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 use of sports analytics has become a large contributor to the success of professional sports teams. The analysis and visualization of game data has allowed teams 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 visualize the predicted trajectory of defensive players in response to offensive players. The tool will take as input both the starting position of each player and the offensive route; then visualize a predicted defensive trajectory in response. We use NFL player tracking data to train an 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 by coaches and managers of teams to gain an advantage against their opponents. With modern technology, player tracking data has become more available in recent years, introducing more opportunities for analytical projects. The National Football League (NFL) and NFL teams are increasingly using sports data analysis for offensive, defensive, and special team play design [14]. The NFL has even hosted annual sports analytics competitions known as the Big Data Bowl, allowing participants to submit projects using a provided NFL dataset that includes player tracking data [8].

Many of the previous Big Data Bowl Competition submissions have produced predictive or analytical models provided the data, which a non-technical individual, such as a coach, cannot conveniently use by themselves. The average NFL coach would benefit from an accessible coaching tool which integrates these predictive/analytical models. This project attempts to solve that problem by coupling a graphical user interface (GUI) with a predictive neural network model.

The developed 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 (Wide Receiver) 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 the 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 [13]. LSTM models are able to accurately predict trajectories by using sequential spatiotemporal data [18]. Related applications of LSTM for trajectory prediction problems include: vehicle trajectory prediction [2], human trajectory prediction [1], basketball shooting trajectory prediction [20], and more. This model uses NFL player tracking data provided from the 2023 and 2021 Big Data Bowl competitions 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 the trajectory of an offensive player, and adjust the starting positions of both offensive and defensive players. 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 Wide Receivers. We display the different trajectories to a user which allows them to select frequently occurring offensive player routes in real NFL games. Additionally, the GUI supports visualization and input features that are provided to aid users in interacting with the model as intended. The combined result is a fully interactive and accessible visualization tool, instead of a stand-alone predictive model.

Evaluation of this tool involves analyzing the accuracy of the model's predictions and exploring the added value of the visualization produced by the GUI to the targeted user (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 datasets, clustering of sequences to identify common offensive player routes, and the attempt to use this data for a trajectory prediction LSTM model. This report also documents the assumptions made during development of this model, the limitations of the data, and highlights 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 [11]. 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 this project.

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 [8]. Each year, the NFL release a dataset collected from previous seasons as well as a specific topic or challenge [9]. 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 [10].

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, specifically for research problems related to American Football.

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 [12]. 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 [10], it is also important to acknowledge 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 [17]. 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 datasets 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 [19]. This particular example by Wang et al. compared LTSM frameworks to alternative machine learning frameworks. In the evaluation, the authors concluded that frameworks utilizing LSTM outperformed those that did not, which supports the effectiveness of LSTM for trajectory prediction problems.

Violos et al. presented a project that utilized LSTM in a Deep Learning (DL) neural network for position prediction of vessels using trajectory data [18]. 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 [20]. 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 trajectory 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 application and the GUI that will visualize predictions created by the underlying LSTM 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 research that use visualization and visual analysis for sports data include; play visualization of American football [16], visual analysis of effective set pieces in soccer [15], and feature driven and large scale visual analysis of player spatiotemporal data in soccer [3, 6]. The play visualization tool for American football [16] 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 model trained on player tracking data, that generates predictions based on user input, providing predictive capabilities in the graphical user interface.

3 METHODS

We develop this project using Jupyter notebooks and Python scripts [5] using Python 3 and supporting libraries such as TensorFlow, tslearn, and others. We use the NFL Big Data Bowl 2023 and 2021 datasets, which are usable for non-commercial purposes including academic research. A detailed overview of each of the used tables in both datasets can be found on the official competition sites [7, 9]. We use the game, play, player, and tracking tables from these datasets for sequence generation, player pair isolation, and GUI components of this project.

