Using Keras Neural Network for Predicting Pressures in the NFL

Ian Pope, *University of Central Missouri*, Warrensburg, USA, ijp74190@ucmo.edu

Abstract

This paper looks at evaluating offensive line play in the National Football league using machine learning. The paper evaluates the performance of offensive lineman during pass plays by examining pressures and creating a new metric of expected pressure. Tracking and play data is gathered from the National Football League's Big Data Bowl 2025 competition. The data is comprised of the first nine weeks of the 2022-2023 NFL season. The experiment utilizes the Keras Library to create a predictive neural network. The output of the network utilizes a sigmoid function to generate a score. That score is then compared across all points in a play for an individual and predicts if a pressure occurs on that play. The results of each play are grouped together and used to evaluate the performance of the metric to actual results and compare the players against each other.

Index Terms

Keras, neural network, NFL, offensive line, pressure rate, prediction.

I. Introduction

THE National Football League or NFL is the largest and most prolific sports league in North America, and as such, teams are always looking for the next way to get ahead. In recent years, this has led to a boom in the field of data analysis within the sport. One of the key aspects of NFL data analysis is in player evaluation, with more and more complex ways of determining the qualities of players. However, for offensive linemen much of their evaluations still rely on more traditional metrics such as sacks allowed, team rushing yards, or just the eye-test. These traditional metrics often fail to capture the complexities around playing on the offensive line and it can be hard to separate what role an individual played in a success or failure [1]. As the NFL continues to evolve, it is important that we continue to develop more advanced ways to evaluate offensive lineman. In the NFL, the broad success of pass heavy offenses has led to passing taking over the NFL. With lots of passing, it is highly important for the offensive line to be able to protect the quarterback. This paper introduces a new way of evaluating offensive lineman on passing plays by combining player tracking data with pressure charting data to create a new metric of expected pressures allowed. The goal is to provide another tool for players, coaches, and decision makers to evaluate the performance of offensive lineman and find ways to extract value.

II. Dataset

A. Data Sources

The datasets used in the analysis of the NFL offensive linemen come from the National Football League's Big Data Bowl 2025 contest that is hosted on Kaggle [2]. The Big Data Bowl is a competition that is hosted yearly to advance the discipline of sports analytics for the NFL. For the 2025 contest, over eight gigabytes of data were released for public use. This consists of five different datasets: games, plays, players, play, playerPlay, and tracking. The data was collected from every game in the first nine weeks of the 2022-2023 NFL season. In total, information exists for every player on every play in every game, every tenth of a second throughout a play.

B. Dataset Details

The games dataset consisted of basic game information, the players dataset consisted of information about a specific player including height, weight, and position played on the field, and the play dataset consists of play wide information such as the time of the play, down and distance information, pre-play scoring information, and more advanced fields such as what type of coverage was ran on the play. From the 'isDropback' field within the plays dataset, the plays were filtered down to only include plays that were dropbacks. This does not mean that the play always resulted in a pass, just that a dropback happened on the play. The fourth dataset is playerPlay that contains information that is specific to a player on a given play, including information regarding if a player had a dropback on a play, allowed a pressure, and who a blocker's primary assignment was on a given play. The final dataset is the tracking data. The tracking data is broken up by week over nine files. This dataset contains data such as location, speed, acceleration, direction, and events for that moment in time. The data is captured every tenth of a second from the breaking of the offensive huddle to a few frames after the play is over.

III. Feature Selection and Design

The datasets provided from the Big Data Bowl 2025 were combined, altered, and had the features selected to create the dataset used for the training of the neural network used to make predictions for expected pressures. Some features are included in the dataset but are not included during the actual training of the model. These features are gameId, playId, frameId, and nflId. Thirty-two other features were used in the training and are listed: quarter, down, yardsToGo, gameClock, playAction, homePointLead, offenseFormationEMPTY, offenseFormationIFORM, offenseFormationJUMBO, offenseFormationPISTOL, offenseFormationSHOTGUN, offenseFormationSINGLEBACK, offenseFormationUNKNOWN, offenseFormationWILDCAT, hadDropback, blockerCount, isCenter, isGuard, isTackle, x, y, s, a, dis, o, dir, passOut, pointDifferential, distanceToQb, distanceToDef, defToQB, beforeSnap. 'pointDifferential' is calculated by subtracting the homePoints by the awayPoints and multiplying by negative one only if the player is part of the away team, while offenseFormationX is a one-hot encoding of the column offenseFormation. The fields isCenter, isGuard, and isTackle are encodings of the field position. Finally, distanceToDef, distanceToQB, and defToQB are calculated by the euclidean distance between the blocker and either the player indicated in the blockedPlayerNFLId field or the player where isDropback is true for a given play and frame, or the distance from that same defender to the player with a dropback. All tracking data was translated so that all plays traveled the same direction, and all data that is used in the training of the neural network were modified using min-max scaling to put the data between zero and one. The last feature used in the network is the predicted value of allowedPressureAsBlocker. This field indicates whether the specified blocker ever allowed a pressure at any point on the given play.

