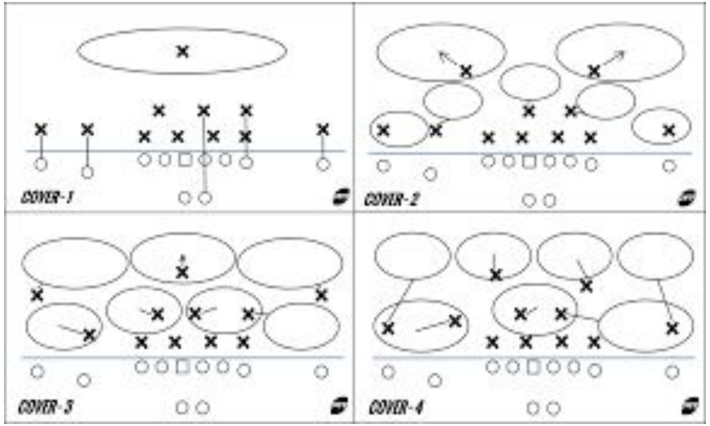
Classifying NFL Defensive Coverages With Graph Neural Networks



Matt Manner, Connor Nickol, Josh Gen

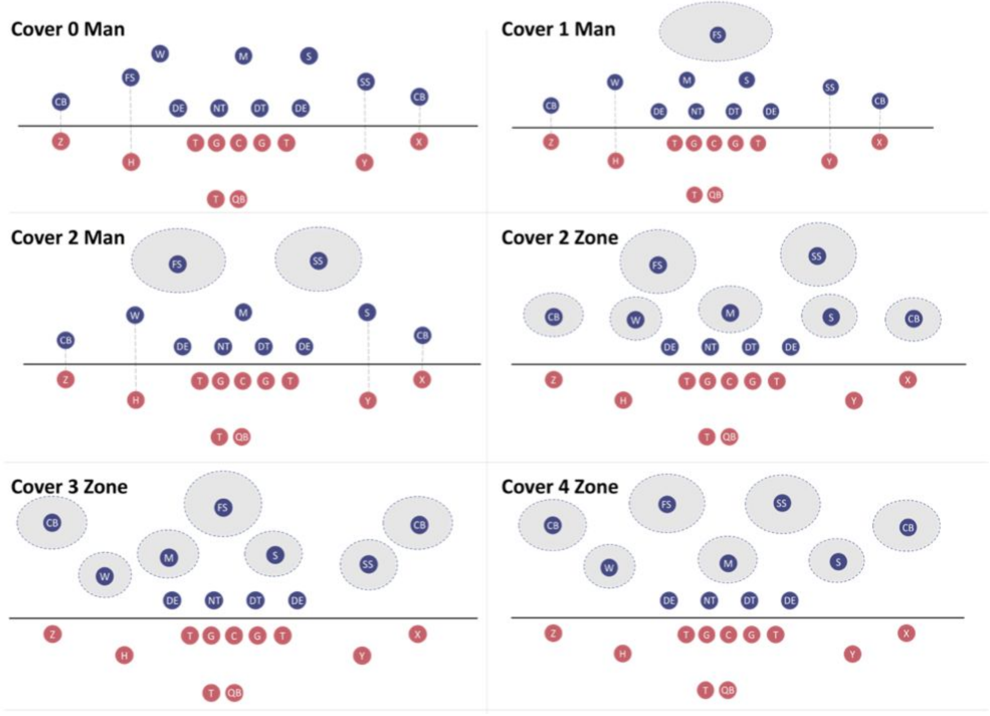
University of Virginia | School of Data Science

Background

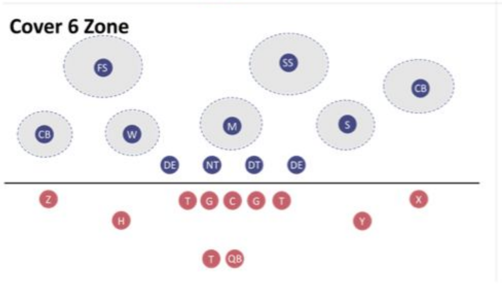
- NFL's yearly Big Data Bowl analytics competition
  - NFL releases Amazon Web Services (AWS) Next Gen Stats player tracking data
  - Our data: first 8 weeks of 2021 season
- Included data sets:
  - Players: name, unique ID, position for 1679 players
  - Plays: unique play ID, offensive/defensive team, pass coverage for 8557 plays
  - PFF: player, play, position for 188254 player-plays
  - Games: 10x/sec player & football location tracking data for each play
- Goal: classify defensive coverages from various post-snap features
  - Motivation: thousands of man-hours spent analyzing film for multi-billion dollar industry
  - Live classification of defensive coverages as they unfold; defensive matchup analytics

