Football Defensive Coverage Classification with Graph Neural Networks

Matt Manner, Josh Gen, Connor Nickol
School of Data Science, University of Virginia
Email: [xkv3na, jdg9vr, can2hr]@virginia.edu

***Abstract*—Graph neural networks (GNNs) are a relatively new technique which often use location data for regression, classification, or generation. Their use in analyzing so-called "invasion games" like soccer or basketball has been previously studied. Our goal was to use these GNNs to classify National Football League (NFL) defensive coverage with high accuracy. We use the publicly available 2023 NFL Big Data Bowl data, utilizing both player locational data as well as play-specific and player-specific data as inputs to our GNN model. Our GNN had 63% accuracy, which is a decent accuracy given that many of these coverages are meant to be disguised as other coverages.**

I. Motivation

The NFL is the world's most profitable sports league, with estimated 2022 revenues of $18 billion. Currently, hundreds or thousands of man-hours are needed for team staffs to annotate game film with gameplay information, such as defensive coverage on a given play. However, the NFL also collects vast quantities of data during games; through the league's partnership with Amazon Web Services (AWS), all players and the football itself are fitted with tracking devices that show real-time location data, speed, acceleration, and more (the parameters that we used are discussed at further detail in the Datasets section of this paper.

We believe this data presents a strong opportunity to automate parts of the film study process. If we can build an accurate classifier, NFL staff resources can be freed up for other, less mundane purposes. In addition, trends in defensive playcalling can be faster analyzed for a competitive advantage. In addition, we hope to demonstrate the utility of GNNs with generalized location data, to show the potential to extend our framework to different domains. GNNs carry enormous potential both in the sports world and even in other fields such as military operations.

II. Datasets

Our data came from the NFL's Big Data Bowl, in which the league releases a portion of its proprietary AWS tracking data for an analytics competition. For the latest Big Data Bowl, the data was collected during the first 8 weeks of the 2021 season. It came in the form of the following CSV files:

- Players: name, unique player ID, position. $n = 1679$ players.
- Plays: unique play ID, game situation (offensive/defensive team, score differential, etc.), pass coverage. $n = 8557$ plays.
- PFF: player, play, position. $n = 188254$ player-plays.
- Weeks: Player and football tracking location data, every 0.1 seconds, for each play.

We used each play's pass coverage as the response variable. Our data was trimmed so each play would start at the snap of the ball. We ended the tracking of our play at either a sack, a thrown pass, or if 3 seconds had elapsed. We felt that 3 seconds was a good choice because any longer than that and it is unlikely the defense retains it shape as the play devolves into more of a scramble drill than a preplanned play. Frames from this point in time in the play would not be as beneficial in classifying what the defense had initially run. We also considered 4 seconds as the cutoff here and that could be a potential improvement point in our model accuracy. We would not expect this gain to be more than a percentage or two, looking at our results.

After these modifications, we were left with around 188,000 frames making up about 7,200 pass plays. These frames were turned into graph objects using Python's Spektral package, as described below in the Technical Approach section.

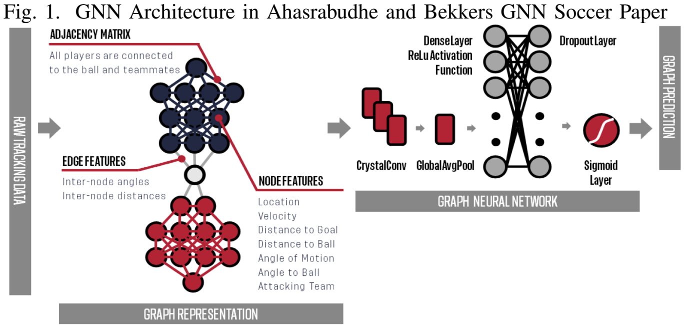
III. Related Work

This paper was motivated by a paper presented at MIT Sloan Sports Analytics Conferece this year, Explainable Defense Coverage Classification in NFL Games using Deep Neural Networks. This paper uses CNN's to predict defensive coverages using the same NFL Big Data Bowl player tracking dataset. This paper combines multiple deep CNN's in a final ensemble model with 88.9% accuracy on the test dataset. It notes in the conclusion and future work section that the model could likely be improved if implemented with a GNN instead of a CNN, which is our proposed method for analyzing defensive coverages.

The paper is based on the winning NFL Big Data Bowl paper in 2020, posted by The Zoo. This was a revolutionary application of CNN in the space of sports analytics. The input for this model was a tensor where one dimension was the offensive player, the second dimension was the defensive player, and the final dimension was the actual feature from the player tracking data (speed, x location, y location, acceleration, etc.). They performed a number of convolution layers and pooling layers, with a final softmax layer to classify the defensive coverage to one of the coverages in the dataset. There have been a number of other papers expanding this original CNN applied to player tracking data, including the MIT Sloan paper that sparked our interest.