IV. Implementation

A. Overview

To predict when a play will result in an offensive lineman allowing a pressure, a model was built using Keras's sequential model. Before training, the training and testing groups were divided, with rows containing the same gameId, playId, and nflId belonging to the same group to maintain individual play integrity and was split with 80% of the data for training and the remaining data for testing and validation. Although the total data includes hundreds of thousands of rows, there are only about 30,000 distinct combinations of game, play, and player. Additionally many of the rows are data from before a play started or could contain misleading data. Even though the data was sent to training frame by frame, because of the low numbers distinct plays multiple measures were taken to prevent over fitting [3].

```
model = Sequential()
model.add(Dense(128, input_dim=32, activation='relu', kernel_regularizer=l2(0.01)))
model.add(BatchNormalization())
model.add(Dropout(0.4))
model.add(Dense(64, activation='relu', kernel_regularizer=l2(0.01)))
model.add(BatchNormalization())
model.add(Dropout(0.4))
model.add(Dense(64, activation='relu', kernel_regularizer=l2(0.01)))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Dense(32, activation='relu'))
model.add(BatchNormalization())
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer=Adam(), loss='binary_crossentropy', metrics=['AUC', 'accuracy', 'precision', 'recall'])
model.summary()

class_weight = {0: 1., 1: 1.}
early_stopping = EarlyStopping(monitor='val_loss', patience=8, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=3, min_lr=0.0001)
history = model.fit(X_train, y_train, epochs=50, batch_size=48, validation_data=(X_test, y_test),
          class_weight=class_weight, verbose=1, callbacks=[reduce_lr, early_stopping])
```

Fig. 1: Code for the creation of predictive model

B. Model Structure

The main components of the model consisted of five layers, an input dense layer, three intermediate dense layers, and an activation layer. Each of the first three layers in the sequential model utilized an input regulator l2 with a value of 0.01 and a were followed by a dropout layer with rates of 0.4, 0.4, and 0.2 respectively. Both of these parts of the model were used to reduce over fitting. Additionally, each of the first four layers utilized a relu activation function and were immediately followed by a batch normalization layer. This was important to be able to converge on an answer quickly, as the dataset was large and required a larger batch size. The final activation layer consists of a single neuron. This node utilizes a sigmoid activation function that results in a value between zero and one. This resulting value acts as a score, where if the predicted value ever reaches over a certain threshold, the play will be marked as having an expected pressure.

C. Compilation and Runtime

The skeleton of the model was then compiled using adam as the optimizer, loss of binary-cross entropy and monitoring area under the curve, accuracy, precision, and recall. The model was fit with an initial epochs of 50, a batch size of 48, and callbacks of early stopping and reduce learning rate on plateau. The early stopping callback monitored the validation loss, has patience set to 8, and restored the best weights. The learning rate reducer also monitored validation loss, had a factor of 0.5, and a patience of 3. Multiple variations of configurations were attempted, however these were the values that produced the best result. In addition, experiments were conducted adjusting class weights, making one's appear between one to ten times more valuable than zero's, however the highest performing configuration was one to one.

