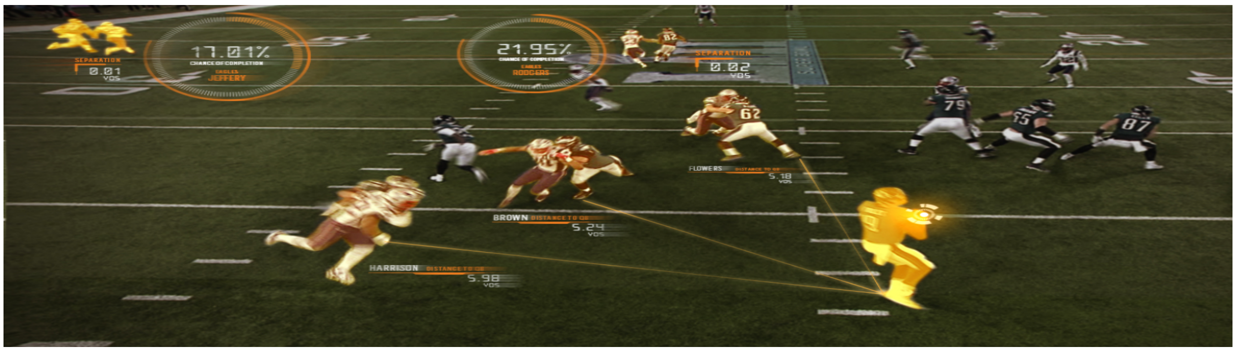
Unlocking the NFL Secret Playbook

A Data Mining Exploration of Winning Strategies in National Football League (NFL) Games

Alexander Meau
Computer Science
University of Colorado, Boulder
alme9155@colorado.edu



ABSTRACT

Predicting American Football game outcomes is a key focus of sport analytics. The National Football League's Next Gen Stats (NGS) provides rich player tracking data, but its volume and complexity challenge classic machine learning and statistical methods. This study aims to illustrate that Graph Neural Networks can serve as an effective self-representation learning tool for data mining.

Using a graph structure, players are represented as nodes, and interactions are represented as edges. Others attributes like velocity, acceleration, and orientation are captured as node features. A GNN aggregates and encodes frame-level graphs into embedding representations. These embeddings are extracted and clustered to reveal recurring tactical strategies. The GNN-based data mining approach can be evaluated in two stages: predictive performance, as measured by AUROC and confusion matrices, and pattern quality, as assessed by silhouette scores. Graph Neural Networks compress complex spatio-temporal interactions into groups of winning tactics, which coaches can use to formulate actionable strategies.

CCS Concepts

• Information systems → Data mining;

• Computing methodologies → Neural networks;

• Applied computing → Sports and entertainment;

Keywords

Sports analytics,

National Football League (NFL),

Graph Neural Network; Machine Learning;

1. INTRODUCTION

The National Football League's Next Gen Stats (NGS) has revolutionized sports analytics. Previously, coaches and analysts manually tracked statistics like yards gained and pass completions. Now, players and football carry RFID tags, with stadium receivers providing precise (x,y) coordinates, velocity, and acceleration 10 times per second. This high-frequency tracking creates a massive volume of data. Carnegie Mellon study [5] estimating 100 million data points in each NFL season. For example, each game tracks 22 players and the football, generating ~5 MB per game, or ~1.3 GB per season. Other than its overwhelming volume, these data capture complex play outcomes, interactions, and strategies, demanding advanced data mining and machine learning for meaningful insights.

Statistical approaches and classical machine learning methods are ill-equipped to the complexity or the sheer volume of NFL Next-Gen data. Statistical methods require aggregating raw tracking data into simplified categories, such as using chi-square tests to compare pass success rates near the sidelines versus midfield. However, this strips away rich spatial and temporal details and ignores multiplayer interactions that drive play outcomes. Classical machine learning methods fail to capture the rich dynamics of player interactions. For example, logistic regression, which relies on feature engineering, will need to calculate player distances to estimate quarterback pressure. The computed distance loses the dynamic geometry of game formation and player interactions. The limitations of statistical and classical machine learning approaches expose the need for deeper learning models that can directly model temporal-spatial tracking data to capture interdependent roles and coordinated movements in team formations.

