Universiteit Leiden

Master Computer Science

Classifying sequences of football play using boosted LSTMs

Name:	Wouter Leeftink
Student ID:	s1730398
Date:	08/07/2020
Specialisation:	Data Science
1st supervisor:	Laurentius A Meerhoff
2nd supervisor:	Stephan van der Zwaard

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Wider availability of tracking data of football matches has increased the interest in predictive models in the sport of football. Predictive models can help the current state of match analysis by providing analysts with insights they would previously not have access to. A problem in the world of football analytics is that finding a success label is a balancing problem between the quality of the label and the imbalance of the data set. This thesis attempts to predict the success of football sequences using a strict success label. To achieve this a boosting model for long-short term memory networks is proposed. This network, in combination with a set of very simple features is then applied to a large dataset of football matches. It is found that using the boosting models in combination with very simple features achieves a performance that outperforms a statistical baseline. Further comparison to existing work using a more relaxed success label shows that the models in this thesis achieve similar performance to previous work.

Acknowledgements

This thesis was prepared as part of the master study in computer science under the supervision of Rens Meerhoff. I would like to thank him for the constructive comments he provided on my thesis throughout the process. Furthermore I would like to thank the KNVB for providing the data and a starting point for this research. Finally I would like to thank Jieming Ye for letting me stay in his apartment during most of the time spent working on this thesis when universities were closed.

Contents

1 Introduction

In recent years, the advancement of tracking data in the sport of football has allowed for the tracking of all players and the ball on the pitch [59]. These tracking data have largely been used to assess the activity of players and use this information to prescribe the correct training load [61]. However, these tracking data also have much potential in the field of tactical analysis [59, 29]. When using these large tracking datasets, there is a lot of room for different machine learning approaches to perform the tactical analysis that would normally be performed manually [59]. This can help the analysts of professional football teams by taking some of the manual work away.

One of the tasks machine learning could help with is the identification of important moments in a sequence. This sequence is a time period in a football game of moments that follow each other, how this sequence is extracted from the data is explained in section 3. Currently, video analysts have to manually review every game from start to finish to identify moments to show their teams. Applying machine learning to this problem could result in a model that is able to rank moments on how interesting they are for a video analyst. To do this, models need to be trained using some measure that is similar to what video analysts use now in determining what sequences are interesting. In this thesis, the measure chosen for this is the *success* of a sequence. A model that can classify these sequences well could be helpful in finding important moments in a sequence. If the probability suddenly strongly increases or decreases, this might be caused by a single action. Investigating these actions can help the analysis of games and make it easier to find impactful players or successful patterns of play.

Within these interesting moments, there may be be single actions that determine whether the sequence of play is a success. Previous work has tried to rate the actions of players by assigning a value to actions or positions during a football game [18, 30, 16]. A model that can predict the probability that a success will occur in this sequence can help with finding these interesting actions. By looking at the probability that a success will happen in the current sequence it may be possible to identify interesting actions. A strong increase of the probability over a certain time interval can be further investigated by looking at the actions that happen in this time span.

One of the biggest challenges in football-related research is finding a good success measure. This success measure is needed to be able to use machine learning methods. The number of goals in a football match is much lower than in an ice hockey or basketball game. This makes it more difficult to use goals as a success metric in machine learning algorithms trying to predict successful moments in football matches. Ideally, goals would be used as a success metric due to the aim of football matches being to score more goals than your opponent. However, the scarcity of goals leads to highly imbalanced datasets that increase the difficulty of training machine learning algorithms. Different approaches to success in football have been attempted in the past. Solutions such as getting within a certain distance of the goal [18] sacrifice some of the quality of the success measure for more successful instances. Whereas using only goals scored as a success [16] results in having very few successful examples, but of high quality. A definition that tries to find a middle ground between goals and distance to the goal is using 'dangerous' moments as a success label, which the model is trained to predict. These dangerous moments include goals, shots that miss the goal and saved attempts. The objective of this thesis is to develop a model that can accurately predict the probability that a sequence leads to a dangerous moment

The definition of success has an impact on the level of class imbalance. However, regardless of the definition of success used, some class imbalance will exist. The first goal of this thesis is to correctly classify sequences of football play. The selected method for dealing with the class imbalance and correctly predicting whether sequences of football play are successful or not is a boosting algorithm in combination with LSTMs.

Another goal of this thesis is to determine at what point in a sequence the success can be correctly classified. The reason for this second goal is that knowing at which point the model is reliable provides information on the use of the model. If a model is reliable from the beginning to the end of a sequence, it can be used to rank the state of the playing field during a football game at all points in the game. If a model is only reliable at certain points in the sequence, this can be because a single moment or a subset of moments influenced the probability of success. Moments that influence the probability of success in this way can be investigated by looking at the players that are involved in these moments, the location that these moments occur and the actions that cause these quick increases in probability.

The structure of the remainder of this thesis is as follows: section 2 outlines the work done in the field of football analysis from both a sports science and a computer science perspective. Section 3 describes the data used in this thesis and some of the difficulties caused by the data. Section 4 describes the methods used in this thesis: the baselines and metrics, as well as the final models. This section also introduces the

experiments and goes more into detail on the implementation. Section 5 describes and explains the results of the experiments. The results are explained both in statistical terms and more practical cases. Finally, section 6 provides a conclusion. This section also gives some suggestions for future work.

2 Related Work

The evaluation of performance in football has been approached both from a sports science and a computer science point of view. Sports science approaches often develop a hypothesis and test this hypothesis during carefully designed controlled experiments, typically small-sided games [55]. This is done by extracting data during these experiments, which are often small-sided games [3, 27, 48]. The data are then transformed into features that attempt to describe the performance of football teams. The features developed by these works might not always fully represent the performance of football teams. These features are an attempt to explain the performance and interaction of football teams with one number. Such a number is unlikely to fully capture all that goes on on the field [29].

Research on the game of football from a computer science perspective focuses more on developing models using existing data [31]. The computer science approach uses data collected during matches that have been played and aims to detect patterns in these data. The models developed are then evaluated by how well they can explain the performance of football teams and how likely it is that the found model performance is achieved by chance [29].

The work in this thesis will mostly follow the computer science approaches to the analysis of football using tracking data. The models in this thesis are developed using a dataset of a full season of football in the eredivisie, the highest Dutch football division. These models are then evaluated using several metrics to measure how well these models can explain the performance of football teams. From the sports science approach the models in this thesis use several features that have been developed using the sports science approach of developing a hypothesis and testing this hypothesis using controlled experiments.

2.1 Football Data

Recently, the availability of more and more data on football matches has allowed for an increased interest in the analysis of tactical performance of football teams using these data [31, 59]. The data used for these analyses can be split into two categories: tracking data [6] and event data [57]. Tracking data is often gathered by optical tracking [48]. Other approaches to collect tracking data exist, such as GPS tracking. The advantage optical tracking has over GPS tracking is that it does not require football players to wear sensors, as well as containing the position of the ball in the data. Additionally, optical tracking has been implemented in several football leagues, whereas GPS tracking has mostly been used for smaller scale experiments. Datasets collected by tracking contain the positions of all players and in some cases the ball at a frequency between 10 and 25Hz. A work that is based completely on tracking data is the work by Dick and Brefeld [18]. This work uses the unprocessed tracking data as the input of the models. The first disadvantage to this type of input is that these data are sparse. Using this type of input for many football games requires a lot of memory, whereas aggregating the state of the football pitch to some features reduces the size of the input data. The second disadvantage of using just the positions, directions and speed of all players and the ball is that it does not incorporate football theory in the models. From the sports science perspective, many features have been designed to explain the performance of teams in games [29, 43, 38, 3]. Using the unprocessed tracking data without this information removes the football theory from the models and assumes that a model would find relevant information on itself. The explainability of a model also suffers when using the tracking data as input, as this requires deep learning methods with no information as to what caused an increase or a decrease in the value of a state of the football pitch. The value of the football pitch can for example be determined by the probability that a goal will be scored in the next n moves or seconds. Using these features would help the interpretability of the models developed as the coefficients attached to these features show whether they are negatively or positively related to the value of the state of the football pitch. Concepts such as dangerousity [43], defensive disruptiveness [38] and the centroids [3] are calculated from the tracking data. Other football theoretical concepts such as what the type of an action was, for example a pass, are much more readily available in annotated event data.

The work by Decroos, van Haren and Davis [16] uses these annotated events as the input data. Their model uses the events and information related to the events, such as the players involved, the start and end time, the start and end location, the type of action and the body part with which the action was performed to predict the probability that a goal is scored in the next k seconds. They use the increase in probability as the value that

this action had and use these increases in probability to evaluate players. The advantage of using the event data over tracking data in this work is that tracking data does not contain information on the type of action and the body part with which the action was performed. These concepts are found by manually annotating the data. However, the disadvantage of this type of data in the work by Decroos et al. is that players not directly involved in the action are left out of consideration entirely. This is because these event data do not contain the information on all players in the game, but just the players that performed the actions. Players not directly involved in the action may still have had an influence on the probability that a goal is scored in the next k seconds, but in works using just event data are not incorporated in the modeling process.

Both data types have their advantages and disadvantages. Tracking data misses football theoretical concepts, some of which can be calculated using these tracking data [43, 38, 3] and others that can be extracted from annotated event data. The advantage of using a combination of tracking data and event data is that actions contained in the event data can be evaluated based on features calculated from tracking data as in the work by Kempe and Goes [38]. This thesis uses a combination of event and tracking data. The event data are used to extract and label sequences from the data as explained in section 3.4. The tracking data are used to calculate the features listed in section 3.3. This combination of tracking and event data ensures that information of all the players on the field is used during the time period determined using the event data.

2.2 Football Analysis

The analysis of football tactics [47] and performance [38] has been approached both from a computer science [18] and sports science side [25]. An area of interest in the field of football analysis is finding features that describe the performance of a team. Features such as team centroid [26], team surface area [26], team stretch index [25] and team spread [50] are team-wide spatial features that can be used to describe a team at a point in time. These features are then investigated on their ability to describe football performance. However, most of the investigation is done by looking at football games, often small-sided, in a controlled experimental setting [29]. These methods are often not compared to other methods in the literature on standardised metrics which makes it hard to compare the results achieved. The features proposed in these sports science papers can also be used to summarise the data collected by spatio-temporal tracking, thereby reducing the memory space needed to store the tracking data and use them in models.

Identifying important sequences in sports matches has also been an area with much research into it. In the context of player ratings, research has been done in identifying important moments so that players who are involved in such moments more receive a higher rating [56]. This approach uses features extracted from a human annotated dataset to rank the performance of players over the last g games. These events do not include off-the-ball actions which can be important to a player's performance. Being in the right place at the right time is something that can determine the success of a football player and this is not incorporated in this approach. Another approach is to give events a score based on whether there is a goal in the next k actions [16]. This approach also uses event data and as such does not incorporate the position of players not currently in possession of the ball. This is detrimental to the explanatory power of the model, as two passes covering the same location can vary greatly in contribution to the success of a team depending on the position of other players on the field. Some approaches attempt to rate the value of a pass by using the position of all players on the field over the course of the pass [38, 30]. The movement of players can then be used to predict the outcome of a match. This metric is only used to evaluate complete matches based on the average value for the metric of the winning and the losing team. This work shows that the defensive disruptiveness aggregated over a whole game is a good indicator for which team wins.