V. Results

During training of the model, the early stopping callback kicked in and stopped the model after the 36th epoch. Because the neural network was fed frame by frame information and we want to determine play by play, it was difficult to determine the success of the model over the epochs ecept through the loss function. Across epochs, the validation loss slowly fell and resulted in a value just below 0.27. Meanwhile, the validation precision and recall only started to perform better sometime after the 20th epoch. During

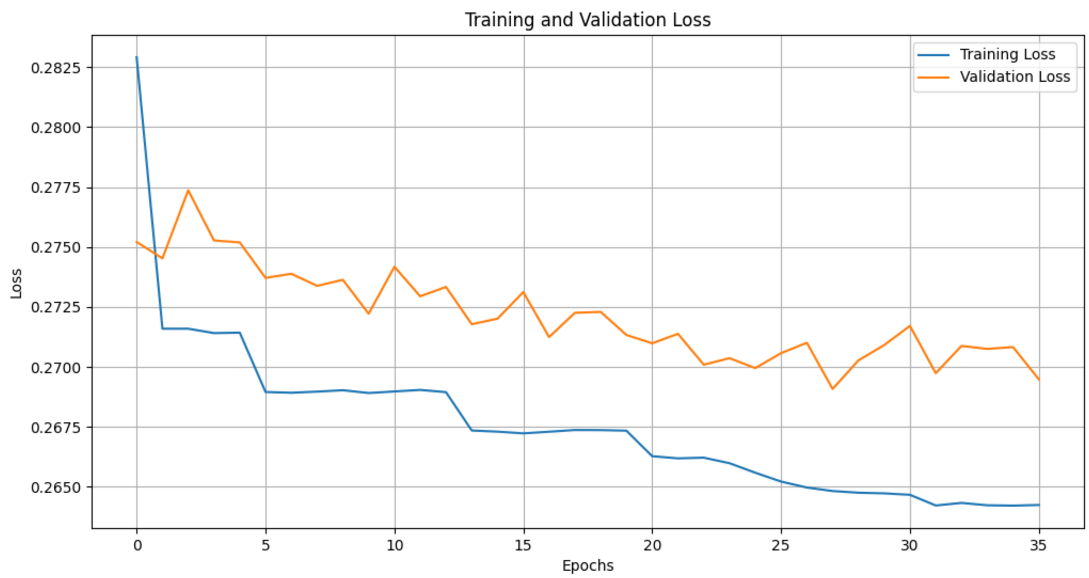


Fig. 2: Training and validation loss over epochs

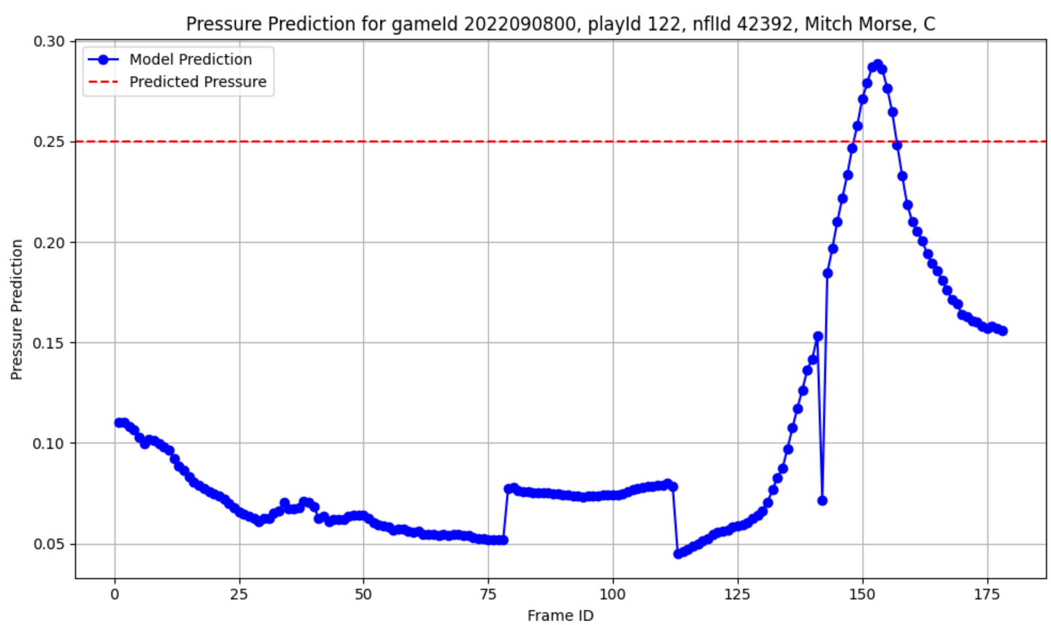


Fig. 3: Example of play that results in predicted pressure

the testing phase each frame was evaluated using the model. Different cutoff points for the scores were tested with the final result 0.25. This means that any play that contained a frame with an predicted value greater than 0.25 would be labeled as a predicted pressure. The value of 0.25 was selected because the total number of plays marked as a predicted pressure aligned relatively closely with the number of actual pressures, although roughly 25% percent less often. Lowering the threshold quickly increased the number of plays as predicted pressures and the precision decreased. When looking at the confusion matrix we can calculate recall of 0.464 and precision of 0.539 on the testing data after grouping plays based on game, play, and nflId.