Neural networks, and especially graph neural networks, provide one promising solution to address these shortcomings. In the sections that follow, this work outlines a data mining approach using a Graph Neural Network (GNN) embeddings to uncover recurring patterns and tactical strategies of successful NFL gameplay.

2. RELATED WORK

The applications of neural networks to sports analytics has grown in recent decades. Purucker [1] introduced neural networks for predicting football outcomes, but early work was limited by the absence of player-tracking data and thus weak spatial–temporal modeling [5]. Building on this, Blaikie et al. [2] used ANNs to predict game outcomes, outperforming traditional baselines. Anyama & Igiri [3] showed ANNs can capture the nonlinear complexity of NFL scores. The NFL Big Data Bowl competition [4] has illustrated tracking data can be used in machine learning to analyze pre-snap tendencies and player behavior. While these studies establish the usefulness of neural networks in sports analytics, few studies directly address the relational dynamics between players' geometry and group movements. Graph-based methods address this gap: Kipf & Welling [6] demonstrated that GCNs effectively learn from relational structure. Inspired by the work of Kipf and Welling, this project aims to explore the application of Graph Neural Networks (GNNs) to the 2025 NFL Big Data Bowl Dataset for data-mining analyses.

Competition Host
The National Football League

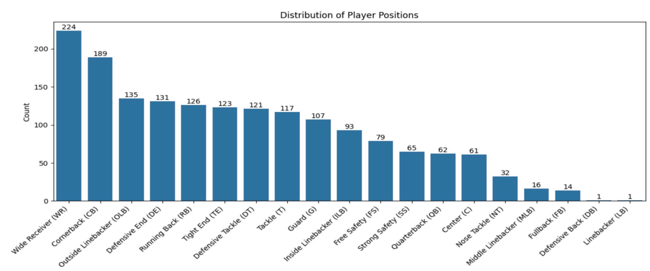
3. PROPOSED WORK

3.1 DataSet

This project uses the NFL Big Data Bowl 2025 dataset from Kaggle, an annual analytics competition run by the National Football League. The dataset combines player movement traces captured via RFID with game event information, providing rich spatiotemporal context. This pairing makes it especially well-suited for graph-oriented data mining of tactical patterns.

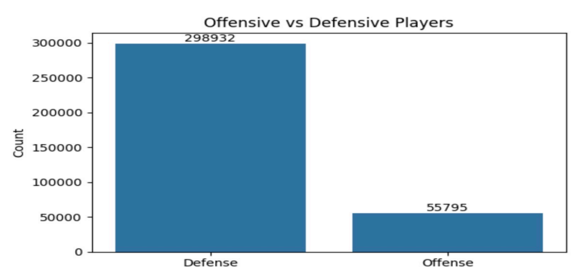
Four categories of data:

- **players.csv**: Contains player metadata, including IDs, names, positions, and player physical attributes.
- **plays.csv**: Describes play-level information, such as game context, play type, and outcomes.
- **player_play.csv**: Links players to specific plays, detailing their involvement and roles i**tracking_week_x.csv**: Provides frame-by-frame player tracking data, capturing positions, speeds, & orientations each week.



Data Size and Dimension:

- **players.csv:** ~2,000 player records
- **plays.csv**: ~10,000-20,000 play records **player_play.csv**: ~100,000 records linking players to specific plays.
- **tracking_week_x.csv**: ~10-100 million rows across weeks 1-9, capturing player coordinates, speed, and orientation at 10 Hz.



3.2 Main Tasks

This project proposes the application of a Graph Neural Network (GNN) to model the NFL NextGen tracking data for football analytics. Raw tracking data will be transformed into a graph structure, where nodes represent players and edges capture interactions. Each node will be enriched with dynamic features, including velocity, acceleration, and orientation. A sequence of frame-level graphs will be constructed for each play, tracking changes in player interactions over time. A GNN neural network model will then process these play-level graphs to generate latent embeddings that represent the tactical structure of each gameplay. The project aims to use neural network embeddings for data mining tasks. Through clustering embedding, this project seeks to identify recurring strategies and their link to successful gameplay.

