and the label ranges between 0 or 1. The Brier Score can be inadequate for very rare events if we train directly as an objective function. If we have many samples, however, our results can be more robust.

When we do a classification problem we have to set a threshold, which will determine which class we have every time for every sample label. If the prediction is above the threshold, then we say that the model predicted a positive class where the $y_i = 1$, if it is lower than the threshold then we say that the model predicted a negative class, where $y_i = 0$. So, we have to define what those terms positive and negative mean when we have a right prediction or a false one for each of them.

- True Positives(TP): True Positives are the cases when the actual Positive class was predicted correct.
- True Negatives(TN): True Negatives are the cases where the actual Negative class was predicted correct.
- False Positive(FP): False Positives are the cases which the actual Negative class was predicted as a Positive class.
- False Negative(FN): False Negatives are the cases which the actual Positive class was predicted as a Negative class.

Now, from those terms we can build some metrics which can tell us the how our model's performance is regarding the correct prediction of each class.

Of course, we want our model to predict every label's class correctly but it is not usually the case.

4.6.2 Accuracy

Accuracy[Pow20] is the metric where it measures the number of correct predictions for all the classes, defined as:



$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

The problem of Accuracy is that in cases of imbalanced data it can be misleading, as the majority class is predicted perfectly. For instance, if we have very few Positive labels and almost every sample is a Negative label then the model will learn to predict only the Negatives ones. So, it is not an appropriate measure for these cases.

4.6.3 Precision

Precision[Pow20] is the measure that indicates how many of the predicted Positives were actual Positives, defined as:

$$Precision = \frac{TP}{TP + FP}$$

That measure may have problems if we have an imbalanced dataset where the Positive labels are very few and the False Positives will dominate them if the model can not learn well some of the Positive ones and learn all the Negative ones. However, it is still reliable when we have used a technique to deal with the imbalanced problem, as the model will be capable of learning the Negatives ones and some of the Positive labels. So, if we do not apply a cost-sensitive or resampling technique to reduce the problem of imbalanced data and then it will have a perfect score.

4.6.4 Recall

Recall[Pow20] is the proportion of the the Positives labels that we predicted correct, defined as:

$$Recall = \frac{TP}{TP + FN}$$

The Recall is very useful as our main concern is to predict the right Positive labels, as in our case is the action that leaded to a goal.

4.6.5 F1 Score

F1 Score[Pow20] is the harmonic mean of Recall and Precision, defined as:

$$F1_{Score} = \frac{2RecallxPrecision}{Recall + Precision}$$

The F1 Score can be used in cases of imbalanced data to determine if the model's performance is good, as it take into accounts both Recall and Precision and it weights them equally. The threshold value can play a major role for the value of F1 Score and depending on the objective of the problem we can adjust it accordingly, if needed.

4.6.6 Area Under the Curve(AUC)

The Area Under the Curve(AUC)[Pow20] is a metric which comes from the Receiver Operator Characteristic curve where we have as axis the True Positive Rate, which is the same as Recall and the False Positive Rate which measures the proportion of Negatives labels that were misclassified as Positive labels. This metric measures the performance of the model for all thresholds and is unaffected from the imbalanced data problem. There have been some ways as imputation to compute the integral of this curve, with the combined performance of the True Positive Rate and False Positive Rate. This, however, sometimes can be overoptimistic, as we may have a very skewed distribution of our labels, like in our case. Nevertheless, we will still use it still indicates if the model was good in its predictions.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

5 Method

In our approach we will utilize well known models of deep learning and apply some techniques to enhance them. The models that we propose are Long Short Term Model with Attention, Fully Convolutional Network [Kar+19], Temporal Convolutional Neural Network and some combinations of them. We will compare them in accordance to the evaluation metrics we mentioned in section 4.

5.1 Long Short Term Model

An Long Short Term Model [HS97] has the architecture of combining some gates which are like hidden activation functions of sigmoid and tanh. So, the Lstm with those gates can control the flow of information and in each Lstm cell the same process takes place. Therefore, the input gate controls the new values flow, the forget gate controls the extent to which the flow of the previous information remains and the output gate is used to compute the output vector in order to then compute the hidden units the Lstm cell, in other words, the values that will go in the next timestamp. Now, we can see the Lstm cell in the below equations:

$$f_t = \sigma_g(W_f X_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma_g(W_i X_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o X_t + U_o h_{t-1} + b_o)$$

$$ci_t = \sigma_h(W_{ci} X_t + U_{ci} h_{t-1} + b_{ci})$$

$$c_t = f_t \circ c_{t-1} + i_t \circ ci_t$$

$$h_t = o_t \circ \sigma_h(c_t)$$

where

$$f_t, i_t, o_t \text{ and } ci$$

are the forget gate, the input gate, the output gate and the cell input gate. Of course, the dimensions of the weight matrices and the gates depend on the input dimension and the the dimension of hidden units.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

The above model can be stacked, meaning that the output of every cell will be an input variable for a next LSTM layer. Of course, the problem is that sometimes the LSTM forgets previous information and it can be tackled with Bidirectional Long Short Term Model [SP97], which as its name implies have two directions from the past information to the future one and the reverse flow. These models can be used for sequential data as well as for sport data which are even more complicated but they can catch the non-linear relationships between the variables. The architecture of the LSTM model we discussed above can be seen in the next figure 5.1.

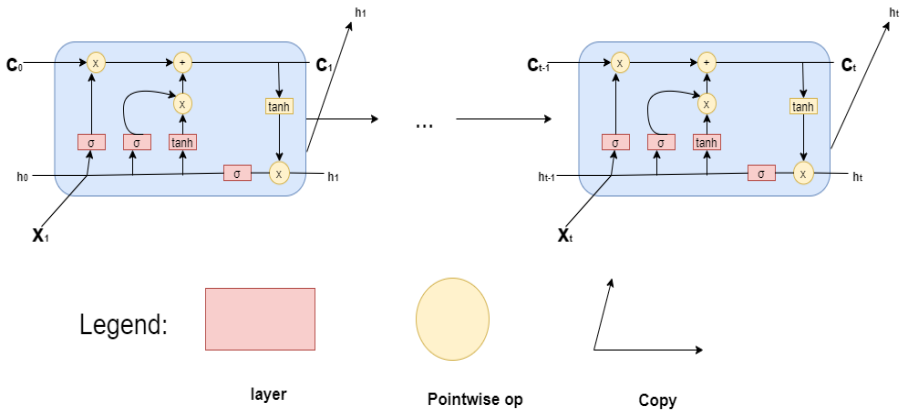


Fig. 5.1: The structure of a LSTM layer

5.1.1 Attention

In the field of Natural Language Processing there is a technique named Attention[Yan+16],[Pap+19]. Of course, this technique give the model the ability to emphasize some the positions or time steps in our case which can be more useful information for the model to process.The Attention can be really useful, as even though the LSTM can catch long relationships it does not give some time steps the importance they have for the predictions of our model. So, there are some scheme that are followed in the literature in the cases of Attention on top of LSTM layers. We use first an simple Multilayer Perceptron layer with Tanh activation function which is the same for every time step, to process the embeddings or the previous hidden units of each time step. If we add more hidden layers, then in the last one we use a simple linear combination to produce values of 1 dimension. Then we use the probabilities of a softmax function with input the previous output values of the last layer, to use them as weights for the initial embeddings or hidden units of the time steps.

Let h_t be the hidden units of the the concatenation of embeddings of categorical variables and the continuous variables of an action at a time step t of the sequence length M. Then we can describe what we said above with the following equations:

