ANTICIPATING THE NEXT ACTION IN A FOOTBALL MATCH USING NEURAL NETWORKS

Candidate Number: 260841

Supervisor: Dr. Colin Ashby

Submitted in partial fulfillment of the requirements for the MSc in Data Science at the University of Sussex

August 2023

ACKNOWLEDGEMENT

This paper serves as the crowning jewel in my quest to step beyond what I previously knew and acquire new knowledge in the field of Data Science and Machine Learning. As this phase comes to an end, I am thankful to God Almighty for providing me the strength to take on this challenge and successfully complete this journey. To my superb supervisor, Dr, Colin Ashby, your guidance, knowledge and feedback throughout the process of this work have played a very pivotal role in ensuring the content of this work is of high quality.

To my family, who have been the backbone of my academic and professional success, their support in providing the opportunity to take up this challenge cannot be overstated. I would love to thank my friends and colleagues for their support throughout this journey and Kenechi Dukor for setting me on this path to discovering my love for data, its applications and constantly providing support to me whenever I call.

ABSTRACT

The landscape of Football has continued to evolve, with the old belief that the game could not be studied with data gradually dying. Football Analytics has revolutionised the way the game is now viewed with advanced analytics providing much more detail about on-field actions and even off-field activities. With this revolution, very little attention has been paid to the concept of Anticipation in football. In this research we take advantage of the little attention in this area and propose a model that can predict the next action in a football match with an F1-Score of 73.9% and is particularly adept at anticipating when a shot is the next action. The model is built on the implementation of an LSTM framework and relies on the introduction of in-game categorial information that can affect decisions and choices of what action to take.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Background

Football, a team-based sport also known as soccer in some parts of the world is a dynamic, fast-paced nature, where the course of a match can pivot in an instant based on a single action or event (Machado et al., 2016). Each pass, shot, tackle, and movement adds to the complexity of the sport, producing a sequence of interconnected events that define the match being played.

The world economy is undergoing rapid development, and the lives of the people are constantly improving with the use of data and the internet (Li, 2020). This has led to the popularity of watching Football matches through streaming services via the internet and the sport becoming the most watched and popular sport in the world. The recent 2022 FIFA World Cup held in Qatar was reported to have received over 5 billion engagements validating how popular the sport has become (FIFA, 2022). In recent time Football has been known to be a major contributor to the economy of western countries especially the United Kingdom, Spain, Germany, Italy and France (Zhang et al., 2022). These countries have various top tier and popular football leagues that are largely responsible for the growth and popularity of the sport. A direct result of this growth in audience of football matches, Football teams now generate a lot more revenue through broadcasting rights (Tunaru et al., 2010).

This increase in revenue and general technological advancement has led to a revolution in football, in which data has become the tool of choice in making decisions. This has led to the growth of football analytics, a field dedicated to extracting meaningful insights from football data. This advancement has led to a rise in Football and Sport Analytics companies, like StatsBomb, WyScout, Statsperform(OptaStats) which specialise in gathering data and tagging events from Football matches while also providing sports analytics services. These companies have improved data collation processes with Player Tracking technologies with the goal of capturing every location, action and movement of footballers and the ball during a match (Eggels, R. et Al., 2016). Football clubs now also have internal analytics departments that use data generated by their teams from medicals, matches, training and so on to

investigate ways to gain a competitive edge through signing the best players, managing injuries, and improving tactical strategies.

The field of football analytics has seen a significant transformation over the past decade, evolving from basic statistics and data collection to the application of advanced statistical methods and machine learning techniques for predictive modelling. Making predictions of football matches is a problem that has been the subject of a wide variety of studies. Early efforts often relied on conventional statistical methods to analyse player and team performance focusing on basic aggregated statistics and metrics of simple actions like passes, shots, tackles carried out during matches by individual players or as a team (Decroos et al., 2019). While these statistics provide summarised information of a team or player's performance, they lack depth needed to understand the intricacies of a match. However, with the advent of machine learning, researchers have begun to apply more sophisticated techniques to this problem.

The introduction and advancement of technology have paved the way for more detailed data collection during matches, leading to a richer understanding of the sport (Perl et al., 2012). Today, football analytics encompasses a wide range of areas, including player performance evaluation (Pappalardo et al., 2019), tactical analysis (Low et al., 2020), injury prevention (Rogalski et al., 2013), and talent identification (Cwiklinski et al., 2021). The advent of detailed event data, which captures specific actions during a match, has further enriched the scope of analysis, providing granular insights into the dynamics of football (Simpson et al., 2022).

With football becoming more complex and an increase in the availability of comprehensive tracking data, so did the need for more sophisticated analytical tools. Advanced analytics, which combines statistical, computational and mathematical techniques, have been adopted to provide deeper insights into performances and team dynamics (Fernandez et al., 2019).

Machine learning techniques have found numerous applications in football analytics, driven by the increasing availability and complexity of match data. Traditional machine learning techniques, such as regression models and decision trees, have been used to predict

outcomes of matches like number of goals scored or the result of a match (Hvattum & Arntzen, 2010).

Machine learning is a subset of advanced analytics which has shown significant promise in football analytics. Its ability to handle large amounts of data and automatic pattern identification makes it a powerful tool for evaluating performance, predicting match outcomes (Gudmundsson et al., 2017) compared to humans who perform poorly at these tasks proven by Franks & Miller (1986).

In any football match, there are two teams, made up of eleven players each. Each team's players work together to score against each other and also prevent goals. Many tactical decisions are made before the match and during the match to provide a team with the best chance of winning (Beal et al., 2020). Tactics is the foundation on which a team plays its football and is key to winning matches. It describes the strategies and decisions made by a team to achieve its goal during a match (Jin, 2011).

A team's tactics can be analysed by its Formation, the organisation of players on the pitch in relation to each other, where there is always one goalkeeper and ten outfield players who are identified as either defenders, midfielders and attackers or forwards (Beal et al., 2020). Tactics can also be analysed as Playing System and Playing Style; A team will implement a tactic using a system of play which influences the style of play. Today in football, football teams have identities, these identities describe the style of play of that team. The most popular system of play today is Total Football implemented by majority of the top European clubs like Manchester City, Barcelona, Bayern Munich e.t.c.

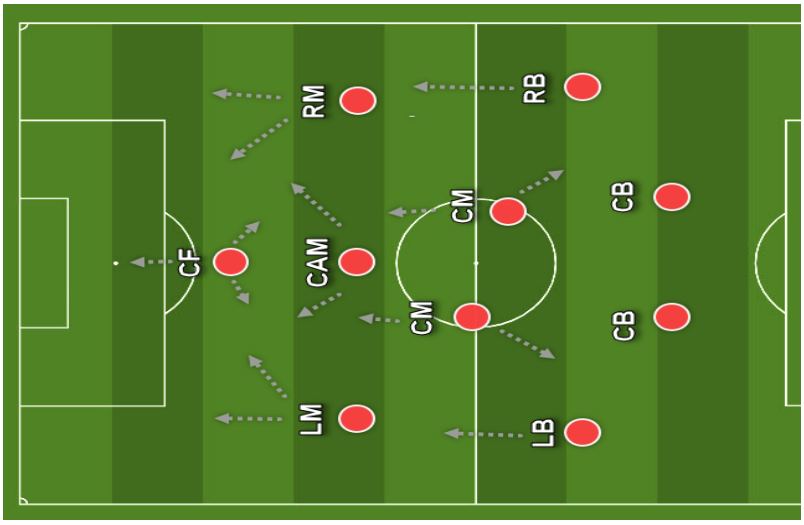


Figure 1.1: *Organisation of team members in an organised structure representing the formation in use (O'Neill, 2020)*

When systems, style of play and formation come together, playing patterns can be observed. These patterns can be team specific and also depend on the individual players on the pitch. A playing pattern is a sequence of actions and movements that a team implements to achieve a goal, usually to score goals or defend to prevent goals. These patterns can be developed through training and repetition during matches.

Despite the rise in attention of football analytics and availability of event data from football matches, the application of advanced analytics in studying pattern of play of teams to anticipate or predict the next event/action by a team in a football match is an area that has been relatively unexplored. Other factors like opposition tactics, individual skills, physical conditions, and even external factors such as crowd noise and weather conditions, can all intertwine to shape each moment on the pitch (Fernandez et al., 2019).

Recognising off-the-ball and on-the-ball movements and patterns of teams in predicting the next action in a football match is indeed a challenging task due to the dynamic and complex nature of the sport. The complexity of this problem stems from a variety of factors that influence each action in a match, lends itself to advanced analytics techniques such as machine learning and data mining. Through the application of these techniques, we can begin to deconstruct these interactions, discover hidden patterns, and gain a deeper understanding of the dynamics of the sport. Predicting the next action or Anticipation as it

more commonly called in football, is the skill of accurately perceiving the intentions of other players and is a valuable trait that influences many aspects of the game (Miller-Dicks et al., 2010).

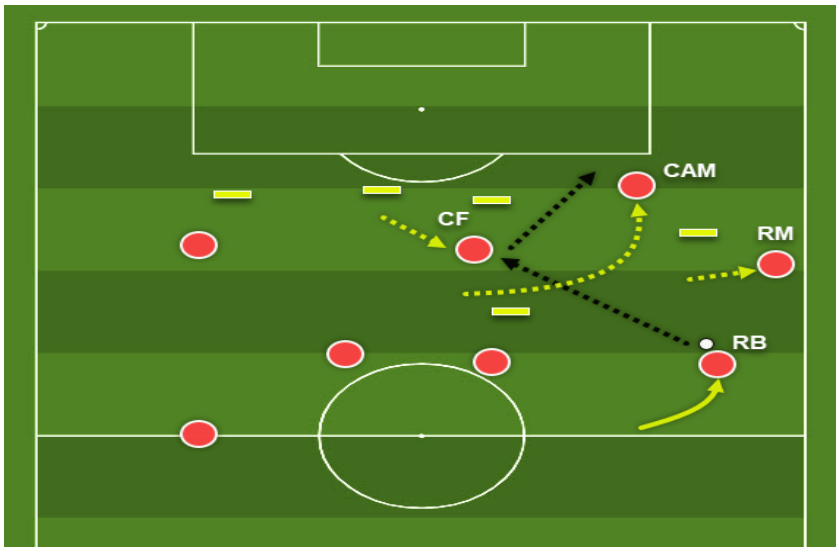


Figure 1.2: *Player movement and potential passes displaying the teams' pattern of play (O'Neill, 2020)*

This allows us to move beyond the simplicity of the eye test and into the use of analytics tools, where we can now rely on mathematics and science to anticipate the next event/action in a football match to a higher degree of accuracy than a human can. While it is possible to relatively predict some actions using the state of play, a shot is more likely when the ball is close to the goal, other actions are likely to be less predictable due to the influence of numerous factors affecting the decision of players. These include a variety of tactics implemented by teams, the technical abilities and decision-making of players, and the unpredictable nature of the game itself.