The following is a list of primary tasks identified:

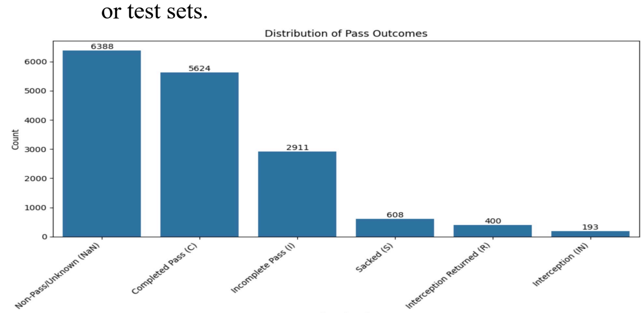
- **Data Preparation and Normalization**: Clean, and pre-process raw tracking data for proper joining and appropriate scaling.
- **Graph Construction:** Transform raw tracking data into a graph structure suitable for GNN processing.
- **GNN Modeling and Training**: Architect a GraphSAGE neural network model suitable as a binary classifier to predict the outcome of an offensive gameplay. The model will be evaluated based on the label for passing competition or the number of yards gained. .
- **Embedding Extraction and Pattern Clustering**: Compute play-level embeddings via global pooling of node features. Reduce dimensionality (e.g., UMAP or PCA), then apply K-Means to cluster recurring tactical patterns.
- **Pattern Evaluation**: Assess the quality of discovered patterns using quantitative metrics to evaluate their effectiveness.

3.3 Data Preparation and Normalization

To prepare the NFL tracking dataset for graph-based modeling, the data must be cleaned, standardized, and partitioned.

Here are a few key steps of preparation:

- **Entity joins:** Merge tracking_week*.csv with players.csv to include player roles (e.g., *'QB'* for quarterback, *'RB'* for running back). Keep key fields: gameId (int), *playId* (int), *nflId* (int), *frameId* (int), team (str), *x*, *y* (float, yards), o, *dir* (float, float), *side* (str), and a canonicalized *position*(str) from raw position fields.
- **Outcome labels:** Join plays.csv to attach play-level targets (e.g., *isWinningPass*, *isCompletedZeroGain*) to the corresponding tracking records.
- **Coordinate standardization:** Standardize player positions (x, y) by setting the line of scrimmage as x=0 and centering the field's width at y=0 (midfield). By adjusting player coordinates, this ensures consistent spatial representation across plays.
- **Filtering & de-duplication:** Keep only player rows (exclude the football), drop duplicate nflId records within a frame, and remove obviously invalid coordinates. (e.g., x < 0, x > 120 yards, y > 53.3 yards, or missing values for x, y).
- **Tracking Record Retention:** For each unique gameplay (gameId, playId), the earliest tracking record keep begins at the moment of the ball snap, or just before it. Discard earlier frames, as they represent pre-play movement noise.
- **Data split & leakage control.** Partition the data by gameId into 70% training, 15% validation, and 15% test sets. In order to prevent leakage of game-specific patterns, normalization parameters are fitted only on the training set to avoid information leakage from validation or test sets.



3.4 Graph Construction

To reduce data volume, only offensive team players are represented in the graph structures. The following are the key steps for creating the graph:

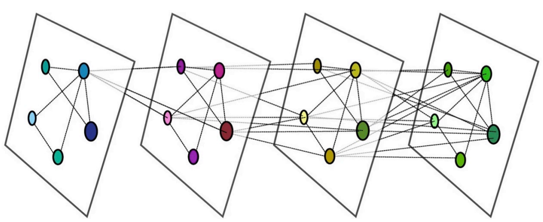
- Node selection: For each frame of tracking data, select offensive team players as nodes. Players from the offensive team are identified by the presence of a quarterback (QB). Records with fewer than 8 offensive players or without a quarterback will be ignored.