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

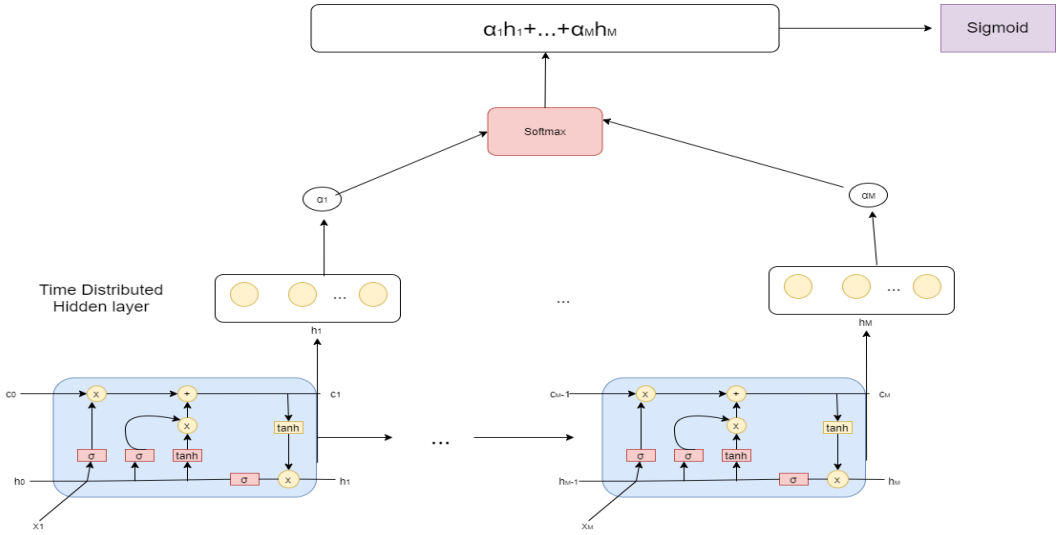


Fig. 5.2: Attention with one Time distributed hidden layer

$$a_t^{(1)} = tanh(W^{(1)} h_t + b^{(1)})$$

$$\dots$$

$$a_t^{f-1} = tanh(W^{(f-1)} a_t^{f-2} + b^{(f-1)})$$

$$a_t^{(f)} = W^{(f)} u_t + b^{(f)}$$

$$a_t = softmax(a_1^{(f)}, a_2^{(f)}, ..., a_M^{(f)})$$

$$s = \sum_{t=1}^{M} a_t h_t$$

where $W^{(1)}, .., W^{(f-1)} \in R^{dwxdw}$ and $b^{(1)}, ..., b^{(f-1)} \in R^{dw}$, $a_t^{(1)}, ..., a_t^{(f-1)} \in R^{dw}$, $a_t^{(f)}, a_t \in R$ and $W^{(f)} \in R^{1xdw}$,$b^{(f)} \in R$, dw is the dimension of h_t. An example of attention with one layer is shown in figure 5.2.

5.2 Convolutional Neural Network

Another model which we will use is Convolutional Neural Network[Mat+92]. Convolutional Neural Networks are currently one of the most prominent model for deep learning with image data. However, they could also be used with sequential data which have one less dimension. A Convolutional neural network is composed of three types of layers, a Convolutional layer, a Pooling layer and dense layer which is a simple layer like the ones that a MultiPercepton neural network we saw in section 4.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

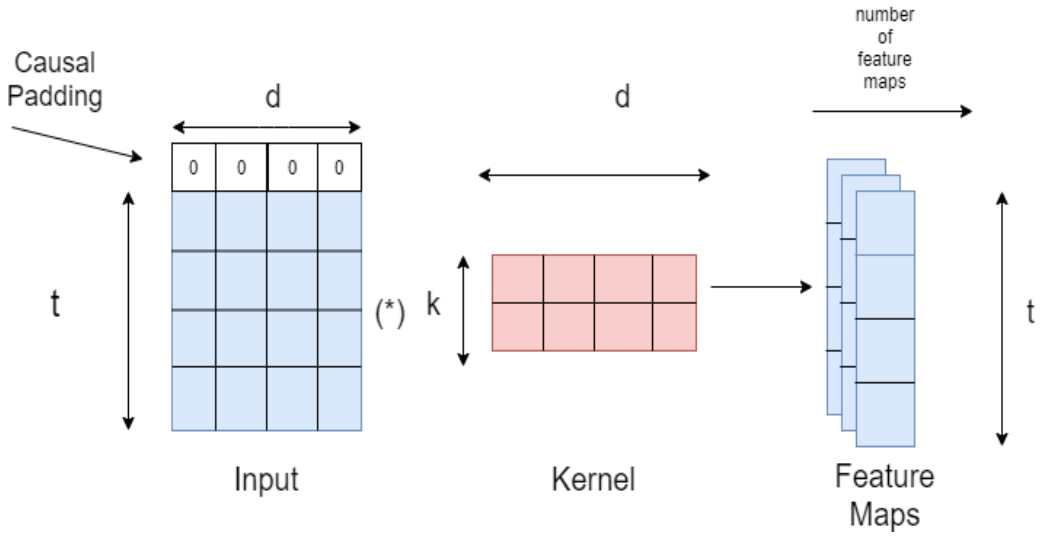


Fig. 5.3: Convolution of an Input and a Kernel

5.2.1 Convolutional layer

In a Convolutional layer the basic operation is the convolution between the input of the layer and a kernel matrix or filter which contains learnable weights. Through the convolution of the input and a different number of kernels we can obtain feature maps which contain spatial information from the input. This happens, as each neuron a feature map is connected to a region of neighbouring neurons in previous layer. Then, the resulted convolution is passed through an element-wise non-linear activation function. Next, we can see the convolution type for 1 dimension:

$$(X[n] * K[n]) = \sum_{i=-\infty}^{\infty} X[i]K[n-i]$$

where X can be an input and K a kernel.

The kernel that is applied on the whole input is weight-shared. That is important, as with that way we have less weights to train and it is more computationally efficient. The convolution usually reduces the dimensions. So, we will use causal padding witch adds zero rows before the input, since we want to maintain our dimensions to use the residual connections we will see in subsection 5.2.3. The Convolution of an input with a kernel for 1 dimension is depicted in the figure 5.3.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

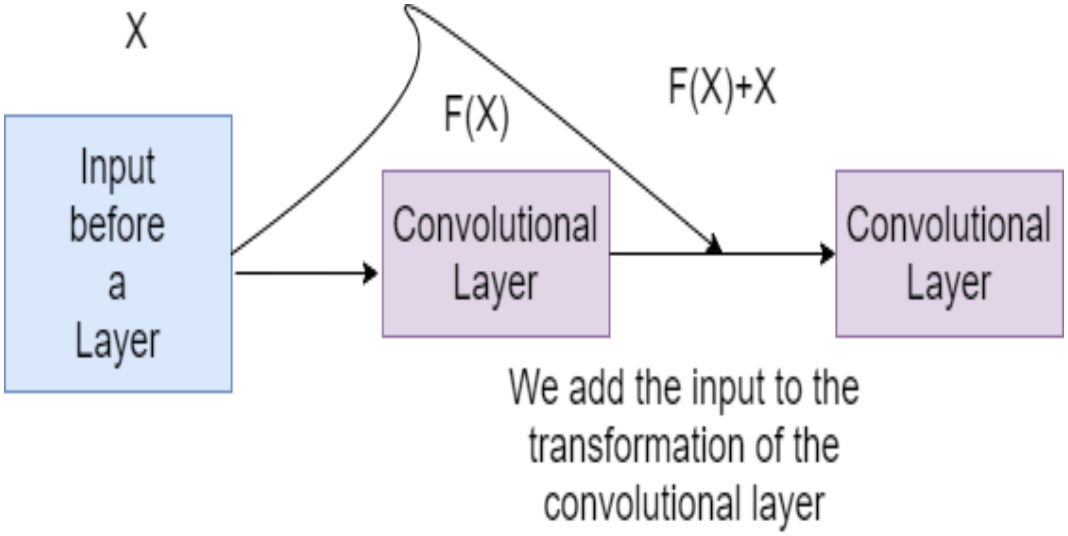


Fig. 5.4: Residual connection for a convolutional layer

5.2.2 Pooling Layer

A Pooling Layer is a layer where we try to reduce the dimensions of the previous output of the Convolutional Layer and retain the most important information. The most popular Pooling techniques are Max-Pooling and Average Pooling, which reduce the dimensions of a matrix or vector by taking the max or average value in a certain region of every feature map with a specified number of dimensions of Pooling which will be applied in the matrix vector. However, in this case we will use Global Average Pooling which will reduce a matrix of txv to 1xv by taking the average of the rows which are time steps and indicate each of them an action.

5.2.3 Residual Connections

Neural networks nowadays can be very deep regarding their depth. Thus, the problem of Vanishing Gradient arises. [He+16] made an important in the neural networks architectures. They proposed Residuals neural networks for computer vision problems, where we can add skip connections from an input of an layer and then add it to a latter input where the initial data has been mapped.This way we keep important information intact. In our case where the dimensions are the same as we use padding and we will use the same number of filters, it can be applied.

$$y = F(x) + x$$

where F is a mapping of input x through an activation function where the input was multiplied with a weight matrix. This procedure can be seen in figure 5.4.