3.1 Data Preprocessing and Analysis

We begin by removing empty rows and unused columns (ie. birth rate of a player) from the play and player dataframes to avoid corrupting the model and using unnecessary features in training. 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 are able to assess the model's ability to predict outcomes for individual player pairs, and evaluate the feasibility of predicting multiple players at once. The offensive position selected is wide receiver (WR) with defensive positions strong safety (SS), free safety (FS) and cornerback (CB). The most prominent offensive and defensive player pairs are for the wide receiver (WR) and defensive back (DB) positions, respectively. Finally, we convert the categorical data in each table into numerical values using one-hot encoding, allowing the model to process the categorical data. We merge the player and tracking table together to contain the necessary tracking data for the restricted positions.

Next, we consider the types of events that occur in the tracking data and more specifically, the initiating and terminating events of a sequence that we wish to capture. The decision of which initiating and terminating events to select was made after considering the number of occurrences of each event, which we captured in a distribution plot available in the project repository [5]. We identify ball snap to be the standard initiating event of each sequence we generate, meaning we start collecting data for each frame in a trajectory 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]. Note that neutral events, such as $pass\_forward$, are not considered as initiating or terminating events. Finally, we consider a sequence length cutoff to allow our model to capture average length sequences. Through another distribution plot, we see that most plays land within a 20-90 frame range, and determine that a sequence must be between 10 to 90 frames to include it in the total list of generated sequences[5].

3.2 Sequence Generation

Next, we generate sequential data from the tracking table with the event and play duration assumptions described in Section 3.1. The required format for our LSTM-based model is a tensor where each entry of the tensor corresponds to one entry of the sequence. Therefore, we generate 2-dimensional tensors by taking every row (frame) from the dataframe for each player, play, and game that are unique by their combined identifiers. Each row of the 2 dimensional tensors will contain the x, y, and o (orientation) features of the dataset, which are the primary features used for model training.

We proceed to isolate offensive/defensive sequence pairs by selecting an offensive player in a given play, selecting a defender in the same play, and keeping track of the largest difference in Euclidean distance between both players at

different frames in the sequence. The defender with the smallest "largest difference" in Euclidean distance is assigned to the offensive player. We use this method in lieu of smallest Euclidean distance to ensure that the defensive player remained near the offensive player throughout the play, rather than for a portion of the play. Additionally, we exclude pairs if the achieved distance is greater than 8 units. 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. After pair detection, we encode the position of the line of scrimmage, the original starting position (x and y) of the defensive player, and the distance from the offensive player to both sidelines and the targeted endzone, into each of the offensive sequences. We do this to use positional features as additional predictors of defensive trajectory, as these normally affect player routes in real games.

We normalize each sequence pair so that sequences are always moving from left to right on the field. Additionally, we reflect pairs from the top to the bottom half of the field. 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 (0,0) coordinate, and shift the defensive sequence accordingly. Note that we only perform normalization on the x and y features of the sequence pairs, to make use of the original coordinate system of the dataset for other features. This is done to ensure that original positioning of the players will impact predicted results on the normalized coordinate system. In summary, the offensive sequences hold 9 features per sequence frame, while the defensive sequences only hold x and y features to simplify the model's prediction. We demonstrate player pair isolation and some normalization steps 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.

