Predictive Tool for American Football Defensive Positioning Using Machine Learning to Aid Coaches in Design of Offensive Formations and Plays

ALEJANDRO ESCOBAR, University of Calgary, Canada
JONATHAN HUDSON, University of Calgary, Canada

With the increasing availability of sports data through modern tracking technologies, the usage of sports analytics has become a large contributor to the success of professional sports teams. The analysis and visualization of game data has provided teams with the ability to extract components of designed plays that result in higher play success rates. This paper proposes a tool for aiding coaches in offensive play design for professional football teams. The tool utilizes a Long Short-Term Memory (LSTM) predictive model, coupled with a graphical user interface (GUI) to predict and visualize the trajectory of defensive players in response to offensive players. The tool will take the starting positions for each player, and the offensive route used as input to predict and plot the defensive trajectory. We use NFL player tracking data to train the LSTM model for accurate predictions of real game scenarios in professional football.

CCS Concepts: • **Computing methodologies → Neural networks**; • **Human-centered computing →** *Visualization.*

1 INTRODUCTION

The use of sports analytics and statistics has resulted in an evolution of the methods used in coaching teams to grant them an advantage against their opponents. With modern technology, player tracking data has become more available in recent years, which has introduced more opportunities for analytical projects. The National Football League (NFL) and NFL teams have used and continue to use sports data analysis for play design and coaching of their offensive and defensive teams [13]. The NFL has even hosted several sports analytics competitions, allowing participants to submit projects using the provided NFL dataset that includes player tracking data [7].

However, many of the previous projects have produced predictive or analytical models using the data, which an individual cannot conveniently use by themselves. The ability to integrate a predictive or analytical model into a straightforward tool would add value to individuals utilizing the tool, such as an NFL coach. This project attempts to provide a solution to that problem by coupling a graphical user interface (GUI) with a predictive neural network model.

The proposed tool in this report focuses on coached plays of single downs in American Football (refer to Section 1.1.1). To be specific, the tool predicts the trajectory of a defensive player on a play in response to the movement of the offensive player they are assigned to. The intention of this tool is to provide users with a visualization of the predicted movement of a defender throughout a play, given each player's starting position and the offensive route as input. This tool, at least in the non-private field of machine learning in sports, is novel and represents a move beyond using machine learning for descriptive statistics towards predictive assistance in adapting strategies.

This tool makes use of a Long Short-Term Memory neural network which is a type of recurrent neural network that is capable of learning order dependence in sequence prediction problems [12]. LSTM models are able to accurately predict trajectories by using sequential spatiotemporal data [17]. Related applications of LSTM for trajectory prediction problems include: vehicle trajectory prediction [2], human trajectory prediction [1], basketball shooting trajectory prediction [19], and more. This model uses NFL player tracking data provided from the 2023 Big Data Bowl competition for training and testing to ensure accurate predictions of real game scenarios in professional football.

The GUI component of this tool allows users to select a starting position for an offensive and defensive player pair, as well as the route of the offensive player. The application will produce a visualization of the predicted trajectory of the defensive player responding to the offensive player's route and other input. We use a K-Means clustering algorithm on the offensive sequences to identify common trajectories of offensive players. We display the different trajectories to a user which allows them to select frequently occurring offensive player routes in real NFL games. This component of the project provides an interactive visualization tool instead of a stand-alone predictive model.

Evaluation of this tool involves analyzing the accuracy of the model's predictions and the added value of the visualization produced by the GUI to target users, such as coaches. The success of this project depends on the applicability, accuracy, and convenience of the tool in its use cases. In addition to being a novel coaching tool, this project documents findings of exploratory data analysis (EDA), data preprocessing steps of the Big Data Bowl dataset, clustering of sequences to identify common offensive player routes, and the attempt to use this data for a trajectory prediction LSTM model. Assumptions on this dataset to build the model which are highlighted in this report also describe the limitations of the data and highlight possible improvements built upon the findings of this report.