- Base edges: Connect the QB to nearby offensive players based on distance, using the 30th percentile of non-zero distances.

- Receiver Priority: Always connect the QB to receiving roles, e.g., Wide Receivers, Tight Ends, & Running Backs, regardless of their distance.

- Nearest Neighbor Fallback: Add nearest neighbor connections to avoid isolated nodes or small graph components.

- Edge Pruning: Limit each node to two edges, prioritizing connections to receivers (TE, WR, RB), for a lean graph.



3.5 Graph Neural Network (GNN)

A three-layer GraphSAGE neural network model is chosen as the binary classifier to learn whether a given offensive play results in a successful outcome. The model architecture includes the following:

- Three GraphSAGE convolutional layers (SAGEConv) with 128 hidden units on each layer, followed by batch normalization and ReLU activation

- Dropout (p=0.3) after each layer to regularize and reduce overfitting during message passing

- A global mean pooling is applied to produce a graph-level embedding for each play.

- The network ends with a sigmoid to produce a probability outcome.

Since the dataset contains more unsuccessful gameplay than complete passes, a weighted, binary cross-entropy loss (*BCEWithLogitsLoss*) loss function has been chosen to mitigate bias caused by class imbalance. An *AdamW* optimizer is used for optimization, with a learning rate of 1e-3 with gradient clipping (L2 norm) to prevent exploding gradients.

Although a deeper architecture may capture more complex interactions between players than my 3-layer architecture, a trade-off is made to use a simpler architecture to reduce computational complexity, as Google Collab has limited GPU hours available to free subscribers. Despite its architectural simplicity, the GraphSAGE model achieved a robust AUROC of 0.721, confirming that the chosen trade-off in favor of computational efficiency is justified.

```
Starting training (CPU only)...
Epoch 01 | lr 1.0000e-03 | loss 0.8824 | val acc 0.685 auroc 0.729 auprc 0.514
Epoch 02 | lr 1.0000e-03 | loss 0.8436 | val acc 0.695 auroc 0.731 auprc 0.517
Epoch 03 | lr 1.0000e-03 | loss 0.8374 | val acc 0.698 auroc 0.744 auprc 0.526
Epoch 04 | lr 1.0000e-03 | loss 0.8238 | val acc 0.701 auroc 0.742 auprc 0.526
Epoch 05 | lr 1.0000e-03 | loss 0.8238 | val acc 0.692 auroc 0.745 auprc 0.530
Epoch 06 | lr 1.0000e-03 | loss 0.8160 | val acc 0.687 auroc 0.744 auprc 0.537
Epoch 07 | lr 1.0000e-03 | loss 0.8133 | val acc 0.687 auroc 0.747 auprc 0.544
Epoch 08 | lr 1.0000e-03 | loss 0.8078 | val acc 0.690 auroc 0.742 auprc 0.526
Epoch 09 | lr 1.0000e-03 | loss 0.8054 | val acc 0.696 auroc 0.745 auprc 0.531
Epoch 10 | lr 1.0000e-03 | loss 0.8080 | val acc 0.683 auroc 0.748 auprc 0.533
Epoch 11 | lr 1.0000e-03 | loss 0.8007 | val acc 0.687 auroc 0.742 auprc 0.528
Epoch 12 | lr 1.0000e-03 | loss 0.7994 | val acc 0.698 auroc 0.740 auprc 0.531
Epoch 13 | lr 1.0000e-03 | loss 0.7992 | val acc 0.689 auroc 0.746 auprc 0.530
Epoch 14 | lr 1.0000e-03 | loss 0.7927 | val acc 0.684 auroc 0.746 auprc 0.537
Epoch 15 | lr 1.0000e-03 | loss 0.7955 | val acc 0.694 auroc 0.747 auprc 0.529
Early stopping triggered.

Best epoch: 7 | best val score: 0.544
Final (best) val – acc 0.687 auroc 0.747 auprc 0.544
Test – acc 0.653 auroc 0.721 auprc 0.476
```

4. EVALUATION

A multi-level approach is used in my project to assess the effectiveness of the tactical gameplay pattern extracted from the GraphSAGE GNN model:

- Model Predictive Accuracy
- Pattern Cohesion Assessment

4.1 Model Predictive Accuracy

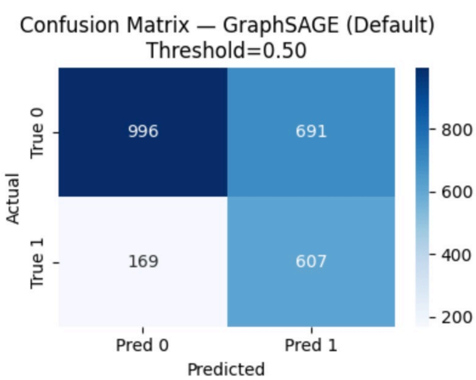
Since the tactical offensive pattern is extracted from the embeddings of the GraphSAGE neural network model, the effectiveness of the Graph Neural Network directly influences the quality of the patterns extracted. The first level of evaluation, therefore, measures the model's ability to correctly classify offensive plays (e.g., successful passes and yardage gained).

We evaluated the predictive performance of the GraphSAGE model for binary classification using both **threshold-free** and **threshold-based** metrics. Threshold-based metrics, in general, provide a comprehensive ranking of model performance, whereas threshold-based metrics measure performance using the default value of 0.5 for the decision boundary.

Threshold-Free Metrics:

- Accuracy (AUROC): 0.721
- Precision (AUPRC): 0.471

The AUPRC score is notably higher than the baseline prevalence of the positive class (**0.315**), indicating that the model can reliably distinguish between successful and unsuccessful plays even under class imbalance.



Threshold-based Metrics (0.5 default value):

- Accuracy: 0.651
- Precision: 0.468
- Recall: 0.782
- F1-Score: 0.586

The best F1-score was achieved at a threshold of 0.5, indicating a good trade-off between precision and recall. The high recall indicates the model successfully identifies most positive plays, while the moderate precision reflects the challenge of making confident positive predictions under an imbalanced dataset.

Overall, these results confirm that the GraphSAGE model outperforms random guessing and demonstrates solid generalization in both ranking and classification, providing a strong foundation for extracting reliable and interpretable tactical gameplay patterns from the learned embeddings.

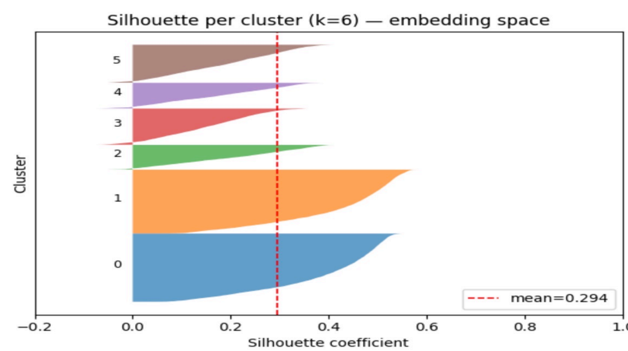
4.2 **Tactical Pattern Extraction**

Using the trained Graph Neural Network (GNN), each play's graph is encoded into an embedding through a forward pass in the following pipeline:

(SAGEConv>BatchNorm->ReLU)3x -> Global Mean Pool. As the SAGEConv layers were trained using supervised learning on the "completed pass" label, their weight and bias parameters capture generalized patterns of successful gameplay. This feed-forward pipeline embeds latent patterns of winning strategies.

4.3 Pattern Cohesion Assessment

After converting all graphs into embeddings in a standardized space, we cluster the embeddings using K-means (with K = 6). The optimal cluster is selected based on its silhouette score, which measures the highest internal cohesion (compactness within the cluster) and external separation (distinction from other clusters). Elements within the clusters represent distinct game plays, enabling the identification of recurring tactical patterns. The significance of the silhouette score lies in its ability to quantify the quality of clusters. In the context of this project, a cluster with the best silhouette score ensures the identification of a highly distinct and representative gameplay pattern of winning strategies.