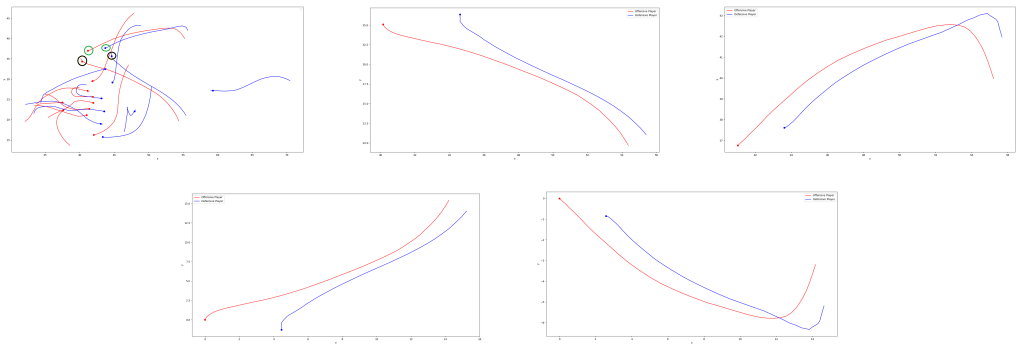


Fig. 1. Dallas (blue) defend an attempted Tampa Bay (red) Tom Brady pass. Here, Tampa Bay attempt to progress forward (left to right) to the opposition end zone and we can see two different starting positions (green and black circles) and trajectories of WR/DB pairings. The bottom two sequences demonstrate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

3.3 Sequence Clustering

To provide users of the coaching tool GUI in Section 3.5 with selectable, pre-recorded trajectories of offensive players, we cluster the offensive sequences generated in Section 3.2 into groups with similar curve shapes. 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. We cluster ~13,000 of all sequences generated in Section 3.2 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 explore the results and evaluation of clustering in Section 4.1.

3.4 Defensive Coverage Path Prediction

We proceed to train a defensive trajectory predictive model for use in the coaching tool GUI in Section 3.5. The goal of this model is to receive offensive sequences with features described in 3.2, to predict a defensive trajectory as output. We use all ~43,000 sequence pairs 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 and Section 4.2, through an iterative process of adding/removing features, and seeing the effects on loss and plotted predictions. We split the generated sequences into training (70%) and testing (30%) datasets, and organize these into batches of size 64 for training and evaluation.

We build the model 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 the KerasTuner Hyperband algorithm to efficiently search for the optimal 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 points in the expected sequence. 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. Exploring changes in model attributes and the model's predictive results are discussed in Section 4.2.

3.5 Graphical User Interface

Next, we develop the Graphical User Interface (GUI) of the coaching tool using Dash Open Source, as shown in Section 4.3, Fig 4. The goal of the Dash application is to provide a user-friendly interface for users to interact with the model and produce different game scenarios, using real NFL data from the regular season. The application displays a 120 x 53.3 unit plot adhering to the dimensions used in the Big Data Bowl dataset. The application plot displays a selected offensive player and predicted defensive player sequence pair at a time, where the sequences are oriented to match the original coordinate system of the Big Data Bowl dataset [7, 9]. 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. In addition to the trajectory plot, the application also displays play information pertaining to the original play and input information to assist the user. The model considers each user adjusted input to produce a new defensive trajectory, which is reflected on the plot.

We provide a user with three different modes to select an offensive trajectory and generate a new predicted defensive trajectory. First, a user can select a sequence by index, from all available sequences generated in Section 3.2, to display the sequence on the plot. Alternatively, a user may tab to "Cluster Selection" mode, where they are displayed with the 8 cluster centers computed in Section 3.3. From here, a user can select a cluster center and select one of eight closest sequences to that cluster center to be displayed on the plot. Finally, a user can tab to "Play Selection" mode, where they are able to choose valid plays from the 2023 Big Data Bowl dataset to display on the plot. These plays are filtered to plays that have been identified to have a valid offensive/defensive sequence pair that can be used by the model. Each of these three changes to the offensive trajectory will visually plot the predicted trajectory of the defender attempting to defend the chosen offensive route.

To produce new scenarios and predicted defensive trajectories, users can adjust the positions of both players. Adjusting the x and y of the offensive player will subsequently shift the defensive player. From here, a user can adjust

the starting position of the defensive player to view the new defensive trajectory in response to starting position. The model considers these changes to generate a new defensive player prediction in response to the new positioning.