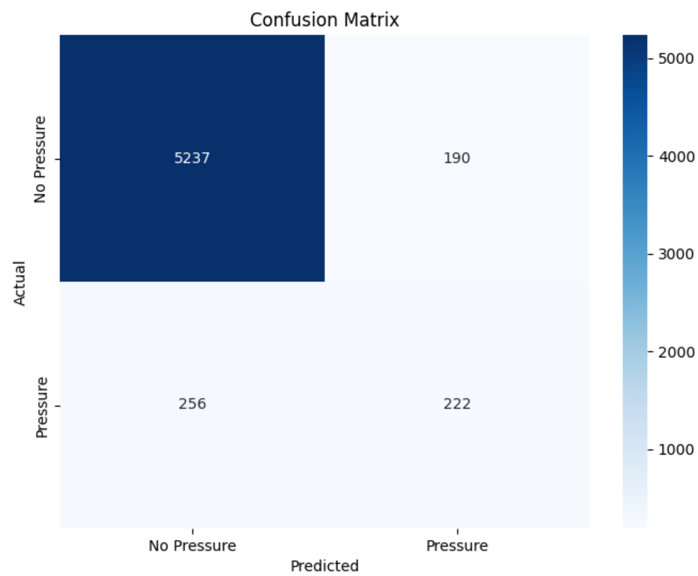


Fig. 4: Confusion matrix for testing data

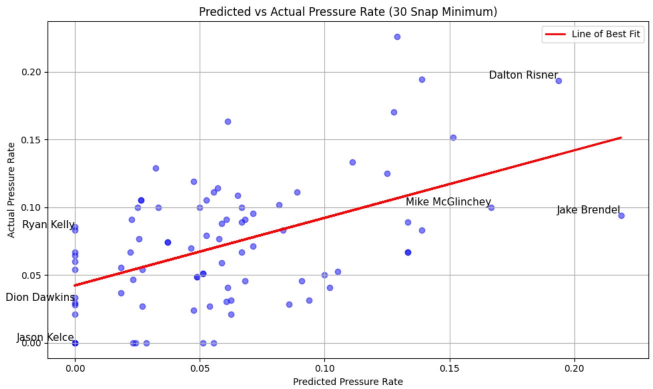
VI. Applications

A. Showcasing Outliers

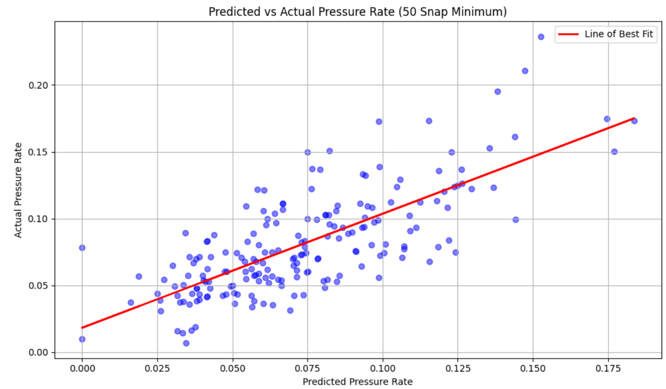
Typical pressure numbers do not account for many things. If a player gets beat on a play but their quarterback throws the ball quickly or if they do their job correctly but because someone else did not, it puts them in a position to give up a pressure that they did not deserve to give up [4]. One of the applications of expected pressure is to compare the value with the traditional pressure rate given up by a player. Because of the limited nature of the data, all of the data has been included in the following evaluations and contains total play minimums of 50. The largest difference between rates with a predicted rate of 0.153 to an actual value of 0.236 is tackle, Bernhard Raimann, meaning that the model was significantly more generous to him than the traditional model. That said, his 0.153 expected pressure rate is still the fourth worst of all lineman in the dataset. Another player the model is much higher on is center James Ferentz, with the model predicting a 0.0 rate to an actual 0.078. Although 0.078 is an average rate for a center, it is quite possible Ferentz was playing better than his actual rate. The model was also lower than the actual rate for many players. The model was lowest on offensive guard Isaac Seumalo, with a predicted score 0.05 higher than the actual. Notable players such as Elgton Jenkins and Connor McGovern were also predicted to have higher rates by the model.