4.4 Possible Pattern Interpretations

The NFL Big Data Bowl Dataset, compiled from data collected by various teams, reveals recurring tactical patterns. Clusters with the highest silhouette scores indicate potential best practice offenses across teams, with success rates listed in the table below. These findings can provide coaches with an advantage in making strategic decisions.

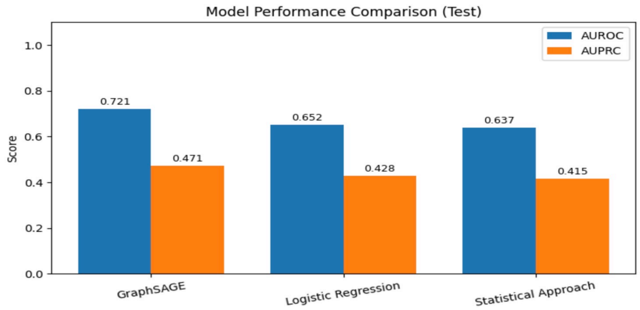
(BEST cluster auto-selected = 1)

Top formations in BEST cluster — ranked by FREQUENCY:

	formation	play_count_in_cluster	success_rate_in_cluster	success_rate_global	play_count_global
4	SHOTGUN	2911	0.508	0.407	8791
0	EMPTY	631	0.566	0.539	1342
5	SINGLEBACK	290	0.441	0.187	3915
3	PISTOL	80	0.463	0.193	641
1	I_FORM	73	0.438	0.157	1035

4.3 Other Modeling Methods

Comparing the best AUROC and AUPRC scores using a statistical method and classical machine learning with logistic regression, GraphSAGE achieves the highest AUROC (0.721) and AUPRC (0.471), outperforming both logistic regression (AUROC: 0.652, AUPRC: 0.428) and a statistical approach (AUROC: 0.637, AUPRC: 0.415). This comparative result validates that the use of GNN-based representation learning is a more effective foundation for data mining patterns from the complex, spatio-temporal NFL dataset.



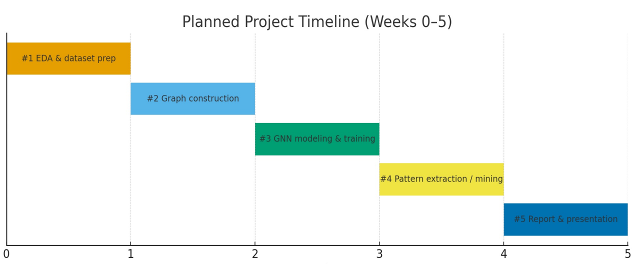
5. DISCUSSION

5.1 Project Milestones and Status Update

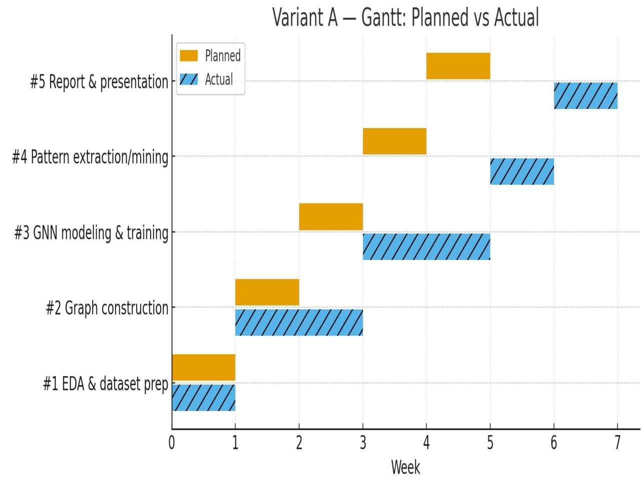
#	Description	Estimate	Actual	Progress
#1	EDA and Dataset Preparation	1 week	1 week	Complete
#2	Graph Construction	1 week	2 week	Complete
#3	GNN modeling and training	1 week	2 week	Complete
#4	Pattern extraction mining	1 week	1 week	Complete
#5	Project report and PPT	1 week	1 week	Complete

5.2 Project Status and Timeline

- Planned Project Timeline



- **Actual Project Timeline**



This data mining project was initially planned to span five weeks. Due to the complexity of GNNs and my limited prior experience with the APIs, the project suffered a two-week delay. Overall, the schedule was 40% over the original budgeted timeline. Despite the schedule delay, the project concluded with a successful final deliverable.