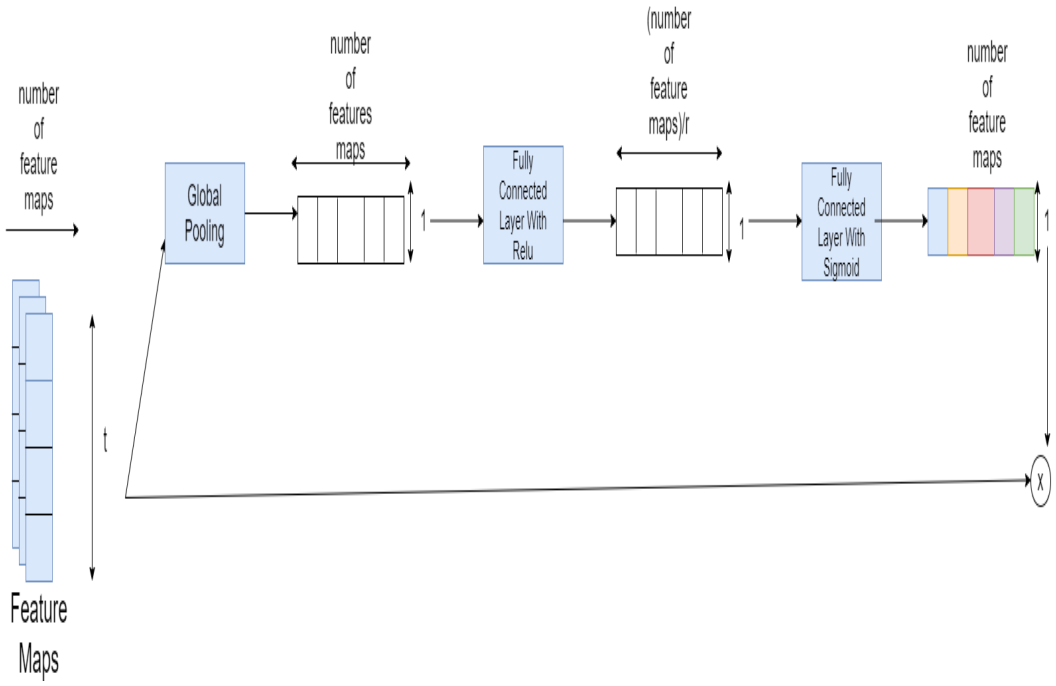


Fig. 5.5: Squeeze and Excitation of Feature Maps

5.2.4 Squeeze and Excitation

Another technique that could be used to enhance the performance of a Convolutional neural networks is Squeeze and Excitation. This is an attention mechanism which was found heuristically by [HSS18]. So, the idea of this technique is first to reduce the dimensions of an txd matrix to a vector 1xd with a Global Average Pooling, then pass it in a simple Fully Connected Layer with Relu with a less number of dimension, that is left as a hyperparameter r which divide the dimensions of the input. Then, this mapping we pass it in a hidden layer with a Sigmoid function, where the dimensions here must be the same as they were before we pass the input to the first hidden layer with relu. Now, we have 1xd dimensions we use these probabilities as weights and multiply them with the original txd feature maps before we use Global Average Pooling. That way we give more importance to specific feature maps and we see how much each of the feature contribute a useful information for the model's training. The above procedure can be seen more clearly in the figure 5.5 above.

5.3 Temporal Convolutional Neural Network

A different approach than the conventional Convolutional neural network is the Temporal Convolutional neural network. The convolution in this architecture are causal, in other words, there is no information that is leaked from future to past. The dimension of the sequence of the input layer is maintained through the layers. That is achieved with causal padding which could be used in any convolution to have the same length in very sample

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

sequence but in our case the padding creates zeros in the border of the input from the first time step. An element of a feature map occurs only from the convolution of the current and previous time steps in the previous layer with the kernel. In this model we also use the techniques we said in subsection 5.2.3 and subsection 5.2.4.

5.3.1 Dilated Convolution

The causal convolution can be difficult to implement, if we want long dependencies because it looks back with size linear in the depth of the network and filter size. So, dilated convolution has been introduced to face this problem. The dilated convolution is like normal convolution but in the kernel it has zeros between the weights and the number of zeros depend on the dilation factor. If we have dilation factor d=2, then each weight has a zero next to it and we will have different matching regarding the weights and the elements of the input matrix or vector. So, it matches more distant element from the previous layer to the weight in order to create an element of the feature map. If we have d=1, then we have the normal convolution. Therefore, we take into account elements that are far away from the current time step and when we get all the feature maps we have all the information of the input in a temporal way. where F is a mapping of input x through an activation function where the input was multiplied with a weight matrix.

5.4 Long Short Term Model- Fully Convolutional Neural Network

Another approach for modeling our problem would be to use the ensemble of an Long Short Term Model and a Fully Convolutional Neural Network, we take the two models and we concatenate them. Of course, we use the techniques that we mentioned before as well in the layers of the Fully Convolutional Network, as we can add residuals connections like we said in subsection 5.2.3 [He+16], which means we can add an initial value after some hidden layers and sum with the output. This helps avoid the vanishing gradient problem. We also use Squeeze and Excitation as we described in subsection 5.2.4 for the conlvolutional part of the model. Also we use the attention in the Long Short Term Model. Now, that we can train these models simultaneously the time to train it the whole procedure will be much more than it was individually. This model tries to take advantage the recurrent nature of the Long Short Term Model and the feature extraction of Convolutional Neural Network as well, in parallel manner.

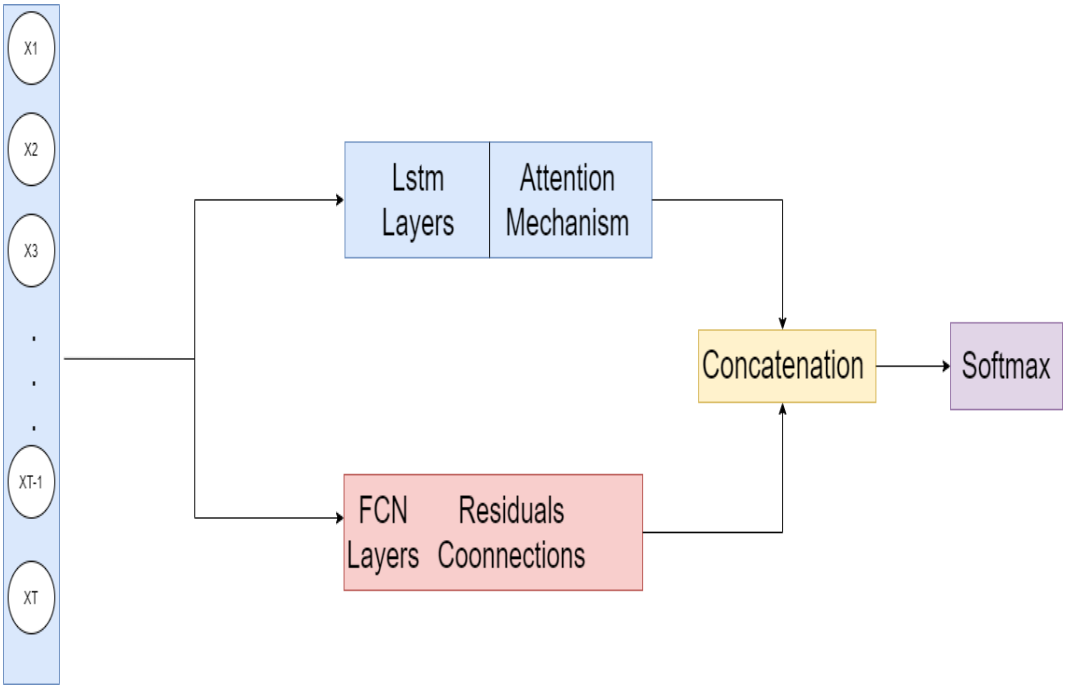


Fig. 5.6: The Architecture of the model

5.5 Temporal Convolutional Neural Network- Long Short Term Model

The last model which we will use propose for the preprocessing of our football spatio-temporal data is a different combination of a Temporal Convolutional Neural Network and a Long Short Term Model. This model will apply the two mentioned deep learning models sequentially. In other words, we will use the output of the Temporal Convolutional Neural Network as an input to Long Short Term Model. That way we utilize the benefits of both models, as with Temporal Convolutional Neural Network we get a feature extraction, which take into account the time aspect and those feature maps are new representations which will be processed as input for the Long Short Term Model in a sequential way. Of course, we will apply the techniques of residuals connections which we described in subsection 5.2.3 and squeeze and excitation which we described in subsection 5.2.4 for the Temporal Convolutional Neural Network and the attention mechanism for the Long Short Term Model. The architecture of the model is presented in the figure 5.6 below.