B. Player Rankings

Another application is to compare players based solely on their predicted pressure rates. When comparing the raw number it is often better to evaluate each position independently, as tackles are more likely to be predicted and have actual pressures occur in a game as opposed to their interior counterparts. Perhaps the most coveted position on the offensive line are the offensive tackles, especially when it comes to pass protection. Among the top performers for offensive tackles are some names of players widely considered to be among the best in Terron Armstead and Tristin Wirfs, but perhaps more interesting are the names that are not as well known. The highest performer among offensive tackles for expected pressure percentage is Cedric Ogbuehi and third is D.J. Humphries, both of whom fans may know but are from household names. At the top of the offensive guards are notable players Michael Onwenu, Justin Pugh, and Joe Thuney. At the bottom of the list there are also some recognizable names, although not as highly thought of as those previously mentioned, in Samuel Cosmi, Dalton Risner, and David Edwards. Finally at center,



(a) Testing data



(b) Total data

Fig. 5: Scatter plot comparing predicted to actual pressure rates

the top of the list holds superstar centers of Rodney Hudson, Jason Kelce, and the previously mentioned James Ferentz. While at the bottom there is Lucas Patrick with an expected pressure rate of 0.175, 0.048 higher rate than the next worst center.

Top and Bottom 5 Pressure Rate Difference (Min 50 Snaps)

Name	Predicted	Actual	Difference
Isaac Seumalo	0.124	0.075	0.05
Cameron Fleming	0.116	0.068	0.048
Rob Havenstein	0.144	0.1	0.045
Elgton Jenkins	0.099	0.056	0.043
Connor McGovern	0.118	0.079	0.039
Joseph Noteboom	0.082	0.151	-0.068
Dennis Kelly	0.099	0.173	-0.074
George Fant	0.075	0.15	-0.075
James Ferentz	0.0	0.078	-0.078
Bernhard Raimann	0.153	0.236	-0.083

(a) Max differences from predicted to actual rate

Top and Bottom 5 Predicted Pressure Rate (Min 50 Snaps)

Name	Predicted	Actual	Difference
James Ferentz	0.0	0.078	-0.078
Rodney Hudson	0.0	0.01	-0.01
Michael Onwenu	0.016	0.037	-0.021
Cedric Ogbuehi	0.019	0.057	-0.038
Jason Kelce	0.025	0.044	-0.019
Larry Borom	0.147	0.211	-0.063
Bernhard Raimann	0.153	0.236	-0.083
Lucas Patrick	0.175	0.175	0.0
Samuel Cosmi	0.177	0.15	0.027
Dennis Daley	0.184	0.173	0.01

(b) Highest and lowest predicted rates

Fig. 6: Tables showcasing variation and performance for predicted pressure rate

VII. Conclusion

With the importance of passing and subsequently pass blocking in this day and age in the NFL, creating new ways to study and analyze offensive lineman can help give evaluators another tool in their tool belt to break down a notoriously difficult position to study. Applying this technique at the entire line level may also be an interesting way to approach evaluating offensive lines as a whole. There are also a number of things I wish I had the time to change or experiment with, however with the time required in pre and post processing of data, was limited to. I would like to see the model change from a play by play to predicting if there is a pressure in the next n seconds instead. I anticipate this would improve the models accuracy and be a better predictor. However, I both hope and believe that this new metric can serve as a valuable tool in predicting and evaluating offensive lineman. As the NFL continues to evolve, our methods of evaluation must evolve as well, ensuring that every aspect of the game, including offensive line play, is assessed with the precision and depth it deserves.