1.1 Background

1.1.1 American Football. American Football is a game played by two teams of 11 players each on the field at one time, with the aim of outscoring the opposition [10]. Games are played primarily in a series of downs, where the offensive team attempts to progress the ball down the field and score a touchdown[1]. Downs are attempts to snap the ball into play and progress the ball down the field or attempt a point scoring play. Players line up in different formations on each play, with various skill-dependent, heterogeneous positions that determine their movements. This indicates that player ability and purpose in a play can differ for individual players and is a factor in our model.

1.1.2 Big Data Bowl. The Big Data Bowl is an annual sports analytics competition provided by the NFL to students, professionals, or aspiring data analysts [7]. Each year, the NFL release a dataset collected from previous seasons as well as a specific topic or challenge [8]. The NFL invites participants to produce innovative solutions to a given problem with the widely available dataset. This dataset includes several tables containing game, player, play, and tracking data collected by the NFL. The tracking data is collected using RFID technology in player equipment and game balls [9].

2 RELATED WORK

2.1 Big Data Bowl

Previous submissions to the Big Data Bowl competitions have utilized a dataset with a similar structure to the one used in this project to create predictive or analytical models, additionally providing the required preprocessing steps.

One example of an analytical model is Kyle Burris' finalist submission in the 2019 Big Data Bowl competition [4]. This model uses a neural network to predict the arrival time of every player at a given play using a time-optimal trajectory. Space ownership is quantified by granting ownership of a space to the player with the fastest predicted time to it, which is demonstrated through a visual example of a play. In another finalist submission of the 2020 Big Data Bowl, Graham Pash and Walker Powell created a cumulative distribution model to predict yard gain in a given play [11]. This model uses spatial control fields to estimate the control a player has on any given point in the field. This was then used to construct a probability distribution model for yard gain predictions, used by a multilayer perceptron, a convolutional neural network (CNN) and a mixed-data model to create predictions and evaluate the performance of the three models, concluding that CNN performed the best. These projects highlight the steps taken to preprocess the player tracking data, as well as form well-defined trajectory and predictive problems by creating assumptions about the data. This serves as inspiration for the data manipulation steps performed in this project. The limitations of these

[1] 6 points

projects include convenience and ease of use, due to the complex models requiring particularly formatted data. This project goes beyond a stand-alone predictive or analytical model by adding a visual component to create a practical tool.

Due to the usage of RFID (Radio-Frequency Identification) chips by the NFL to collect the player tracking data [9], it is also important to mention the effectiveness of RFID technology in detecting and collecting spatiotemporal data of moving entities. RFID has been used in the past for the collection of spatiotemporal data such as in the tracking of autonomous entities [16]. This particular example evaluated a system using RFID technology to track moving autonomous robots, which proved to be more effective and accurate at capturing tracking data than other existing alternatives. This example supports the accuracy and integrity of a spatiotemporal dataset collected using RFID, which provides motivation for the usage of the Big Data Bowl dataset in our project.

2.2 LSTM Neural Networks

Previous LSTM-related works investigate the usage of LSTM for trajectory prediction in comparison with alternative models [18]. This particular example by Wang et al. compared machine learning frameworks using and not using LSTM. Through the evaluation of this report, the authors concluded that frameworks utilizing LSTM outperformed those that did not, which suggests the effectiveness of LSTM for trajectory prediction problems.

The report by Violos et al. highlights a project that utilized LSTM in a Deep Learning (DL) neural network for position prediction of vessels using trajectory data [17]. Upon evaluation of DL models with LSTM neurons in comparison with other state of the art solutions, the authors discovered that the DL model using LSTM outperformed its competitors. The authors also implemented the solution using Python and other supporting libraries such as TensorFlow, which are used for the implementation of this project. Other applications of LSTM include solutions to specific trajectory problems such as: vehicle highway trajectory prediction [2], human trajectory prediction in crowded spaces [1], and basketball trajectory prediction [19]. Each of these examples use LSTM for trajectory prediction problems with data from different sources ranging from sports data to highway vehicle data. Each of these projects explore trajectory prediction and provide motivation for the effectiveness of models using LSTM in comparison to other state of the art solutions.