5.6 Regularization Techniques

In deep learning due to the number of parameters and the architecture of the models, overfitting is a usual phenomenon. So, we will use some regularization techniques in order to tackle this problem.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

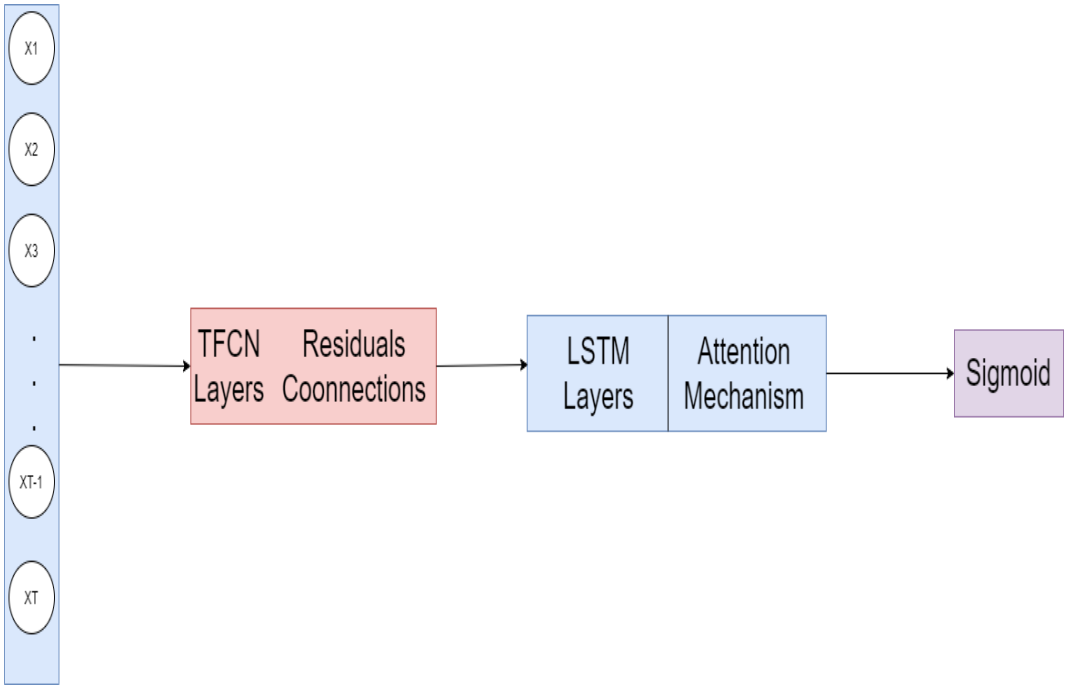


Fig. 5.7: TFCN-LSTM Model

5.6.1 Early Stopping

Early Stopping[Pre98] in which we keep track of the performance in the validation dataset by a certain metric we choose. If the performance of the model in the validation dataset does not get better in a certain number of iterations of the training phase, then the procedure will stop and we keep the weights of the epoch(iteration through all training dataset) with the best performance. For instance, if we choose a loss function then we will the minimum value, if we choose accuracy we will take the maximum value.

5.6.2 Dropout

Another good regularization technique is Dropout[Sri+14]. Dropout selects some random nodes at each iteration and deactivate them from the training process of the iteration. It samples from a Bernouli distribution based on a Drop rate and then it multiplies the units selected with zero values. So, we have an ensemble of different neural networks and Dropout prevents the network from becoming depended on the neurons.

5.6.3 Recurrent Dropout

Dropout in a Long Short Term Model may not work well, as Gal and Chaharmani(2016)[GG16] found that it still leads to overfitting, since applying standard Dropout to recurrent net-

works tends to limit the ability of the network to retain their memory. So, we apply Dropout to the recurrent connections, in order to improve the performance of the model.

5.7 Hyperparameters

Another important aspect of a machine learning algorithm and especially of deep learning neural networks is the hyperparameters. Hyperparameters are the parameters of the network that need to be configured before the training phase of the model. The values of those parameters effect the model's performance in great degree. The hyperparameters that we will use are the following:

- The size of the sliding window of our data.
- The learning rate of our optimization algorithm.
- The batch size of our model.
- The embedding size of the categorical variables.
- the drop rate of the Dropout.
- The drop rate of the Recurrent Dropout.
- The number of epochs that we will train the model.
- The dilation rate

5.8 Hyperparameter Tuning

The next step of our approach is tuning the hyperparameters in order to gain the best result for the metric we want to optimize and get the best performance for the model. We will use a well known technique which is named Bayesian Optimization.

5.8.1 Bayesian Optimization

Bayesian Optimization is a technique that is based on bayesian inference[SLA12]. We can use a subjective prior probability for our metric like the objective function and since it is a function here we will have the Gaussian Stochastic Process which will multiply

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

with the Likelihood which are the values of the metric we have for the first iterations of this technique. Each iteration is the training and validation of the model with random initialized hyperparameters values. Then the last iterations we use the values we found as data and with the Gaussian Stochastic Process we get the Posterior Probability. From it we make an utility function which we want to optimize as the number that it will occur from the optimization will be the proposed new data point of the metric by taking into consideration the previous values that they occurred randomly and gave a metric value. With that way we search the hyperparameter space sequentially as the initial values are random and then the new values that we will occur every iteration after the first iterations which produce random values of the hyperparameters will take into consideration which combination of hyperparameters gave the best result for the metric.

5.9 Reduction Technique

In section 3 we described the embedding encoding technique that we used for our categorical variables. However, if we want to interpret the results of the weights for a categorical variable we must have 2 dimensions to explain them in more detail. So, we decided to use a reduction technique, and specifically the Principal Component Analysis.

5.9.1 Principal Component Analysis

The Principal Component Analysis[JC16] is a technique which is used for representation purpose or to reduce dimension in order to tackle the problem of curse of dimensionality for numeric input variables. It produces a linear combination of the variables to create new variables that are uncorrelated to each other other. So, in our case we will use it to represent the actions in 2-dimensional euclidean space. We want to see how the model can interpret the action and relates to others actions in regards to contribution for the action quality and if those actions are from the same type.


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

6 Evaluation

For the implementation of our models for the Premier league data of the season 2017-2018 we will use the framework Tensorflow. The dataset that we will use will be split into training, validation and test sets, like we mentioned in section 3. The 70% will be the training set which are the first 266 games and the next 15% the validation which are the next 57 games and the next 15% will be the test set which are the last 57 games of the season. The split was done this way, as we did not want hurt the temporal aspect of the data. We will use the number of time steps in which a goal was scored or conceded for the team in possession to be 10. The reason we do that is to have similar results with the paper of [Dec+19] which is mentioned that this number is used in the literature.

Now we will focus on the metrics that we will use to evaluate our models in the training and validating phase, in order to select the suitable hyperparameters and to avoid the overfitting problem for each training early stopping is implemented, so we will have different number of epochs for each model. When we have the final form of the models we will use them to assess their performance in the independent test set.

In the next figure 10 we see the performance of our proposed models in accordance to the Binary Cross Entropy, Auc and Precision. We observe that the objective function Binary Cross Entropy is the same for all near 0.06, but as in all the metrics, each model has a different number of epochs that it i trained due to the Early Stopping. The Long Short Term Model (LSTM) with attention does not learn well in the beginning as its curves shows, but it gets better results after the epochs. In the Auc and the Precision metrics we see that the models are close to each other as well. We can observe that the Precision reaches near 1 for the training and validation phase as well, so our models can predict the negative label very well, in other words, the probability of an action which do not lead to a goal in the next time steps or actions.

In figure 11, where we have the other three metrics of Recall, F1 Score and Brier Score or also mentioned Mean Squared Error in this case of binary classification. The metrics here show the same behavior as the previous three for the training and validation set alike. However, we must mention that the value of 13.5% of recall where all the models reach is expected, as the number of shots that leaded to a goal is about that percent of the actions that belong to the positive labels. Well, in every model which we train the threshold is over 0.5, so the action that has such a large probability to lead to a goal are shots. The other actions like passes, ground attacking duel, crosses and acceleration have smaller