Approaches focusing on sequences of play instead of single actions also exist. Some of these focus on rating player position [18] by finding sequences of play that strongly increase the value of the game state. This approach uses the positions, directions and speed of all players and the ball at every frame and trains a deep reinforcement model with the goal of predicting whether a sequence is successful. This work uses two separate success definitions, coming within 25 meters of the goal and coming within 18 meters of the goal. By doing this, the work evaluates the balance between a narrower success definition, leading to a more imbalanced dataset and a broader success definition which labels more unsuccessful sequences as a success. However, neither coming within 25 meters from the goal or coming within 18 meters of the goal is a true success. Both of these definitions still include many unsuccessful sequences as the goal of football is scoring goals and not getting within a certain distance from the goal. By using this success definition the authors are able to achieve an area under the ROC curve of 0.85 on the task of classifying sequences of football play, indicating that the models can be used for the classification of football sequences. Using sequences of play can also help with highlight detection and prediction in football matches [17]. This approach uses event data and k-nearest

neighbours to find sequences close to the current sequence and use this sequence to predict whether the current play will be interesting or not. This task can be applied to the task of classifying football sequences on their success as well. However, k-nearest neighbours is known to perform poorly in high-dimensional spaces [20]. The sequences in this thesis have many calculated features meaning that the data in this thesis are high-dimensional and k-nearest neighbours might not be all too suited as a model for the data in this thesis.

2.3 Sequence Classification

The models developed in this thesis need to determine whether sequences of football play are successful. The task this thesis tries to solve can be framed as a sequence classification problem. Sequence classification [64] is the task of classifying sequential data. These data do not contain independent feature vectors but rather have their features change over time. Tasks that fall under the umbrella of sequence classification are natural language processing tasks such as sentiment classification [67], classification of time series [22] and the classification of DNA sequences [53]. The main common denominator between all these tasks is that the sequences contain a list of ordered observations. To apply that to this thesis: the observations in the data in this thesis are an ordered set of frames. Sequence classification tasks often leverage information from the previous observations to predict what comes next. In the case of this thesis, models that use the ordered set of frames will be able to extract the trajectory of the feature values over time from the data. Models that do not use this sequential information, but try to predict the outcome of a sequence by looking at a single frame such as the work by Decroos and van Haaren [16] do not use the development of the values in the sequence. Long-short term memory networks [34] (LSTMs) have achieved promising performance on various sequence classification tasks [36, 65, 63]. Predicting the class of multivariate time series can also be seen as a sequence classification task on which architectures containing LSTMs have achieved promising results [37].

In this classification task there are many more unsuccessful cases than there are successful cases, due to the choice of success label. This leads to a class imbalance. Class imbalance is a problem in classification tasks that is widely studied [44]. Because most datasets do not have a perfect balance between classes some measures often need to ensure a classifier predicts that all examples come from the majority class. Work on data imbalance is often separated into two categories: algorithm- and dataset level solutions [9]. Dataset level solutions focus on rebalancing the data before feeding these data to a classifier. Examples of this include oversampling of the minority class, undersampling of the majority class or synthetic resampling methods, such as SMOTE [10, 32]. Synthetic resampling is more difficult for the sequential case as the synthetic samples are generated by randomly changing some of the values in an example. In a sequence this causes problems because the model can not correctly capture the temporal dimension. Solutions for this have been developed using techniques such as generative adversarial networks [42]. On the algorithm level, ensemble methods such as different boosting algorithms [21, 11] have shown to improve performance of classification on imbalanced datasets [35]. Boosting algorithms have also shown to improve the performance of LSTMs on time series forecasting [4].

In this thesis, a stricter definition of success is used than in previous similar work using tracking data [18]. The goal of this thesis is to find a classifier that is able to identify successful sequences in an early stage. To achieve this goal, a stricter definition of success is used, such that sequences labeled with the success label are more likely to actually be successful sequences. However, with a restricted definition of success, the class imbalance becomes stronger. The problem that is trying to be solved in this thesis can be modeled as a sequence classification task with highly imbalanced classes where the features used in the sequence classification task are taken from the literature of previous research on football analysis.

3 Data

This section will go over the data used in this thesis. In the previous section, the difference between tracking data and event data was introduced. This section will go into further detail on the two types of data, as well as go through the steps taken to go from the data provided to the data used in the model. Finally, the features used in training the models will be described.

3.1 Tracking Data

The tracking data are collected by cameras around the pitch at 25Hz, afterwards these data are downsampled to 10Hz. These data points include the x and y coordinates of all the players and the ball. The data also

contains the acceleration and speed of all the players and the ball plus the team that the players play for and the timestamp. The advantage of these data is that it is complete. For every timestamp all the columns are filled in, making it convenient for calculations over time. Another advantage is that these data have not been subject to human interpretation, which the event data have. A disadvantage is that these data are not labelled, meaning that these data do not contain any information related to football theoretic concepts. The data also contains moments where play is not continuing for example between a foul and the subsequent free kick. During this period the coordinates, speed and acceleration of the players and the ball are still recorded. This means that these data need some editing before being useful in an application. However, tracking data are important to this thesis as they contain all the information that is needed to calculate the features taken from the football literature.

3.2 Event Data

Event data are collected by human observation. Companies such as Opta, Stats and ChyronHego provide such data. The data contain the type of event that happens, such as a pass or a dangerous moment. These types are then further refined by assigning a classification to them. A pass can be successful, meaning that it reached its intended target, or unsuccessful for example and a dangerous moment is a *goal*, a *miss* or a *keeper save*. These data also contain the timestamp of the start and the end of the event as well as the x and y coordinates of the start and end of the event. Furthermore, these data also keep track of the players involved in an event and the team these players play for.

In this thesis, sequences are extracted and the data are labelled using these events. A sequence is defined by the 'ballwin' event, which will be further explained in section 3.4. This event type has a start and an end timestamp that are used to define the sequence. The label of these sequences is then determined based on whether the sequence ends in a dangerous moment or not.

3.3 Calculation of features

Both of these data sets only give an overview the position of all players and the ball at every frame and the events that follow each other during a game. However, to apply existing research on football theoretic concepts to the analysis of football games, some additional information is needed. Much of the literature on football works on developing features that hope to give more meaningful information on the state of a football game. Features such as dangerousity [43], defensive disruptiveness [38], space occupation gain [24] and centroids, length and width [3, 25] are all features that have been developed to give indicators for successful football play. In order to enrich the data used to build the models in this thesis, some of these features are used in this thesis as well. An overview of the features and a short description can be found in table 1. These features exist for both the team that starts the sequence as well as the opposing team at every timestamp, as they are not dependent on the reference team being in possession of the ball. Other features such as I-Mov and D-Def [38], that may perhaps be more informative, do depend on a team being in possession to exist at a timestamp. The simple calculated features at every timestamp are then connected to the event data to obtain the input to the models in this thesis.

3.4 Data Preparation

Sequences need to be properly prepared to be useful for predictive modeling. Not all sequences of frames in a football game are suited for predictive modeling. Sometimes teams capture the ball and immediately lose it due to player error. These sequences that are so short only because a player made a mistake are not relevant for research into tactical behaviour of football teams. For this reason, the *ballwin* event is used to extract the sequence. This *ballwin* event is used by the KNVB, the Dutch football association, to indicate a moment of capturing the ball after which multiple actions of the same team follow. This means that this only includes longer periods of possession of one team. A *ballwin* includes a start and end time in the event data. The start time indicates the moment the ball was captured, the end time indicates the end of the possession started by the *ballwin* as identified by the KNVB. The first step of the data preparation process is collecting the start and end times of these *ballwin* events.

After the start and end times have been determined the sequences need to be restricted to some interval. The goal of this thesis is to find the success probability over time of a sequence to highlight events that increased the probability of success. To compare the model performance over time, the sequences need to be restricted to be similar in length. The sequences shorter than 10 seconds are dropped from the data, as these sequences

Table 1: Features used to predict the probability of success of a sequence

Feature	Description
TeamCentX	The X-coordinate of the centroid, indicating the position along the length of the field.
TeamCentY	The Y-coordinate of the centroid, indicating the position along the width of the field.
Length	The distance from the player furthest behind to the player furthest ahead (in meters).
Width	The distance from the player furthest left to the player furthest right (in meters).
TeamCentX_gkExcl	TeamCentX but the goalkeeper is left out of the calculation
TeamCentY_gkExcl	TeamCentY but the goalkeeper is left out of the calculation
Lenght_gkExcl	Length but the goalkeeper is left out of the calculation
Width_gkExcl	Width but the goalkeeper is left out of the calculation
Spread	Measure of how spread out a team is.
stdSpread	Standard deviation of the spread.
Surface	The surface of the pitch that a team covers (in square meters).
SumVertices	Circumference of surface area of a team (in meters)
ShapeRatio	Ratio between the length and width of a team

are likely to be less informative. With the sequences shorter than 10 seconds dropped, the first and last 10 seconds of a sequence of play are taken. The model performance over the first 10 seconds of a sequence will indicate whether the models are able to identify sequences in an early stage. The performance in the final 10 seconds of a sequence is compared to previous work to see how the models in this thesis compare to existing models. The sequences of 10 seconds and longer are selected, because taking longer sequences would remove too many sequences from the data.

After the intervals of the sequences used have been collected the sequences need labels. The sequences have labels assigned to them based on the event data. If in the interval found in the previous step there is a *danger* event, the sequence gets a positive label. If such a *danger* event is not in the interval, the sequence gets a negative label.

The next step is to attach the calculated features to the intervals. The event data collected so far only contain the start and end time based on the *ballwin* events. In this step, the features at every frame are attached to the sequence. The features at every frame are collected based on the raw positional data, which has full information at every time step. Because the features used in this research are chosen such that they exist at every time step there are no missing data in the final sequences.

The features added to the data exist for both the defending and attacking team. In the next step, the features were made relative to the teams' role (attacking or defending) and normalised for the playing direction.

The final step is adding the relative time of the sequence. The relative time of the sequence is computed by subtracting the start time of a sequence from the timestamp of a frame.

As all the features are calculated based on the coordinates of the players, there are no outliers that need to be removed from the data. Because all the players are constrained to the pitch, these spatial features have a small range in which they always fall. For this reason, the only preprocessing done on the data is making sure all the values are correct, meaning no negative values in features that can not be negative such as the surface, and all features exist at every time step. Furthermore, an inspection of a subset of the data showed that no values had to be removed in this subset and that this subset of the data was complete and correct.