Finally, we provide a user with several visual adjustments and features. A user can toggle to display the trajectories of all other players of the original play or toggle to display the original offensive/defensive player trajectories to compare predictions with the actual defensive trajectory, and to view the original positioning of the offensive player. Furthermore, a user may choose to adjust the line of scrimmage which will shift the entire play with it. Finally, a user may animate all of the sequences displayed on the plot, to see the real time movements of actual and predicted sequences. The animation will update the plot using sequence frames, allowing users to view play progression for every 500 ms.

4 RESULTS

This section highlights the findings and results from the methods used in Section 3. Here, we acknowledge the outcomes of different components of the project, limitations, and the assessments used to determine a successful outcome.

4.1 Sequence Clustering

We first evaluate the results of clustering highlighted in Section 3.3. 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, as each sequence closest to a cluster center was differently shaped and inconsistent.

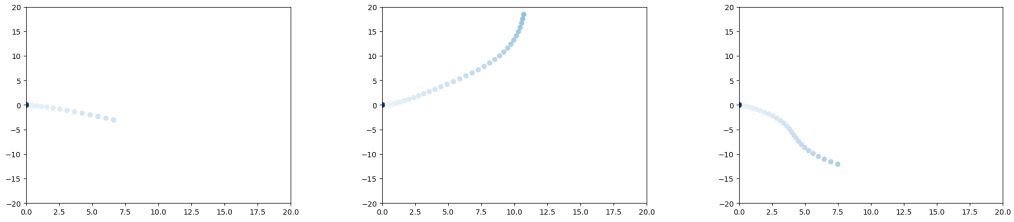


Fig. 2. Sequences closest to cluster center 1, 3 and 5, respectively. This represents cluster centers computed from the 2023 Big Data Bowl dataset (GitHub).

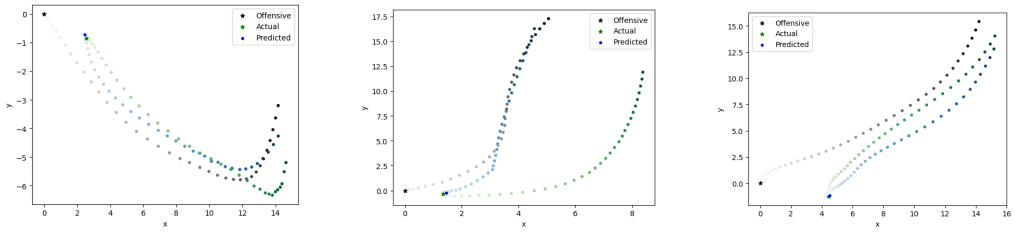
We evaluate the quality of the clustering by graphing the closest sequences to each cluster center and noting similarities or differences between sequences belonging to a cluster. Using Euclidean distance as the distance metric or a KShape model typically yielded inconsistent results, as the closest sequences were seen to be significantly different from one another. In comparison, DTW with a KMeans model produced cluster centers that had similarly shaped sequences closest to each center, which was sufficient to accept the cluster centers for use in the application. This was expected as DTW is able to compare different length sequences which is necessary for this dataset, as plays of different lengths can contain similarly shaped player trajectories. Additionally, it is important to note that the clustering is only performed on trajectories ran by Wide Receivers (WR), as this is the only offensive position we consider in this project. This is an obvious limitation to the clustering component of this project, which can be expanded on by performing the sequence generation in Section 3.2 with a different positional filter, and clustering the position(s) of interest. This is a possible area of expansion, however, for the purposes of the GUI in this iteration of the project, the current results are acceptable.