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

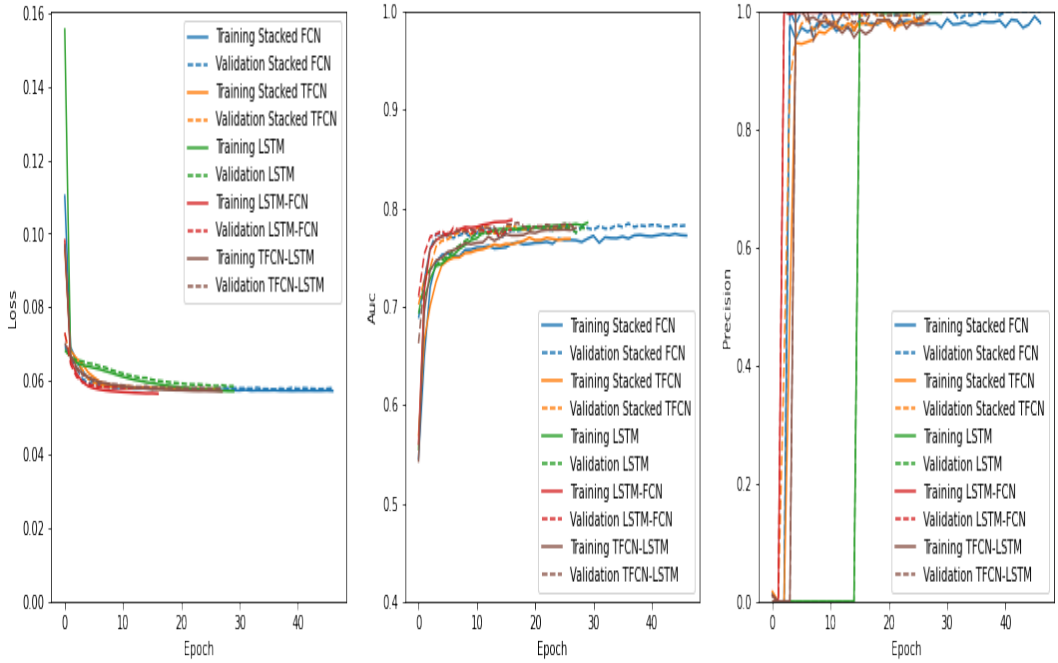


Fig. 6.1: The 3 metrics for probability of scoring and awarding the actions of team in possession

probabilities from the 0.5, as it is very difficult for the model to distinguish the quality of those actions, since there are many actions of this type in the game that they will not lead to goal in the next 10 actions. The f1 score is 0.23 for all the models, as we have a low recall and a high precision. However, in our case it is not a bad result, as for the most actions, aside from all kinds of shots that lead to a goal, we care only for the probabilities themselves and not the hard predictions. That is important, as we do not want our model to be confused and assign a high probability in action that did not contributed or did not have good potential to help the team reach a goal. That is the reason we did not use class weights to help the model learn the positive labels better, as with that way we would alter the distribution of our labels and the probabilities of some actions that do not help the team would have higher probabilities than they were supposed to have. The Brier Score very low and that is expected, since the models predict the negative labels so well. In addition, that means we have a good calibration, as our probabilities are also close to the empirically realistic probabilities of the actions. So, we can notice that our models need a few epochs to train before then stop training due to overfitting. The early stopping that we mentioned earlier is based on the Auc metric, as the others metrics show less deviation between the models. Moreover, the Auc is computed with different thresholds so even if the the distribution of the labels is highly skewed it remains a reliable method to choose for evaluation.

In figure 12 we can see the results of our models for the training and validation phase, but this time for the estimation of the probabilities of conceding a goal in the next 10 time steps or actions. Here, we can notice that the 3 metrics give similar results for all the models as

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

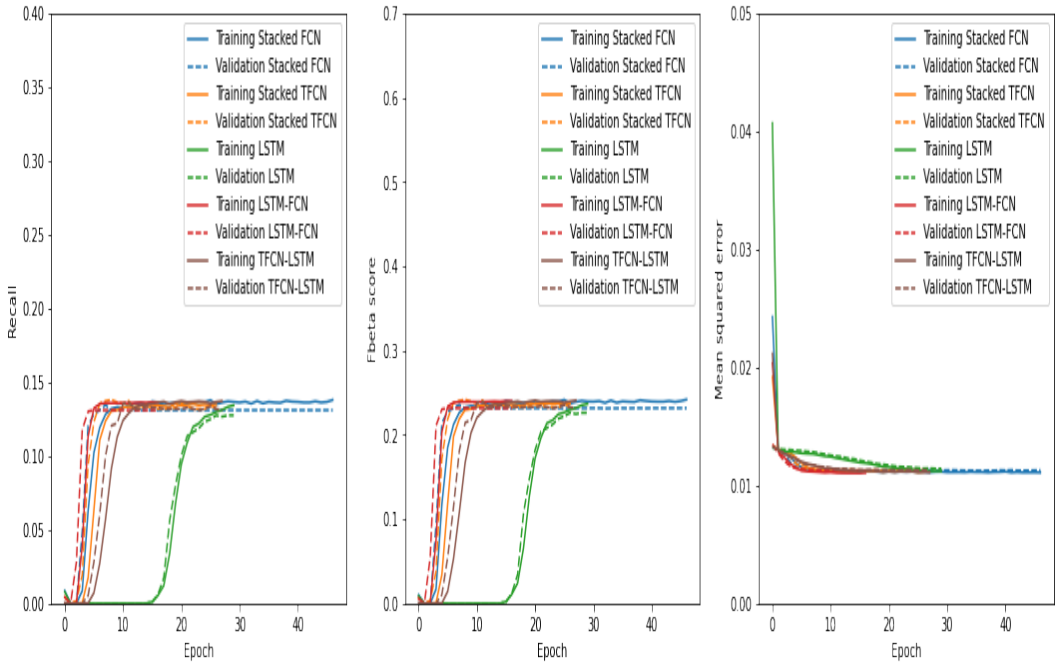


Fig. 6.2: The rest 3 metrics for probability of scoring and awarding the actions of team in possession

well. As, we expected the numbers are different, we have smaller Binary Cross Entropy loss but a lower Auc. That happens due to the models' difficulty to identify the actions that in the next 10 time steps leaded to concede a goal for the team in possession of the ball. The probabilities here are smaller as the shots here do not contribute for conceding, so we have probabilities which are lower than the threshold of the 0.5. Hence, the precision metric here is unstable as it goes from 0 to 1, but since we do not have true positive, it is natural to have a zero value.

In figure 13 we can notice, what we mentioned before for the lower values of probabilities. So, the metrics of recall and f1 score are 0. The Brier Score is smaller here again due to the distribution of the labels, where the deviation from the true values is small, as we have more actions that they do not lead to concede a goal in the next 10 time steps or actions. The models have similar values for the metrics so it is good to see first the hyperparameters that we used for each of the them and then we will evaluate them with the metrics in the independent set.

In the table 6.1 we can observe the values of hyperparameters of our scoring models. The sliding window for the data is 3 from our tuning, this confirms the claim of [Dec+19], who the have chosen 3 as well. The learning rate was good for values near 0.001. The dropout rates for the convolutional layers was found high, and especially for the Temproral Convolutional Neural Network with value of 0.7. The reccurrent dropout rate we found to be good at the value of 0.2628 for the Long Short Term Model and for the TFCN-LSTM we found it near 0.30. The batch size was configured manually, as small sizes made the

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

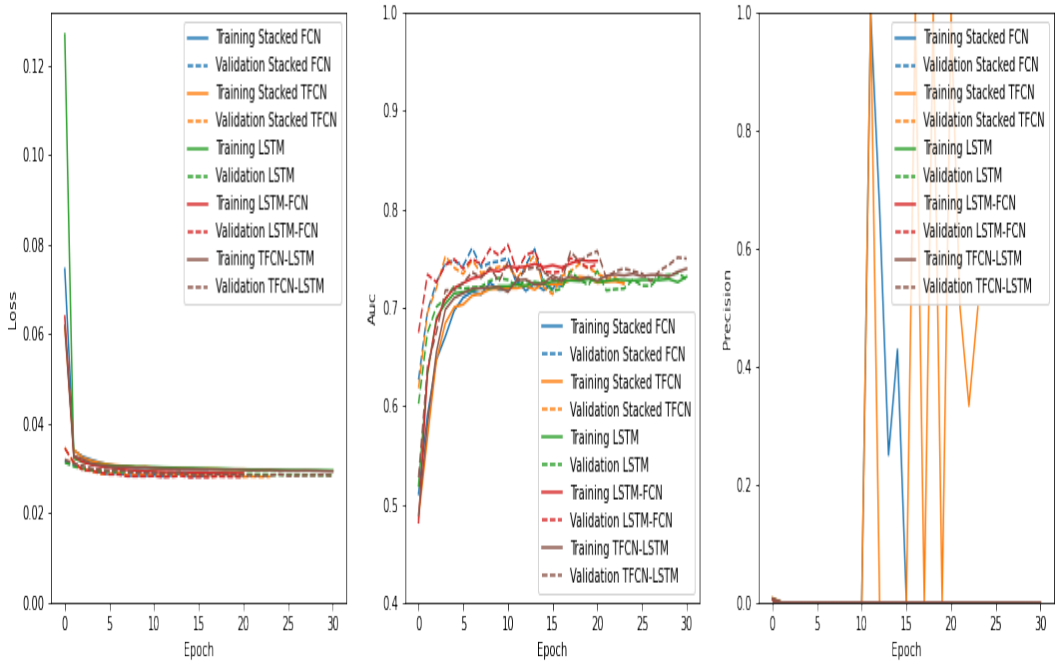


Fig. 6.3: The 3 metrics for probability of conceding a goal and the punishing the actions of the players.