University of Virginia | SCHOOL of DATA SCIENCE

7 COVERAGE OPTIONS



- 3 Man Coverages
- 4 Zone Coverages
- Cover 2 Man is also known as Cover 5



Data Processing

- Consider 22 players (11 offense, 11 defense) + 1 football as nodes in an undirected graph
  - Information encoded in nodes and edges
  - Look at each frame individually and then aggregate predictions for a play level prediction
- Input data: adjacency matrix, edge matrix, node-feature matrix
  - **Adjacency matrix**: 23 x 23 (number of nodes x number of nodes)
    - Values: Binary 1/0 whether 2 nodes are connected by an edge (<10 yds apart)
  - **Edge Feature matrix**: #Es x 1 (number of edges x number of edge features)
    - One edge feature: Distance between the connected nodes
  - **Node Feature matrix**: 23 x 4 (number of nodes x number of node features)
    - Values: X & Y coordinates, defense/offense, frame number, player orientation, score differential
  - **Y matrix:** 1 x 7 (one for each play x number of label options)
  - Edge and node feature matrices were normalized

NODE FEATURE MATRIX

```
[6]: node_feat
```

[6]:

uniqueplayId	frameId	nflId	new_x	new_y	Defense	o	score_d	frames_after_snap
202109090097	**6.0**	**25511.0**	-0.980974	-0.008857	-0.957427	0.434343	-0.334252	-1.590890
	31.0	**25511.0**	-0.782499	0.114917	-0.957427	0.546177	-0.334252	-1.470628
	32.0	**25511.0**	-0.769268	0.138913	-0.957427	0.597867	-0.334252	-1.350365
	33.0	**25511.0**	-0.757690	0.169225	-0.957427	0.597867	-0.334252	-1.230103
	34.0	**25511.0**	-0.742804	0.203326	-0.957427	0.620312	-0.334252	-1.109841
...	**...**	**...**	...	...	...	...	...	...
20210919081081	**25.0**	**football**	-0.405397	0.056819	-0.957427	-1.660279	-0.628234	0.333308
	26.0	**football**	-0.403743	0.066923	-0.957427	-1.660279	-0.628234	0.453570
	24.0	**football**	-0.413667	0.049241	-0.957427	-1.660279	-0.628234	0.573833
	7.0	**football**	-1.012399	0.027770	-0.957427	-1.660279	-0.628234	0.694095
	6.0	**football**	-0.908200	0.027770	-0.957427	-1.660279	-0.628234	0.814358

4337018 rows × 6 columns

- Other potential features:
  - Down & Distance
  - Defensive Team (One hot encoded)
  - Quarter
  - Yardline

ADJACENCY MATRIX

[12]:
```
pd.DataFrame(adj_mat[0])
```

[12]:

	0	1	2	3	4	5	6	7	8	9	...	13	14	15	16	17	18	19	20	21	22
0	1	1	0	0	1	1	1	1	1	1	...	1	1	1	0	1	1	0	0	0	1
1	1	1	0	0	1	1	1	1	1	1	...	1	1	1	0	1	1	0	0	0	1
2	0	0	1	0	1	0	1	0	0	0	...	0	0	1	0	0	1	1	0	0	0
3	0	0	0	1	0	1	0	0	0	0	...	0	0	0	1	1	0	0	0	0	0
4	1	1	1	0	1	0	1	1	1	1	...	1	1	1	0	0	1	1	1	0	1
5	1	1	0	1	0	1	1	1	1	1	...	1	1	0	1	1	0	0	0	1	1
6	1	1	1	0	1	1	1	1	1	1	...	1	1	1	0	1	1	0	1	0	1
7	1	1	0	0	1	1	1	1	1	1	...	1	1	1	0	1	1	0	1	0	1
8	1	1	0	0	1	1	1	1	1	1	...	1	1	1	0	1	1	0	1	0	1
9	1	1	0	0	1	1	1	1	1	1	...	1	1	1	0	1	1	0	0	0	1
10	1	1	0	0	1	1	1	1	1	1	...	1	1	1	0	1	1	0	0	0	1
11	1	1	0	0	1	1	1	1	1	1	...	1	1	1	0	1	1	0	1	0	1
12	1	1	1	0	1	1	1	1	1	1	...	1	1	1	0	1	1	0	1	0	1