3.5 Descriptive Statistics

After all preparation steps were performed on the full dataset containing x matches. These matches contained 29,688 ballwin sequences. 1660 of these ballwin sequences were successful according to the definition used in this thesis, meaning that 5.5% of the sequences was successful and 94.5% of the sequences were unsuccessful. The full dataset contains 302 matches. This means that a match contains on average 98.3 *ballwin* moments longer than 10 seconds. A full football season in a league of 34 teams takes 306 games to complete, meaning that 4 games are missing from the data.

This preparation is also done on all the individual teams to build models for the individual teams. The number

Table 2: The distribution of the sequences over the different teams

Team	Sequences	Positive labels	Fraction positive
Team00	1619	72	0.044
Team01	1592	82	0.052
Team02	1513	77	0.051
Team03	1665	92	0.055
Team04	1666	95	0.057
Team05	1732	103	0.059
Team06	1705	85	0.050
Team07	1615	101	0.063
Team08	1697	90	0.053
Team09	1742	124	0.071
Team10	1595	77	0.048
Team11	1649	90	0.055
Team12	1613	91	0.056
Team13	1715	100	0.058
Team14	1566	102	0.065
Team15	1569	83	0.053
Team16	1625	87	0.054
Team17	1810	109	0.060

of sequences, the number of sequences with a positive label and the fraction of sequences with a positive label are shown in table 2. This table shows that there is some difference between the smallest and the largest fraction. Because the percentage of successful sequences is so small, the difference between 0.044 and 0.071 could have a large impact on the performance of the model for the different teams.

4 Methods

This section will go over the methods used in this thesis, both the baseline methods and the more complex methods. Finally, the evaluation metrics used in this thesis to analyse the results numerically.

4.1 Baseline Methods

To place the results obtained by the final model into perspective, some comparison methods are needed. These methods are widely used statistical machine learning methods. In the final results section these methods will serve as a means to compare the final results with other classifiers.

4.1.1 Logistic Regression

Logistic Regression [51] is a technique that aims to classify a set of data based on a linear combination of the features of this dataset. Logistic regression predicts the probability that an observation is in a class y with the following formula

$$P(Y = y) = \frac{1}{1 + e^{-(\alpha + \sum \beta_i x_i)}}$$

where α and the vector β are parameters that need to be learned from the data. The features of this observation are in the vector x. The parameters α and β are learned by optimising a certain loss function. The loss function for the logistic regression function used in this thesis is the loss function used by scikit-learn [58], which is as follows:

$$\min_{\beta,\alpha} \beta^T \beta + C \sum_{i=1}^{n} log(e^{-y_i(X_i^T \beta + \alpha)} + 1)$$

In this function if y_i and $X_i^T \beta + \alpha$ have the same value, under the assumption that y_i can be -1 or 1, the loss will be low. When the value is different, meaning an observation is wrongly classified, this value will be higher. The loss function is minimized with respect to α and β, meaning that these values can change while

the learning is in process. This will result in a final combination of parameters α and β that are used to predict new observations.

The advantage of this method is that it is relatively simple to interpret the model by looking at the coefficients produced by the model. A large positive coefficient means that a feature has a strong positive effect on the probability that a sequence is positive. However, logistic regression underestimates the probability of rare events [39] and since dangerous moments are a rare event in the dataset this could lead to low predicted probabilities. These lower probabilities can still be used, but a lower prediction threshold needs to be found that optimally separates negative and positive examples.

4.1.2 Naive Bayes

Naive Bayes is a generative model [52]. This means that it tries to build a joint probability distribution $p(y, x)$ which it uses in combination with Bayes' rule to calculate $p(y|x)$ and predict y based on the features x. Naive Bayes is called naive because the assumption is that the features x are independent from one another. Despite its naivety, Naive Bayes is known to be a good classifier [66] when looking at the class labels that are predicted by a Naive Bayes classifier.

The Naive Bayes method similarly to logistic regression has as an advantage that the model is easily explainable. The feature importance can be found by taking the probability that a feature has a certain value given that the class is positive. This makes these models more interpretable and therefore more believable. Although predictions made by a Naive Bayes classifier have shown to be of high quality, the same can not be said of the probabilities resulting from this method [66] which can mean that metrics that use these probabilities such as area under the curve can produce less trustworthy results.

4.1.3 Random Forest Classifier

Random forest classifiers [7] are an ensemble method. A random forest is a set of decision trees that are grown by random selection of training examples and features. Each tree in the ensemble is grown on a random subset of the training data. The features that are used to grow this tree are also randomly selected from the full set of training data. In the end all these trees are combined in an ensemble. A new observation that needs to be classified is classified using voting between all the trees in the ensemble. The class that gets the most votes is the class of the new observation.

The advantage of a random forest classifier in this thesis is that it will only use features that are important for the classification performance. Section 2 showed that there is the possibility to add many features to a model that can be calculated at every frame of a football game. Using a random forest classifier will ensure that only the most important features will be used in the classification and that unimportant features do not generate any noise [49]. On the other hand, random forest classifiers are ensemble methods, which means that they can be less interpretable than methods that consist of one classifier. Furthermore, random forest classifiers calculate probabilities based on the number of positive and negative samples in a leaf node, giving many examples the same probability if they end up in the same leaf nodes. This can be a problem when using evaluation methods that rely on different prediction thresholds. Ensemble methods also require more effort when determining feature importance. As the random forest classifier consists of many different decision trees it can be difficult to determine what features have a larger or smaller impact on the prediction based on where in the decision trees they appear.

4.2 Recurrent Neural Networks

All of the previously mentioned methods do not fully fit the task presented in this thesis. The data in this thesis are of sequential nature, meaning that every frame has a relation to the frame before and after. For this reason, the observations are not entirely independent, which can impact the performance of the statistical models. On top of this, the statistical models do not classify the whole sequence, but rather calculate the probability that a frame is in a successful or unsuccessful sequence. This method of classifying the individual moments in a football match has previously been done by [16], but fails to capture the fact that the current state of a football pitch depend on the previous moment.

Recurrent neural networks [60] have shown promising results on sequence classification tasks [36]. These recurrent neural networks keep a hidden state. Then when inputting the elements of a sequence, the output of one node is the input for the next node. Saving the hidden state as input for the next state makes it so that

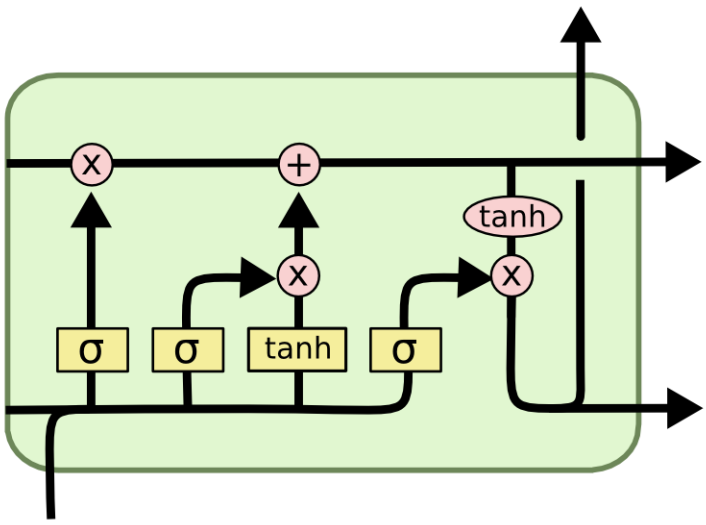

Figure 1: An LSTM cell [1]

the network is able to backpropagate through time. In the backpropagation through time the model computes the gradients over the whole sequence and update the weights that count for the whole sequence.

These traditional Recurrent Neural Networks are known to suffer from the vanishing gradients problem. If the number of timesteps in a recurrent neural network becomes larger, the gradient is also computed using more and more factors. These factors are small and a product of many small values gets closer and closer to zero. This means that the events many timesteps back from the final timestep have very little impact on the result. In the task presented in this thesis this is not desirable, as the goal is to get a correct classification as early as possible.

To solve this problem, LSTMs [34] were developed. The LSTM architecture contains a collection of 'gates' that determine what information in the sequence is important to the task. An example of an LSTM cell can be seen in figure 1. The gates in an LSTM cell can be described using the following formulas:

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

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{6}$$

where x_t is the input to the LSTM, h_t is the hidden state of the LSTM and W_f, W_i, W_C, W_o are the weights that are learned during training. Equation (1) describes the forget gate of the LSTM, in 1 this is the leftmost σ. This gate ensures that information that is no longer relevant is forgotten by the network. This allows the network to retain important information for a long time, whilst forgetting unimportant information. The output of this gate is a number between 0 and 1 where 0 represents that the information is completely unimportant, whereas a 1 represents that the information is very important. Equation 2 calculates the input gate value. This gate determines which values in the cell state C_t should be updated. In Figure 1 this is the second σ from the left. Equation 3 calculates the candidate values $\tilde{C}_t$ for the cell state. In Figure 1 this is the $\tanh$ in the yellow rectangle. Equation 4 calculates the new cell state C_t based on the values calculated in the first three equations. The old cell state is multiplied element-wise with the value for the forget gate to remove irrelevant information from the cell state. The values for the input gate are multiplied element-wise with the candidate values for the cell state to get new values for the cell state. These two states are then summed and become

the new cell state. Equation 5 calculates the output value of the LSTM cell. This value is used for the next hidden state. The output gate is shown in Figure 1 as the rightmost σ. Finally, in equation 6 the next hidden state is calculated. This is done by element-wise multiplication of the output gate calculated in equation 6 and the hyperbolic tangent of the cell state calculated in equation 4. This hidden state is then forwarded to the next LSTM cell and the whole process is repeated.

LSTMs are optimised by computing a loss function. The loss function used in this thesis is a binary crossentropy loss function. The binary crossentropy loss function looks as follows:

$$L(y) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i)) \tag{7}$$

To minimise this loss function, the probabilities of positive instances need to be as high as possible and the probability of negative instance needs to be as low as possible. The loss function is then backpropagated [60] through the network using an Adam optimizer [40].

In this thesis the LSTM architecture is used because of their ability to retain information over a longer time. The football sequences can contain a large number of time steps and using the oldest version of RNNs could lead to the information of the beginning of the sequence not being taken into consideration for the final classification. This is something that should be avoided, as one of the goals of this thesis is to correctly classify these sequences as early as possible.

4.3 Boosting

Boosting is an ensemble method that is known to work well on imbalanced datasets [9]. The objective of a boosting algorithm is to train multiple weak classifiers and combine these to a single strong classifier. AdaBoost [28] is such a boosting method. AdaBoost achieves this by training a sequence of classifiers where each sequential classifier focuses on examples that were previously misclassified. AdaBoost does this by re-weighting the misclassified examples to make them more important in the next classifier. In the initial classifier the weights of all the observations are set to $\frac{1}{N}$ where N is the number of observations. After each iteration the correctly classified examples have their weight decreased and the misclassified examples have their weight increased. This ensures that each subsequent classifier focuses more on harder to classify observations.