"Next action" refers to the subsequent event that occurs in a football match following a previous event. This could be any action taken by any player on the pitch. For example, if a player receives the ball as a pass, the next action could be another player receiving the pass, or it could be an interception by an opposing player.

For this research, we will not focus on the fine-grained actions that occur in a match. The focus of this study will be on actions that concern a team when in possession of the ball,

actions like passes, crosses, dribbles and shots. These are actions that directly influence how a team attacks in a match and can be used to find patterns and capture a team's regular style of play. Also, from a practical perspective, focusing on these actions helps limit the complexity of the problem. Football matches involve many possible actions, and attempting to predict all these actions can lead to overcomplication of the task. Focusing on a defined subset of actions, developing a manageable and focused prediction model.

Priority is to learn patterns of play identities of football team and correctly anticipate their next action. This would be done through exploring and understanding complexities in football teams' movements on the pitch. To achieve this, extensive experimentation was carried out using sequential football event data containing important information about each event. These features include and are not limited to action types, location of action on the pitch, team identities and so on that define the conditions around that event taking place.

1.2 Significance of Research

The significance of tackling this type of problem cannot be overstated and it has the potential to revolutionize football. By predicting the sequence of events that occur in a game, we are provided the opportunity to gain a granular understanding of how a match progresses, which can serve as a valuable tool for football teams. While this problem is related to more popular areas of football analytics like predicting match outcomes and expected goals (xG), it is not necessarily subordinate to them. It should be seen as a complementary approach that provides additional detail and granularity.

This level of prediction can create a way for more sophisticated and complex tactical strategies. For example, if a model can correctly predict that an opposing player is likely to pass the ball to a particular teammate or a particular region of the pitch, the defending team can use this knowledge to adjust their positioning to intercept the pass. This proactive style can provide teams with significant competitive advantage by disrupting the opponent's strategy, enabling them to respond to events preemptively rather than being reactionary (Memmert et al., 2017).

Next action prediction can contribute towards enhancing player performance. From understanding the sequence of events that led up to a goal, we can gain more nuanced information on specific players' contributions towards the team's success. This could potentially lead to fairer and more accurate player evaluations, therefore influencing player training and development (Rampinini et al., 2009).

This part of football analytics can contribute towards improving fan experience. Advanced game analytics can provide the opportunity for enthusiastic fans to gain a deeper understanding of the game, exposing them to the intricacies of how their favourite teams play and the skills involved. This could lead to increased fan engagement and enjoyment of the sport.

In summary, the ability to predict the next action in a football match holds significant potential for improving tactical strategies, player evaluations, and fan experiences. However, while a considerable amount of work has been undertaken in predicting match outcomes or player performance, there is a gap in the quantity of literature concerning predicting the next action/event in a football match using sequential event data. This research aims to contribute to filling that gap, applying machine learning algorithms to predict future actions or events during a football match, potentially providing valuable insights for coaching, tactical decision-making, and fan engagement.

1.3 Research Overview

For this research, we performed a comprehensive exploration of predicting the next action in a football match, implementing advanced machine learning techniques. We used sequential event data made openly available by StatsBomb. This deviates from the largest available football data, provided by WyScout, used for previous research on similar topics.

The StatsBomb data was chosen to carry out this research because of the richness of information contained in each event. The data consists of important information about each event and ranges from the action being done, location of players on the field during the action, the formation being used and player roles which the WyScout dataset lacks. Using

this StatsBomb data with the features mentioned, we believe it will provide more context around the decision made by players in-game as a team.

From the data gathered, our research focuses on events that carry information on location of players during an event. Features were merged where possible and the data was further filtered to features based on their potential to influence the outcome of the next action in sequence.

This research takes a page out of the previous work done on this problem and focuses on building our model on a type of Recurrent Neural Networks called Long-Short Term Memory (LSTM) and a more advanced form of Neural Network called Transformers. The two models are designed to effectively handle the nature of our data, which is sequential. The two models have shown great success in their applications in various domains including timeseries and sequential problems.

LSTMs possess the ability to remember past information relying on their internal cell and hidden state, making them suitable for historical based predictions. They rely heavily on the order or historical sequence which makes them capable of recognising team patterns.

Transformers on the other hand are different from LSTM as they do not have an understanding of order without the introduction of positional encodings. They leverage an attention mechanism to weigh the importance of different parts of sequences to make predictions. They can handle complex interactions of data with their multi-head attention mechanism, and they possess the tools to process data in parallel.

The task, predicting the next action of a football match, focuses on 4 classes and is a classification problem. In matches Shots and Crosses tend to be less common than Passes or Dribbles, this is reflected in the dataset as an imbalance of the target variables. To properly evaluate the performance of the model on the loss function, Sparse Categorical Cross Entropy and Sparce Categorical Accuracy specifically designed to handle multi-categorical problems were used to evaluate how well the model was learning. F1-Score, Accuracy and Confusion Matrix also helped in providing a holistic view of how well the model could tackle the task. The evaluation process showed promising results, with the model performing better than

the results obtained in previous studies. Detailed results and discussions obtained from the training and testing are presented in the following sections.

This paper takes a structured format in describing using comprehensive details, the steps and methods carried out and the results achieved during the research stage. We start by discussing research methodologies into the application of machine learning processes in football and works directly related to the problem of predicting the next action. A detailed description of the dataset used is then provided along with comprehensive information on the machine learning architectures used for the task. Following the description of data and architecture, results obtained from the research process are presented. Here, the results are discussed focusing on the insights drawn from analysing our models' performances. Finally, the paper is concluded by providing a summary of the research, findings and potential future improvements.

2. LITERATURE REVIEW

2.1 Football Analytics and Machine Learning

Reep & Benjamin (1968) work is believed to be the first introduction of analytics to football. They analysed data gathered from over 500 football matches with the goal of investigating passing movements, using pencil and paper. Their findings showed over 90% of plays took less than four passes, concluding that the success of passes in a sequence of plays is dependent on the quantity of passes. This theory went on to influence English football for years creating a style of play called Long Ball or Route One.

Analytics in football has advanced since then, becoming more complex. Hvattum & Arntzen (2010) investigated the use of ELO ratings to predict the results of football matches. The ELO rating system is most used in chess tournaments to calculate the skill levels or strength of players. Covariates were produced from implementing this system and then modelled using logit regression methods, team were provided ELO ratings based on their current strength and using information from previous matches played. Their results showed that this technique was competitive against six benchmark prediction methods.

Decroos et al., (2019) studied player evaluations using event data from football matches and developed the Valuing Actions by Estimating Probabilities framework. Their goal was to address the subjective nature of traditional player ratings by taking advantage of the power of machine learning in quantifying the relative importance of every action done on the ball. They by calculating scoring and conceding probabilities which are then used to compute action values. These action values are then converted to player ratings which represent their total offensive and defensive contributions. All of this is done by training data for 3 and 4 years using CatBoost algorithm and predicting the next years.

The Expected Goals (xG) metric initially used in Hockey (Macdonald., 2012), to correctly estimate the number of goals a team should score using various in-game information has become one of the most influential metrics in modern advanced football analytics. Pollard et al. (2004) first provided proof of factors that affect the probability of scoring a goal from a single shot. They applied Logistic Regression to develop a model that estimated the

probability of a shot being scored. Their model showed that the probability of scoring a goal was greatly influenced by distance from goal (for every one yard away from goal, odds of scoring reduced by 15%), the angle from which the shot is taken relative to the goal post (for every one degree away from goal, odds of scoring reduced by 2%) and the distance from the nearest opponent at the time of the shot. Their results further emphasized the importance of moving the ball into high scoring positions on the pitch before taking a shot.

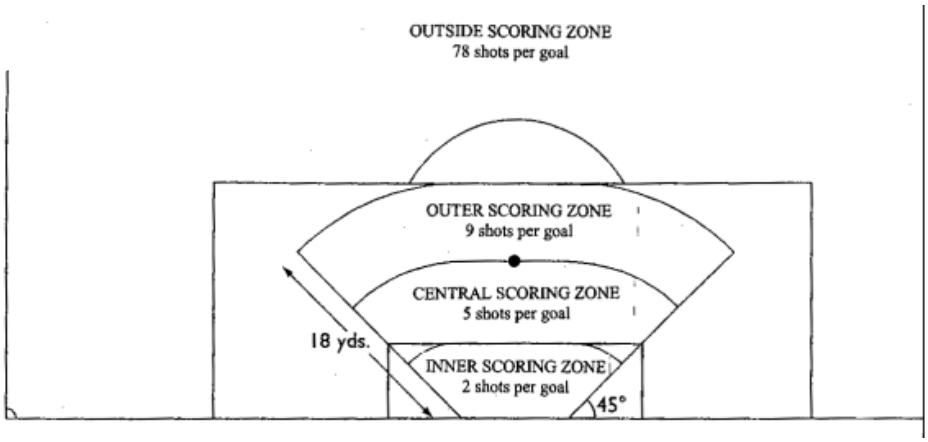


Figure 2.1: *Scoring zone with average number of shots per goal from different locations (Pollard et al., 2004)*

Eggels, H.H. (2016) was one of the first to introduce xG as a metric in his paper "Expected Goals in soccer: Explaining match results using predictive analytics". He investigated the quality of goal scoring chances by applying different algorithms including Random Forest and ADA-Boost on a combination of Tactical, Spatio-Temporal and Player data. His model went beyond the distance metrics described by Pollard et al. (2004), adding information like the state of play, body part used etc. His results showed that his model could accurately predict the score of a match in with over 50% accuracy, falling short in matches that mostly ended in draws.

Le, Carr, Yue & Liu (2017) employed deep learning to investigate the tactical behaviour of teams. Their research titled “Data-Driven Ghosting Using Deep Imitation Learning” used tracking technology to analyse how players behave in several scenarios, especially defensive situations. Their work used a technique called deep imitation learning applied to player tracking data, which produced results using a data focused system that answered the

question “how should this player or team have played in a given game situation compared to the league average?”.