- 23 x 23
- Binary encoded
- 1 if players are less than 10 yards apart
- Could be adjusted to 15 or 20 yards potentially
- Both offensive and defensive players are connected

EDGE FEATURE MATRIX

```
[18]: e_temp

[18]: array([[ 0.05186691],
             [ 0.03718468],
             [ 0.00805848],
             [-0.01114553],
             [ 0.0289336 ],
             [ 0.0318355 ],
             [ 0.03305182],
             [ 0.03303774],
             [ 0.02983895],
             [ 0.02507425],
             [ 0.02405333],
             [ 0.01397416],
             [ 0.00791901],
             [ 0.01448363],
             [-0.01168777],
             [-0.00257598],
             [ 0.03218069],
             [ 0.03718468],
             [ 0.05186691],
             [-0.00576094],
             [ 0.00272568],
             [ 0.01713086],
             [ 0.02279601],
             [ 0.02914762],
             [ 0.03330279],
             [ 0.03647011],
             [ 0.02288395],
```

- Number of edges x number of edges features
- 345 x 1 here for the example shown
- Our feature is normalized distance the 2 players making up the edge are from each other
- Consider adding other features such as angle or speed differential

Y LABEL MATRIX

```
[19]: y
```

[19]:

uniqueplayId	frameId	0	1	2	3	4	5	6
20210909001078	**6.0**	0.0	0.0	0.0	0.0	1.0	0.0	0.0
	7.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
	8.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
	9.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
	10.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
...	**...**	...	...	...	...	...	...	...
2021102500933	**34.0**	0.0	0.0	0.0	0.0	1.0	0.0	0.0
	35.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
	36.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
	37.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
	38.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0

188699 rows × 7 columns

- One hot encoded coverage (1 - 7)
- Same coverage label for each play, one for each frame of the play