4.2 Defensive Coverage Path Prediction

Next, we evaluate the accuracy of a model with the architecture described in Section 3.4. We show sample predictions in Fig 3. The features selected in Section 3.2 yielded the lowest error and improved the plotted predicted defensive trajectories produced by the model. The primary predictor features of the offensive sequences are x and y coordinates, which allow the model to roughly determine what the trajectory of the defensive player will be relative to the offensive player. These features are minimal for training a model to produce an expected defensive trajectory in response to an offensive trajectory. The offensive x and y coordinates were not sufficient for our tool to provide completely accurate defensive predictions. The defensive starting position relative to the offensive player is important in determining a defender's expected trajectory. Additionally, features such as the offensive player's distance to either sideline and endzone they target, will affect the predicted defensive trajectory, as these features limit the range of which a player is moving in a real game of football. For the purposes of this project, we determine this more full set of selected features to be the minimal set that allows the model to be responsive to user adjustments in input.

We assess the success of the model by the achieved error (MAE) in comparison with other iterations, and through plotting the predicted trajectories of the model in comparison with the expected/actual trajectories that occurred during the game. Minor changes in model architecture did not significantly impact the error or predicted trajectories of the model. The achieved error slightly increased when training the model with additional sequences from the 2021 Big Data Bowl dataset, but this is acceptable to improve generalization of the model. Therefore, to our knowledge, the model yielding the lowest error of 0.36845 that used all ~43 thousand generated sequence pairs for training is currently used. A brief visual comparison of the predicted and actual defensive trajectories implies that the model produces trajectories that would be typical of a defender placed in the real game situation. It is important to note that defenders on a play do not always run the "most optimal" trajectory to cut off an offensive player or stick close to them. Therefore, we attempt to achieve predicted routes that are roughly what a defender in that position would take.

Fig. 3. Predicted sequences (blue) for offensive sequences (grey) in comparison with the actual sequences ran in the play (green) for real NFL plays. 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).



In summary, the model is able to predict the rough trajectory of a defensive player in a real game scenario, however, this model is still limited in that it is incapable of handling multiple offensive sequences at once. The current capability of the model provides an accurate one to one prediction from offensive player (Wide Receiver) to a defensive player, with no other player data considered as a predictor of defensive trajectory. Ideally, this model would be able to predict the defensive sequences of all players on the defensive team (11 players), considering the data of all players as input of prediction. However, for the sake of this project, the current model fulfills the requirements of responsiveness to changes in positioning by a user, to accurately predict defensive trajectory.

4.3 Graphical User Interface

The Graphical User Interface (GUI) of the project provides an interactive platform that plots the predictions of the model described in Section 3.4 and Section 4.2. All features described in Section 3.5 are available in this application, which allows a user to visualize the predictive capabilities of the model, and provide their own input to see the effects of player positioning on the model's predictions. Through the application, a user is able to modify input features of the original sequence, to produce a new defensive prediction that represents a simulation of the same play with different positioning of players on the field. The application is live[2] and is subject to updates.

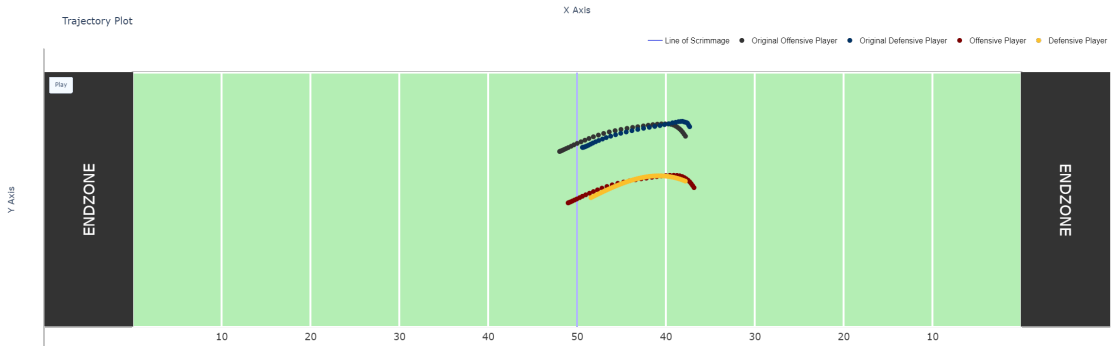


Fig. 4. The GUI of the application is available as a Dash web application. We show the offensive (maroon) and predicted defensive (yellow) sequences, divided by the line of scrimmage (blue). Additionally, we show the original offensive player sequence (black) and original defensive player sequence (dark blue), allowing a user to compare with the original sequences. The play in this visual moves from left to right. From here, a user may make adjustments through input features, as described in Section 3.5 (Application).