Hirano & Tsumoto (2005) applied rough clustering together with multi-scale structural matching to efficiently cluster spatio-temporal pass trajectories in football matches. Their combination of methods was to handle problems of local disturbance of a dissimilarity matrix and the irregular sampling intervals and irregular lengths of playing sequences. The combination of these methods resulted in the successful discovery of patterns in passing during football matches.

Vidal-Codina et al., (2022) introduces a deterministic decision-tree based algorithm for automating the identification of events in football tracking data. The consists of Possession Step able to identify which players had the ball for every frame of the tracking data and identify unique player arrangements when the ball is not in play, aiding the detection of set-pieces. It also possesses an Event Detection Step that combines ball possession change with the rules of the sport to identify in-game events. Their model was able to correctly identify events over 90% of the time when compared to manually annotated events across multiple football competitions.

2.2 Predicting Actions/Events in Football

While the prediction of actions or events in football matches remains a challenging problem, some effort and progress have been made towards tackling this problem using sequential event data. Fernandez et al. (2019) have proposed a deep learning framework for Football analytics that provides an Expected Possession Value (EPV) for each event in a sequence. This model has the potential to be improved to predict the next event based on the current EPV and the context of the game.

The EPV model uses logistic regression to estimate the probability of passing or loss of possession. It uses CNN architecture with a combination of information to predict the most likely action to be carried out. Deep Neural Network designed with caution is used to learn passes and running with the ball. The decoupled nature of the proposed model provides the opportunity to investigate decision-making process on a situation-by-situation case.

One of the pioneering works that tackles this specific problem is the ***Seq2Event model*** proposed by Simpson et al. (2022). This model applies a Neural Network based approach to predict the next location and type of action of a match event in a football match. The paper also introduces a novel metric called ***poss-util*** which evaluates attacking possession utilisation of any given possession at any time during a match through the application of a context-aware model. The model was able to demonstrate some ability towards characterising team attacking behaviour using real-world data from what are considered the top 5 league competitions.

Simpson et al. (2022) designed models using Auto Regressive and Markov chain models to serve as baseline for the research. They found that these models performed worse than the Transformer and LSTM models designed. Their unidirectional LSTM model with one layer was found to be the best performing model in predicting the next action in a sequence of attacking actions. Their model focused on predicting four actions, Pass, Dribble, Cross and Shot. Interestingly, their results show their highest accuracy on any of these actions was 54%, predicting Shots as the next event.

Another important and recent contribution to this field is the Transformer-Based Neural Marked Spatio Temporal Point Process Model (NMSTPP) proposed by Yeung et al. (2023). This research is built on the work done by Simpson et al. (2022), the model was designed to handle three different problems; temporal, spatial, and action type, of football events simultaneously and dependently. The authors also introduced a new performance evaluation metric for teams called ***Holistic Possession Utilisation Score (HPUS),*** this improves on the ***poss-util*** in order to evaluate the effectiveness and efficiency of a team's possession during a match.

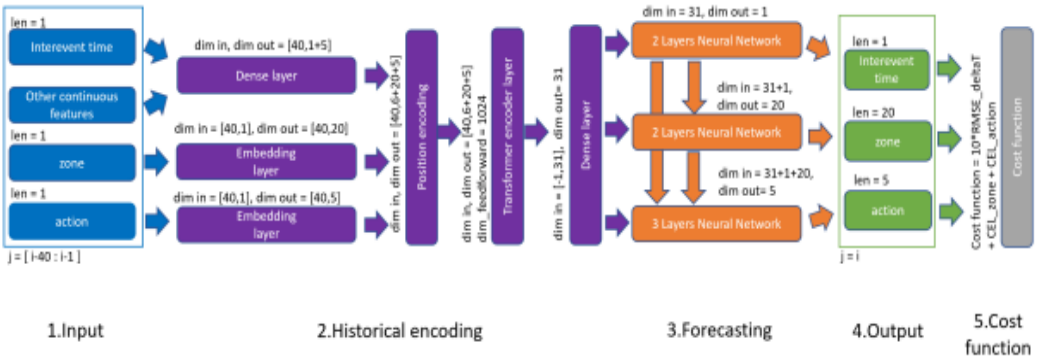


Figure 2.2: *The 5 stages of the NMSTPP Transformer model used in predicting the next action of a football match (Yeung et al., 2023).*

The NMSTPP research made use of the same data as used in the Seq2Event model, data provided by WyScout containing event data from 2017/2018 Season from five European leagues (Pappalardo et al., 2019). In their research, instead of predicting the exact co-ordinates done by Seq2Event, they divided the pitch into zones using the Juego de Posicion philosophy made popular today by Pep Guardiola. The performance of the model was observed and found to have a total loss of 4.40, while the confusion matrix produced from testing shows that while the model had a high accuracy of 65% in correctly predicting Shots, the model had less than 50% accuracy in predicting Passes, Dribbles and Crosses.

However, this remains a relatively new and less explored area compared to other aspects of football analytics, such as match outcome prediction and player performance evaluation and metric designs which have for a very long time been the focus of attention in Football Analytics. Therefore, there is still considerable room for exploration and improvement in the prediction of next actions/events in football matches using available event data logs.

A comprehensive summary of how football analytics has evolved from the use of pen paper to the application of sophisticated algorithms has been covered. These works has been instrumental in the choice of this task, allowing us to opt for a problem which has been paid little to no attention while having a potentially large impact on changing the way football teams play their football. The application of deep learning techniques implemented by Simpson et al. (2022) and Yeung et al. (2023) have shown promising in predicting the next

action, this research intends to improve their methods by learning playing pattern of play or team style in predicting the next action.

This work hopes to differ and intends to critic the work done by these two papers with the use of a different dataset and changes in the data processing stages. This difference stems from the hypothesis that the introduction of more information about events will be beneficial in increasing the accuracy of a model, in predicting the next action in a football match. This change is implemented by introducing team-dependent information into the training model, player information as well as some in-game information like "pressure" which informs us if an action is carried out while under pressure from an opponent.

While this work focuses on just the next action, this research can be potentially important in improving the possession evaluation metrics of Simpson et al. (2022) and Yeung et al. (2023). Expected Goals metric is now a very important statistic in football, as shown by Yeung et al. (2023) work and HPUS metric, possession evaluation through predicting the next action can surpass the power of xG in predicting and evaluating team performance. In essence, the study aims to serve as a source of information and inspiration by providing innovative insights that can change the way some metrics are evaluated.

3. METHODOLOGY

The essence of this research goes beyond the results and findings but in the systematic methods used to achieve those findings. This methodology section breaks down the procedures, techniques and tools used in this research to answer how machine learning can be effectively applied to predict the next action in a football match.

The objective of this section is to describe in detail the data collection, data preparation, preprocessing, model architecture, training steps and evaluation metrics used in this research. By providing this information we have ensured that this research can be reproducible while also contributing to the wider discourse on the best practices in tackling similar problems or research. These details provided will help you, the reader, understand how from raw match data, we were able to draw and provide meaningful insights from our findings.

All the experimentation done in this research was done using Python and its libraries. Work moved between working locally in Jupyter Notebook for data collection and preparation. While Google Colab and Kaggle Virtual Machines were used for training and testing because of their GPU availability, to speed up training times on the models.

3.1 Data Processing

3.1.1 Data Collection and Data Description

The data used for this research is football event data sourced from the StatsBomb Open Data Repository available on GitHub. StatsBomb is one of the most popular Football and Sports analytics companies today. They are known for their commitment to the open data initiative, constantly sharing new data to drive research and understanding of Football for researchers and lovers of the game.

The decision to pivot from the more popular available WyScout dataset used in previous football analytics research was driven by the quality and comprehensiveness of the data provided by StatsBomb. StatsBomb data provides much more detailed information for each event in every match event data made available, capturing intricate details that are essential

for predicting the next action. One of these standout features is the provision of 360-event information. This provides spatial data that highlights the location of teammates and opposition players within the camera frame during a particular event. This type of information is important in understanding the context in which an action takes place, providing insights into the decision making of players and team dynamics. Incomplete 360 event information. Their data gathering processes involve the use of computer vision algorithms as well as human input as an extra layer of validation. The information collected is then tagged with metadata that appropriately describes events.

StatsBomb Open Data Repository provided a vast amount of match data for different leagues and competitions. For the purposes of this research, the data was filtered to contain only match event data with 360 information. All the datasets were provided in JSON format and documentation was provided to avoid ambiguities about the meaning of features and variables.

The available 360 event datasets were manually downloaded from the GitHub repository and stored in a single folder which was then accessed for pre-processing. Each 360 event JSON file contained only the 360 information previously described as well as the event id. Each file name was named using the match id of the specific match.


```
[ {
  "event_uuid": "97a8f158-6d73-42c8-a49c-6cd186b3a106",
  "visible_area": [37, 80, 25, 0, 60, 0, 42, 80, 37, 80],
  "freeze_frame": [
    {
      "location": [ 104.2, 54.0 ],
      "teammate": true,
      "actor": true,
      "keeper": false
    },
    {
      "location": [ 104.2, 54.0 ],
      "teammate": false,
      "actor": false,
      "keeper": true
    }
  ]
}]
```


Figure 3.1: *Data, JSON structure of the 360-event dataset containing, visible area and team member or opposition player locations (StatsBomb)*

StatsBomb python library, statsbombpy, was installed and used to process and transform the data. The library contains several callable methods to provide information and data. Competitions, matches methods along with Python's Pandas library were used with the file names of downloaded files, to produce full match event data of matches that contained 360-event data. This left us with data from three National team competitions; FIFA World Cup - 2022, UEFA Euro – 2020, UEFA Women's Euro – 2022 and 146 from these three competitions. These three competitions served as the basis for the building the model used to tackle the problem of predicting the next action.