Research conducted on LSTM models trained with trajectory data has provided supporting evidence for using LSTM in our player tracking prediction model. The contribution of this project in comparison with the aforementioned works is the application of LSTM in a model for a novel trajectory prediction tool that visualizes the predicted trajectories of a defensive player in response to the positioning and movement of an offensive player. Similarities between this project and the mentioned works lie in the usage of LSTM for trajectory prediction. However, this project is unique in its problem and visualization component that will use predictions created by the underlying LSTM-based model.

2.3 Sports Data Visualization and Analysis

Visualization and analysis provides context to sports data and derives insights and patterns in the data that teams use for strategy design. Topics of projects that use visualization and visual analysis for sports data include; play visualization of American football [15], visual analysis of effective set pieces in soccer [14], and feature driven and large scale visual analysis of player spatiotemporal data in soccer [3, 6]. The play visualization tool for American football [15] provides an effective method for representing player tracking data taken from several plays of a football game, with the ability to classify plays and output videos from the data. The visual analysis examples each provide techniques for visualizing data from different elements of a soccer game including; set pieces, single player, multi-player, formations, and more.

Each of these examples provide different methods of capturing components of a sport through the analysis of player tracking data. This project takes inspiration from these related works for providing sports analysts and coaches with tools or models that offer value through the visualization of sports data. The contribution of this project will differ from previous works through the use of an underlying LSTM based model trained on player tracking data, to generate predictions based on user input. This will offer coaches with a method to predict and visualize the movement of defensive players in response to an offensive formation and route.

3 PROGRESS TO DATE

We develop this project in a Jupyter notebook [5] using Python 3 and supporting libraries such as TensorFlow, tslearn, and others. A detailed overview of each table in the NFL Big Data Bowl 2023 dataset can be found on the official competition site [8]. This dataset is usable for non-commercial purposes including academic research. We use the tracking and player data tables from this dataset for this project. The tracking table contains player tracking data collected using RFID technology in NFL games, with spatiotemporal data for any player on any given play for every frame (tenth of a second). We use player position information in the player table for pair isolation in Section 3.2.

3.1 Data Preprocessing and Analysis

We begin by removing approximately 48 thousand empty rows and a number of unused columns (ie. birth rate of a player) from the dataframes to avoid corrupting the model and using unnecessary features when training it with the data. Next, we filter players in the player dataframe by position to select players with unique trajectories that are easily isolated with their respective defender. By restricting player positions, we can assess the model's ability to predict outcomes for individual player pairs, providing insight into the feasibility of predicting multiple players at once. In this iteration, the most popular expected offensive defensive positions we wish to use are wide receiver (WR) and defensive back (DB), respectively. Finally, we convert the categorical information in each table into numerical values using one-hot encoding, allowing the model to process categorical data in a numerical format. We merge the player and tracking table together to contain the necessary tracking data for the restricted positions.

Next, we consider the types of unique events that occur in the tracking data and more specifically, the initiating and terminating events of a sequence that we wish to capture. A distribution of the number of occurrences of each event is available in the notebook for a better understanding of the occurrences of events [5]. We identify ball snap to be the standard initiating event of each sequence we generate, meaning we generate each sequence created in Section 3.2 after a *ball_snap* event. Next, we identify terminating events of a sequence that we define as any event that causes a defender to switch from a planned defensive strategy, to an event based defensive strategy. For example, a *fumble* event would cause defensive players to attempt to recover the ball, affecting their original trajectory. We interpret terminal events through their descriptive names and list the selected events in the project repository [5]. Finally, we consider the length of unusually long plays which we choose to ignore to allow our model to capture the most frequently occurring sequences. We can capture the different lengths of plays in the dataframe through a distribution plot available in the project repository [5]. In the plot, we can see that most plays are between 20-80 frames, from which we determine a cutoff of less than or equal to 90 frames to include a play in the generated sequences.

3.2 Sequence Generation

Next, we generate sequential data from the tabular dataframe with the event and play duration assumptions described in Section 3.1. The required format for our LSTM-based model is an array where each entry of the array corresponds to one entry of the sequence. Therefore, we generate 2-dimensional arrays by taking every row (frame) from the dataframe for each player, play, and game that are unique by their combined identifiers.