We measure the success of this component of the project by considering the effectiveness of the GUI in abstracting the usage of the underlying LSTM model, to allow users to interact with the interface and simulate real game scenarios. Visualization features help the user understand how to use the tool, and compare predicted results with the actual play. In summary, the tool succeeds in allowing users to interact with the predictive model in an intuitive manner. Additionally, the limitations of this tool stem from limitations of the model, highlighted in 4.2. The current model and GUI are incapable of multiple player prediction, and a user is limited to viewing a single predicted defensive sequence at a time. While this is useful for the isolated defensive positions, there is a lot to gain from allowing multiple player/position prediction. This type of feature would provide a user with a method of viewing player predictions and simulating play trajectories in isolation, or as a whole. We explore expansions of the model and GUI in Section 5.

5 DISCUSSION

In this project, we use real player tracking data from the NFL regular season to produce an interactive coaching and predictive tool, that is responsive and allows users to simulate real game scenarios, visualizing expected behaviour of defenders on a play. The results of this project have implications in several fields, specifically sports analysis, machine learning, and sports data visualization. It is important to note that changes in the technology used for clustering, the predictive model, and GUI may affect the results of researchers replicating this work.

Given the methods and analysis in Section 3.2, we provide an outline of required data processing steps and assumptions for researchers looking to use the tracking data provided in Big Data Bowl competitions for similar projects. Additionally,

[2]Available at: https://cpsc502application.onrender.com/

we provide the required sequence normalization steps for training our model, allowing readers to replicate this work, expand on it, and form different assumptions. In particular, data processing is key for future iterations of this project, as similar steps will have to be performed to expand the model for multi-player sequence prediction.

With the clustering described in Section 3.3 and 4.1, we provide an outline of the model and distance metric used to cluster offensive player sequences into groups, to acquire the most popular offensive (Wide Receiver) trajectories in real games. Additionally, the steps taken to obtain these results can be replicated with sequences of different positions. Doing so results in cluster centers representing the most common trajectories of players in that position.

Given the model described in Section 3.4, a researcher can replicate the architecture of this model, and train it with the same data as this project, or their own data. Similarly, the model architecture and parameters can be changed to view the affects on model predictions. The results of this model described in Section 4.2 can be used as a benchmark for future improvements of this project. Additionally, the results provide evidence that LSTM models produce adequate results in sequence to sequence prediction, particularly for the problem statement of this project.

The most prominent limitation of this model is its inability to predict more than one defensive player sequence at a time. In future iterations of this project, the improvement of the model would be the first priority. This would involve modifying the sequence pairing process in Section 3.2, to create multiple unique offensive/defensive player sequence pairs, for each play. This is a complex task due to the requirement of identifying different defensive sequences per offensive sequence. Additionally, the model would have to consider all positional features of the selected subset of interest from the defensive team. The iterative process of training the model with several sequences and combinations of input features is lengthy, and a researcher is required to ensure the data is being processed by the model as expected.

Finally, the GUI of this project described in Section 3.5 and Section 4.3 can be replicated by a reader with the required input features to produce an interactive interface that uses an underlying predictive model. The model used in this GUI can be replaced, meaning that expansions to the model could be configured in the GUI in future iterations of the project. Additional requirements or features would follow after improvements to the model are made. In practice, the GUI can be used to interact and visualize the predictive capabilities of the current model. The produced GUI is a proof of concept demonstrating that a more complete coaching tool including all players during a play is feasible.