GNN MODEL ARCHITECTURE

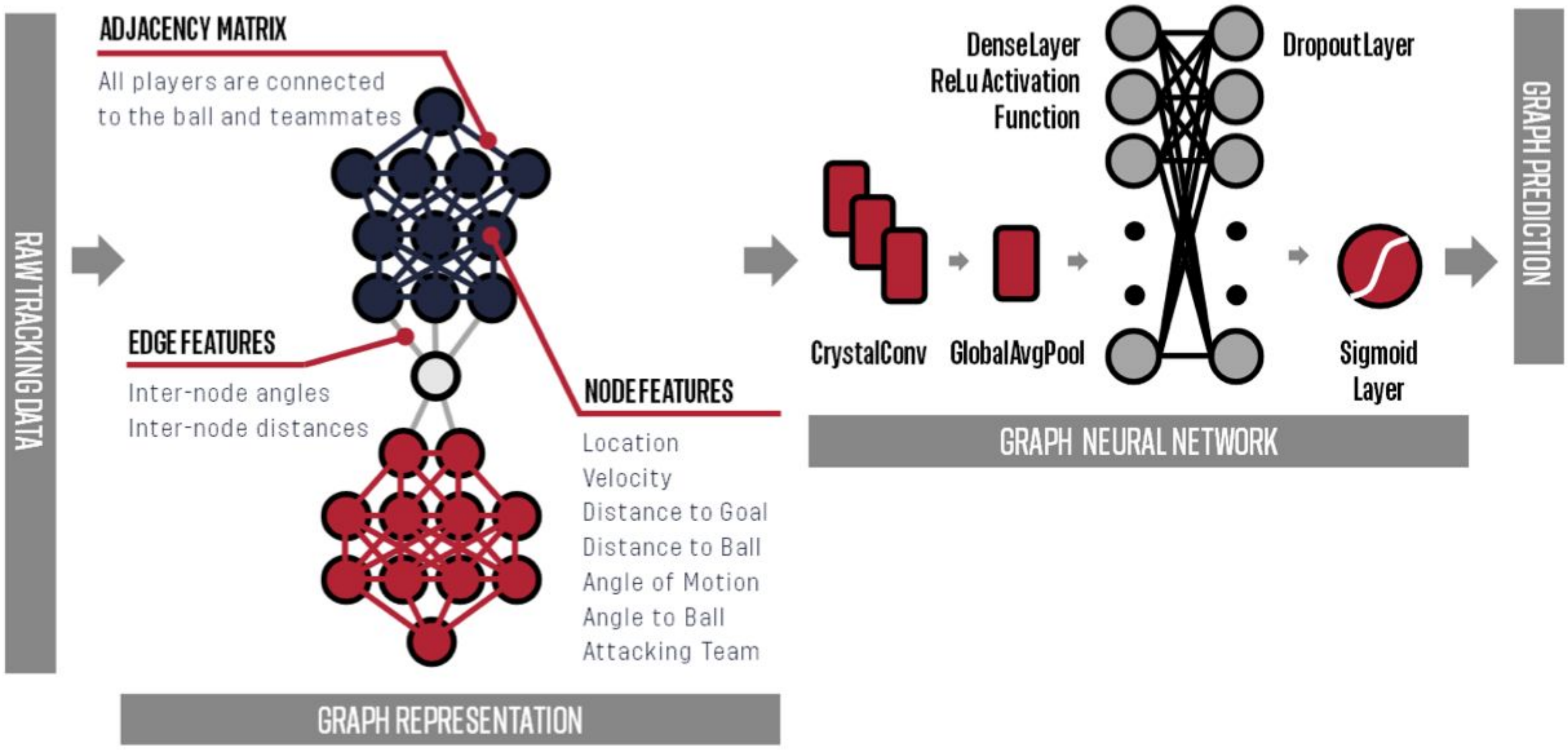


Figure 2: GNN model architecture

Methodology

- Inspiration from a soccer paper using GNN's attempting to classify successful counter attacks
- Used the Spektral Package for creating our graph data
  - Transforms the data into a suitable format for analysis in TensorFlow
- Model Information:
  - GNN with 3 hidden layers, 128 hidden nodes for each layer, 100 epochs
  - 0.01 learning rate, ADAM optimization, Categorical Cross Entropy loss function, Leaky ReLu & Softmax activation functions
  - Used an 80/20 train test split
    - About 188,000 frames making up 7,100 plays

Results and Analysis

- Our GNN correctly classified 57% of the frames
  - With aggregation this resulted in 63% accuracy at the play level
  - Training issues and time limited our ability to further improve accuracy
- Amazon ML Solutions Lab CNN achieves 85% accuracy using 8x the data
- Our Bayesian Logistic Regression Model from last semester was 48% accurate
  - The increased accuracy is due to including post-snap frames and using deep learning methods over Bayesian methods

[47]:

	uniqueplayId	coverage	predicted_coverage	count
0	202110170483	3	5	27
1	202110170673	3	5	13
2	202110170762	1	1	20

UNIVERSITY of VIRGINIA | SCHOOL of DATA SCIENCE

Conclusions

- Takeaways:
  - Graph neural networks can be more powerful than Bayesian methods
  - GNN’s are growing in popularity and usability (growth of Spektral)
  - Model updates **live, frame-by-frame predictions** of defensive coverage
- Limitations:
  - Network depth and parameters (relatively shallow, small network due to available resources/time)
  - Limited node features due to technical errors with Spektral
  - Lack of documentation/example work regarding Spektral and GNN’s in general
- Future Work:
  - Expanded node features and increased network size
  - Longer training/more epochs with further GPU resources
  - Add in features such as man coverage indicators to help the model with coverages commonly confused

UNIVERSITY of VIRGINIA | SCHOOL of DATA SCIENCE

REFERENCES

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

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