In this thesis, two different boosting methods are trained. As part of the statistical baselines, an AdaBoost model is trained with a decision tree as its weak learner. In this baseline each iteration a decision tree is trained and the weights are changed based on the observations that were correctly classified by this decision tree. Based on the quality of the classification the decision tree is also assigned a weight. A final classification is assigned to each observation by taking the weighted votes of each weak learner. The main model of interest of this thesis is a boosted LSTM, which will be further explained in the next subsection.

Boosting has as an advantage that it works well on imbalanced data [11] which are one of the challenges in this thesis. By focusing on the incorrectly classified examples, the minority class examples would get a higher weight until they are classified correctly. This is due to the weak classifier predicting that every example is part of the majority class. At a certain point the minority class instances will receive such a high weight that they receive a different classification. This will then increase the probability of these instances being of the positive class.

One of the drawbacks of the AdaBoost algorithm is that the exponential weight updates make it sensitive to outliers. The AdaBoost algorithm will continue increasing the weights of instances that are not classified correctly. This should not be a problem in this thesis, as all the data are constrained to the pitch. The players and the ball being constrained to the pitch causes there to be no outliers in the data.

4.3.1 Boosting LSTMs

Because the data has such a large class imbalance, boosting might be a good choice to help dealing with this problem. The weight of incorrectly classified instances is increased after each estimator is trained. Due to the class imbalance initially the positive instances will mostly be classified incorrectly. The weights of these instances will be changed by the AdaBoost algorithm:

The positive instances will be classified positively in later classifiers, when their weights are sufficiently increased. These positive classifications will contribute to a higher probability in the final prediction. This higher probability should ensure that most positive instances are ranked above negative instances, allowing for good classification of sequences, provided the correct threshold is chosen.

Algorithm 1: The AdaBoost algorithm for boosting LSTMs

Result: Ensemble of weak classifiers H and classifiers weights α
Initialize all sample weights w_i to $\frac{1}{n}$;
for $t \in T$ **do**
  Train weak classifier h_t to minimize cross-entropy
  Calculate classifier error $\epsilon_t = \sum_{i=1}^{N} (h_t(x_i) \neq y_i) \cdot w_i$
  Choose classifier weight $\alpha_t = \frac{1}{2} \frac{1-\epsilon_t}{\epsilon_t}$
  Update weights $w_{i,t+1} = w_{i,t} e^{-\alpha_t \cdot (h_t(x_i) \neq y_i)}$
  Normalize weights $w_i = \frac{w_i}{\sum_{j=0}^{N} w_j}$
end
Predict test observations using $\hat{y_i} = \sum_{t=1}^{T} \alpha_t h_t(x_i)$;

Boosting has been used before to improve the performance of an LSTM model [4] and has shown to improve classification performance on imbalanced datasets [9]. However, a big disadvantage of boosting deep neural networks is the additional time it takes to train these algorithms. As boosting is often done using many estimators, boosting LSTMs requires the training of many LSTMs. Training a single LSTM can be a costly process, so training many LSTMs is often not preferable. However, because the model predictions made by pretrained deep neural networks are fast, this should not be a big issue. When implementing a deep neural network in a pipeline that is useful for football associations, the trained model is placed in this pipeline. This model should be retrained every so often, because of new insights or a change in personnel, but this can be done during downtime.

4.4 Metrics

To assess what methods perform better than others some metrics are needed for this thesis. Some metrics are suited better for the data than others, in this section some metrics will be explained and whether they are well suited to the task in this thesis.

4.4.1 Accuracy

The accuracy is the simplest metric for a classification task. It is defined by the number of correctly classified observations divided by the total number of samples.

$$Acc = \frac{C}{N}$$

Where C is the number of correct classifications and N is the size of the test set. The accuracy may not necessarily be the best evaluation method for this task as the false negatives could still be of interest. The imbalance in the data also makes the accuracy a poor evaluation method for this task, as the accuracy for a model that predicts only negative classes would still achieve a very high accuracy.

4.4.2 Precision

Precision measures how many positive predictions are in fact positive. Precision uses the following formula:

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

Where TP is the number of true positives, the positive examples classified as positive. FP is the number of false positives, the negative examples classified as positive. Precision may also not be the most important metric in this thesis. This is because of the imbalance in the data. There is a higher chance of too few positive values being predicted than that too many positive values will be predicted.

4.4.3 Recall

Recall is a measurement for the number of positive observations are found by a model. Recall is calculated using the following formula:

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

Where TP is the number of true positives, the positive examples classified as positive. FN is the number of false negatives, the positive examples classified as negative. Recall could be very important in this thesis. Because of the imbalance in the data models trained on these data may tend to predict too many negative values. The recall measures whether the models in this thesis do a good job finding what few positive instances there are. However, only prioritising the recall is also not desirable, as a model that only classifies examples as positive would also get a perfect recall.

4.4.4 Area under the ROC curve

The area under the ROC curve scores a model based on the trade-off between true positives and false positives. At various thresholds for a positive prediction the number of false positives and false negatives is different. For example, predicting every instance with $P(X = 1) > 0.5$ as positive leads to fewer false positives than when $P(X = 1) > 0.1$ is used as threshold. However, the higher threshold could also lead to some of the true positives being missed by the model. The ROC curve plots the false positive rate and the true positive rate against each other using these different thresholds. The area under the curve summarises the ROC curve in one number. On top of this, when using normalised units, the area under the curve reflects the probability that a random positive observation ranks higher than a random negative observation. [23]

A disadvantage to the area under the ROC curve is that it might reflect too positively on models which use highly imbalanced data [15]. The reason for this is that the true negatives are involved in the calculation of the ROC curve. The number of true negatives in highly imbalanced data is much larger than any of the other values, so changes in these other values are not properly reflected in the result.

4.4.5 Average Precision

Davis and Goadrich [15] suggest the use of precision-recall curves at various thresholds. These curves plot the recall and precision against each other at various thresholds. When lowering the prediction threshold, the recall should increase as more positives are found. However, the fraction of positives that are actually positive should decrease as more and more negatives will end up in the set of retrieved sequences. The area under the precision recall curve is given by the average precision. The precision is calculated at each threshold that is used to plot the precision-recall curve and the average of this is returned as the average precision.

4.4.6 F1 Score

The F_1-score combines the precision and recall in one score. F_1 is a specific version of the F_β where the precision and recall contribute equally to the score. The F_1-score being the harmonic mean of the precision and recall. The score is calculated using the following formula:

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

This score gives more balanced perspective on the precision and recall. This is important because of the concern raised in the section on recall, the example being that classifying all examples as positive leads to perfect recall.

4.4.7 Matthews Correlation Coefficient

A measurement that considers all four possibilities of a classification is Matthews' Correlation Coefficient [46]. Accuracy and the F_1-score have as a limitation that they do not fully use the size of all four possibilities of a classification: false negative, false positive, true negative and true positive. The score of this coefficient is only high when it performs well on both negative and positive examples. [13] Matthews Correlation Coefficient is calculated as follows:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Matthews Correlation Coefficient could be well suited for the task in this thesis, as the data are highly imbalanced. This can have a negative impact on the accuracy and the F_1-score as exemplified in the work by Chicco [13]. This coefficient should perform better in this case and should give a more accurate representation of the performance of all models.

4.5 Implementation

The models are implemented using Python 3.7. The statistical baseline methods as well as the metrics used in this thesis are implemented using the scikit-learn library [58]. The LSTMs are implemented using tensorflow [2] and keras [14]. A custom implementation is made for boosting the LSTMs.

The LSTMs are a stacked 2-layer LSTM with 32 nodes in each layer, this architecture is determined by observation. Afterwards there was a fully connected layer with 16 nodes and a relu activation function and a final fully connected layer to come to a prediction using a sigmoid activation function. The hyperparameters for this model were found by performing a grid-search. The number of boosting estimators chosen for the final experiments in this thesis were a decision made based on the amount of time training all the models on the full dataset would take.

5 Experiments and Results

The evaluation of the models in this thesis is separated into three different experiments. The first experiment evaluates how well the models in this thesis compare to models from previous work. To the best of our knowledge this thesis is the first time sequences of football play are classified based on the *danger* success definition. However, there is some previous work that reports a numerical evaluation of a sequence classification task using a different success measure [18]. In section 2.2 it was mentioned that the work by Dick and Brefeld also classifies sequences of football team over time, but that their success measure was simpler than the one used in this thesis. For a good comparison between their work and the models in this thesis a comparative experimental setting is required. Therefore, in the first experiment the models in this thesis will be trained using a success definition similar to the work by Dick and Brefeld and the performance of these models will be compared to the performance of their models.

The second experiment aims to compare the statistical baselines to the LSTM implementations. This experiment will compare the performance of the models when trained on the same dataset and will then train the LSTM models on a larger dataset. The goal of these experiments is to find whether the LSTM models benefit from being able to be trained on a larger dataset and to find how the statistical baselines and the LSTM models compare in terms of performance.

The third experiment investigates whether models trained on the data of a single team perform better than models trained on the full dataset, containing sequences of play of all teams. This experiment compares the performance of all models when trained on the data of a single team and also compares the performance of the models trained on a single team to the models trained on all teams.

5.1 Experiment I

The goal of this thesis is to train classifiers on football data that are very imbalanced. To the best of our knowledge there has been no previous work that attempts to classify sequences of football play with this definition of success. However, the work by Dick and Brefeld [18] does also classify sequences of football over time. This allows for a comparison to be made, as such a comparison is needed to verify that the models in this thesis achieve a performance similar to previous work. To fully unlock the potential of big data, computer science applied to the sports science domain should strive for easier to compare outcome measures [29].

The difference between this thesis and the work by Dick and Brefeld is in the success measure as outlined in section 2.2. Therefore this experiment will use the success label of Dick and Brefeld applied to the data used in this thesis. The models used in this thesis are then trained on the dataset with the more relaxed success measure and the results are compared to the results achieved by Dick and Brefeld. This comparison will help place the model in the literature. If the models in this thesis are strongly outperformed by the models in the work of Dick and Brefeld then the models in this thesis should be reevaluated on their suitability for the task of classifying sequences of football play.

The comparison between this thesis and the work by Dick and Brefeld is no perfect comparison. The sequences in this thesis are collected by looking at the *ballwin* event, whereas Dick and Brefeld simply look for episodes of uninterrupted ball possession. Furthermore, the features used in this thesis and the work by Dick and Brefeld are different. In this thesis, calculated features are used to represent the playing field at every timestep, whereas Dick and Brefeld use the X and Y coordinates of all players and the ball at every frame. Finally, the data used in this thesis and the the data used by Dick and Brefeld come from different competitions. This can have an impact on how well the sequences of play can be modeled. Despite these three differences, the work by Dick and Brefeld is the closest comparison that can be made with the models in this thesis.

