CHAPTER 2
State-of-the-Art

The analysis and study of football patterns and strategies that make a winning team is an activity that has been carried out for more than six decades. Back in the early 1950s, Charles Reep collected by hand statistics which suggested that the key to score goals was to pass the ball as quickly as possible from back to front, which indirectly led to the beginning of the "long-ball movement" in English football. [Reep and Benjamin, 1968].

However, in recent years, technology has made it possible to collect large volumes of data, allowing for a more exhaustive analysis, and thus for algorithms to be designed to extract information and learn from this data. There are three current technologies that allow the collection of the events that take place during a football match: *soccer-logs*, *video-tracking* and *GPS data* [Pappalardo et al., 2019]. *Soccer-logs* technology describes the events that occur during a match and are captured through proprietary *tagging* software. *Video-tracking data* describes the movement of the players during the games and is collected from match recordings. Finally, *GPS data* describes the players' trajectory during training sessions and is obtained from GPS tracking devices embedded in the players' equipment. This vertiginous increase in the amount of data has led to the opposite problem. Today, massive amount of data itself has become an obstacle for data analysis, due to a lack of methodological guidelines and theoretical models of tactical decision-making in football [Rein and Memmert, 2016].

Despite the wealth of data generated, these data bases are often difficult to access for scientific research [Pappalardo et al., 2019]. However, this has not prevented studies on tactical analysis or outcome prediction. Many researchers have questioned the relationship between performance and sporting success, whether in football or other sports [Yucesoy and Barabási, 2016]. The conclusion of these studies is that there is a strong relationship in sport between performance and success or failure. However, Pappalardo, for instance, talks about the fact that, while victory or defeat can be explained by performance, it is difficult to detect draws using Machine Learning techniques [Pappalardo and Cintia, 2018, Ulmer et al., 2013, Tax and Joustra, 2015].

2.1 Tactical analysis

One of the main areas of study in football is the analysis of tactics to improve the team's performance. Tactics and strategy, although they are inter-conditioned, are different concepts. Rein and Memmert define them as follows: "while the team strategy describes the decisions made before the game with respect to how the team wants to play, the tactic is the result of the ongoing interactions between the two opposing teams" [Rein and Mem-

mert, 2016].Tactics determine how a team manages space, time and individual actions in order to win the match, and are dependent on the status of the team and the opponent, and on external factors such as who is playing at home or away, the weather or the half-time break.

Automatic detection of a team's tactics and behaviour has been object of research over the last few years. Factors such as "possession directness" correlated with ball possession and passes from the defensive third of the field to the offensive third are important in identifying playing styles [Fernandez-Navarro et al., 2016]. For the study of tactics, many studies have focused on identifying team formations and the distribution of players on the field. Bush et al., for example, investigated the relationship of formation with players' physiological performance and technical skills, and found that players ran longer distances when playing in 4-2-3-1 formations compared to 4-4-2 [Bush et al., 2015]. Another approach, given by Silva et al., is based on analysing numerical superiority in a particular part of the field. This approach resulted in the control of space being a central aspect of tactics [Silva et al., 2014].

Related to this approach is the Team Centroid method, which is based on the geometric centre of a team's player positions, and is used to analyse team behaviour during key match events such as goals [Rein and Memmert, 2016]. Recently, this method has been developed further by calculating the Approximate Entropy (ApEn), a non-linear time-series measurement technique. Goncalves et al. used this method to investigate team coordination between and within defenders, midfielders and attackers. The research showed that movements were more regular with respect to the centroid of their respective subgroups compared to the other groups [Gonçalves et al., 2014]. All these works, however, focus on a very specific aspect, so at the moment it is not clear how team formations interact with individual technique or tactics [Rein and Memmert, 2016].

Two other emerging approaches in football tactics analysis are based on network approaches and Machine Learning methods. The core idea of network approaches is to model the players as nodes and the passes as vertices, with weights which represents the number of passes between them. This representation allows to easily find the key players in a team by considering their connections [Gama et al., 2014]. Wang et al. used a Bayesian latent model that was able to automatically identify tactical patterns which, combined with information about successful plays, detected the most effective tactics [Wang et al., 2015]. Machine Learning methods , such as EM, have also been used for automatic formation detection, showing for example that teams tend to use defensive formations during away matches [Bialkowski et al., 2014]. It is, however, in the field of football results prediction where Machine Learning-based methods are most widely used.