After sequence generation, we then proceed to isolate offensive/defensive pairs to train a single defensive trajectory predictive model. We isolate pairs by selecting an offensive player in a given play and detecting the defensive player in the same play with the shortest Euclidean distance from it at different points in the sequence. We exclude pairs if the shortest distance is greater than 8 units to avoid model corruption. The most popular offensive/defensive pairings are for the wide receiver (WR) and defensive back (DB) positions. Note that the position of players in pairings is not unique by play, and there may be multiple players in a play that fill the same position. We demonstrate the isolation in an example through Fig 1, where the generated sequences for a play in a game of Dallas versus Tampa Bay is plotted on the left, and two isolated WR/DB player pairs are plotted on the middle and right.

After pair detection, we encode the position of the line of scrimmage and the starting positions of the defensive and offensive players into each sequence. We do this to use these features as additional predictors of defensive trajectory since starting positions on the field and distance from the endzone normally affect player trajectories. We also normalize each pair so that the offensive player is always moving to the right of the field. Additionally, we reflect pairs on the top half of the y-axis to the bottom half of the axis. We do this to normalize the routes ran by players on opposite starting positions in a formation, which is normally affected by their position relative to the middle of the field and the closest sideline. We also adjust the starting position of the pair of sequences so that the offensive sequence starts at the coordinate (0,0) and shift the defensive sequence accordingly. We show some of these steps in Fig 1.

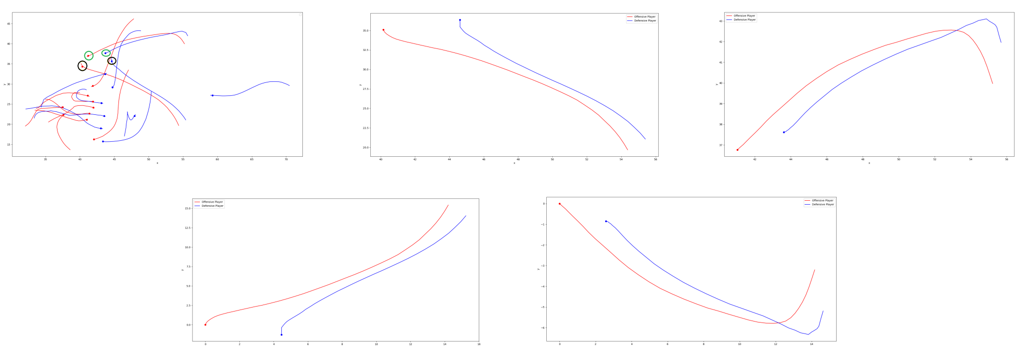

Fig. 1. Dallas (blue) defend an attempted Tampa Bay (red) Tom Brady pass. Here, Tampa Bay attempt to progress forward (left to right) to the opposition end zone and we can see two different starting positions (green and black circles) and trajectories of WR/DB pairings. The bottom two sequences demonstrate the normalization of the top middle and top right sequences, performed in Section 3.2. (GitHub)

3.3 Sequence Clustering

In the application developed in Section 3.5, our goal is to present the users with common trajectories of offensive players to improve the GUI design. To achieve this, we cluster the sequences of offensive players generated in Section 3.2 into groups with similar curve shapes, that we can then display in the application and allow a user to select from. We do this using a K-means clustering algorithm (TimeSeriesKMeans) for time-series data in tslearn. We use the x,y features of the sequences to ensure the clustering algorithm only considers the shape of the routes for the clustering algorithm.

We cluster the sequences with a K value of 8 and using the Dynamic Time Warping (DTW) metric in tslearn. This indicates that we compute 8 centers and that the model will use a distance metric that considers the different lengths of offensive sequences in our dataset (DTW). We retrieve the closest sequences to each cluster center, as shown in Fig 2, to measure the results of the clustering algorithm. We made other attempts to cluster the same sequences with different models and distance metrics using combinations of KMeans and KShape models with Euclidean distance and DTW as the distance metrics. These yielded inadequate results when plotting the sequences closest to cluster centers. When plotting sequences belonging to each cluster center produced by other models, we noted inconsistencies and sequences of significantly different shapes, which justified the decision to use TimeSeriesKMeans over the other options.