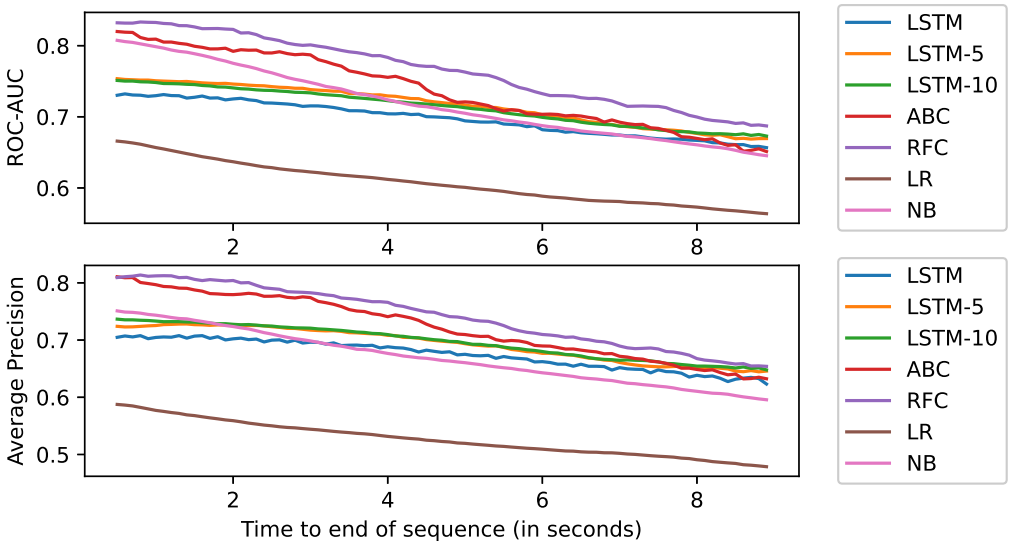

Figure 2: Area under the ROC curve (top) and the average precision (bottom) of the models trained on a dataset containing the sequences of *Team17* with reaching the final 25 meters of the pitch as a success label.

5.1.1 Results

Figure 2 shows the results of the experiments using the success definition that Dick and Brefeld [18] use. The graphs show the area under the ROC curve and the average precision of both the statistical methods: the AdaBoost Classifier (*ABC*), the Random Forest Classifier (*RFC*), Logistic Regression (*LR*) and Naive Bayes (*NB*) and the models based on LSTMs: a normal LSTM (*LSTM*) and the boosted LSTMs with five (*LSTM-5*) and ten (*LSTM-10*) estimators. The bottom graph in this figure serves as a point of comparison to the later experiments, where all models will be trained with a more strict definition of success. The average precision of the models in this experiment peaks just above 0.8 for the statistical baselines and around 0.75 for the boosted LSTMs. Considering that the expected performance of a model with no skill is $\frac{833}{1810} = 0.47$, the models are able to explain some of the football performance. This graph also shows that the performance of the LSTM models is slightly below that of the statistical baselines, with the exception of the Logistic Regression model. For most of the sequence the performance of the statistical models and the models using LSTMs is very similar, only in the latter few seconds the performance of the LSTMs plateaus whereas the statistical model improve 0.05 more.

To compare properly to the work of Dick and Brefeld the area under the ROC curve is also calculated. This metric shows that the performance ranking in terms of area under the ROC curve is similar to when looking at the average precision. The maximum area under the ROC curve achieved by the statistical models is 0.85, whereas the maximum area under the ROC curve achieved by the models based on the LSTMs is 0.76.

5.1.2 Discussion

The area under the ROC curve reported by Dick and Brefeld [18] when using the success metric used in this experiment is similar to the area under the ROC curve achieved in this experiment. The peak of the model by Dick and Brefeld is similar to that of the statistical baselines, at 0.85. The trajectory of the area under the ROC curve of the statistical models in this thesis is also similar to that of Dick and Brefeld, meaning that at 8 seconds before the end of a sequence, the LSTM models achieve a performance that is similar to previous work.

It should be reiterated that the data on which these two models are trained and validated are different. Therefore, the results found in this experiment do not show that the models in this thesis are strictly better or worse than the models in the work by Dick and Brefeld. However, the results in this experiment show that the models in this thesis are able to process sequences of football play at a level that has previously been achieved in the literature. This is an indication that the models are suited to the task of classifying football sequences.

As good as the model quality looks when predicting whether a sequence of football play gets within 25 meters

of the goal, the question remains whether this success label is relevant to the game of football. It is unlikely that the average football fan would say that an attack is successful when it gets within 25 meters of the opponent goal. Most fans would not say that a shot on goal is a success either, but when a shot is missed it can be attributed to player error, whereas an attack that reaches 25 meters from the opposing goal that does not result in a goal has a lot more potential reasons for the lack of goal.

5.2 Experiment II

The goal of this second experiment is to compare the LSTM models to the statistical baselines mentioned in section 4.1. Because the data can not be streamed through the statistical models, only the matches from the first four match days are used in this experiment. Using the first four match days meant that 35 matches were used for training the model. Choosing a set number of match days ensures that all the teams appear a similar number of times in the results of the experiments. These experiments compare the average precision over time to see how well boosted LSTMs perform in comparison to simple statistical methods. The accuracy, recall, precision, F1-score and Matthews correlation coefficient at various prediction thresholds will also be compared. This comparison is done at various thresholds because the class imbalance causes most of the predictions to be below 0.5 which is the standard threshold for classification. These low predicted probabilities can lead to low recall, F1-score and Matthews correlation coefficient scores. Lowering the threshold leads to more positive classifications, but will most likely also lead to more false negatives.

After all models have been trained on the first four match days, the models using LSTM networks are then trained on the full dataset containing all 34 match days. The LSTM models are better suited to be trained on this whole dataset because they do not need all the data to be loaded into memory during the whole training process. In this part of the second experiment the impact additional training examples have on the performance of the models.

5.2.1 Results

The aim of this thesis is to correctly predict sequences and find how the model performance develops over time. To evaluate when the models are able to achieve good performance on this task the average precision is plotted against the time in a sequence. The top graph of Figure 3 shows the average precision of all models used in this thesis over the first nine seconds of the sequences of play. The Logistic Regression and Naive Bayes methods perform the best when the models are trained on the first four match days. The AdaBoost classifier has some higher peaks than the Logistic Regression and Naive Bayes methods, but the performance of the AdaBoost classifier drops off near the end of the interval shown in the top graph of Figure 3.

The top graph in Figure 3 also shows that on the smaller dataset, the LSTM methods perform poorly. The average precision of none of the methods is particularly high, but the normal LSTM method achieves a performance that is only slightly better than a model with no skill near the end of the interval. This figure does show that the boosted LSTM methods achieve a better average precision than the standard LSTM over the whole sequence.

The bottom graph of 3 shows the average precision of the LSTM models trained on the larger dataset and the LSTM models trained on the dataset containing four match days. The performance of the models trained on the larger dataset is worse in the early stages of the sequence and similar performance at the end of the sequence. The graph also shows that the methods trained over time are more consistent than those trained on the smaller datasets.

Figure 4 shows the value of Matthews' correlation coefficient, the F1-score and the accuracy at different prediction thresholds. This figure shows that the Naive Bayes models achieves the highest score in both the F1-score and Matthews' correlation coefficient. The figure also shows that the models using LSTM networks have a very narrow interval of prediction thresholds where the performance of the models is higher than a model with no skill. All models except the Naive Bayes model achieve a performance equivalent to a model with no skill at some prediction thresholds except for the Naive Bayes model. However, the interval for the LSTM models is much narrower than that of the statistical baselines.

5.2.2 Discussion

The results of the second experiment show that the LSTM models are outperformed by the statistical baselines when trained on a dataset containing sequences of all teams. Overall the Naive Bayes classifier seems to perform the best over the whole interval. There are points in the interval where the AdaBoost classifier achieves a

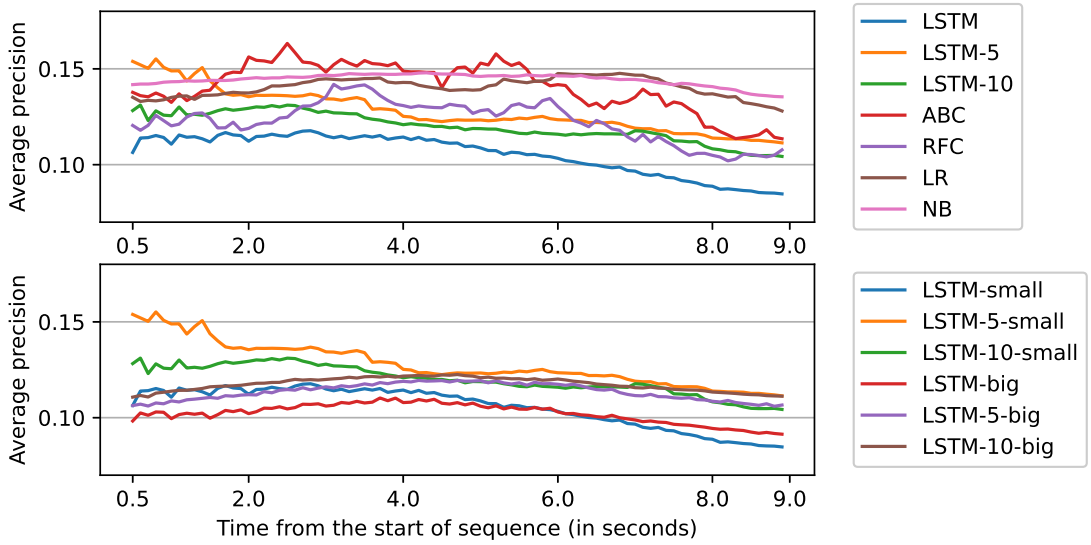

Figure 3: The top graphs shows the average precision of all methods in this thesis over time during the first 9 seconds of the sequence. The models in the top graph are trained on the matches of the first four match days. The bottom graph shows a comparison between the LSTM networks trained on the first four match days (denoted by the small suffix) and the models trained on the full dataset containing 34 match days (denoted by the big suffix)

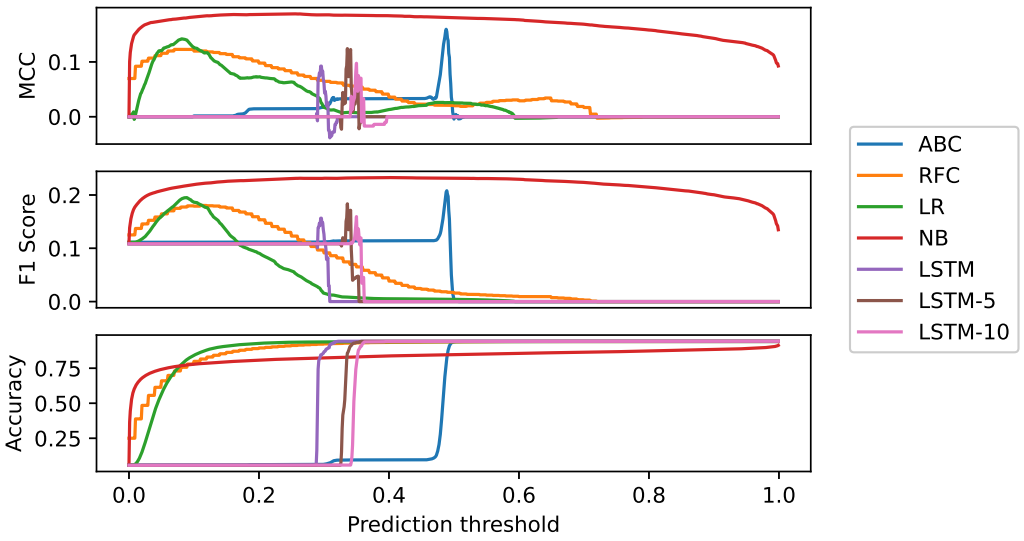

Figure 4: Matthews' Correlation Coefficient (Top), F1-score (Middle) and Accuracy (Bottom) for different prediction thresholds of all models trained on the dataset containing the first four match days.

better performance than the Naive Bayes classifier, but the AdaBoost model drops off in performance towards the end of the sequence.