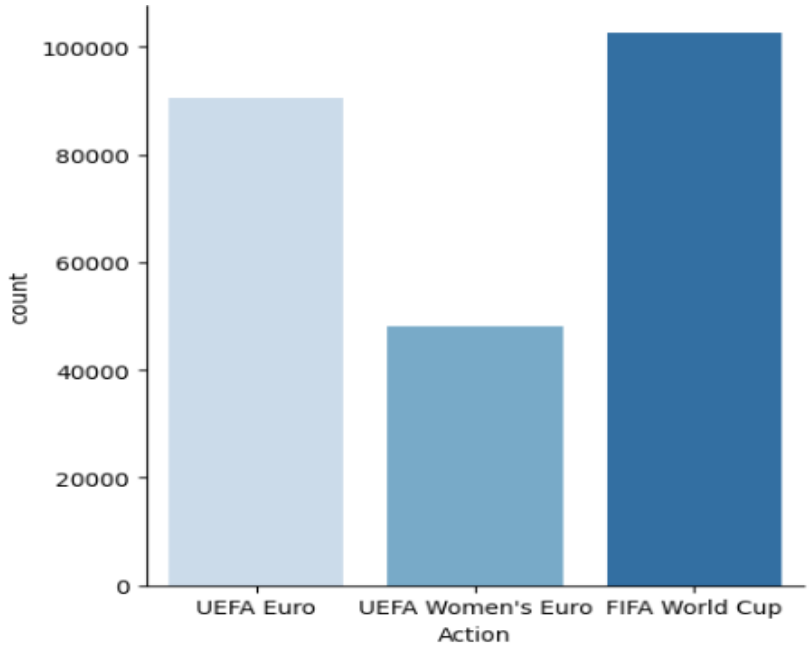


Figure 3.2: *Distribution of the number of matches contained in the 3 competitions used in this research.*

3.1.2 Data Cleaning and Feature Extraction

This step plays a very important role in the success of the research. The quality of preparation done on any dataset is determinant of how well the model produced can be generalised in the real world. In this step we work on carefully curating the dataset to ensure that it is pristine, contains no errors or miscalculations before being fed into the model for training and testing.

The entire match event dataset contained over 100 features which were largely categorical in nature, describing each event where necessary and over 530 thousand events. While the 360-event dataset had just 4 features; the match id, event id, the visible area of the pitch in camera and freeze frame describing the location of players in the free frame.

Due to the size of the match event data set, NaN values were prevalent in a lot of events. The documentation provided by StatsBomb provided meaningful descriptions of each feature, combined with domain knowledge of football helped in dropping features that would be of no importance to the task at hand. Various cleaning and preprocessing techniques from splitting columns, replacing null values and calculations for new features were carried out to ensure the dataset contained the appropriate information needed for modelling.

The action type of each event which is the main feature of importance contained over 40 types of actions, ranging from actions carried out by players, referees and substitutions. Using domain knowledge and consulting with other football enthusiasts, some events were combined and events with action type not important to this task were excluded, leaving Passes, Crosses, Dribbles, Shots.

- **Pass:** This is the act of moving the ball from one location to another between players. It is done by kicking the ball with the intention of just moving the ball.
- **Cross:** This is a type of pass usually made with the aim of attacking towards the opponent's goal. It usually involves moving the ball off the ground from one side of the pitch to the opposite side.
- **Shot:** This involves hitting the ball with great power, usually with the aim of scoring. It can also be used as a means of kicking the ball away from danger, i.e a team clearing the ball away from their goal.
- **Dribble:** This is the act of moving, running with the ball. It can also involve the use of special skills to evade opposition players.

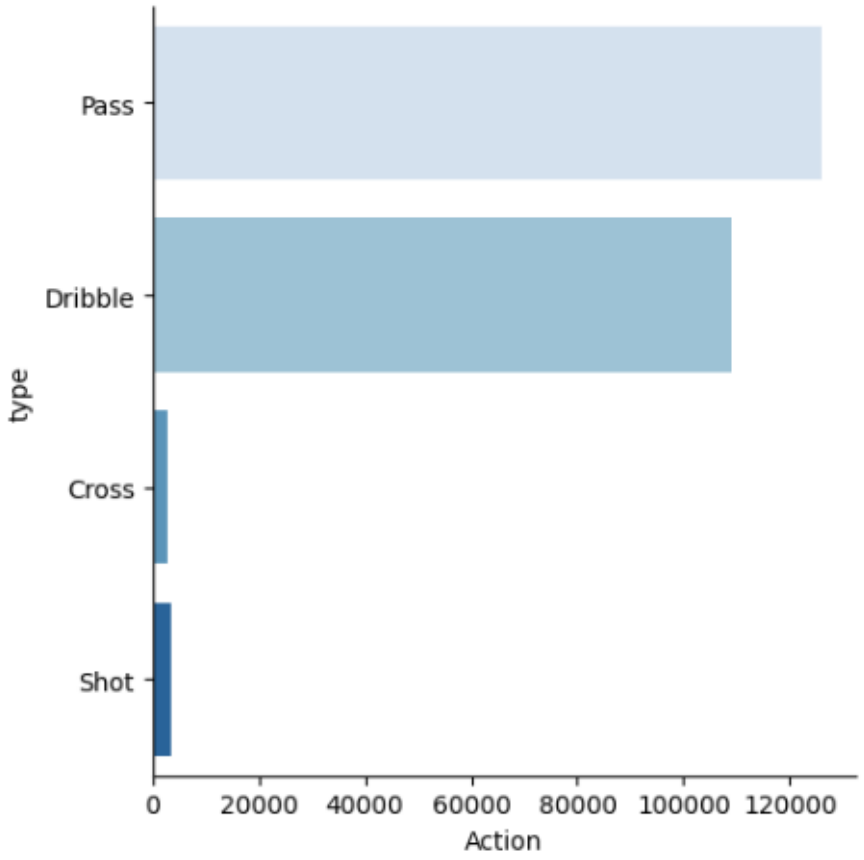


Figure 3.3: *Frequency distribution of the target classes, highlighting data imbalance in these variables.*

The 360-event data was merged with the match event data on match id and event id. Rows or events where 360 information was unavailable were removed from the dataset. To reduce dimension size, the 360 information was aggregated into 4 columns; number of teammates, average distance of teammates from the action, number of opposition and average distance of opposition from the action. A new column was also created to also represent the next action after a preceding event.

Integer Encoding was done to replace string values in some features with integer values suitable for machine learning models to understand. Different techniques were used here like the rank method, combining mapping with the get indexer method on previously created dictionaries. Integer encoding was opted for over One-Hot encoding to avoid over-extending the dimension of the dataset which would be beneficial in reducing training time and other preprocessing techniques. These integer encoding techniques were applied to categorical

columns that described the following; action type, formation, team id, position, play pattern etc.

Feature	Description
match_id	This is a unique number given to differentiate between matches
under_pressure	Describes if an action was done under pressure from an opposition player or players
zone	Describes the square out of 30 squares where an event occurred.
type	Describes the action that has been done during an event.
possession_team_id	Unique number to differentiate between teams in possession of the ball.
position	Describes the position or role of the player in possession of the ball.
formation	Describes the tactical formation employed by a team during the event.
duration	the time taken to complete an action.
avg_teammate_dist	Average distance of teammates of the player in possession.
avg_opponent_dist	Average distance of opposition players from the player in possession.
num_teammates	Number of teammates in the frame during an event.
num_opponents	Number of the opposition players in the frame during an event.
x0, y0	Coordinates of the player in possession on the x and y axis
Zone_deltay	Change in coordinate on y-axis using the zone
Zone_deltax	Change in coordinate on x-axis using the zone
Zone_dist	Distance between zones
Zone_dist2g	Distance from zone to goal
zone_angle2g	Angle from zone to goal

Table 3.1: *Cleaned, final data features and their descriptions used in the modelling of the LSTM and Transformer models.*

3.1.3 Splitting and Sequence Structure

To ensure the development of an effective model and its evaluation, data has to be divided into three parts; training set, validation set and test set. The data was split using the competition name and match id to avoid breaking the sequence of matches which is very important for the goal of the research.

80% of the cleaned data allocated to the training set. This portion was used for modelling the framework capable of predicting the next action. A large amount of training data is required for the model to have enough information to find patterns and learn from. 10% was allocated to the validation set. This set serves as the initial evaluation set after every epoch of training.

The evaluation results from this set are used to adjust hyperparameters to avoid underfitting and overfitting of the model on the training set and ensure generalisation of the model. The test set was allocated the remaining 10% of the cleaned data. This set provides us with the final and actual evaluation of the model, the performance metrics obtained on this set inform us about how well our model performs on new, unseen and possibly real-world data.

After splitting, scaling is done on the three datasets. A MinMaxScaler method is used here, the model is fitted on only the training dataset and used to transform the training, validation and test sets. The approach is done to avoid data leakage from the test and validation set into the training set.

Due to the sequential nature of football matches and the task in predicting the next action after a series of actions, each of the three datasets was restructured. Windowing is a technique used to divide data into overlapping or non-overlapping windows of fixed sizes. It is commonly used in time series data where prediction of the future is needed using past observations. A window length is set as well as time step of 1. This means that every window contains rows equal to the size of window length and the window is moved by one step or row to create a new window.

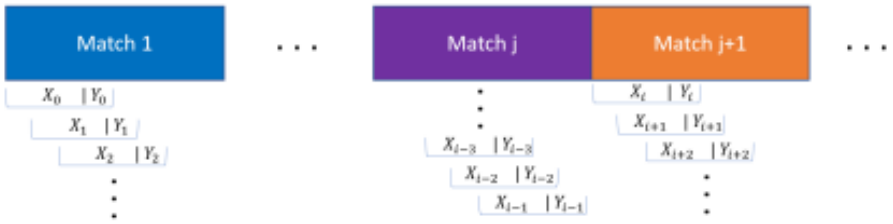


Figure 3.4: *Windowing method for creating sequences of events to input into the model (Yeung et al., 2023).*

3.2 Primary Model Development

In machine learning there are a variety of algorithms available to handle a wide range of problems. Our model selection was based on the type of data, time series data, being used as well as work that has been previously done in prediction next action in football matches. The LSTM framework (Long Short-Term Memory) was selected based on these considerations, its ability to handle time series data. LSTMs belong to the family of Recurrent Neural

Networks (RNN), they are suited for predicting problems that rely on sequential information because of their ability to remember patterns over long sequences. This makes them ideal for the task.

The LSTM deep learning model was designed using the Keras library in Python. The model was designed to handle the categorical and numerical input features of the StatsBomb dataset. The model was designed to have 7 inputs and 4 outputs. 6 of inputs were the categorical features and the last all the numerical features combined as one. The final model was trained over 16 epochs and had a total of 22,848 trainable parameters and 0 non-trainable parameters.