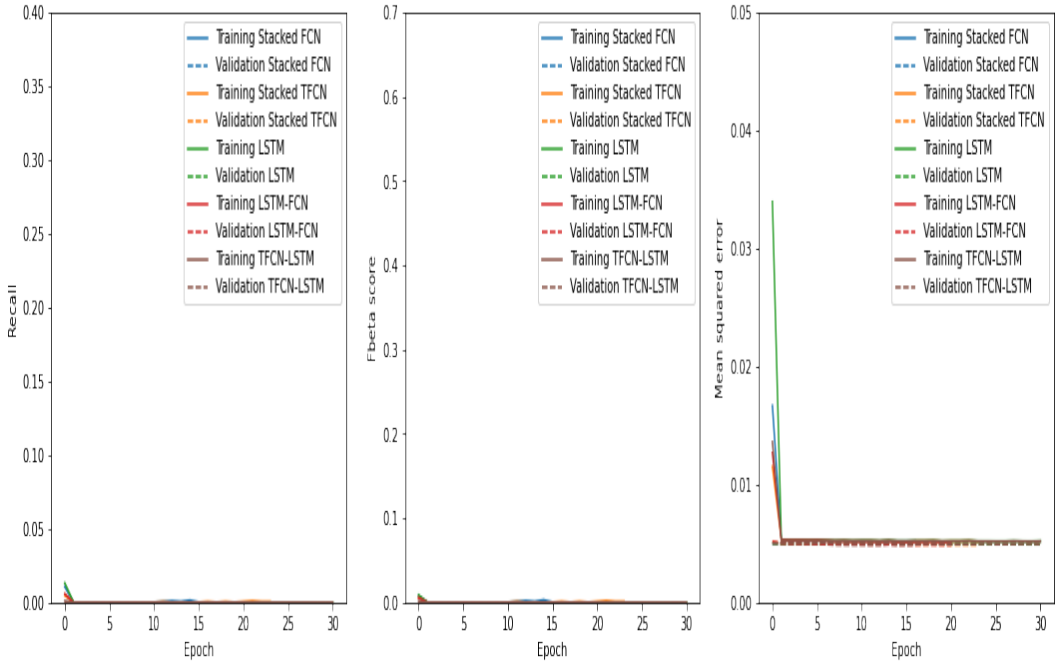


Fig. 6.4: The rest 3 metrics for probability of conceding a goal and the punishing the actions of the players.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Hyperparameters of the scoring models					
Hyperparameter	FCN	TFCN	LSTM	LSTM-FCN	TFCN-LSTM
Sliding Window	3	3	3	3	3
Learning Rate	0.001	0.001	0.001	0.001	0.0001
Dropout Rate 1	0.5	0.7	-	0.5	0.5
Dropout Rate 2	0.5	0.5	-	0.5	0.5
Recurrent Dropout	-	-	0.2628	-	0.30
Batch Size	1024	1024	1024	1024	1024
Embedding size Actions	4	4	4	4	4
Kernel Size 1	2	2	-	2	2
Kernel Size 2	2	3	-	2	3
Dilation Rate	1	2	-	1	2
LSTM Hidden Units	-	-	32	-	32
Number of Kernels for All Layers	64	32	-	32	64

Tab. 6.1: The hyperparameters of the for both the Scoring Models

models to be extremely slow and there would be a chance to lose some of the positive labels in some batches due to the imbalance problem. The Kernel sizes for the Stacked Convolutional models with 2 layers were 2 for all the models in the first layer and 3 in the second layer of Temporal Fully Convolutional Network and TFCN-LSTM. The hidden units of the LSTM were chosen to be 32 and the number of kernel for the Convolutional models were 64 for the FCN and the TFCN and for the rest three 32. With these combinations we achieved our best performance like we showed in the previous figures. However, we have to evaluate the models now in a independent test set, to see the ability of the model to make good predictions of probabilities for unseen data.

In table 6.2 we have similar results for the the models we use for conceding with small changes in the hyperparameters..

We have seen the aggregated results for the models in the training and validation phase with our 6 metrics. We will use the exact same metrics for the test set and make comparisons which model is the best, but as we have seen in the training and validation set, we do no expect much of difference between them. In this case we will also use the approach that [Dec+19] use with their representation of the data and the use of machine learning models of Exgboost and Catboost. For the implementation of this method we will utilize the package of Socceraction provided by the authors which is public in the python package manager. We can not use statistical inference here, since there is a difference in the sample dataset as we have kept the air duels in contrast to their preprocessing steps. So, we have a little different data due to the drop of air duels and the transformation of touch to other actions such as pass or dribble.

Hyperparameters of the conceding models					
Hyperparameter	FCN	TFCN	LSTM	LSTM-FCN	TFCN-LSTM
Sliding Window	3	3	3	3	3
Learning Rate	0.001	0.001	0.001	0.001	0.0001
Dropout Rate 1	0.4	0.7	-	0.5	0.5
Dropout Rate 2	0.5	0.5	-	0.5	0.5
Recurrent Dropout	-	-	0.23	-	0.35
Batch Size	1024	1024	1024	1024	1024
Embedding size Actions	4	4	4	4	4
Kernel Size 1	2	2	-	2	2
Kernel Size 2	2	3	-	2	3
Dilation Rate	1	2	-	1	2
LSTM Hidden Units	-	-	32	-	32
Number of Kernels for All Layers	64	32	-	32	32

Tab. 6.2: The hyperparameters of the for Conceding Models

The proposed deep learning models seem to outperform the approach of Decroos et al.[Dec+19] which their best models are the Xgboost and Catboost and the most notable difference with one of the metrics they use is the Auc metric but even that is in a small degree. Aside from that, we must mention that these results do not necessarily mean that the models have always better results with a similar dataset, since the statistical inference can not be applied, like we explained before, due to the different approach for the preprocessing of the dataset. The results are illustrated in the table 6.3, for every type of the probabilities of scoring and conceding a goal in the next 10 actions, as we said the differences between the models in regards to the evaluation metrics are small and we can not make a certain decision which model is the best for new data or data with similar structure fir the same problem, like a different season of the Premier League or a different League.

Now, we can use the predictions, we made for the probabilities and compute the ratings for each player for all the season, by taking into account the expected number of actions they make in 90 minutes game. We focus on the players, that have played over 900 minutes, since like in [Dec+19], we found that in the experiments some players have larger values that they are supposed to have and that is explained from the minutes they play. If they play for a few minutes, then the metric will overestimate their ability and the the expected attempts the will make in a 90 minutes match. In figure 14 we notice that most players have a mediocre rating, even if they have many actions per 90 minutes game. There are, however, players with high ratings but their expected number of actions may vary. For instance, Continho seems to have the best rating, but De Bruyne seems to be less efficient in his actions but with more attempted actions per game. The players that have made

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Test Set								
Type	Metric	FCN	TFCN	LSTM	LSTM-FCN	TFCN-LSTM	XG-BOOST	CAT-BOOST
Scoring	Loss	0.05	0.05	0.05	0.05	0.05	0.054	0.054
Scoring	Auc	0.7745	0.7725	0.7764	0.7763	0.7761	0.7650	0.7688
Scoring	Precision	0.99	1	0.99	1	1	0,99	0.99
Scoring	Recall	0.1315	0.1325	0.1328	0.1325	0.1323	0.13	0.13
Scoring	F1 Score	0.2321	0.2339	0.2341	0.2339	0.2336	0.2299	0.2299
Scoring	Mean Squared Error	0.0115	0.0114	0.0115	0.0114	0.0114	0.0129	0.0127
Conceding	Loss	0.035	0.035	0.035	0.035	0.035	0.45	0.40
Conceding	Auc	0.7597	0.77	0.7509	0.7727	0.7741	0.77	0.7720
Conceding	Precision	0	0	0	0	0	0	0
Conceding	Recall	0	0	0	0	0	0	0
Conceding	F1 Score	0	0	0	0	0	0	0
Conceding	Mean Squared Error	0.005	0.005	0.005	0.005	0.005	0.005	0.005

Tab. 6.3: The evaluation metrics for the test set

many actions but also have high ratings, are those that can contribute and be a main factor for their team's performance.