A potential cause for this is that deep learning methods often seek to find hidden patterns in the data [5]. Using the calculated features in these data instead of the raw input data may cause statistical methods to outperform deep learning methods as these deep learning methods extract features by themselves. Previous work has shown that a set of human-engineered features in a statistical classifier can outperform deep learning methods [54]. In this thesis, the deep learning methods were not chosen to construct features, but because of their ability to work with data similar to timeseries. In this domain statistical methods have also shown to outperform deep learning methods [45], especially on shorter time intervals [8].

Boosting the LSTM models does seem to have a positive effect on their average precision. The LSTM model that has not been boosted achieves a performance that is the worst of all models in the experiment at any point in the sequence. The boosted models are more similar to the lower end of the statistical baselines. This observation is according to expectation, as boosting was applied to the LSTM models to improve the model performance on highly imbalanced data. This is consistent with previous work, where boosting shown to improve the performance of classifier on datasets with class imbalances [11]. Boosting recurrent neural networks has also shown to improve time-series forecasting [4], which is in agreement with the observation that boosting the LSTMs in this thesis improves their classification performance.

The bottom graph in figure 3 shows that the performance of the LSTM models trained on the larger datasets have similar performance to the models trained on the dataset containing the first four match days. This may be an indication that the sequences of all teams together are not well suited to the task of classifying these sequences. If combining sequences of all teams in one dataset is not a good input for the model, then adding more data might not improve the models either. Another explanation is that the first four match days contain sufficient data to get close to the best model performance a model trained on all teams can. Previous work on increasing data volume in human activity recognition [12] has shown that when increasing the data volume there is a rapid increase initially, but that this increase slows down when more data is added. The observation that the performance on the large dataset is not better than on the small dataset may indicate that the small dataset contains enough data to reach the best performance the models can using the features in this thesis.

Compared to a model that predicts the same probabilities for every sequence, these models perform better. A model without any skill would achieve an average precision score of $\frac{\#ofpositiveinstances}{\#ofinstances} = 0.056$. All the models in figure 3 have an average precision that is higher than such a model at every point in time.

The graphs in Figure 4 show that the range of probabilities predicted by most of the models is smaller than the range from zero to one. The reason that most models predict value that are not centered around 0.5 is that there are many more negative than positive examples. This causes the most models to undervalue the probability that a sequence is positive.

The model using LSTMs are better than models with no skill only in a small interval. This means that the boosted LSTMs and regular LSTM only predict probabilities in a small range. This narrow range of predictions makes these models difficult to use for new data, as the threshold needs to be carefully selected to get a performance out of the models that is better than the performance of a model that predicts the same for every example.

Previous work has shown that machine learning methods struggle to predict probabilities over the whole interval when the data on which these models are trained are imbalanced [62, 39]. This explains the distribution of the predicted probabilities. Deep learning methods have also shown to struggle to predict informative probabilities when data are imbalanced [33]. The boosting method proposed in this thesis does help the maximum performance in terms of these statistical methods, but does little to combat the small range in which the deep learning methods predicts the probabilities.

5.3 Experiment III

In the third experiment all models are trained for individual teams. As teams all have their own preferred play style, it is unlikely that a model that is trained on data from all teams can correctly predict whether a sequence is successful for a team. In this experiment all models will be trained using the data of a single team. This experiment will be repeated for all teams present in the dataset. After the models have been trained they are evaluated using the same metrics as experiment I.

The models trained on the data of a single team will be compared to the models trained using the data containing all teams and the models of the individual teams will be compared to each other. The results obtained from the comparison of the team specific models and the model using all data shows whether using

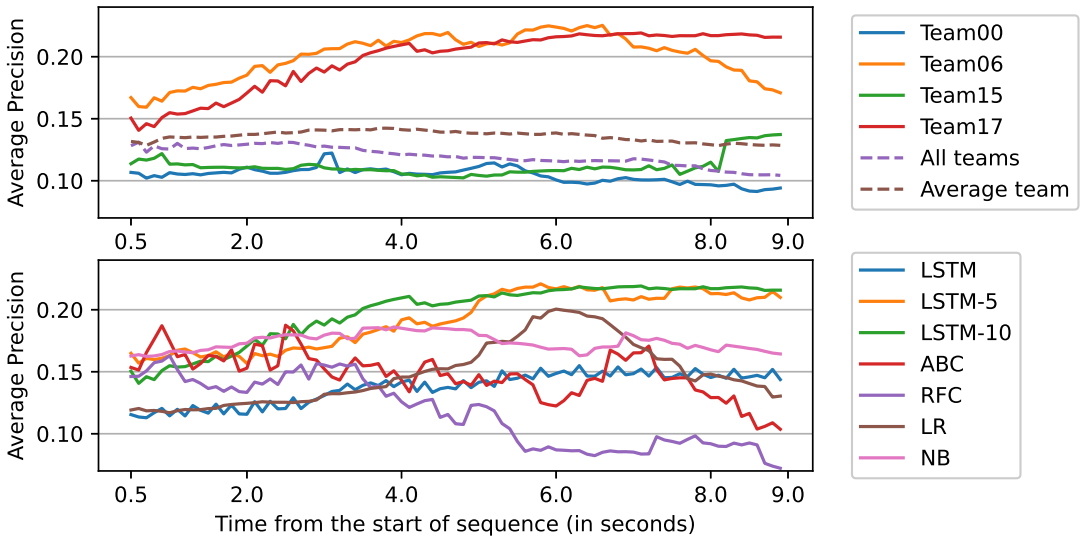

Figure 5: The top graph shows the average precision of *LSTM-10* over the time for the best and worst two team-specific models as well as the *LSTM-10* models for all teams and the average precision of the team-specific *LSTM-10* averaged over all teams. The bottom graph shows the average precision over time for all methods in this thesis, trained on the data containing sequences of *Team17*.

the data of a single team improves model performance, despite being trained on fewer data. The comparison between teams gives an indication of the range of performance these models fall into.

5.3.1 Results

The top graph of 5 shows the best two team-specific *LSTM-10* models and worst two team-specific *LSTM-10* models. The worst two team models are *Team00* and *Team15*, the best two team models are *Team17* and *Team06*. The purple dotted line is the average precision of the model trained on data of all teams over time. The brown dotted line titled *Average team* is the average precision over time averaged over all team-specific models. The graph shows that the average performance of a team-specific model is better than the model trained on all teams.

The bottom graph shows the performance of all models similar to the results of experiment I, but with all models only trained and verified on sequences of *Team17*. The graph shows that in this case the boosted LSTM models outperform the statistical baselines, with the *LSTM-5* and the *LSTM-10* achieving very similar results. From the statistical models, the Naive Bayes classifier and the Logistic Regression classifier achieve the best performance.

Overall, all models except the Random Forest Classifier improve when being trained on the data of one team, but the strongest improvement is achieved by the *LSTM-5* and *LSTM-10* models, which almost doubled the average precision they achieve.

5.3.2 Discussion

The models trained on specific teams outperform the models trained on sequences from all teams. This observation was expected beforehand, because models trained on a specific team are better able to capture the conditions each individual team has for success. Teams are successful in different ways, some teams prefer over the right, whereas other teams prefer to play over the left flank. When training the models on data from all teams this information relating to the play style of a team will not be properly captured in the model.

The average team-specific model is also better at classifying sequences of football play than the model trained on sequences of all teams. Once again, this is according to the expectation that models trained for a specific team do a better job at classifying sequences than a more general model trained on data from all teams.

However, there are some teams for which the team-specific model performs worse than the model trained on sequences from all teams. This result does not say that a model trained on the sequences of a specific team makes worse predictions for this specific team than the model trained on the full dataset. For example, the

model of *Team00* performs worse than the model trained on the full data. However, this may be caused by the sequences for *Team00* being harder to predict by a boosted LSTM rather than the model for *Team00* being worse at predicting sequences of *Team00* than the model trained on all teams.

The work by Decroos, Bransen, van Haaren and Davis [16] mentions that it is difficult to compare players across leagues and teams and show this by referring to some examples of this observation. This thesis also finds that using one model for all teams results in a worse model performance than when the models are team-specific. Transfer learning has shown that a model trained on a specific task can be repurposed to a new task, but that retraining an entire new model is more effective [19]. This is comparable to predicting the sequences of a single team with a model trained on all teams or a model trained on the data of that single team.

The results achieved by models trained on different teams vary greatly. The worst models are outperformed by the model on all teams, whereas the best models achieve an average precision around twice that of the model of all teams. This variety of model performance can be caused by many things. Inspecting Table 2 shows that it is unlikely that this difference in model performance is caused by a different balance in successful and unsuccessful sequences. The teams whose models achieve the best average precision are not the teams with the highest rate of positive sequences. Table 2 shows that in data of the teams with the best performing models, *Team06* and *Team17*, the fraction of positive sequences is 0.05 and 0.06 respectively. In the data of the teams with the worst performing models, *Team00* and *Team15*, these fractions are 0.044 and 0.053. While out of the four teams *Team17* has the highest fraction of positive sequences, there is little difference between the fraction of positives in the data from *Team06* and *Team15*. Table 2 also shows that the teams with the highest fraction of positive sequences *Team09* has neither the best nor the worst performing model.

The top graph in Figure 5 shows only the average precision achieved by the *LSTM-10* models. The previous experiment showed that when training the models on data from all teams, the *LSTM-10* model was not the best performing model. The bottom graph in Figure 5 shows the average precision of all models when trained on a single team, *Team17*. This figure shows that for this task the boosted LSTM models outperform the statistical baselines. A potential cause for the stronger improvement of the boosted LSTM models compared to the statistical baselines is that deep learning methods are generally better at finding complex patterns in the data than the statistical methods [41]. Some teams may benefit from the models' ability to find more complex patterns, whereas other teams have more straightforward conditions for a successful attack. This would explain the difference in improvement of the LSTM models for the different teams.

5.4 Discussion

The experiments have shown that the models presented in this thesis struggle to correctly classify sequences based when the sequences are labeled using the danger event as a success. All models do outperform a model that predicts randomly, but the average precision of the models is low and the best models trained on this success achieve a maximum average precision of 0.22. One of the potential reason for this poor performance is the large imbalance of positive versus negative sequences in the first two experiments. Some of the models were able to cope with this class imbalance better and as such achieved better results in the experiments, but the class imbalance is large enough to cause the models to make poor predictions.