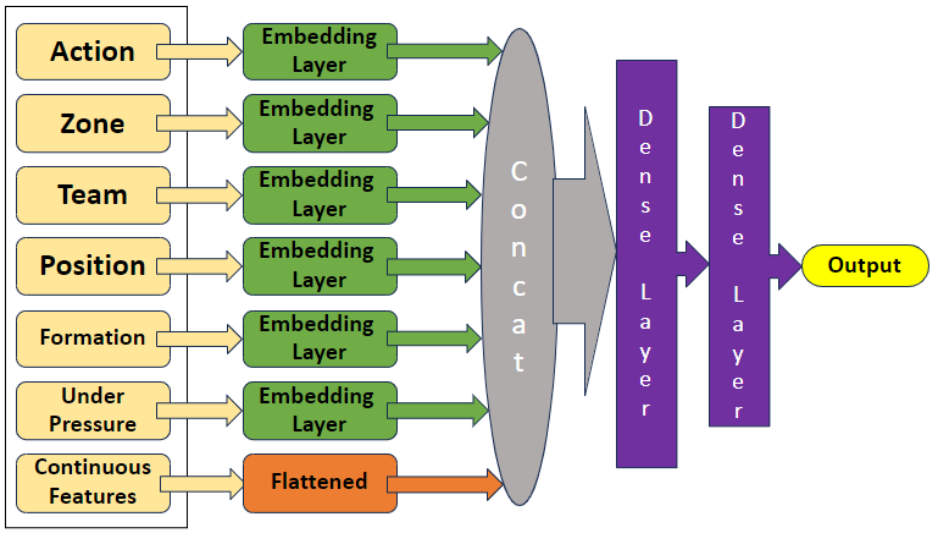


Figure 3.5: *Our LSTM architecture displaying the input features and the various processing stages before producing an output, next action.*

3.2.1 Embedding Layer

Considering that the data of choice provided by StatsBomb contains a series of categorical information valuable to predicting next action, embedding was employed to convert this categorical information into dense vector representations. Columns that went through embeddings are zone, action type, possession team id, position and formation.

Embedding was the best choice to handle these features over One-Hot encoding. Embeddings provide a reduction in dimensionality that One-Hot encoding does not which allows for

lower training time. Embeddings also have the potential to provide context and meaning representation of categories to the model which can help improve the accuracy of the model.

3.2.2 Model Architecture

- **Embedding Layer:** Each of the features represented by categorical variables were passed through their respective embedding layer. The output from each embedding process was then passed through a single LSTM layer with 16 units and a ReLu (Rectified Linear Unit) activation function. Passing each of the embeddings through an LSTM layer was done to allow the model capture temporal dynamics of each feature, allowing the model to learn patterns in the sequence of each feature. Each embedding was set to have an input equal to the number of unique variables in that feature, while embedding output was set to be the minimum number between half of the number of unique variables and 50, this was done to reduce the size of the embedding vectors. Feature embedding shape and input was dependent on the size of the window used.

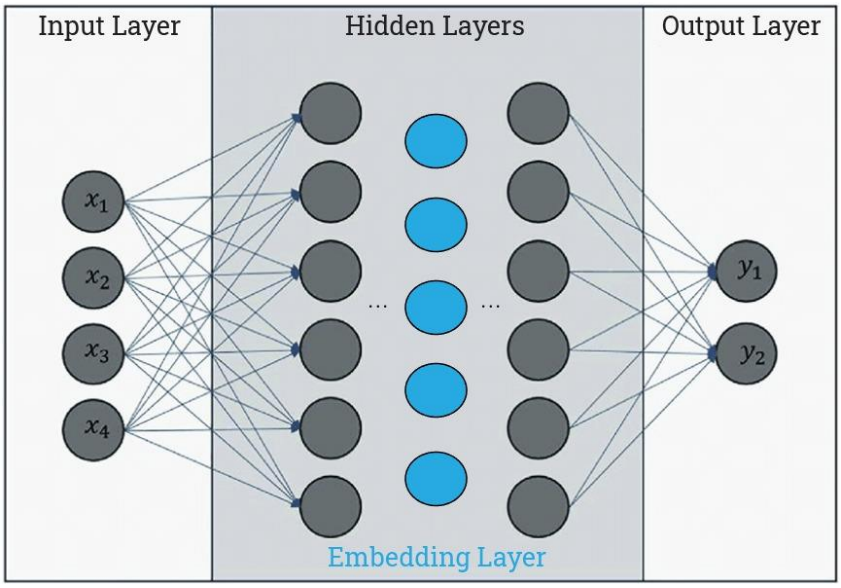


Figure 3.6: *Embedding process of the Embedding Layer in the LSTM model, outputs half the dimension from the input (Zvornicanin, 2023).*

- **Numerical Inputs:** A different input layer was set for features represented by numerical inputs, floats. This input was flattened after shaping as (window, number of numerical features). Flattening was done to enable concatenation of the numerical

sequences to the vector outputs generated from the LSTM embeddings. This method allowed us to disregard temporal relationships between time steps. Doing this appeared to perform better than when passing the numerical inputs into a separate LSTM layer to capture temporal relationships.

- **Dense Layer:** Outputs from the embedding LSTM and flattened numerical data were concatenated to form a single representation of the dataset. This concatenated data was passed through two dense layers. The first dense layer was fed the concatenation of features, it consists of 16 nodes as well as ReLu activation function. The output of the first dense layer is passed into the second dense layer with 8 nodes and a ReLu activation function. The purpose of the ReLu activation function is to capture more complex patterns by introducing non-linearity into the model.

$$f(x) = \max(0, x) \qquad \textit{(Equation 3.1, ReLu Activation)}$$

- **Output Layer:** This layer is the final layer and is responsible for producing the prediction from passing the data through all the previous layers. This layer is designed as a dense layer with 4 nodes, representing the four target actions. In each sequence, four outputs are generated, probability is assigned to each of the outputs and the variable with the highest probability is collected as the final prediction by the model. These probabilities are produced by implementing a softmax activation function in the output layer.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad \textit{(Equation 3.2, Softmax)}$$

3.2.3 Training

1. **Class Weight:** The target class in the dataset used for this research contained a very distinct class imbalance. Class weights were calculated using the classweight function provided in the sklearn library. This function uses the inverse proportion weighting method to calculate the weights of each class. Application of this weight was to reduce prediction bias towards the more common target classes, Passes and Dribbles. These

weights allow the minority class, Crosses and Shots, more importance in order to handle the class imbalance and also improve prediction accuracy of the model.

$$w_j = \frac{n}{k \times n_j}$$ *(Equation 3.3, Inverse Proportion)*

Action	Weights
Pass	0.47707
Dribble	0.55549
Cross	20.99232
Shot	17.83910

Table 3.2: *Calculated weights, CEL bias for each target class.*

2. **Model Compilation:** Adam optimiser, a common optimisation algorithm in deep neural networks was used to compile the model. It combines two other optimisation algorithms. The loss function used to track how well and if the model was doing a good job at learning, is Sparse Categorical Entropy. This was used due to the multi-class nature of the target variables. Sparse Categorical Accuracy was used to measure the quality of prediction in the model. These two metrics were applied to training and validation set, with the performance on validation used judge the model's ability during compilation state.

$$L_i = -\log(p_i, y_i)$$ *(Equation 3.4, Sparce Crossentropy)*

3. **Early Stopping:** Early stopping callback was implemented during the training stage. This is a method of regularisation that stops the model from training further when performance on the validation set stops improving after a certain number of epochs. This helps to handle overfitting or model degradation which was a common observation during training before the introduction of callback. Validation loss was tracked, and patience was set to 10 epochs. This stopped the training process if there was no improvement after obtaining the best loss up to that point. The model's weight at the final epoch of training was retained because of fluctuations of validation loss during training. This prevented bias of say picking the best weights from the 4th epoch of training, retaining the weights from the final epoch of training provided balance.

3.2.4 Hyperparameter Tuning

This process of the experimentation involved changing certain components of the training process to find the best hyperparameters that optimise the model on a metric of interest. Training and Validation loss were monitored as the metric to optimise. This process was done manually, iterating hyperparameters based on random values, subjective knowledge of the model and task. This method was chosen because it managed computation cost better over exhaustive grid search or random search.

Hyperparameters tuned to varying degrees and different combinations include;

- *Number of Layers & Nodes*: These hyper-parameters were tuned to find a balance and manage underfitting & overfitting, computational resources as well as results.
- *LSTM Direction:* To find the model with the best loss and accuracy, different types of LSTM were tested. We weighed the benefits of processing sequences in both directions compared to one direction. Bidirectional LSTMs, during training look forward and backwards, relying on information in both directions in making predictions.
- *Embedding Process:* This part of the Hyperparameter Tuning was the longest. Here, exhaustive considerations were made by trying different methods of passing the data into the training model. This was done to ensure that the best possible predictive model was evaluated.
    - *Each Categorical Embedding to its own LSTM Layer*
    - *Each Categorical Embedding to its own Dense Layer*
    - *Concatenated Categorical Embedding to an LSTM Layer*
    - *Concatenated Categorical Embedding to a Dense Layer*
- *Dropout Rate:* Dropout rate was experimented with. Randomly dropping between 0 to 50% of nodes after an LSTM or Dense layer. This served as method of regularising the model to prevent overfitting. This was not included in the final model.
- *Learning Rate:* Multiple learning rate techniques were applied to the model to help improve loss during training and on validation. Learning rate was tuned by adjusting the Adam optimisation to several fixed values and also implementing a learning rate

scheduler. This scheduler was applied to help reduce degradation or stagnation of validation loss. It works by systematically reducing the learning rate of the model, allowing it to take smaller steps during training when the validation loss appears to have stagnated. The best learning rate was found to be the default adam optimiser, the learning rate scheduler caused the training of the model to stagnate for long epochs immediately after the first to third epochs.

$$\theta = \theta - \frac{\alpha}{\sqrt{v}+\varepsilon} m \qquad \textit{(Equation 3.5, Adam Optimiser)}$$

- *Window/Sequence Length:* The length of sequential events used in embedding and training the model was also tested. Using an initial starting point of 40 sequences per batch and randomly varying to higher and lower sizes. Shorter windows were found to not perform well on the model while longer windows varied depending on the model architecture while slowing increasing training time. Window size of 40 was settled on because of the reasonable results obtained and training time being better when compared to larger sequence sizes.

3.3 Secondary Model Development

In addition to the LSTM model, a second model was designed using the Transformer architecture in predicting the next action in football matches. Transformers, just like LSTMs are advanced machine learning algorithms. They gained popularity due to their application in natural language tasks. Here we briefly describe the data processing and model framework for training our dataset on a Transformer model. This model was built using Pytorch, providing freedom in the designing of the model's complexities.

3.3.1 Data Processing

Categorical features were encoded for compatibility with the transformer model. These were converted into tensor forms and numerical features were standardised. These outputs were merged to form the input data for the model. The dataset underwent similar preprocessing as with the LSTM preprocessing, the dataset was structured into a series of sequences using a sliding window technique. This was done by using a class to ensure accurate windowing of

the input data. Pytorch's DataLoader class was used to facilitate feeding the data for training, validation and test set.