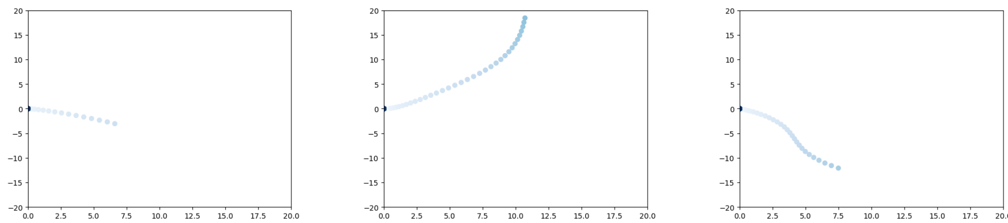

Fig. 2. Sequences closest to cluster center 1, 3 and 5, respectively (GitHub).

3.4 Long Short-Term Memory Model

Next, we use the sequences generated in Section 3.2 to train a Long Short-Term Memory (LSTM) model in TensorFlow 2 for Python. We select all features, highlighted in Section 3.2, including encoded features, through an iterative process of adding/removing features from the dataset, and seeing the effects on loss and plotted predictions. We use the **tf.data.Dataset** TensorFlow API to create a dataset from our sequences that we then batch and split into test and train datasets for our model to process. We create this dataset by specifying the list of offensive sequences and defensive sequences as an input and output pair, respectively. We split the dataset into training (70%) and testing (30%) datasets, and organize these into batches of size 64 for model processing and evaluation.

We build the model used in this project with two LSTM layers, a batch normalization layer to normalize each input feature for a given batch, and a dropout layer to improve generalization of the model on test data [5]. We use KerasTuner to optimize the hyperparameters of the chosen model architecture. We use the Hyperband algorithm to efficiently search for the best hyperparameters to use for this model. This model uses a loss function of Mean Absolute Error (MAE) to determine the cumulative sum of the distance of each point in a predicted sequence from the true point in the expected sequence. Changes to the model architecture did not improve the Mean Absolute Error (MAE) of predictions on the test dataset, therefore to our knowledge, the model yielding the lowest error of 0.302 with this data is currently used. We select other parameters such as number of epochs, optimizer, and batch size from manually training models and selecting the combination of values that reduced the error upon evaluation. We show sample predictions in Fig 3.

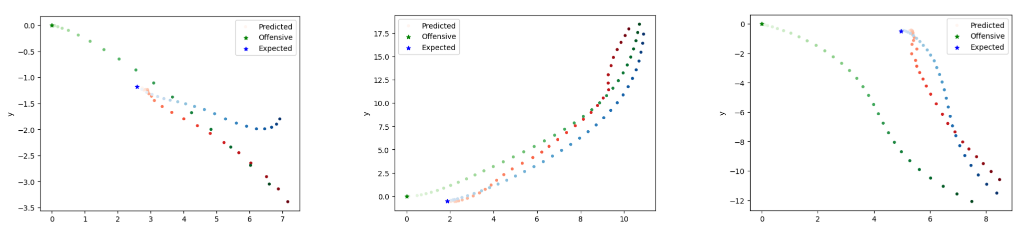

Fig. 3. Predicted sequences (red) for offensive sequences (green) in comparison with the actual sequences ran in the play (blue) in clusters 1, 3, and 5, respectively. Here, the starting positions for the offensive and actual sequences are marked with stars, and the colors of each sequence become darker to represent increasing time (GitHub).

3.5 Graphical User Interface

Next, we develop the Graphical User Interface (GUI) of the tool using Dash Open Source, as shown in Fig 4. The application displays a 120 x 53.3 unit plot adhering to the dimensions used in the Big Data Bowl dataset. The GUI displays an offensive player and predicted defensive player sequence pair at a time in this plot, where the defensive end zone is at the right side at all times. We plot the line of scrimmage (LOS) as a straight line between the starting position of both players to represent the natural separation in lineups of opposing players before ball snap in a real football play. We use the model in Section 3.4 to generate the defensive trajectory for selected offensive sequences. This model considers each of the adjustable inputs when producing a defensive trajectory, meaning user adjustments will produce a new defensive trajectory which is reflected on the plot. We highlight the adjustable inputs below.