There are some examples of using GNN's in other sports, which we relied on to validate the work we were doing of applying GNN's to NFL tracking data. One specific paper, presented in the 2023 MIT Sloan Sports Analytics Conference, applied GNN's on tracking data in professional soccer to classify the live success rate of counterattacks. They included specific node and edge features in their paper, which we used in addition to football context to select our features for our GNN. Their architecture can be seen in Figure 1 and presents a basis for our model.

Fig. 1. GNN Architecture in Ahasrabudhe and Bekkers GNN Soccer Paper



In addition, in our Bayesian Machine Learning course last semester, we attempted to use Bayesian methods to classify defensive coverages using the same data. Essentially, we were tackling the same problem as this project, but using a different framework, Bayesian logistic regressions. Here, we hope to see how deep learning and neural network methods compare in performance to statistical Bayesian methods. This is an opportunity to tackle a similar problem from a different framework. Our model in this project will utilize the tracking data from after the beginning of each play, in contrast to our previous Bayes project which only used the alignment of the defensive players pre-snap (No time component).

IV. Technical Approach

As mentioned in the above section, AWS's approach to solving this problem involved the use of Convolutional Neural Networks. They mentioned a potential extension to their work being the use of Graph Neural Networks instead, so that is the approach we decided to take. For the creation of our graph data, we made use of Python's Spektral package and associated documentation. We made a graph for each of the 10 frames per second recorded in our data. For each of these frames, we treated them as a separate case where we would make a prediction on the defensive coverage for that point in time. After all the predictions were made, we would then aggregate the results for a play, and pick the most predicted coverage as the play wide prediction.

Important to note in how we approached this problem is that we flipped all plays to be going to the same direction on the field. We felt this was important so that the x, y, and player orientation values would be consistent across all plays. We also made the ball position at the start of the play the origin (0,0) and adjusted the x and y coordinates of each player on the play accordingly.

Our data consisted of four matrices. The first was a node feature matrix with the dimension number of nodes by number of nodes. Ours was 23 x 6 with a node for each of the 11 players on offense and defense along with the ball. The features associated with these nodes were the player/ball's x and y coordinates, orientation on the field, whether they are on offense or defense, and the score differential at that point in time. We initially also included variables such as quarter, down and distance, defensive team, and yardline on the field. However, when we began training our GNN, we had issue with model performance and decided to scale back the number of predictors we were using. We feel that adding a few of these variables back in, particularly a defensive team variable that has been one hot encoded, would help us to improve our accuracy.

The second matrix was a adjacency matrix binary encoded to be a 1 if two players or the ball were considered to be an edge, which we decided would be if the players were less than 10 yards from each other at that point in time. This matrix was a 23 x 23 sparse matrix, which is number of nodes by number of nodes. One thing to note here is we debated whether or not an edge should exist between an offensive and defensive player. We decided that we would have these edges as we felt it may be important to include for wide receiver defensive back interactions. We also debated the distance at which we would consider a player connected but ultimately choose 10 yards as our number. Optimizing this choice could again be a point of improvement for our model.

Our third matrix was an edge feature matrix. For any of our edges, as defined above, we had a feature which was the euclidean distance the players were apart from each other. This matrix was number of edges by features, which was around 300 x 1 for an average frame of ours. We talked about other potential features to include here, such as speed differential between the players or angle apart they were. We decided to only include distance because of the training issues mentioned earlier.

The last matrix was our label or y matrix and this consisted of a one hot encoded matrix for each of our frames. With 7 potential coverages, this meant we had 7 columns with a 1 in that column if it was the coverage for that play. This matrix was a 1 x 7 matrix for each frame of a play. Potential coverages were covers 0 through 6.

Using the Spektral package, we were able to combine these matrices into a graph object intended to be used in combination with the tensorflow package in Python. Interesting to note here is that before the data is fed into our model, it is converted back into the matrice format. We created a 3 layered GNN with 128 hidden nodes per layer that utilized ADAM optimization, Leaky ReLu and Softmax activation, along with a 0.01 learning rate and approximately 100 epochs. We attempted to improve our model by changing the learning rate, activation functions, and the number of layers and nodes.

While we didn't find much success with these changes, we do believe that their are improvements to our model to be made using different values for some of these options.

V. RESULTS

Our best GNN model had 57% accuracy at the frame level. We also classified the defensive coverage for one whole play, which is a collection of frames. In order to implement this, we summed each of the defensive coverage prediction vector outputs for each frame on a play, then applied a softmax to the summed vector to get the final prediction probabilities for each coverage. Using this per-play method, our test accuracy was 63%. Expected accuracy from random guessing would be 14.3% (1/7), while guessing Cover 3, the most common coverage, every play, would yield an accuracy of 33.2% based on the actual frequency for each coverage. Of note is that our most commonly misclassified coverage was a mistake between cover 1 and 3. Both of these coverages have a single high safety with the main difference being the corner backs and line backers playing man coverage for cover 1 and zone coverage for cover 3.