We have the results of the models, but we would like to see the way the models interpret the actions in the embedding space in 2 dimensions using Prinipal Component Analysis. We will use the Temporal Fully Convolutional Network for both the embeddings of the probabilities of scoring and conceding a goal in the next 10 actions. We observe in the figure 15 that the model for the probabilities of scoring in the 10 actions assigns the shot, penalty and free kick shot near each other. These actions give the best probabilities of scoring. The goalkeeper actions like Relflexes and save attempt are near each other and are in the left side of the figure. Those actions does not contribute for scoring, so it is natural that they are not near any other action. That indicates that the model can understand in a degree to assign similar actions together in regards of contribution and type. The other actions are more difficult to understand as they are many different actions near each other with different contribution for probability of scoring in the next 10 actions. We can see of course, actions like passes and crosses, free kick crosses and acceleration are near each other and they are similar in regard of passing to a player with different way except for acceleration which is one of the actions like the three mentioned before which can contribute greatly to score a goal in the next 10 actions.

On the other hand, in figure 16 we can see that the model is having a hard time to interpret the actions in regards to conceding a goal in the next 10 actions. We can notice that reflexes

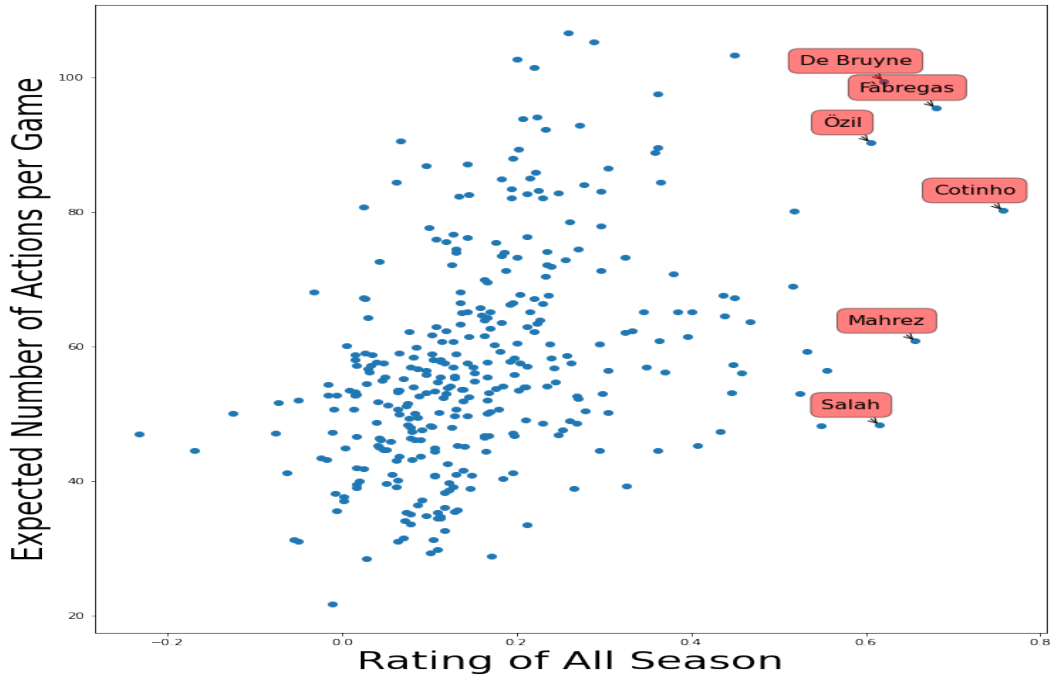


Fig. 6.5: The Ratings of the Players and the attempts the take by 90 minutes

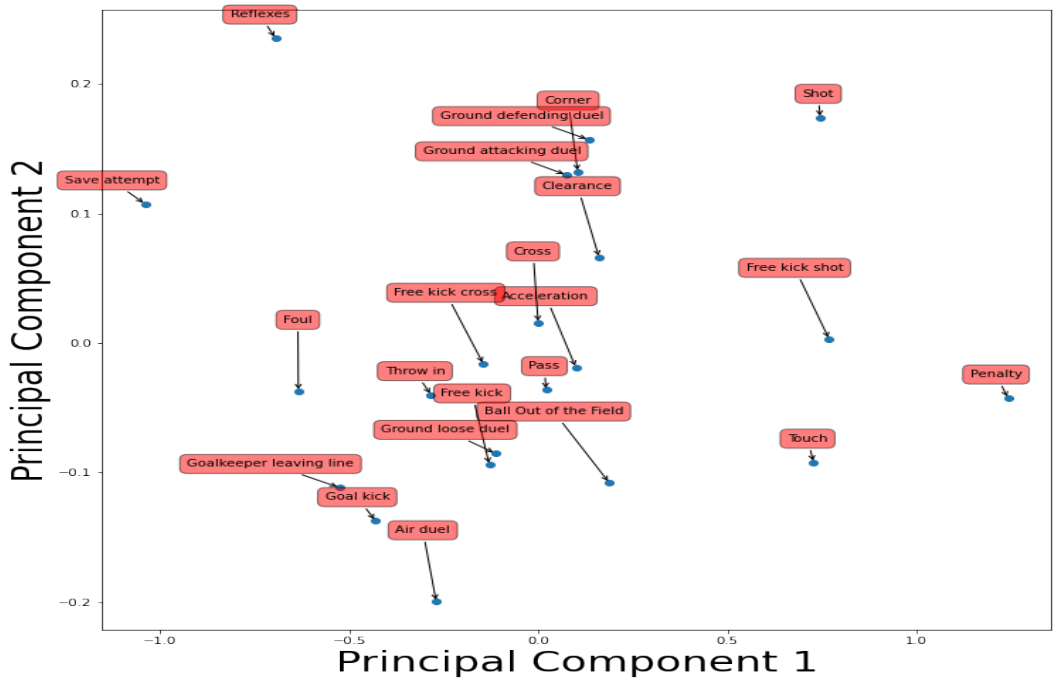


Fig. 6.6: The embedding space of actions for the model of scoring

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

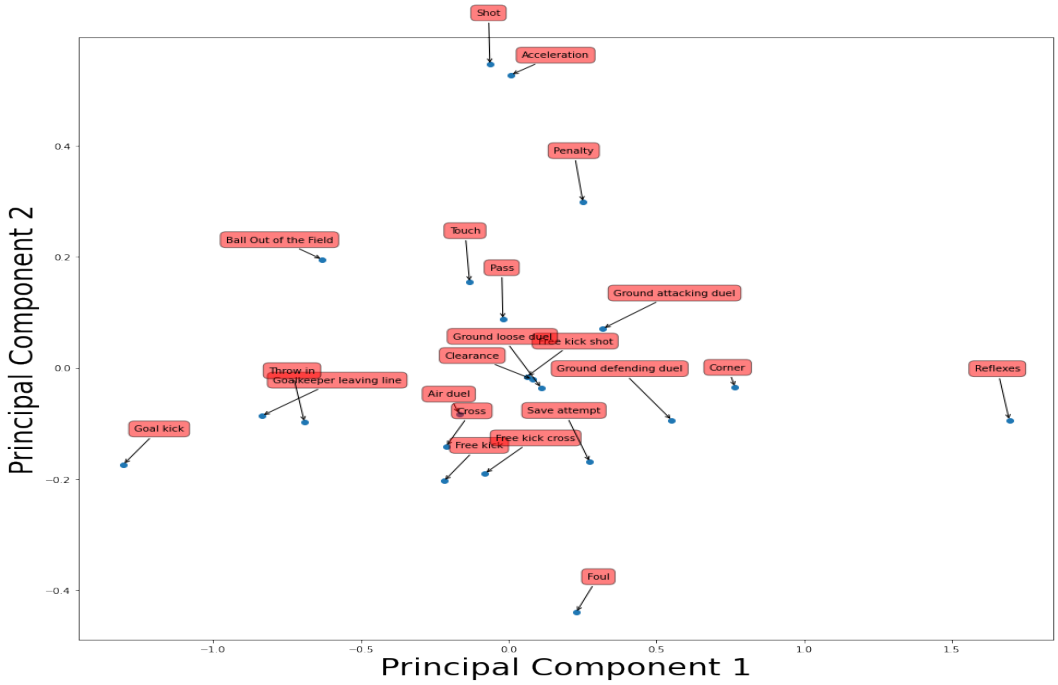


Fig. 6.7: The embedding space of actions for the model of conceding

is the one in the right side of the figure alone, ground defending duel and corner are near each other which may indicate that those actions are punished, since if the the team that was in possession in the next 10 actions concede a goal and made bad decisions to defend against the other team. The most actions are near each other, but as there can be many bad decisions for many actions that lead for to concede a goal in the next 10 actions.

It is sure interesting to explore the categorical variable in a euclidean space and see the correlations between the actions depending on the goal we have every time.