1. **Offensive Sequence Selection:** A user can select the trajectories displayed by entering the index value for an offensive sequence generated in Section 3.2 that they wish to display on the plot. Changing the offensive sequence will update the plot with the chosen route and a new defensive trajectory produced by the model, which corresponds to the predicted trajectory of the defender attempting to defend the chosen offensive route.
2. **Player Pair Positioning:** A user can adjust the starting x positions of the sequence pair using the slider below the plot. This subsequently updates the position of the LOS, which is between the two players at all times. A user can also adjust the y position of the displayed sequence pair with the slider to the right of the plot.
3. **Defensive Player Offset:** A user can adjust the offset of the defensive player's starting position relative to the offensive player. By default, both players have the same y value, and the defensive player x value is opposite of the offensive player in the LOS. Adjusting the offset allows a user to dictate the starting position of both players.

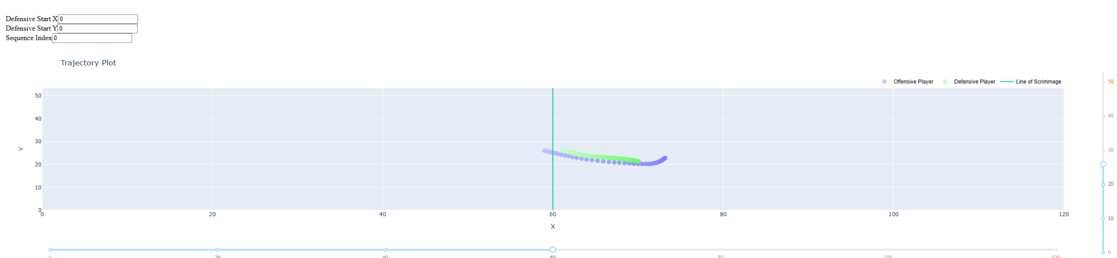

Fig. 4. The GUI of the application is available as a Dash web application. We show the offensive (purple) and predicted defensive (green) sequences, divided by the line of scrimmage (blue). Here, a user can change the sequence by entering an index value in the third text input (top left). A user may also change the positioning of each sequence by adjusting each slider and entering offset values of the defensive sequence in the other two text inputs. Each change will result in a new predicted defensive sequence. (GitHub)

4 DISCUSSION AND REVISED TIMELINE

The next steps in this project involve improving the application created in Section 3.5, since the limitations of this tool are evident with the basic GUI and limited single player prediction. Upon completion, we will evaluate the effectiveness of the current iteration as a coaching tool, and consider whether future possibilities such as multi player prediction are feasible. This revised timeline includes progress made, submission dates for CPSC 502.01A[2], and tentative milestones from the submission date to April 12^{th}, corresponding to the University of Calgary academic schedule[3].

[2]Retrieved from CPSC502.1A Fall 2022
[3]Retrieved from University of Calgary Academic Schedule 2022-2023

1. **CPSC 502.1A:** Project proposal completed on September 28th, literature review completed on October 21st, and interim report completed on December 7th. Completed data preprocessing, EDA (Section 3.1) and sequence generation (Section 3.2) of the Big Data Bowl competition dataset [8], with version control through GitHub [5].
2. **Clustering, LSTM Model and GUI:** Clustering sequences and creating LSTM model (Section 3.3 and Section 3.4). Created a visualization tool for displaying team trajectories which we use with the single-player trajectory model, and will allow future models to use an adjusted visualization component (Section 3.5). Reassessing project scope with the evaluated accuracy of the single-player trajectory model. Completed Midterm Report on Feb. 10th.
3. **Project Scaling and Final Presentation/Report:** Finalize the current GUI and scale the tool to the scope of the redefined problem after assessment. This could be for single or multiple players and depends on the evaluation of the previous tool's limitations. Finalize project and final report/presentation with project supervisor feedback.

REFERENCES

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 961–971.