2.2 Forecasting football matches

Humans are still better at predicting the outcome of a match, as they take into consideration the technical quality of the players as well as the emotional factors that affect the final result. Generally, humans, provided they have knowledge of the match and the teams, have the ability to generalise and summarise this knowledge well enough to make their bets; however, they have a limited amount of information that they can summarise, and their predictions are often influenced by their emotions, which often leads to mistakes [Jain et al., 2021]. The accelerated progress of Machine Learning, and more specifically Deep Learning, over the last ten years has contributed to the increase in research on the

development of predictive models of football matches.

Ulmer and Fernandez conducted a study consisting of a 3-class classification problem (home-win, away-win and draw) using Naive Bayes, Hidden Markov models, SVM, Random Forest and OneVsAll SGD [Ulmer et al., 2013] with English Premier League matches. Ulmer and Fernandez say the biggest challenges they faced were the large random component of data and the large number of outliers in football, such as Leicester's 2016 title. In terms of their methodology, they first followed a feature selection process based on previous literature and intuition. They selected characteristics such as whether the team played at home or away, the ELO Ranking or the "streakness" of the last n games. Given the teams' streaks, they faced the dilemma of what to do with the first n games. The two approaches they considered were to scale the data or to ignore the first n matches, which gave better results. The optimal number of matches, n^*, was trained as a hyperparameter of the model in a range between 2 and 7. The best performing models was the OneVsAll SGD with an error of 0.48. As for the SVM, an RBF-SVM (0.52 error) was used first, but it overfitted, so a simpler model, a Linear SVM (0.49), was tried, which solved the overfitting and allowed more features to be added to the model. Naive Bayes (0.56) and Hidden Markov (0.56) were the worst performing models, in the first case due to the false assumption of sample independence and in the second case due to the model assumption that past states (the result of the matches) are hidden when, in fact, they are known.

More recently, Deep Learning models have been used to predict match outcomes. Rudrapal et al. proposed an MLP-based prediction model, and compared it with classical Machine Learning models [Rudrapal et al., 2020]. Using an MLP model with 10 hidden states, they achieved an accuracy of 73.57%, compared to 72.92% for Random Forest, 58.77% for SVM and 65.84% for Gaussian Naive Bayes. The models were trained on 11,400 Premier League matches, between the 2000-01 and 2015-16 seasons, but they performed feature selection prior to training. The input of the models consisted of features associated with each team, such as attacking, midfield and defensive ability, a team rating and an indicator of the team's streak; features associated with each player, such as their/his rating, potential, market value and indicators of goalkeeper, defence, midfield and attack; and features associated with the match context, such as average home and away points scored, performance over the last five matches and goal difference.

Ogunseye et al. attempted to predict goals scored by Manchester United between 2009 and 2018 using a MLP of 6 input units and 5 hidden layers that achieved a 73.72% accuracy [Ogunseye et al., 2019]. However, it is not only the English Premier League that has been the subject of study, other leagues such as the Dutch league have also been studied. Tax and Journal, used thirteen seasons of the Dutch Eredivise to train several classification algorithms (Naive Bayes, LogitBoost, MLP, RandomForest, CHIRP, FURIA, DTNB, Decission Tree (J48) and HyperPipes) combined with dimensionality reduction techniques. The best results were using the combination of PCA (with 15% variance) with Naive Bayes or a MLP (56% of accuracy). The models were trained with a hybrid dataset of betting odds and public data features. However, they do not use Cross-Validation to train their models, due to the temporal nature of the data, and they used the first seven seasons for training and the other six for testing. The high success with Naive Bayes, they say, may be due to the fact that the assumed dependence between features disappears with the application of PCA with low variance. [Tax and Joustra, 2015].

Another interesting study is the one conducted by Guan and Wang using a very innovative approach, a combination of grey prediction algorithm and extreme learning-

machine algorithm [Guan and Wang, 2022]. A grey-box model combines a partial theoretical structure with data and is opposed to black-box models where no theoretical model is assumed, and to white-box models which are purely theoretical. On the other hand, extreme learning machine is a single hidden-layer feed-forward neural network with a completely different learning method from traditional iterative methods. In this model the input weights are randomly generated, while the output weights are obtained by analysis and computation, thus avoiding the difficulties of non-linear optimisation of the input weights[1]. The two models are combined using a function that uses as input the output of the other model. The combination can be linear (equal weighted average, weighted based on the prediction error or based on the covariance) or non-linear (weighted geometric average or weighted harmonic average). [Guan and Wang, 2022].