6 CONCLUSION

In this project, we use NFL player tracking data from the NFL Big Data Bowl competition datasets to produce an interactive coaching tool that uses an underlying LSTM model for defensive player trajectory prediction in response to a single offensive player. This report highlights the data processing steps and assumptions made to obtain sequential data in a format for training our LSTM model, with a specified architecture. Additionally, we cluster offensive sequences and view the most popularly ran routes of wide receivers from plays in the dataset. The clustering results and LSTM model are then used in the Graphical User Interface of the tool, to allow coaches/users to select offensive sequences and produce expected defensive sequences on the plot of the tool. In addition to displaying the predictive capabilities of the tool, the user is provided with additional input and visualization features, to alter the predicted defensive trajectory and resulting plot. The underlying model of this tool produces new predicted defensive trajectories in response to user input, allowing users to simulate real game scenarios from real plays that occurred in a regular NFL season. Overall, this project demonstrates the capabilities of an interactive tool with an underlying LSTM model to allow users to display real game scenarios and witness the effects of player positioning on predicted trajectories of defenders. In conclusion, this work makes a valuable contribution to the non-private fields of sports analytics, machine learning, and sports data visualization, and provides a prototype coaching tool that is a solid foundation for future work in this project.

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. NFL Trajectory Prediction. https://github.com/alejooescobar/NFL-Trajectory-Prediction

[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. 2021. NFL Big Data Bowl 2021. Retrieved November 13th, 2022 from https://www.kaggle.com/competitions/nfl-big-data-bowl-2021/data

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

[9] 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

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

[11] 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

[12] 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

[13] 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 arXiv:1909.09586

[14] 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

[15] 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

[16] 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.

[17] 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

[18] 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

[19] 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

[20] 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

A APPENDIX: CLUSTERING

In this section, we explore further results of the clustering component of this report. Originally, the cluster centers computed in Section 3.3 and Section 4.1 only used the sequences generated from the 2023 Big Data Bowl dataset. This was done to keep consistent with the data available in the GUI, as users are able to select the sequences from only the 2023 dataset, due to a slight formatting difference between both datasets.

In addition to the 2023 Big Data Bowl sequences, we decided to cluster the sequences generated from both 2021 and 2023 Big Data Bowl datasets to explore differences in clustering. The new cluster centers did not significantly differ from the cluster centers achieved previously with only 2023 data. As seen in Fig 5, the 3 achieved cluster centers (3, 4 and 5) are similar to the previous cluster centers seen in Fig 2. This implies that similar trajectories are seen from different sets of play data for the selected offensive position (Wide Receiver). These results are secondary to the goal of this research project, but interesting nonetheless.

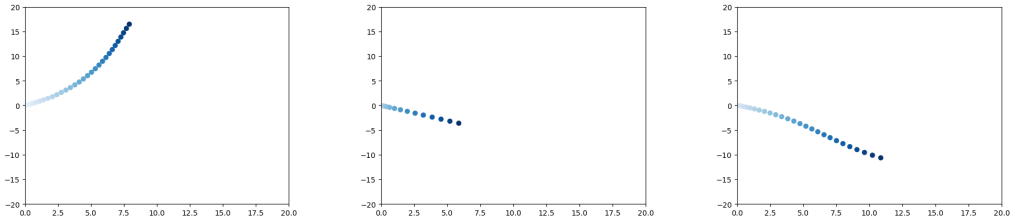


Fig. 5. Sequences closest to cluster center 3, 4 and 5, respectively. This represents cluster centers computed from the 2023 and 2021 Big Data Bowl datasets (GitHub).

B APPENDIX: GUI

In this section we explore the capabilities of the application further, providing an example of the possible adjustments a user can make, and the affect on model predictions. The full capabilities and input features of the application are displayed below in Fig 6 and Fig 7. These figures display the same selected offensive sequence from Fig 4, with user input adjustments through the GUI.