5.3 Key Challenges and Accomplishments

A major challenge of the project was the substantial memory footprint of the GNN model. Kaggle's free environment has limited GPU and RAM. GNNs keep many nodes and edges in memory. Execution often failed with unknown platform errors. Multiple attempts to reduce the memory footprint were unsuccessful. These challenges led to a critical decision to narrow the project scope to offense-only data and moved to paid Google Colab. After dropping more than half of the data, the runs stabilized.

The learning curve associated with graph modeling and Graph Neural Networks (GNNs) was steep. Converting tracking geometry into nodes and edges required a new mindset and substantial experimentation. PyTorch Geometric API is powerful but complex. The project schedule is delayed by 2 weeks in the GNN phase. Despite a two-week schedule slip, the payoff was significant. The GNN prediction outperforms logistic regression's 0.70 AUROC by 21% and 0.65 AUPRC by 23%. It also surpassed statistical methods by 31% in AUROC and 33% in AUPRC. In retrospect, overcoming the learning curve through persistence is among my greatest achievements. The knowledge I have gained stands as my biggest reward.

6. CONCLUSION

6.1 Project Summary

This project utilizes a Graph Neural Network (GNN) to extract game strategies from the NFL Big Data Bowl dataset. Gameplay tracking data is modeled as player-interaction graphs, where players are represented as nodes and interactions are represented as edges. Dynamic attributes, such as position, speed, and acceleration, are learned using a GraphSAGE neural network model to produce play-level embeddings. Trained embeddings are extracted and clustered into recurring patterns, enabling coaches to perform tactical analysis. This approach transforms complex, spatio-temporal tracking data into compact, interpretable information.

6.2 Key Findings

The concept of extracting neural network embeddings in data mining applications is a form of representation learning, where raw data is mapped into a lower-dimensional latent space to support downstream analysis. Although the concept of extracting information from neural network embeddings is not entirely new, it has numerous applications in domains such as social network analysis and molecular chemistry research. Applying GNNs to sports tracking data is a relatively unexplored area. This new application addresses a gap in sports analytics. Instead of focusing on the prediction evaluation, the internal embeddings of the neural network model are equally valuable as features for data mining purposes.

6.3 Future Work

The scope of this project is limited to the 2025 NFL Big Data Bowl dataset on Kaggle, which provides a usable foundation for evaluating data mining techniques with Graph Neural Networks (GNNs). A natural extension of this work will involve leveraging the richer and more recent NFL Next Gen Stats (NGS) data to capture broader and more representative game contexts. Another future research is to explore the integration of visual and spatio-temporal transformers. While this project focuses on modeling player interactions through graph structures, transformers have shown strong potential in capturing both fine-grained local interactions and global play context.

REFERENCES

[1] Purucker, M. C. 1996. Neural network quarterbacking: Comparing different training schemes with application to the football game. IEEE Potentials, Vol. 15, Issue 3 (Aug.-Sept. 1996), 9-15. DOI: 10.1109/45.535226.

[2] Andrew D. Blaikie, Gabriel J. Abud, John A. David, and R. Drew Pasteur. 2020. NFL & NCAA football prediction using artificial neural networks. *College of Wooster and Virginia Military Institute Technical Report*.

[3] Oscar Uzoma Anyama and Chinwe Peace Igiri. 2015. An application of linear regression & artificial neural network model in the NFL result prediction. *International Journal of Engineering Research & Technology (IJERT)* 4, 1 (Jan. 2015).

[4] NFL Data & Analytics. 2025. 2025 Big Data Bowl Finalists. National Football League, New York, NY.

[5] Timothy C. Chartier, Kevin Quealy, et al. 2019. Big data of small things in football. *arXiv preprint* arXiv:1906.03339.

[6] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR '17)*, Toulon, France, April 24–26, 2017.