3.3.2 Model Architecture

Yeung et al. (2023) work served as the foundation in the design of this model. The model was designed as a class set to inherit from the nn.Module. Categorical features, the same as in the LSTM were embedded in separate layers. These embeddings help to capture the relationship between different categories. The embeddings are then concatenated with the numerical features which have gone through linear transformation to be standardised. To provide the model with an understanding of how order and position of actions in a sequence of events, positional encoding was added to the combined features.

The transformer encoder which serves as the most important part of this model is responsible for processing the concatenated inputs. It is made up of layers stacked on top of each other to form a deep network. Each layer is made up of a Multi-head Self Attention and Feed-Forward Neural Networks. These were set as values for hyperparameter tuning of the model. The multi-head head attention created multiple heads that are tasked with learning different patterns in the data. Feed-Forward Networks handle the output of the attention layer through hidden layers, introducing non-linearity to capture complex relationships and patterns. The output from the encoder is then passed through some linear transformation, this prepares the encoder's output for the prediction task. The last process is to pass through a linear layer that again transforms the processed information and outputs the 4 classes.

Training was done over 50 epochs using the adam optimiser to update parameters along with a learning rate scheduler to automatically adjust learning rate during training. The loss monitoring was done by focusing on the Cross Entropy Loss generated during each epoch of training.

Hyperparameter tuning was tested on the model with little freedom due to computing power limitations. Embedding outputs of each categorical feature, the number of hidden dimensions, number of attention layers as well as the number of encoder layers were adjusted as much as possible given the limited freedom. The final model used for predictions

was set up with 521 hidden dimensions, 2 layers and 29 multi-head attention layers, a window length 40 was also set.

4. RESULTS & DISCUSSIONS

In this section we discuss the performance of the model used for final prediction in predicting the next action a team will make in a football match. We analyse and discuss the observations and findings made throughout the research process. We also compare the results obtained from our model against the results obtained from the two previous research, Simpson et al. (2022) and Yeung et al. (2023).

4.1 Training Evaluation

As discussed previously, the Sparce categorical cross-entropy loss and Sparce categorical cross-entropy accuracy were both used for evaluation during training of the LSTM model. During training hyperparameters were adjusted continuously to find the model that has the best training loss and performance on validation dataset.

Here we will focus on how different model structures performed on the training set and how this influenced their performance on the validation set in predicting the next action.

1. **No Layer after Embedding:** For this variation of the model, all the embeddings created from individually embedding each categorical value were concatenated with the numerical features. This was done without passing the embeddings through any form of Neural Networks layer first. The concatenated results were then passed through a series of LSTM and Dense layers.

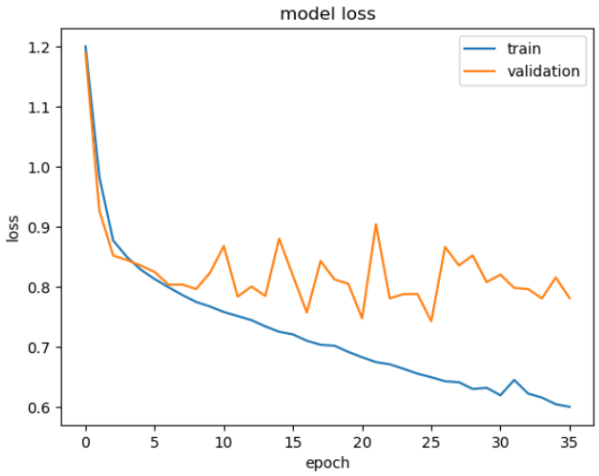


Figure 4.1: *No Layer after Embedding Sparce CEL on training and validation set*

This model showed too many fluctuations on validation set. This validation loss did not show that the model was consistently learning and improving on validation set.

2. **Dense Embeddings to LSTM Layers:** This model architecture involved passing individual embeddings of categorical features through a dense layer. These individual dense embeddings were merged with numerical features and passed through an LSTM layer.

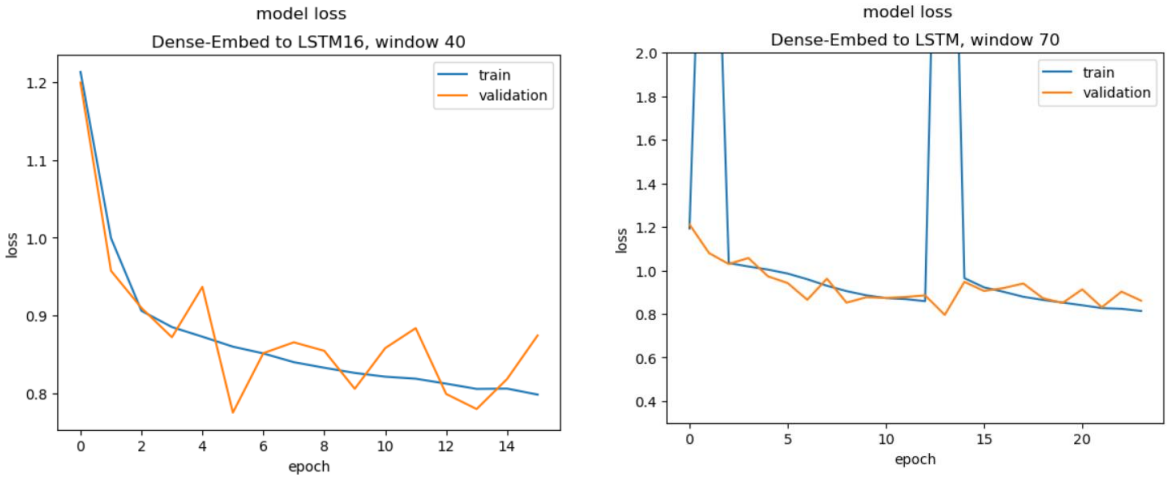


Figure 4.2: *Dense Embeddings to LSTM Layers Sparce CEL on training and validation set.*

These charts show that this model did a poor job at learning patterns when provided with team information and player roles. It performed worse in window size of 40 when compared to a window size of 70.

3. **LSTM Embeddings to LSTM Layers:** This model structure relied on the power of LSTM layers by first passing each individual embedding through an LSTM layer. Concatenating all the embeds with the numerical features before passing it through several/single layer of LSTM and Dense layers.

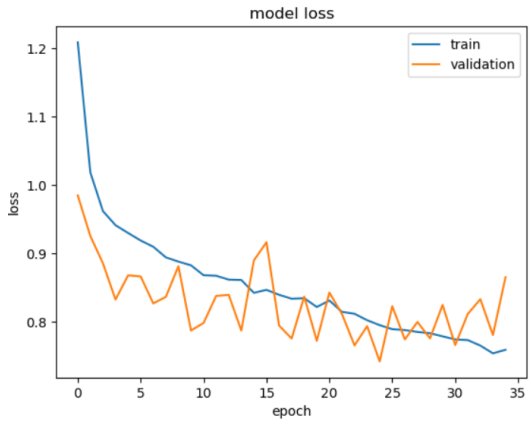


Figure 4.3: *LSTM Embeddings to LSTM Layers Sparce CEL on training and validation set.*

Similar to the first model discussed above, this model's loss contained too many fluctuations. The model appeared to be learning but the unstable nature of the model and increase on the last epoch made it a poor performing model.

4. **LSTM Embeddings to Dense Layer:** This was the best and final model used for prediction. This model architecture was designed by embedding each categorical feature and immediately passing through an LSTM layer. These LSTM embedding were concatenated with the numerical features and fed to Dense layers, excluding any form of LSTM layer before the output.

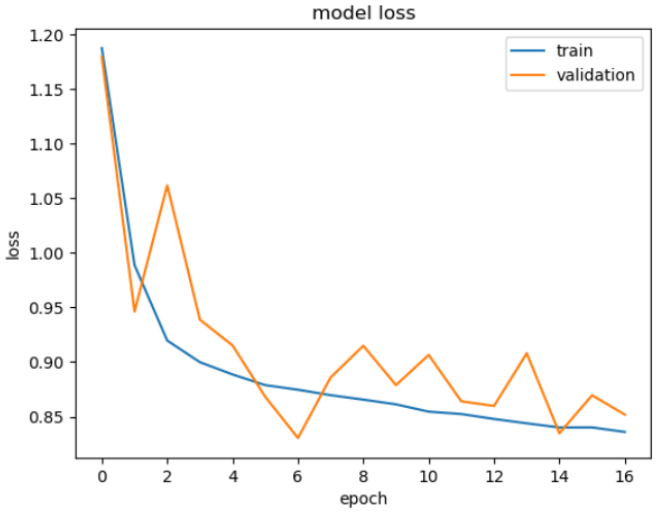


Figure 4.4: *LSTM Embeddings to Dense Layers Sparce CEL on training and validation set.*

The model selected as the best for this research in predicting the next action a team will make. This model also showed fluctuations when evaluated on validation set but unlike the previous models, the validation loss was largely above the training loss.

Model Description	Validation Loss
No Layer after Embedding	**0.7983**
Dense Embeddings to LSTM Layers	0.8762
LSTM Embeddings to LSTM Layers	0.8651
LSTM Embeddings to Dense Layer	*0.8519*

Table 4.1: *Validation CEL for all 4 models tested during hyperparameter tuning.*

The table above shows the loss values for all the models. The selected model had the second-best loss while. This was selected due to the learning plot of the model as well as its performance on test data, performing better on test data even with the higher loss value.

4.2 Test Evaluation

Evaluating the performance of the model on test set is the most important part of any research. This allows us to observe how effective the model is or would be on unseen data, further confirming the results gotten during training on validation set. This research implemented evaluation metrics most suitable for evaluating classification algorithms. The test set was evaluated using the following metrics;

1. **Accuracy:** Accuracy measures the proportion of predictions that have been predicted correctly regardless of the distribution of the target classes.

$$Accuracy = \frac{Number\ of\ Correct\ \text{Predictions}}{Total\ Number\ of\ \text{Predictions}} \quad \textit{(Equation 4.1)}$$

2. **Recall:** This is an evaluation metric that calculates the proportion true positives have been correctly predicted.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad \textit{(Equation 4.2)}$$

3. **F1-Score:** This metric combines Recall and another metric, Precision, to evaluate the harmonic mean of both.

$$F1 - Score = 2 \times \frac{\text{Pr}\,ecision \times Recall}{\text{Pr}\,ecision + Recall}$$ *(Equation 4.3)*

These three metrics combined to serve as the judging criteria for determining how well the model will perform on unseen data. Confusion matrix was also used to visualise the model's performance on test set. It will provide a pictorial representation of how the model performs in predicting each of the actions.