In Fig 6, the full play that the selected offensive sequence belongs to is also displayed. Additionally, the original offensive/defensive sequence pair that is isolated in this play remains shown on the plot, allowing a user to compare with the original positions of players, and the original trajectory ran by a defender on the play.

When a user makes adjustments using the input features described in Section 3.5, the plot updates with a new plotted predicted sequence. In Fig 7, we can see that a user has adjusted the line of scrimmage (los), the starting position of both players, and has chosen to hide the original offensive/defensive sequence pair. Through a brief comparison, it is clear that the predicted trajectory of the player in Fig 7 is different from the trajectory plotted in Fig 6, suggesting that the model considered the user inputs in its prediction, as expected.

These are examples of changes that a user can make to the original plot, to aid them in understanding the predictive capabilities of the underlying model. Coaches can use the tool to analyze the new predicted sequences from updates to position and offensive trajectory, allowing for improved insight in offensive play design. These figures highlight a subset of visualization and input features existing in the application.

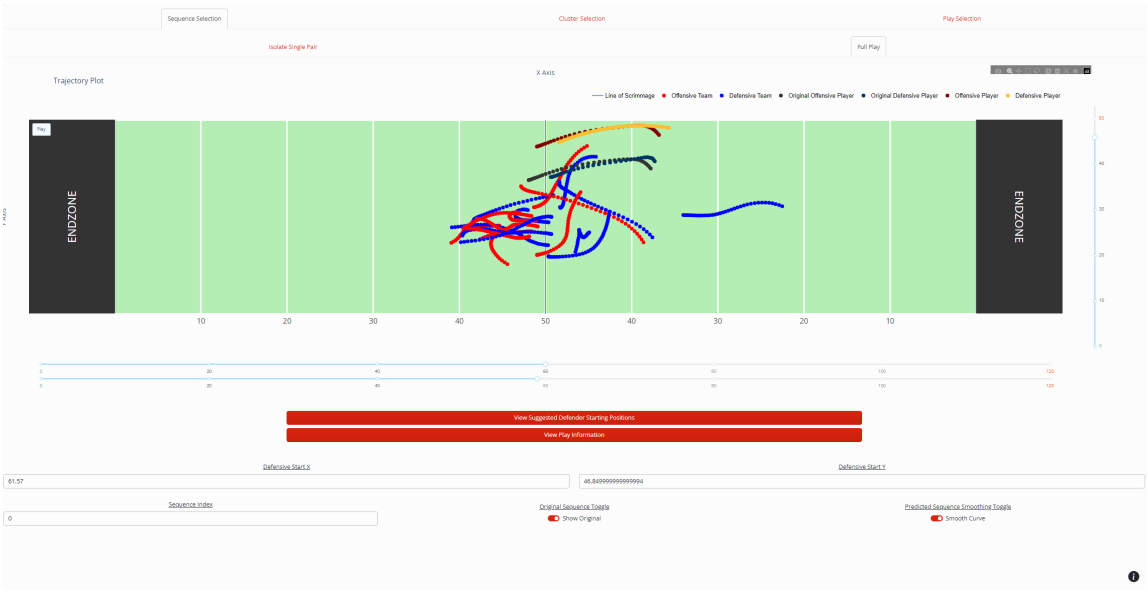


Fig. 6. We show the offensive (maroon) and predicted defensive (yellow) sequences, divided by the line of scrimmage (blue). Additionally, the original offensive team (red), offensive player (black), defensive team (blue), and defensive player (dark blue) sequences are shown in the plot (Application). Here, a user has toggled an input in the application to show the full play on the plot.

Fig. 7. We show the offensive (maroon) and predicted defensive (yellow) sequences, divided by the line of scrimmage (blue). Additionally, the original offensive team (red) and defensive team (blue) sequences are shown in the plot (Application). Here, a user has adjusted the positioning of players on the field to simulate a new scenario, producing a predicted defensive trajectory.