[2] Florent Altché and Arnaud de La Fortelle. 2017. An LSTM Network for Highway Trajectory Prediction. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC).* 353–359. https://doi.org/10.1109/ITSC.2017.8317913

[3] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, Sridha Sridharan, and Iain Matthews. 2014. Large-Scale Analysis of Soccer Matches Using Spatiotemporal Tracking Data. In *2014 IEEE International Conference on Data Mining.* 725–730. https://doi.org/10.1109/ICDM.2014.133

[4] Kyle Burris. 2019. *A Trajectory Planning Algorithm for Quantifying Space Ownership in Professional Football.* Technical Report. NFL. https://operations.nfl.com/media/3665/big-data-bowl-burris.pdf

[5] Alejandro Escobar. 2022. CPSC 502. https://github.com/alejooescobar/CPSC502

[6] Halld'or Janetzko, Dominik Sacha, Manuel Stein, Tobias Schreck, Daniel A. Keim, and Oliver Deussen. 2014. Feature-driven Visual Analytics of Soccer Data. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST).* 13–22. https://doi.org/10.1109/VAST.2014.7042477

[7] The National Football League. 2022. NFL Big Data Bowl. Retrieved November 13th, 2022 from https://operations.nfl.com/gameday/analytics/big-data-bowl/

[8] The National Football League. 2022. NFL Big Data Bowl 2023. Retrieved November 13th, 2022 from https://www.kaggle.com/competitions/nfl-big-data-bowl-2023/data

[9] The National Football League. 2022. NFL Next Gen Stats. Retrieved November 13th, 2022 from https://operations.nfl.com/gameday/technology/nfl-next-gen-stats/

[10] The National Football League. 2022. Official Playing Rules of the National Football League. Retrieved September 17, 2022 from https://operations.nfl.com/media/5kvgzyss/2022-nfl-rulebook-final.pdf

[11] Graham Pash and Walker Powell. 2019. *A Mixed-Data Predictive Model for the Success of Rush Attempts in the National Football League.* Technical Report. NFL. https://operations.nfl.com/media/4207/bdb_pash_powell.pdf

[12] Ralf C. Staudemeyer and Eric Rothstein Morris. 2019. Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks. (2019). https://doi.org/10.48550/ARXIV.1909.09586

[13] Raymond T. Stefani. 1987. Applications of Statistical Methods to American Football. *Journal of Applied Statistics* 14, 1 (1987), 61–73. https://doi.org/10.1080/02664768700000006

[14] Manuel Stein, Halldór Janetzko, Andreas Lamprecht, Daniel Seebacher, Tobias Schreck, Daniel Keim, and Michael Grossniklaus. 2016. From Game Events to Team Tactics: Visual Analysis of Dangerous Situations in Multi-match Data. In *2016 1st International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW).* 1–9. https://doi.org/10.1109/TISHW.2016.7847777

[15] Toshihiro Tani, H Huang, and Kyoji Kawagoe. 2015. Sports Play Visualization System for American Football. In *Proceedings of the international multiconference of engineers and computer scientists*, Vol. 1.

[16] Ricardo Tesoriero, Jose A. Gallud, Maria D. Lozano, and Victor M. R. Penichet. 2009. Tracking Autonomous Entities Using RFID Technology. *IEEE Transactions on Consumer Electronics* 55, 2 (2009), 650–655. https://doi.org/10.1109/TCE.2009.5174435

[17] John Violos, Stylianos Tsanakas, Maro Androutsopoulou, Georgios Palaiokrassas, and Theodora Varvarigou. 2020. Next Position Prediction Using LSTM Neural Networks. In *11th Hellenic Conference on Artificial Intelligence* (Athens, Greece) *(SETN 2020).* Association for Computing Machinery, New York, NY, USA, 232–240. https://doi.org/10.1145/3411408.3411426

[18] Chujie Wang, Lin Ma, Rongpeng Li, Tariq S. Durrani, and Honggang Zhang. 2019. Exploring Trajectory Prediction Through Machine Learning Methods. *IEEE Access* 7 (2019), 101441–101452. https://doi.org/10.1109/ACCESS.2019.2929430

[19] Yu Zhao, Rennong Yang, Guillaume Chevalier, Rajiv C. Shah, and Rob Romijnders. 2018. Applying Deep Bidirectional LSTM and Mixture Density Network for Basketball Trajectory Prediction. *Optik* 158 (2018), 266–272. https://doi.org/10.1016/j.ijleo.2017.12.038