Because the models in this thesis achieve similar results as models from previous work, the models should not be discarded as being of low quality. It seems more likely that the success measure used in this thesis, the *danger* event containing near misses, goals and shots on goal, is too difficult to predict using the data available in this thesis. As mentioned in the data section, the features used in this thesis are simple features that can be calculated at every timestamp. It is possible that these simple features are not able to fully represent how football teams play. The simple features could be enough to predict an easier target, such as getting within a certain distance of the goal, but the experiments show that they do not manage to predict the more difficult target correctly.

Another possible problem in the data is the selection of the sequences. In this thesis, the dataset was reduced to only sequences longer than 10 seconds. This was done arbitrarily, with the trade-off between informative sequences and the number of sequences kept in mind. Perhaps the chosen sequence length of 10 seconds was incorrect, as valuable information in the sequences longer than 10 seconds is missed or because informative sequences shorter than 10 seconds were discarded. Proper experimentation with different sequence lengths could provide an answer to the question whether the way the sequences were selected in this thesis has a harmful effect on the model performance.

The last two experiments also showed that boosting the LSTMs helped them deal with some of the problems in the data, evidenced by the boosted LSTMs achieving better performance than the standard LSTM. In the

experiment looking at a single team, the boosted LSTMs outperformed all the other models. The *LSTM-5* and *LSTM-10* in this experiment were also the two methods that achieved the highest average precision on the data with the *danger* event as a success label.

Finally, the first experiment showed that especially the statistical baselines reach similar performance when evaluated using a dataset with a more relaxed definition of success. A comparison with the work by Dick and Brefeld [18] shows that the statistical baselines reach an area under the ROC curve that matches the results in their work. The LSTM network and boosted LSTMs show a slightly worse area under the ROC curve in the final stage of a sequence, but earlier in the sequence they also achieve an area under the ROC curve that is similar to previous work. This shows that the methods in this thesis are able to classify sequences of football play, but that the simple features in this thesis combined with the strict definition of success result in a poor performance by these models.

6 Use Case

To show what the goal of the models in this thesis is in practice, a use case will be presented in this section. The goal of the models in this thesis is to take the feature values of the sequences at every time step and predict the probability that the current sequence will result in a *danger* event. Abrupt changes in probability can then further be investigated to see what resulted in this abrupt change in success probability. The probability of sequences can also be used to investigate whether passes between to players or other actions are valuable. All the passes between two players can be extracted and the average increase in probability of all these passes can be calculated. This can help coaches determine whether two players play well together.

In this section, three different models will be presented and their ability to help in the use case presented in the previous paragraph will be analysed. The models that will be looked at in this section are an *LSTM-10* model trained on both the dataset containing sequences from *Team17* and the full dataset. Another model that will be investigated on its ability to give useful insights is a Naive Bayes method trained on the dataset containing sequences from *Team17*.

Figure 6 shows two plots. The top plot shows the progression of the probabilities predicted by the *LSTM-10* model. The starting probability for all sequences predicted by these models is the maximum probability over the whole sequence. Afterwards, the probability only drops. This is likely due to the imbalanced data which causes all predicted probabilities to be lower than 0.5. After starting with a relatively high probability the probability of the sequences drops quickly. There After reaching the lowest probability value there seems to be very little change in probability. The cause of this can be the memory of the LSTM model. The model might strongly take into account what happens early in the sequence and unless something drastic changes in the data, this information will be kept all the way throughout the sequence.

The *Team17* model seems to be able to separate the two sequences better than the model trained on the full dataset. The highest probability over the course of the whole interval is the probability of success predicted by the *Team17* model. The model trained on the full dataset is also able to separate the successful and unsuccessful sequences over the whole dataset. Between the two *LSTM* models the model trained on the dataset for *Team17* predicts probabilities that are slightly further apart, giving a little more room to pick a decision threshold for a classifier.

However, the results achieved by the would be difficult to apply to the use case described in this section. Finding actions that contributed strongly towards the success of a team is hard to do when there is little variation in probability. Furthermore, the probability of success in these sequences is ever decreasing, even in the positive sequences, so finding moments that were responsible for this sequence being a success will be hard to find.

On the other hand, because the unsuccessful sequences are below the successful sequences over the whole interval, this model can still be used to find conditions for success. If the features contributing strongly to the sequence being a success can be found, then teams can still be set up to exploit these findings.

The Naive Bayes predictions in the bottom graph seem more promising should the models be applied to the use case in this section. The unsuccessful sequence starts off with a very high probability of success and drops off around 5 seconds into the sequence. The successful sequence has the opposite. The Naive Bayes classifier predicts low probabilities early on in the sequence, but at around 5 seconds this probability strongly increases. These two moments can be investigated and insights can be drawn from the actions that often have a similar effect on sequences. If for example a certain type of pass often strongly increases the probability of success, then this can be touched on during the preparations for a match.

It should be noted in the Naive Bayes model that Naive Bayes often tends to predict extreme probabilities

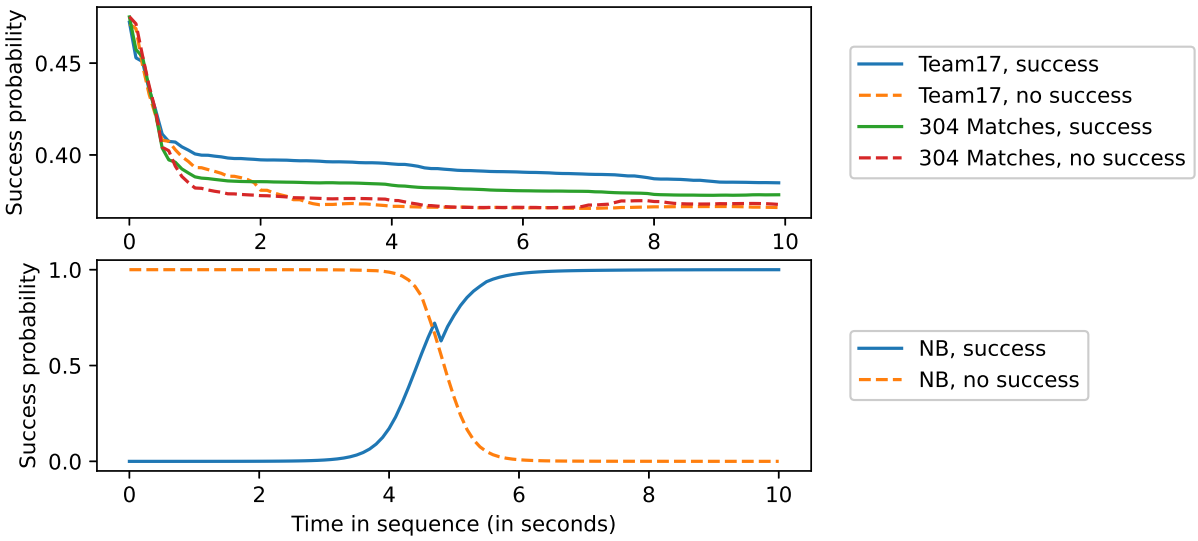

Figure 6: The top graph shows the probability predicted by the *LSTM-10* models trained on the dataset with sequences of *Team17* and the full dataset. The full lines are successful sequences and the dashed lines represent an unsuccessful sequence. The bottom graph shows another sequence, predicted by a Naive Bayes model.

[66]. This means the graph in Figure 6 does not always accurately reflect the actual probability of success. However, the predictions made by Naive Bayes in terms of class label are reliable, so a sudden switch from a negative to a positive classification is still worth investigating. A model that takes the probability of Naive Bayes as absolute truth instead of a pointer that something is worth further investigating might not be as accurate when Naive Bayes is used as a classifier.

Of course, the numerical results shown in Figure 6 are of little use by themselves. For a thorough analysis these sequences need to be connected to a set of annotated events to find what event is taking place during the time period that the trajectory of the predicted probabilities goes through a sudden change. When the annotated events are connected to the trajectories of the probabilities will the predicted probabilities point towards the events that warrant further investigation.

7 Conclusion and Future Work

The experiments have shown that the model performance in this thesis is similar to that in previous work when using an easier simpler success definition. However, when applying these models to a more complicated definition of success the model performance becomes poor. As the first work in using this success definition, the exact reason for the poor performance is unknown.

Because the methods in this thesis achieve results that are on par with results from previous work, it seems unlikely that the solution to the problems that came up during this thesis should be searched for in the models. The methods in this thesis are able to model sequence of football play better than a model with no skill whatsoever. This leaves the data as a potential reason for the poor results of the methods used in this thesis. It is possible that the simple features are not able to effectively predict a strict success definition such as the *danger* event.

The experiments also showed that boosting the LSTMs improved the performance of the LSTMs on a strongly imbalanced data set. This is in line with the expectation that boosting is a good technique to improve the performance of an LSTM model when it used to classify or predict highly imbalanced data. This insight is valuable as datasets used for football tasks can become imbalanced, depending on the success measure chosen.

The practical analysis showed that the *LSTM* models do not allow for a easy analysis of the actions in a football game. The probability of success these models predict remains similar over the whole sequence. The Naive Bayes model does allow for better analysis of a football sequence, as in both the negative and the positive example there was a single moment that swung the prediction from negative to positive and the other way around.

For future work the first suggestion is to attempt to train similar model with different features. The current features are simple and football analysts would generally not say that these features are very indicative of the probability of success. More sophisticated features, such as the available space may be better suited to predicting whether a sequence of football play will be successful.

A second point of interest for future work would be the small interval of probabilities produced by the model. This can be caused by the aforementioned simplicity of the data. These data may not enable a model to model success. However, this may also be caused by the fact that the model is underfit on the data. This can be improved in multiple ways. The first way to improve the fit of the boosted LSTM models on the data would be to increase the number of estimators. This would put even more emphasis on the misclassified examples, because each new estimator focuses more on misclassified examples. The misclassified examples will mostly be positively labeled examples, as these are the minority class. Using more estimators will eventually lead to many examples being classified as positive in the latter estimators. This will cause the probabilities to become more spread out as more instances will receive a positive classification at some point. The final way to improve the underfitting of the boosted models is to improve the base estimator. The base estimator in this thesis is severely underfit, because it is a small LSTM which is trained for very few epochs. Trying to improve the base estimator, either by training it for more epochs, or by using a more complex model could improve the performance of the overall model and would also wide the narrow window in which the predicted probabilities fall now.

The final suggestion for future work to improve the model proposed in this thesis is to make the models even more specific. The results in this thesis have shown that by making the task more specific through training the model only on one team, the performance of the models improved. By specifying the task even more, for example by training models for a team against certain opponents, the task could be even more specified. A challenge here is that specifying the task reduces the number of training data. The way the task is further specified needs to be selected carefully, which could be done by clustering opponents on play style and training a model on each group of opponents.