We believe that we could add in some feature engineered variables that would help the model to better predict whether a play was man or zone coverage, leading to an improvement in the classification of cover 1 and 3. There are some clear man/zone giveaways we could take a look at including. For example, whether a player followed a motion across a formation presnap or if a linebacker is lined up on a tight end or running back out wide where a corner would traditionally be.

The paper from Amazon ML Solutions Lab had an 84%, but we believe that GNN's have the power to increase beyond this accuracy. We note that while other methods referenced in the related works section had higher accuracy, we were using data for half an NFL season, while those papers use tracking data from over four NFL seasons. In addition, our network was very shallow and small compared to other comprable CNN models due to resource constraints, and we believe that the accuracy can be greatly increased by creating comprable sized GNN's.

Furthermore, our Bayesian classification model had 48% accuracy, indicating that the GNN model outperformed the Bayesian model. This is due to two factors. First of all, this GNN model used both snap and post-snap tracking data, whereas the Bayesian model only had data at the snap. Secondly, neural networks and deep learning methods are generally more powerful than Bayesian networks in terms of overall accuracy, and we see that our project follows that trend.

Our Bayesian classification model and AWS's CNN followed similar approaches to solving this problem. Our Bayes model consisted of 400 one hot encoded grid boxes which encoded where each defensive player was at the start of each play. The CNN was fed image data which also worked to inform the model where each player was positioned on the field. Our GNN and the AWS CNN had distinct advantages over our Bayes model. While we had x and y coordinates as a feature of our model, we were also able to have various other features associated with each player on the field. We believe this to be a massive advantage pointing to the potential of GNNs over the other attempted methods. We feel that we did not have the necessary quantity of data or computing resources available to us to realize this potential.

VI. CONCLUSION

The important takeaways from our project are the power of deep learning models as well as the benefit of frame-by-frame prediction for a play consistiting of multiple frames. Deep learning methods have significant power in their prediction accuracy, and we see the quantitative increase in accuracy due to the deep learning model compared to our previous Bayesian model. Also, we note that calculating a new defensive coverage prediction for each frame can allow for mid-play live changes to defensive coverage predictions. This is extremely useful for coaches and analytics departments in the NFL, as it can provide key insights into exactly where and how the defensive coverage becomes obvious in a football play. For example, our model can be used to determine exactly what actions of the players lead the model to predict the coverage easier. Defensive coaches can use this to disguise their coverages more efficiently and offensive coaches can use this to identify coverages in more efficient, standardized ways with analytics.

There are multiple limitations to our work, opening the door for future work in this area. Our model only had three hidden layers, one edge feature, six node features, and a limited amount of epochs. We believe that with more resources, time, and correspondingly a larger neural net structure as well as more fine-tuned feature engineering, our model could have performed extremely well. For example, we had the hypothesis that we could add multiple feature engineered categorical variables such as number of safeties and location of cornerbacks at snap in relation to the wide receivers, could have significantly increased performance for our model. This paper serves as a benchmark for the power of GNN's in football defensive coverage classification, and we hope that future work can expand on the accuracy of our model.

VII. CODE

All of our code from this project can be accessed in this github repository.

https://github.com/cnickol26/DefensiveCoverageGNN

REFERENCES

[1] H. Song et al., "Explainable Defense Coverage Classification in NFL Games using Deep Neural Networks," 42 Analytics, Mar. 2022, Accessed: Winter 04, 2023. [Online].

[2] P. Singer and D. Gordeev, "The Zoo- NFL Big Data Bowl," kaggle.com, 2020. https://www.kaggle.com/c/nfl-big-data-bowl-2020/discussion/119400 (accessed Apr. 24, 2023).

[3] J. Gen, C. Nickol, and M. Manner, "Bayesian Classification for NFL Defensive Coverages," GitHub, Jan. 16, 2023. https://github.com/jdg9vr/bayesian-nfl-coverage/blob/main/BayesFinalReport.pdf (accessed Apr. 24, 2023).

[4] A. Rehman, "Implementation and Understanding of Graph Neural Networks(GNN)," Red Buffer, Dec. 26, 2022. https://medium.com/red-buffer/implementation-and-understanding-of-graph-neural-networks-gnn-54084c8a0e24 (accessed Apr. 24, 2023).

[5] D. Raabe, R. Nabben, and D. Memmert, "Graph representations for the analysis of multi-agent spatiotemporal sports data," Applied Intelligence, vol. 53, no. 4, pp. 3783–3803, Jun. 2022, doi: https://doi.org/10.1007/s10489-022-03631-z.

[6] A. Sahasrabudhe and J. Bekkers, "A Graph Neural Network deep-dive into successful counterattacks," 42 Analytics, Mar. 2023, Accessed: Apr. 24, 2023. [Online].

[7] KIH004, "Football 101: Coveragenbsp;shells," The Phinsider, 15-Aug-2014. [Online]. Available: https://www.thephinsider.com/2014/8/15/6000113/football-101-coverage-shells. [Accessed: 26-Apr-2023].