Model Description	Accuracy	F1-Score
No Layer after Embedding	0.6831	0.7204
Dense Embeddings to LSTM Layers	0.6895	0.7121
LSTM Embeddings to LSTM Layers	0.6535	0.7128
LSTM Embeddings to Dense Layer	***0.7020***	***0.7393***

Table 4.2: *Performance of all 4 LSTM models on Test set.*

The final and selected model showed that it could do a great job at predicting the next action of a team in a football match. The confusion matrix produced by this model on the test is shown below. The model showed its best performance in predicting a player attempting to run or dribble with the ball as the next action, recalling 81% of such actions. The model was also adept at predicting when a shot would be taken, this is particularly interesting given the class was the least represented in the dataset.

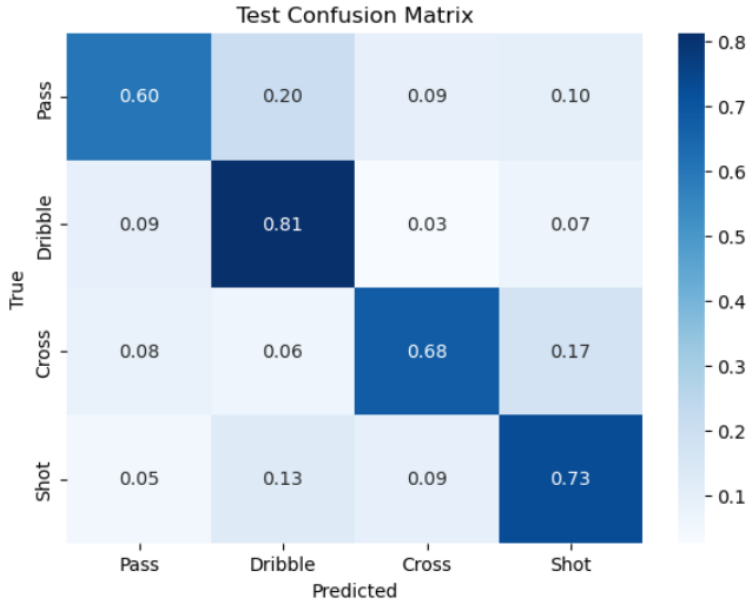
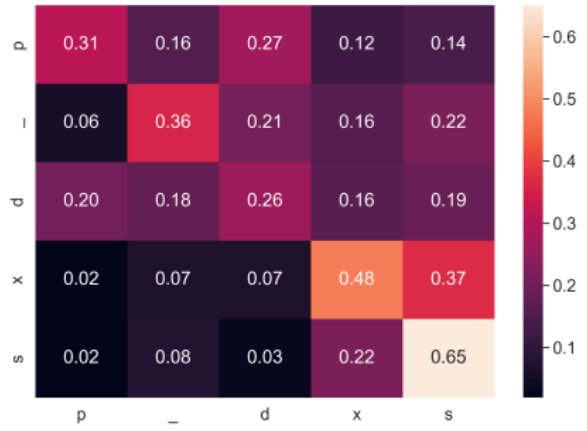


Figure 4.5: *Confusion Matrix highlighting proportion of accurate predictions against True values.*

Furthermore, the ability of the model to accurately recall the next action as a Pass or Cross was at an above average rate. Even with the overall strong performance of the model and high representation of Passes, the confusion matrix showed the model struggled in predicting Passes, misrepresenting 20% of them for dribbles.

From the context of football matches, this confusion might be attributed to the possibility that passes in attacking situations are used to move the ball closer to the oppositions goal, which is also usually the main reason for running with the ball or trying to dribble. It can also be because of the data structure, the sequence of most events in the dataset, where dribbles are usually followed by passes. This common sequence could have resulted in the model picking up a sequence bias. Also, because of the creativity that comes from players in playing football, the mislabeling of passes might be because of each player's unique decision making in breaking out patterns and drawing on their own inspirations.



	Predicted Action			
Actual Action	Pass	Dribble	Cross	Shot
Pass	0.36	0.38	0.16	0.10
Dribble	0.32	0.35	0.20	0.13
Cross	0.13	0.19	0.40	0.29
Shot	0.07	0.12	0.27	0.54

Figure 4.6: *Confusion Matrix displaying proportion of predictions for* ***(L)*** *NMSTPP (Yeung et al., 2023) and* ***(R)*** *Seq2Event (Simpson et al. 2022).*

In comparing the results obtained from our experiments with the results gotten from previous studies, we observed that our model does a better job of predicting the next action in a football match. While it is important to note that our research uses a different dataset as opposed to the WyScout dataset used in the previous papers, our model showed it had a better understanding of what would happen next with extra information provided in the data.

Our data processing steps also excluded the introduction of buffer events to indicate when possession has changed as well as end of matches. With the introduction of team information, the LSTM model is capable of understanding this nuance without the extra action indicating a change of possession. The lowest recall of any of our predicted classes, Passes, was just 5% lower than the highest recall observed by NMSTPP model, Shots.

Action\ Model	LSTM Model	NMSTPP	Seq2Event
Pass	0.60	0.31	0.36
Dribble	0.81	0.26	0.35
Cross	0.68	0.48	0.40
Shot	0.73	0.65	0.50

Table 4.3: *Recall comparison of our model, NMSTPP (Yeung et al., 2023) and Seq2Event (Simpson et al. 2022).*

One interesting observation is that all three models performed very well in predicting when a shot would be the next action. While it is generally believed that location is a big influence on taking shots in a football match, we believe the introduction of player roles further helped in improving the recall ability in this class. Since Strikers, Centre Forwards etc. are more likely to take shots the model could have picked up on this.

Another consideration could be the effects of the weighting system applied to the model due to the class imbalance of the target features. The weights designed gave more importance to Shot actions as it was the least represented class in the predicted feature.

4.3 Transformer Model Evaluation

The transformer models trained performed poorly when compared against the LSTM model. All the models trained had validation loss consistently increasing while accuracy continued to increase. The two figures below show how training and validation loss diverge but the accuracy continues to improve.

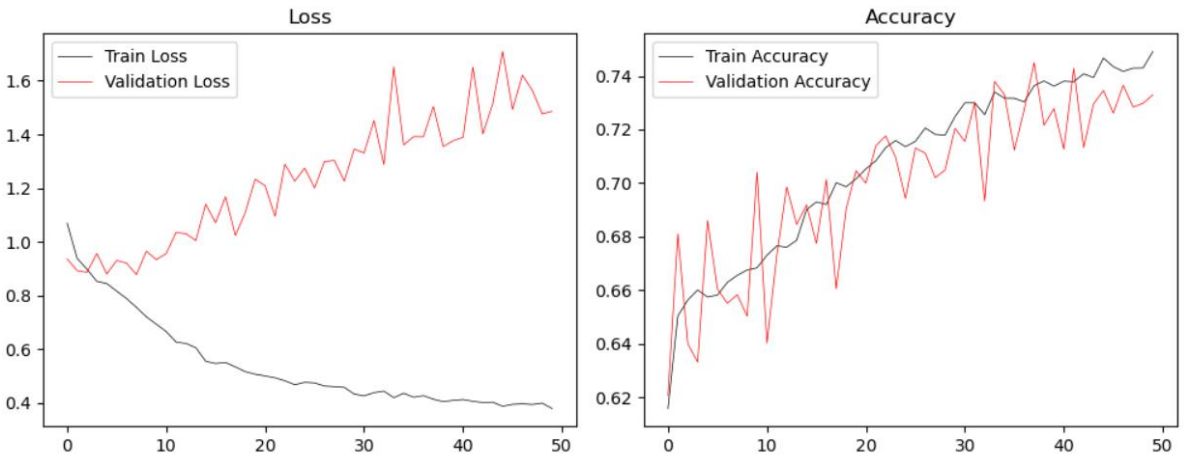


Figure 4.7: (L) *Loss vs Epoch and* **(R)** Accuracy vs Epoch, *during training of the Transformer model.*

This phenomenon displayed can be explained by a variety of reasons. The model could be learning too much on the training set and then it begins to overfit, causing CEL on the validation set to continuously increase on every epoch. The high accuracy can be down to the class imbalance in the dataset, making it look like the model is doing a good job of accurately predicting the next action. Another possible reason could be the model predicting with low confidence over the 4 target classes, the probability distribution amongst the four classes is fairly balanced, causing the CEL loss to increase.

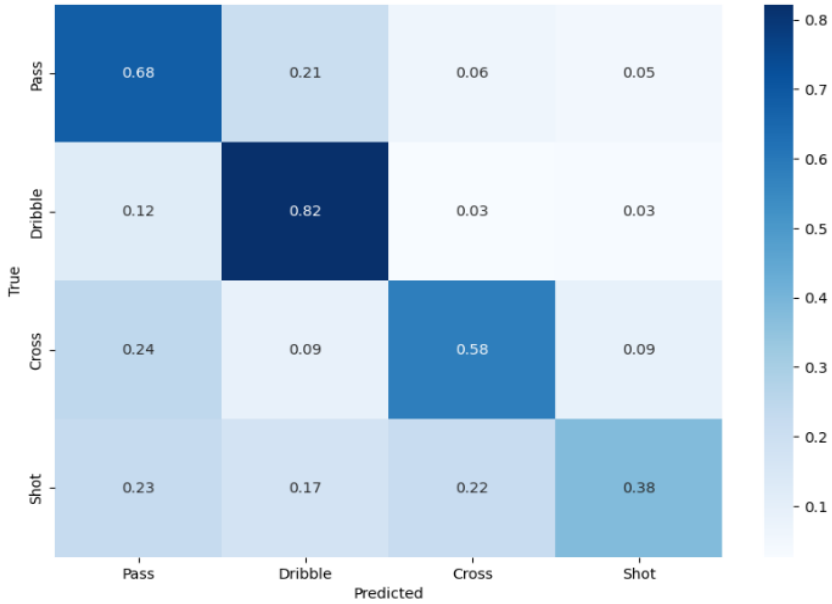


Figure 4.8: *Confusion matrix of Transformer model on the test set.*

The chart above shows the result of testing the Transformer model on the dataset. It produced an accuracy score of 74.08% and an F1-Score of 0.4761. The F1-Score confirms that the model is not adept using the information provided to learn patterns that can help to correctly predict the next action. The evaluation metrics combine to show that the model trained is biased in predicting the next action. From the test results, it is more probable that the data imbalance in the dataset is causing the prediction bias making the model predict most actions as a Pass or Dribble.