Overall, this thesis has shown that while the models used are competitive with similar models from previous work, the task in this thesis is more difficult than that of previous work. The models in this thesis perform poorly on this more difficult task. The results in this thesis also show that boosting LSTMs has a positive effect on the average precision these models achieve on a highly imbalanced datasets. For future work it is suggested to investigate the potential of different features to predict the success measure used in this thesis, as the current features do not manage to produce predictions of a high quality.

References

[1] Understanding lstm networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed: 17-06-2020.

[2] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[3] Aguiar, M., Gonçalves, B., Botelho, G., Lemmink, K., and Sampaio, J. Footballers' movement behaviour during 2-, 3-, 4-and 5-a-side small-sided games. *Journal of sports sciences 33*, 12 (2015), 1259–1266.

[4] Assaad, M., Boné, R., and Cardot, H. A new boosting algorithm for improved time-series forecasting with recurrent neural networks. *Information Fusion 9*, 1 (2008), 41–55.

[5] Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning* (2012), pp. 17–36.

[6] Bialkowski, A., Lucey, P., Carr, P., Yue, Y., Sridharan, S., and Matthews, I. Large-scale analysis of soccer matches using spatiotemporal tracking data. In *2014 IEEE International Conference on Data Mining* (2014), IEEE, pp. 725–730.

[7] Breiman, L. Random forests. *Machine learning 45*, 1 (2001), 5–32.

[8] Cerqueira, V., Torgo, L., and Soares, C. Machine learning vs statistical methods for time series forecasting: Size matters. *arXiv preprint arXiv:1909.13316* (2019).

[9] Chawla, N. V. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 875–886.

[10] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research 16* (2002), 321–357.

[11] Chawla, N. V., Lazarevic, A., Hall, L. O., and Bowyer, K. W. Smoteboost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery* (2003), Springer, pp. 107–119.

[12] Chen, H., Xiong, F., Wu, D., Zheng, L., Peng, A., Hong, X., Tang, B., Lu, H., Shi, H., and Zheng, H. Assessing impacts of data volume and data set balance in using deep learning approach to human activity recognition. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2017), IEEE, pp. 1160–1165.

[13] Chicco, D. Ten quick tips for machine learning in computational biology. *BioData mining 10*, 1 (2017), 35.

[14] Chollet, F., et al. Keras. https://keras.io, 2015.

[15] Davis, J., and Goadrich, M. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning* (2006), pp. 233–240.

[16] Decroos, T., Bransen, L., Van Haaren, J., and Davis, J. 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.

[17] Decroos, T., Dzyuba, V., Van Haaren, J., and Davis, J. Predicting soccer highlights from spatio-temporal match event streams. In *Thirty-First AAAI Conference on Artificial Intelligence* (2017).

[18] Dick, U., and Brefeld, U. Learning to rate player positioning in soccer. *Big data 7*, 1 (2019), 71–82.

[19] Djeddi, C., Jamil, A., and Siddiqi, I. *Pattern Recognition and Artificial Intelligence: Third Mediterranean Conference, MedPRAI 2019, Istanbul, Turkey, December 22–23, 2019, Proceedings*, vol. 1144. Springer Nature, 2019.

[20] Domingos, P. A few useful things to know about machine learning. *Communications of the ACM 55*, 10 (2012), 78–87.

[21] Fan, W., Stolfo, S. J., Zhang, J., and Chan, P. K. Adacost: misclassification cost-sensitive boosting. In *Icml* (1999), vol. 99, pp. 97–105.

[22] Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery 33*, 4 (2019), 917–963.

[23] Fawcett, T. An introduction to roc analysis. *Pattern recognition letters 27*, 8 (2006), 861–874.

[24] Fernandez, J., and Bornn, L. Wide open spaces: A statistical technique for measuring space creation in professional soccer. In *Sloan Sports Analytics Conference* (2018), vol. 2018.

[25] Folgado, H., Lemmink, K. A., Frencken, W., and Sampaio, J. Length, width and centroid distance as measures of teams tactical performance in youth football. *European Journal of Sport Science 14*, sup1 (2014), S487–S492.

[26] Frencken, W., and Lemmink, K. Team kinematics of small-sided soccer games: A systematic approach. In *Science and football VI*. Routledge, 2008, pp. 187–192.

[27] Frencken, W., Lemmink, K., Delleman, N., and Visscher, C. Oscillations of centroid position and surface area of soccer teams in small-sided games. *European Journal of Sport Science 11*, 4 (2011), 215–223.

[28] Freund, Y., and Schapire, R. E. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory* (1995), Springer, pp. 23–37.

[29] Goes, F., Meerhoff, L., Bueno, M., Rodrigues, D., Moura, F., Brink, M., Elferink-Gemser, M., Knobbe, A., Cunha, S., Torres, R., et al. Unlocking the potential of big data to support tactical performance analysis in professional soccer: A systematic review. *European Journal of Sport Science* (2020), 1–16.

[30] Goes, F. R., Kempe, M., Meerhoff, L. A., and Lemmink, K. A. Not every pass can be an assist: a data-driven model to measure pass effectiveness in professional soccer matches. *Big data 7*, 1 (2019), 57–70.

[31] Gudmundsson, J., and Horton, M. Spatio-temporal analysis of team sports. *ACM Computing Surveys (CSUR) 50*, 2 (2017), 1–34.

[32] Han, H., Wang, W.-Y., and Mao, B.-H. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing* (2005), Springer, pp. 878–887.

[33] Hasibi, R., Shokri, M., and Dehghan, M. Augmentation scheme for dealing with imbalanced network traffic classification using deep learning. *arXiv preprint arXiv:1901.00204* (2019).

[34] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[35] Joshi, M. V., Kumar, V., and Agarwal, R. C. Evaluating boosting algorithms to classify rare classes: Comparison and improvements. In *Proceedings 2001 IEEE International Conference on Data Mining* (2001), IEEE, pp. 257–264.

[36] Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.-E., He-Guelton, L., and Caelen, O. Sequence classification for credit-card fraud detection. *Expert Systems with Applications 100* (2018), 234–245.

[37] Karim, F., Majumdar, S., Darabi, H., and Chen, S. Lstm fully convolutional networks for time series classification. *IEEE access 6* (2017), 1662–1669.

[38] Kempe, M., and Goes, F. Move it or lose it: Exploring the relation of defensive disruptiveness and team success. Tech. rep., EasyChair, 2019.

[39] King, G., and Zeng, L. Logistic regression in rare events data. *Political analysis 9*, 2 (2001), 137–163.

[40] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[41] LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature 521*, 7553 (2015), 436–444.

[42] Lim, S. K., Loo, Y., Tran, N.-T., Cheung, N.-M., Roig, G., and Elovici, Y. Doping: Generative data augmentation for unsupervised anomaly detection with gan. In *2018 IEEE International Conference on Data Mining (ICDM)* (2018), IEEE, pp. 1122–1127.

[43] Link, D., Lang, S., and Seidenschwarz, P. Real time quantification of dangerousity in football using spatiotemporal tracking data. *PloS one 11*, 12 (2016).

[44] Longadge, R., and Dongre, S. Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707* (2013).

[45] Makridakis, S., Spiliotis, E., and Assimakopoulos, V. Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one 13*, 3 (2018), e0194889.

[46] Matthews, B. W. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure 405*, 2 (1975), 442–451.

[47] Meerhoff, L., Goes, F., Knobbe, A., et al. Exploring successful team tactics in soccer tracking data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2019), Springer, pp. 235–246.

[48] Memmert, D., Lemmink, K. A., and Sampaio, J. Current approaches to tactical performance analyses in soccer using position data. *Sports Medicine 47*, 1 (2017), 1–10.

[49] Mentch, L., and Zhou, S. Randomization as regularization: A degrees of freedom explanation for random forest success. *arXiv preprint arXiv:1911.00190* (2019).

[50] Moura, F. A., Martins, L. E. B., Anido, R. D. O., De Barros, R. M. L., and Cunha, S. A. Quantitative analysis of brazilian football players' organisation on the pitch. *Sports biomechanics 11*, 1 (2012), 85–96.

[51] Nelder, J. A., and Wedderburn, R. W. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General) 135*, 3 (1972), 370–384.

[52] Ng, A. Y., and Jordan, M. I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems* (2002), pp. 841–848.

[53] Nguyen, N. G., Tran, V. A., Ngo, D. L., Phan, D., Lumbanraja, F. R., Faisal, M. R., Abapihi, B., Kubo, M., Satou, K., et al. Dna sequence classification by convolutional neural network. *Journal of Biomedical Science and Engineering 9*, 05 (2016), 280.

[54] Oakden-Rayner, L., Carneiro, G., Bessen, T., Nascimento, J. C., Bradley, A. P., and Palmer, L. J. Precision radiology: predicting longevity using feature engineering and deep learning methods in a radiomics framework. *Scientific reports 7*, 1 (2017), 1–13.

[55] Olthof, S., Frencken, W., and Lemmink, K. A match-derived relative pitch area facilitates the tactical representativeness of small-sided games for the official soccer match. *JOURNAL OF STRENGTH AND CONDITIONING RESEARCH 33*, 2 (2 2019), 523–530.

[56] Pappalardo, L., Cintia, P., Ferragina, P., Massucco, E., Pedreschi, D., and Giannotti, F. Playerank: data-driven performance evaluation and player ranking in soccer via a machine learning approach. *ACM Transactions on Intelligent Systems and Technology (TIST) 10*, 5 (2019), 1–27.

[57] Pappalardo, L., Cintia, P., Rossi, A., Massucco, E., Ferragina, P., Pedreschi, D., and Giannotti, F. A public data set of spatio-temporal match events in soccer competitions. *Scientific data 6*, 1 (2019), 1–15.

[58] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[59] Rein, R., and Memmert, D. Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science. *SpringerPlus 5*, 1 (2016), 1–13.

[60] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *nature 323*, 6088 (1986), 533–536.

[61] Sarmento, H., Marcelino, R., Anguera, M. T., CampaniÇo, J., Matos, N., and LeitÃo, J. C. Match analysis in football: a systematic review. *Journal of sports sciences 32*, 20 (2014), 1831–1843.

[62] Wallace, B. C., and Dahabreh, I. J. Class probability estimates are unreliable for imbalanced data (and how to fix them). In *2012 IEEE 12th International Conference on Data Mining* (2012), IEEE, pp. 695–704.

[63] Wang, Y., Huang, M., Zhu, X., and Zhao, L. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (2016), pp. 606–615.

[64] Xing, Z., Pei, J., and Keogh, E. A brief survey on sequence classification. *ACM Sigkdd Explorations Newsletter 12*, 1 (2010), 40–48.

[65] Yildirim, Ö. A novel wavelet sequence based on deep bidirectional lstm network model for ecg signal classification. *Computers in biology and medicine 96* (2018), 189–202.

[66] Zhang, H. The optimality of naive bayes. *AA 1*, 2 (2004), 3.

[67] Zhang, L., Wang, S., and Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8*, 4 (2018), e1253.