In the figure 17 we have taken an arbitrary attack phase of Manchester City with 4 actions which leaded to goal against Stoke City. For every action we have the value score since it is an attack the probability of scoring in the next 10 actions and the probability of conceding in the next 10 actions and the name of the players. We compare the results between the first epoch of the model and the epoch with the best weights in regards to early stopping. It is evident, that the model at first did not learn well to assign probabilities, especially for the shot. However, it managed to approximate the probabilities of the two previous passes before the shot. The quality of the actions here indicates that the model gives importance to actions besides the shot, but with less probability of course.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Epoch with Best Weights

	Actions	valuescore	prob_score	prob_concede
0	Pass	0.028103	0.028103	0.002117
1	Pass	0.101976	0.130079	0.002172
2	Pass	0.008047	0.138126	0.001829
3	Shot	0.710684	0.848811	0.002248

Epoch 1

	Actions	valuescore	prob_score	prob_concede
0	Pass	0.047252	0.047252	0.006040
1	Pass	0.050165	0.097417	0.008402
2	Pass	0.046469	0.143886	0.008826
3	Shot	0.060669	0.204555	0.009911

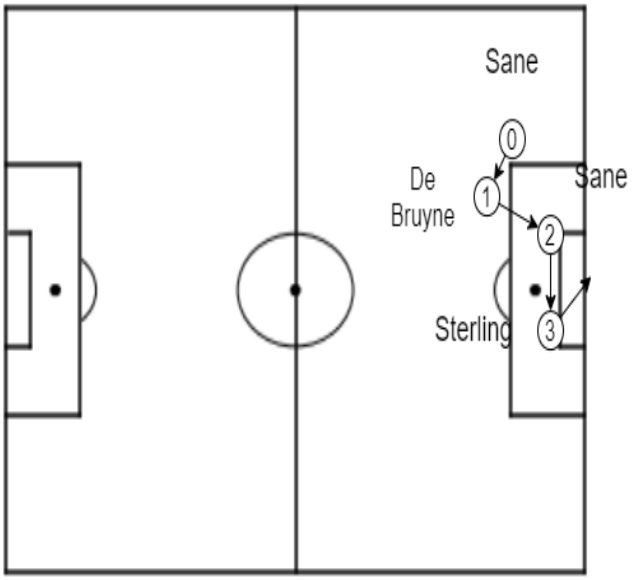


Fig. 6.8: Comparison of probabilities between first and best epoch

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

7 Conclusion

This thesis proposed deep learning models like Long Short term Model and Convolutional Neural Networks and their combinations in order to process on-ball stream data and evaluate actions in the football sports. We started with some preprocessing steps to our dataset before we use it for the evaluation of the models. We saw how the probabilities of scoring and conceding a goal in next near future actions can be used in order to evaluate an action compared to a previous action. We used techniques including embedding encoding, sliding window but also attention mechanisms to facilitate the training process of the model along with regularization techniques. We wanted to use deep learning models in this field and see how they will compare to conventional or modern machine learning models like Xgboost and Catboost. With our experiments we observed that with a great care in preprocessing steps and the architecture of the models, the deep learning models can be applied to structural data. They even outperform the machine learning models,but in a small degree and it is not certainly statistical significant. The techniques that accompany them, like embedding encoding give us the chance to explore the euclidean space of the actions and their relation. On the other hand, we have to admit that these models have many weights in contrast to the machine learning models. However, in the aspect of time the Convolutional Neural Networks which have 5 seconds per epoch need the same time with the machine learning models due to early stopping and of course the gpu which accelerate the process for the Tensorflow framework. Nevertheless, we think that deep learning models with the great advancement in the field and the big amount of data, which we have in our era, can be utilized and give good results in sports analytics field.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Bibliography

[ACL00] Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas. "Score normalization for text-independent speaker verification systems". In: *Digital Signal Processing* 10.1-3 (2000), pp. 42–54.

[BV18] Lotte Bransen and Jan Van Haaren. "Measuring football players' on-the-ball contributions from passes during games". In: *International Workshop on Machine Learning and Data Mining for Sports Analytics*. Springer. 2018, pp. 3–15.

[Cer+16] Daniel Cervone, Alex D'Amour, Luke Bornn, and Kirk Goldsberry. "A multiresolution stochastic process model for predicting basketball possession outcomes". In: *Journal of the American Statistical Association* 111.514 (2016), pp. 585–599.

[Dec+19] Tom Decroos, Lotte Bransen, Jan Van Haaren, and Jesse Davis. "Actions speak louder than goals: Valuing player actions in soccer". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 1851–1861.

[GB16] Cheng Guo and Felix Berkhahn. "Entity embeddings of categorical variables". In: *arXiv preprint arXiv:1604.06737* (2016).

[GG16] Yarin Gal and Zoubin Ghahramani. "A theoretically grounded application of dropout in recurrent neural networks". In: *Advances in neural information processing systems* 29 (2016), pp. 1019–1027.

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[HSS18] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-excitation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141.

[JC16] Ian T Jolliffe and Jorge Cadima. "Principal component analysis: a review and recent developments". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202.

[Kar+19] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. "Multivariate LSTM-FCNs for time series classification". In: *Neural Networks* 116 (2019), pp. 237–245.

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[Lee20] Quint van Leeuwen. "Master Thesis: Data science and Marketing Analytics". In: (2020).

[Luc+14] Patrick Lucey, Alina Bialkowski, Mathew Monfort, Peter Carr, and Iain Matthews. "quality vs quantity: Improved shot prediction in soccer using strategic features from spatiotemporal data". In: *Proc. 8th annual mit sloan sports analytics conference*. 2014, pp. 1–9.

[Mat+92] Ofer Matan, Christopher JC Burges, Yann LeCun, and John S Denker. "Multi-digit recognition using a space displacement neural network". In: *Advances in neural information processing systems*. 1992, pp. 488–495.

[Pap+19] Luca Pappalardo, Paolo Cintia, Alessio Rossi, et al. "A public data set of spatio-temporal match events in soccer competitions". In: *Scientific data* 6.1 (2019), pp. 1–15.

[Pow20] David MW Powers. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation". In: *arXiv preprint arXiv:2010.16061* (2020).

[Pre98] Lutz Prechelt. "Early stopping-but when?" In: *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

[SLA12] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. "Practical bayesian optimization of machine learning algorithms". In: *Advances in neural information processing systems* 25 (2012), pp. 2951–2959.

[SP97] Mike Schuster and Kuldip K Paliwal. "Bidirectional recurrent neural networks". In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[SPG19] Anthony Sicilia, Konstantinos Pelechrinis, and Kirk Goldsberry. "Deephoops: Evaluating micro-actions in basketball using deep feature representations of spatio-temporal data". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 2096–2104.

[Sri+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[Yan+16] Zichao Yang, Diyi Yang, Chris Dyer, et al. "Hierarchical attention networks for document classification". In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*. 2016, pp. 1480–1489.

[ZL05] Zhi-Hua Zhou and Xu-Ying Liu. "Training cost-sensitive neural networks with methods addressing the class imbalance problem". In: *IEEE Transactions on knowledge and data engineering* 18.1 (2005), pp. 63–77.

ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

List of Figures

1.1 The evaluation of actions in a attack phase of a team . . . 3

3.1 Barplot of the Actions. . . . 11
3.2 The barplot of our goals . . . 13
3.3 The data's concatenation . . . 17
3.4 The Preprocessing Steps Procedure . . . 18

5.1 The structure of a LSTM layer . . . 28
5.2 Attention with one Time distributed hidden layer . . . 29
5.3 Convolution of an Input and a Kernel . . . 30
5.4 Residual connection for a convolutional layer . . . 31
5.5 Squeeze and Excitation of Feature Maps . . . 32
5.6 The Architecture of the model . . . 34
5.7 TFCN-LSTM Model . . . 35

6.1 The 3 metrics for probability of scoring and awarding the actions of team in possession . . . 40
6.2 The rest 3 metrics for probability of scoring and awarding the actions of team in possession . . . 41
6.3 The 3 metrics for probability of conceding a goal and the punishing the actions of the players. . . . 42
6.4 The rest 3 metrics for probability of conceding a goal and the punishing the actions of the players. . . . 42
6.5 The Ratings of the Players and the attempts the take by 90 minutes . . . 46
6.6 The embedding space of actions for the model of scoring . . . 46
6.7 The embedding space of actions for the model of conceding . . . 47
6.8 Comparison of probabilities between first and best epoch . . . 48


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

List of Tables

3.1 The actions of a player 12
3.2 Distribution of the Sets 16

6.1 The hyperparameters of the for both the Scoring Models 43
6.2 The hyperparameters of the for Conceding Models 44
6.3 The evaluation metrics for the test set 45


ATHENS UNIVERSITY OF ECONOMICS & BUSINESS
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