Nevertheless, most of the recent publications have focused on the application of RNN and LSTM models to predict the outcome of football matches. A success rate of 80% has been achieved with LSTM models. [Jain et al., 2021]. LSTM models are RNN models that are designed to remember both the oldest and the most recent values. If the model decides that a feature is important for deciding the output, it stores that feature (the model remembers it) for a longer period of time [Jain et al., 2021]. In this work, the approach was to infer the best features from the results of previous matches, so they made use of an LSTM model trained on data from the English league between 2010 and 2018. For each match the data included the goal difference of each team and data generated from other attributes, such as the generation of goals scored by each team, the streak of each team in the last 3 and 5 matches or the point difference of each team.

A very different approach but also based on LSTM networks was done by Pettersson and Nyquist. The paper describes their study of different possible formats for network input [Nyquist and Pettersson, 2017]. They trained the LSTM model with data from 63 different countries, and the data consisted of events that occurred during matches, such as goals, fouls or penalties. To use this data as input for their model they had to transform the events so that they all had the same size. The options they propose are, in the first place, to use Deep Embedding, inspired by *word2vec*, which work very well for NLP; in the second place, One-Hot Vector, with all attributes of all events, which means that for each event there are many empty columns as all attributes except player and team identification are one-hot encoded; the last option is Concatenated Embedding Vectors for all attributes, a slight variation of the one-hot vector, as we use one embedding for each event type and one lookup for all values in the attribute vector and concatenate the resulting embedding. The format finally chosen was the latter, and each sample entered in the model is an event of a match. The input consists of a vector of ten variables. These variables are then transformed either to a one-hot encoding for each attribute and then concatenated, or to an embedding lookup for each feature and then concatenated to form a larger vector [Nyquist and Pettersson, 2017, Rahman et al., 2020].

In this paper two different approaches are also applied to calculate the error during training. In the first one, called "Many-To-One", the error is only calculated once, in the last event of the match to predict the outcome, and then used for the BP. The second, called "Many-To-Many" or sequence loss, is an average of the error over the entire sequence, calculating the error for each match event and averaging it before using it in the BP [Nyquist and Pettersson, 2017, Rahman et al., 2020]. During the training and evaluation of the model they propose seven case studies, the first six are LSTM models with different architecture (different number of hidden layers and LSTM units) but all of them use the

[1] read more about Extreme learning machine here: https://link.springer.com/article/10.1007/s11042-021-11007-7

"Many-To-Many" error, while the last case study uses "Many-To-One". In addition, in order to evaluate and validate the models, they predict the final score for each match every fifteen minutes from minute zero. Up to minute 45 the best results are obtained by a model with 1 hidden layer and 256 LSTM units using "Many-To-Many", with accuracy results of 44% (min 0), 45% (min 15), 47% (min 30) and 52% (min 45). From min 45 to the end the best model is the one using "Many-To-One" with 2 hidden layers and 256 LSTM units, with results of 63% (min 60), 74% (min 75), 88% (min 90) and 98% (at the end of the match). Following this work, Rahman conducted a study to predict the outcome of the FIFA World Cup 2018 matches using the same architecture, input format and error functions with a success rate of 63% [Rahman et al., 2020].

Other approaches have been taken to attempt to predict high-impact events during matches or the results of football matches, such as by analysing the sentiment of tweets posted during the match [Godin et al., 2014, Yu and Wang, 2015]. Also, statistical models have been developed to predict the results [Koopman and Lit, 2015], as well as hybrid models based on more than one algorithm that increase the overall accuracy of the system [Guan and Wang, 2022, Jain et al., 2021].

Another interesting question is to keep in mind what the purpose of these models is. Some of these models are evaluated by comparing them with bookmakers' odds, also with the aim of checking whether the model can compete with the bookmakers and make money. However, in order to do so, it is necessary to take into account which matches to bet on. For example, there will be odds with a threshold where, even if the model predicts a win for a team, the odds are so low that it will not be worth betting on the match at all [Bunker and Thabtah, 2019]. Models have been proposed that manage to beat the bookmakers and achieve a certain profitability [Koopman and Lit, 2015, Tax and Joustra, 2015]. Godin et al. claim that with their model they have been able to make a profit of 30% in matches in the second half of the 2013-2014 English Premier League season [Godin et al., 2014].