When compared against the LSTM model, all the LSTM models do a much better job of correctly predicting the next action. The LSTM's ability to recognise, remember and prioritise order might have played a key part in it outperforming the transformer model.

5. LIMITATION & FUTURE WORKS

The aim of this research was to improve on the work done by researchers in predicting the next action in a football match by introducing extra information. While this goal was achieved, the biggest challenge faced was access to bigger computing power. This lack of computing power affected the ability to programmatically experiment with exhaustive hyperparameter tuning. This also affected the ability to train all of the available data on some complex model structures, not allowing us to stretch experimenting beyond the tested LSTM frameworks. This was also a great limiting factor while training the Transformer model.

A lack of deep understanding of Neural Networks also was a limiting factor in this research. Although knowledgeable about the framework used, other Neural Network frameworks could have been implemented with the LSTM model. Applying Convolutional Neural Networks in handling player locations could better improve the results obtained from this research by providing richer and unaggregated data about location.

With access to larger dataset and computing power, player identity can be included in modelling a better model capable of capturing individual player nuances like their common actions, their style of play in certain situations.

6. CONCLUSION

In this research we took on the challenge of predicting the next action in football matches by implementing advanced machine learning techniques. This research focused on the introduction of team, player and in-game related information in achieving the goal. After a reasonable exhaustive evaluation on the model framework of these models, we found that a model built on the LSTM algorithm which processes categorical features by embedding them into LSTM layers before merging them with the numerical features produced the best results when evaluating predictive performance. This LSTM model showed that it could understand the introduction of team dynamics and player roles, outperforming our Transformer model and previous study in this area.

The final evaluation on previously unseen dataset showed the model had a strong affinity for predicting running/dribbling with the ball, shots and crosses with great accuracy. We successfully demonstrated the potential of introducing this new information in enhancing prediction accuracy while facing the challenge of limited computing power.

This research could be extended in future by including player identity information, providing the possibility of understanding more nuanced patterns and information of individual playing styles and decision making. Improving on the complexity of the model as well, by considering Convolutional Neural Networks, could improve prediction by capturing precise and actual spatial information, Overall, the body of work put together on this topic can contribute to the advancement of predictive analytics in sports by showing the efficiency of neural networks in capturing sequential patterns and the common knowledge that more data/features can improve performance.

7. REFERENCES

1. FIFA. (2022). FIFA World Cup Qatar 2022™ in numbers. FIFA Publications. https://publications.fifa.com/en/annual-report-2022/tournaments-and-events/fifa-world-cup-quatar-2022/fifa-world-cup-qatar-2022-in-numbers/ [Accessed 25 August 2023].
2. Machado, J., Alcântara, C., Palheta, C., Santos, J., Barreira, D., & Scaglia, A. (2016). The influence of rules manipulation on offensive patterns during small-sided and conditioned games in football. Motriz-revista De Educacao Fisica.
3. Miller-Dicks, Matt & Button, Chris & Davids, Keith. (2010). Availability of Advance Visual Information Constrains Association-Football Goalkeeping Performance during Penalty Kicks. Perception. 39. 1111-24. 10.1068/p6442.
4. Fernandez, J., Bornn, L., & Cervone, D. (2019). Decomposing the Immeasurable Sport: A deep learning expected possession value framework for soccer. In MIT Sloan Sports Analytics Conference.
5. Memmert, D., Lemmink, K. A., & Sampaio, J. (2017). Current approaches to tactical performance analyses in soccer using position data. Sports Medicine, 47(1), 1-10.
6. Rampinini, E., Impellizzeri, F. M., Castagna, C., Coutts, A. J., & Wisløff, U. (2009). Technical performance during soccer matches of the Italian Serie A league: Effect of fatigue and competitive level. Journal of science and medicine in sport, 12(1), 227-233.
7. Eggels, R. van Elk, and M. Pechenizkiy. (2016). Expected goals in soccer: Explaining match results using predictive analytics. In The Machine Learning and Data Mining for Sports Analytics Workshop (Vol. 16).
8. Macdonald, Brian. (2012). An Expected Goals Model for Evaluating NHL Teams and Players. Proceedings of the 2012 MIT Sloan Sports Analytics Conference.
9. Pollard, R., Ensum, J., & Taylor, S. (2004). Estimating the probability of a shot resulting in a goal: The effects of distance, angle and space. International Journal of Soccer and Science, 2(1), 50.
10. O'Neill, I. (2020). Coaching: Patterns of play in 4-2-3-1. Total Football Analysis. https://totalfootballanalysis.com/training-analysis/coaching-patterns-of-play-in-4-2-3-1-tactics [Accessed 28 August 2023].
11. Gudmundsson, J., & Horton, M. (2017). Spatio-temporal analysis of team sports. ACM Computing Surveys (CSUR), 50(2), 1-34.
12. Gudmundsson, J., & Wolle, T. (2010). Towards Automated Football Analysis: Algorithms and Data Structures.
13. Hvattum, L. M., & Arntzen, H. (2010). Using ELO ratings for match result prediction in association football. International Journal of Forecasting, 26(3), 460-470.
14. Pappalardo, L., Cintia, P., Rossi, A., Massucco, E., Ferragina, P., Pedreschi, D., & Giannotti, F. (2019). A public data set of spatio-temporal match events in soccer competitions. Scientific Data, 6. https://doi.org/10.1038/s41597-019-0247-7

15. Yeung, C. C., Sit, T., & Fujii, K. (2023). Transformer-Based Neural Marked Spatio Temporal Point Process Model for Football Match Events Analysis. arXiv preprint arXiv:2302.09276.
16. Zvornicanin, E. (2023). What Are Embedding Layers in Neural Networks? Baeldung. https://www.baeldung.com/cs/neural-nets-embedding-layers [Accessed 27 August 2023].
17. Simpson, I., Beal, R., Locke, D., & Norman, T. (2022). Seq2Event: Learning the Language of Soccer Using Transformer-based Match Event Prediction. Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.
18. Decroos, T., Bransen, L., Van Haaren, J. and Davis, J., 2019. Actions speak louder than goals: Valuing player actions in soccer. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp.1851-1861.
19. Lawrence, T., 2018. Introducing xGChain and xGBuildup. Statsbomb. Available at: https://statsbomb.com/articles/soccer/introducing-xgchain-and-xgbuildup/ [Accessed 17 August 2023].
20. Franks, I. M., & Miller, G. (1986). Eyewitness testimony in sport. Journal of Sport Behavior, 9(1), 38-45.
21. Rogalski, B., Dawson, B., Heasman, J., & Gabbett, T. J. (2013). Training and game loads and injury risk in elite Australian footballers. Journal of Science and Medicine in Sport, 16(6), 499-503.
22. Herold, M., Goes, F., Nopp, S., Bauer, P., Thompson, C., & Meyer, T. (2019). Machine learning in men's professional football: Current applications and future directions for improving attacking play. International Journal of Sports Science & Coaching, 0(0), 1–20. DOI: 10.1177/1747954119879350.
23. Le, H.M., Carr, P., Yue, Y., & Lucey, P. (2017). Data-Driven Ghosting using Deep Imitation Learning.
24. Hirano, S., & Tsumoto, S. (2005). Grouping of soccer game records by multiscale comparison technique and rough clustering. Fifth International Conference on Hybrid Intelligent Systems (HIS'05), 6 pp.-.
25. Vidal-Codina, F., Evans, N., El Fakir, B., & Billingham, J. (2022). Automatic event detection in football using tracking data. Sports Engineering, 25, 181 https://doi.org/10.1007/s12283-022-00381-6
26. Xin, L., & Xu, Y.-W. (2022). Application of the Multiple Regression Method in Football Tactical Analysis. Wireless Communications and Mobile Computing, 2022, Article ID 3787086, 10 pages.
27. Hillmer, S. C., & Tiao, G. C. (1982). An ARIMA-model-based approach to seasonal adjustment. Journal of the American Statistical Association, 77(377), 63-70.
28. Pappalardo, L., Cintia, P., Ferragina, P., Massucco, E., Pedreschi, D., & Giannotti, F. (2019). PlayeRank: Data-driven Performance Evaluation and Player Ranking in Soccer via a

Machine Learning Approach. ACM Transactions on Intelligent Systems and Technology, 10(5), 59
29. Low, B., Coutinho, D., Gonçalves, B. et al. A Systematic Review of Collective Tactical Behaviours in Football Using Positional Data. Sports Med 50, 343–385 (2020). https://doi.org/10.1007/s40279-019-01194-7
30. Cwiklinski, B., Giełczyk, A., & Choras, M. (2021). Who Will Score? A Machine Learning Approach to Supporting Football Team Building and Transfers. Entropy, 23, 90
31. StatsBomb,. StatsBomb Open Data. GitHub. https://github.com/statsbomb/open-data. [Accessed 30 June 2023]
32. Reep, C., & Benjamin, B. (1968). Skill and Chance in Association Football. Journal of the Royal Statistical Society. Series A (General), 131(4), 581-585.
33. Soccerment Research. (2020). The Football Analytics Handbook. Available at; https://soccerment.com/wp-content/uploads/2020/07/handbook fa.pdf. [Accessed 18 August 2023]
34. Beal, R., Chalkiadakis, G., Norman, T. J., & Ramchurn, S. D. (2020). Optimising Game Tactics for Football. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020)* (pp. 9 pages). Auckland, New Zealand, May 9–13, 2020. IFAAMAS.
35. G. Jin, "Teamwork for referees in football match," in 2011 International Conference on Physical Education and Society Management (Icpesm 2011), vol. 1no. 9, pp. 86–88, 2012
36. Beal, R., Chalkiadakis, G., Norman, T. J., & Ramchurn, S. D. (2020). Optimising Game Tactics for Football. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020)* (pp. 9 pages). Auckland, New Zealand, May 9–13, 2020. IFAAMAS.
37. Tunaru, R. S., & Viney, H. P. (2010). Valuations of Soccer Players from Statistical Performance Data. Journal of Quantitative Analysis in Sports, 6(2), Article 10. https://doi.org/10.2202/1559-0410.1238.
38. Zhang, Q., Zhang, X., Hu, H., Li, C., Lin, Y., & Ma, R. (2022). Sports match prediction model for training and exercise using attention-based LSTM network. Digital Communications and Networks, 8(4), 508-515. https://doi.org/10.1016/j.dcan.2021.08.008.
39. Li, H. (2020). Analysis on the construction of sports match prediction model using neural network. Soft Computing, 1-11.