reference	data	model	classes	acc.
[Tax and Joustra, 2015]	13 seasons, odds + features	PCA + Naive Bayes, PCA + MLP	3	54.7
[Ogunseye et al., 2019]	Man.Utd matches 2009-2018	MLP 6 units input, 5 hidden, 2 output w/sigmoid	3	73.72
[Rahman et al., 2020]	World Cup 2018 10 embedded feats.	LSTM 2 layers, 10 & 5 units	3	63.3
[Jain et al., 2021]	2010-2018 EPL past matches features	LSTM 1 layer, 64 units	2	80
[Rudrapal et al., 2020]	11,400 EPL matches 20H-20A features: team, player and head-to-head features	MLP 1 layer, 10 units	2	73.57
[Guan and Wang, 2022]	Not specified	Grey prediction method + Extrene Learning Machine	-	over 80
[Ulmer et al., 2013]	Train: 10 EPL sns. Test: 2 EPL sns. match features and time-dependence	NB, Markov, SVM (lin. & RBF), RF and SGD (best). Underpredic. draws	3	error: 0.48
[Nyquist and Pettersson, 2017]	Embedded events of 63 leagues Note: got underpredicted draws	LSTM M-to-1 1 layers 256 units	3	min 0: 44
		LSTM M-to-M 2 layers 256 units		end: 98

Table 2.1: Results of related works

2.3 Critique

It is convenient to make a critical analysis of the current state of the scientific study of football before presenting a proposal. As mentioned above, the analysis and study of football and its patterns and strategies, although it has been carried out for several decades, only recently has it become more common and has been updated to more up-to-date Machine Learning techniques.

However, and probably as a consequence of this recent boom, there are still no precise methodological guidelines for this field of study, and each researcher, in the lack of references, must define own methodology. Related to this, the areas of study within sports analytics and sports forecasting based on Machine Learning methods are barely well-defined and most of them are of very recent creation. This is why many studies have a lack of references and are independent studies between which there is no flow of information whatsoever, which means that many studies end up covering similar topics or proposing similar solutions.

Another problem is the secrecy of a lot of projects in sports analysis. This is due to the fact that most high-level projects are carried out by private entities with strong interests in football. This is the case for football teams, which are increasingly performing advanced studies at all levels on the performance of their teams and their opponents. There are also studies, slightly more available to the public, by private entities such as the big media and producers of sports audiovisual content, but they are still quite discreet in their publications. This means that most of the latest work does not reach the academic field, so conducting research in this field is not straightforward.

Closely related to this is the lack of open-access databases. It is impossible to draw solid conclusions without data. The fact is that there are very few available open-access football databases, and those that are available contain very little data or offer little variability of information that focuses on mere general match statistics. In order to access more complex data such as match events, player positions or team formation, databases must be purchased and a considerable licence fee must be paid to access them.

There is a lack of studies applying cutting-edge Machine Learning methods, most likely due to the issues mentioned so far. As a consequence of the lack of accessible data and related previous studies, many researchers decide to apply well-known - and not very new - Machine Learning methods in order to perform well, such as Logistic Regression, SVM or Random Forests. Few papers decide to make use of techniques and methods that have never been applied to football match prediction before.

Regarding the content of the papers related to the forecasting of results, most of them focus on building models that are capable of predicting match results and evaluate them with the overall accuracy or error that they have. However, many of the papers describe how the error distribution of matches is not uniform and happens mostly in matches that end in a draw, in matches that are, a priori, evenly matched, or in matches with a very unexpected outcome. Few studies focus on these more critical and complex matches.

An additional aspect that is rarely treated in most works is the explanation of the factors that influence the result of a match. Emphasis is put on the random factor as an essential piece in the puzzle; however, not many conclusions have been taken regarding the influence of tactical or strategic factors such as the line-up, the level of competitiveness or the absence of key players in a match.

2.4 Possible solutions

In this research work, in order to achieve the proposed objective, some of the aspects previously discussed in related works will be addressed. To do so, useful ideas and results from some previous works will be used as a starting point and new methods and ideas not previously proposed will be introduced.

The solution proposed in this paper starts with the creation of a dataset from several open-access data sources. This dataset should contain as many matches as possible, and for each of them as much information as possible. The idea is that each match should contain information on different historical and strategic aspects. Based on these characteristics, an analysis and experiments would be done to evaluate which Machine Learning algorithms and which characteristics would give the best results.

When evaluating the results, the goal is to obtain good general results, but especially to improve the accuracy of the model in the most difficult matches to predict, the tied matches. To do so, the results will be evaluated by applying different appropriate metrics to measure the accuracy in multi-class classification problems. In the experiments, dataset re-balancing and weighting techniques will be applied in order to give more importance to the matches that end in a draw so that the model is able to learn better the relationships that occur in these matches; regularisation methods will also be applied so that the ML algorithms do not overfit the training data and generalise correctly when new data is provided to them. During the experiments, we will optimise the configuration parameters of the algorithms so that they perform as we want them to.

Examples of Machine Learning algorithms that we will use for prediction are weak classifiers as baseline, such as decision trees or SVM, trained with bookmakers' odds. Later, in the experiments, more sophisticated algorithms will be used for training, namely ANN, with different architectures and complexity, and the gradient boosting algorithm, LightGBM, which is widely and successfully used with tabular data.

2.5 Chosen technologies

Different tools and technologies have been used to carry out this research work. Each one has been used for a specific reason and context, some have been used throughout the whole work and others have been necessary only for a part of it. All of them, independently and together, have formed the working environment.

The work has been mainly undertaken on a personal computer, on which all the necessary software has been installed and configured. Windows has been the main Operating System I have worked on, but the experimental phase was carried out on a virtual machine provided by VRAIN running on a Linux system.

Python has been the programming language used, and it was chosen because it is currently the most used language in Data Science projects and, therefore, it has many advantages when you use it as there is more documentation and more libraries available. Along with Python, a virtual environment manager for Python, called Anaconda, has been used to create an independent environment for this project with the libraries and their respective versions, without affecting the versions of other environments in other projects.

Anaconda has made it possible, in a simple way, to use the same Python environment on both Windows and Linux. The parts of data analysis and exploration, dataset generation, proof of concept and analysis of results were carried out on Windows. For these tasks we used Jupyter Notebooks integrated with Visual Studio Code. Jupyter Notebooks is an application that allows you to implement and run code by execution cells, so it is very useful for analytical tasks or code proofs, and it also permits to take notes in Markdown and Latex language.

However, Notebooks are not the most suitable tool for running experiments, so the necessary code was implemented in a self-made Python library to have an automated pipeline in which the data is preprocessed, the training is executed and the results are generated. This pipeline is executed by receiving as input the configuration and parameters of the experiment. At the end, the different results of each configuration are generated, as well as running logs to verify that it has worked correctly and to analyse its performance. Finally an Excel file is generated to summarise the whole experiment. All this pipeline is also made to be executed in the Linux BASH terminal, passing the configuration through parameters in the terminal call or through a JSON configuration file. We have also used TMUX, a multiplexer for UNIX systems that allows a terminal to be divided into multiple independent sessions and thus be able to run several experiments in parallel on a single terminal.

Some of the Python libraries that have been used throughout the work have been NumPy, an open source library that supports the creation of vectors and multidimensional arrays as well as providing a large collection of high-level mathematical functions to operate with them; and Pandas, a library specialised in the manipulation and transformation of data in the form of tables, called Dataframes. Both libraries are generic libraries that are widely used for data analysis and manipulation, and therefore have a large documentation and support community behind them. They have been used both in the database creation stage, cleaning and transforming the original data and creating new features, as well as for data analysis and exploration and data processing before and during model training. In addition, Pandas has been used for more in-depth analysis of the results.

On the other hand, there are libraries that have been used for more specific functions. This is the case of the graphic generation libraries, Matplotlib and Plotly. These two libraries have been used to visually analyse data from the matches as well as data about the performance of the prediction models, such as the learning curve. Matplotlib has been used more for data analysis, as well as for creating the graphical content shown in this document. On the other hand, Plotly, which provides interactive graphics, has been used more for the analysis of the data in a more interactive and comfortable way.

The main Machine Learning library that has been used is Scikit-Learn, which provides a wide palette of functions for data preprocessing and scaling, which have been used during data preparation before training or for feature selection, such as with ANOVA or Variance Threshold. This library has also been used for Baseline models, as Scikit-Learn offers an API for classical ML algorithms such as SVM or Decision Trees. We have also used Scipy, a library that offers a collection of mathematical functions and algorithms similar to Matlab, which has been used mainly during the data analysis phase.

Two libraries have been used to implement the predictive models, each one specialised in a different Machine Learning algorithm. On the one hand, PyTorch has been used for the ANN-based models. This library is one of the two most widely used libraries for ANN, and it was decided to use PyTorch rather than the other, Tensorflow, because we already

had some academic experience with PyTorch. On the other hand, for predictive models based on Gradient Boosting models, the LightGBM library has been used, which implements a super-efficient version of this algorithm and has recently outperformed other versions.

Finally, for the optimisation of the hyperparameters of the ANN and GB models, we have used Optuna, a framework that automatically performs a Bayesian search for hyperparameters by simply defining and passing an objective function to be minimised or maximised.