APPENDIX A
TOP AND BOTTOM OFFENSIVE TACKLES

See Fig. 7

Top and Bottom 5 Pressure Rate Difference (Tackles Min 50 Snaps)

Name	Predicted	Actual	Difference
Cameron Fleming	0.116	0.068	0.048
Rob Havenstein	0.144	0.1	0.045
Terence Steele	0.106	0.071	0.035
Marcus Cannon	0.081	0.048	0.032
Jawaan Taylor	0.08	0.053	0.027
Kolton Miller	0.058	0.122	-0.063
Joseph Noteboom	0.082	0.151	-0.068
Dennis Kelly	0.099	0.173	-0.074
George Fant	0.075	0.15	-0.075
Bernhard Raimann	0.153	0.236	-0.083

(a) Max differences from predicted to actual rate

Top and Bottom 5 Predicted Pressure Rate (Tackles Min 50 Snaps)

Name	Predicted	Actual	Difference
Cedric Ogbuehi	0.019	0.057	-0.038
Terron Armstead	0.026	0.039	-0.013
D.J. Humphries	0.03	0.065	-0.035
Brandon Shell	0.034	0.05	-0.017
Tristan Wirfs	0.036	0.016	0.02
Nicholas Petit-Frere	0.144	0.161	-0.017
Rob Havenstein	0.144	0.1	0.045
Larry Borom	0.147	0.211	-0.063
Bernhard Raimann	0.153	0.236	-0.083
Dennis Daley	0.184	0.173	0.01

(b) Highest and lowest predicted rates

Fig. 7: Tables showcasing variation and performance for offensive tackles

APPENDIX B
TOP AND BOTTOM OFFENSIVE GUARDS

See Fig. 8

Top and Bottom 5 Pressure Rate Difference (Guards Min 50 Snaps)

Name	Predicted	Actual	Difference
Isaac Seumalo	0.124	0.075	0.05
Elgton Jenkins	0.099	0.056	0.043
Connor McGovern	0.118	0.079	0.039
Jamaree Salyer	0.122	0.084	0.038
James Daniels	0.085	0.053	0.032
Mark Glowinski	0.061	0.095	-0.034
Kevin Dotson	0.036	0.071	-0.036
Bobby Evans	0.099	0.139	-0.04
Jeremiah Kolone	0.067	0.107	-0.04
Ben Bartch	0.067	0.111	-0.044

(a) Max differences from predicted to actual rate

Top and Bottom 5 Predicted Pressure Rate (Guards Min 50 Snaps)

Name	Predicted	Actual	Difference
Michael Onwenu	0.016	0.037	-0.021
Ben Powers	0.031	0.049	-0.019
Justin Pugh	0.032	0.042	-0.011
Nick Leverett	0.032	0.016	0.016
Joe Thuney	0.033	0.037	-0.005
Matt Farniok	0.125	0.125	0.0
David Edwards	0.126	0.137	-0.011
Dalton Risner	0.126	0.126	0.0
Royce Newman	0.129	0.122	0.007
Samuel Cosmi	0.177	0.15	0.027

(b) Highest and lowest predicted rates

Fig. 8: Tables showcasing variation and performance for offensive guards

APPENDIX C
TOP AND BOTTOM CENTERS

See Fig. 9

Top and Bottom 5 Pressure Rate Difference (Centers Min 50 Snaps)

Name	Predicted	Actual	Difference
Jake Brendel	0.069	0.031	0.038
Austin Blythe	0.074	0.043	0.031
Evan Brown	0.093	0.064	0.029
Will Clapp	0.07	0.042	0.028
Frank Ragnow	0.034	0.007	0.028
Drew Dalman	0.044	0.088	-0.044
Jon Feliciano	0.034	0.089	-0.055
Nick Martin	0.115	0.173	-0.058
Brian Allen	0.061	0.121	-0.061
James Ferentz	0.0	0.078	-0.078

(a) Max differences from predicted to actual rate

Top and Bottom 5 Predicted Pressure Rate (Centers Min 50 Snaps)

Name	Predicted	Actual	Difference
James Ferentz	0.0	0.078	-0.078
Rodney Hudson	0.0	0.01	-0.01
Jason Kelce	0.025	0.044	-0.019
Robert Hainsey	0.026	0.031	-0.004
Erik McCoy	0.027	0.054	-0.027
Nick Martin	0.115	0.173	-0.058
Hjalte Froholdt	0.119	0.136	-0.017
Billy Price	0.122	0.108	0.014
Tyler Larsen	0.137	0.123	0.014
Lucas Patrick	0.175	0.175	0.0

(b) Highest and lowest predicted rates

Fig. 9: Tables showcasing variation and performance for centers

REFERENCES

[1] N. Byanna and D. Klabjan, “Evaluating the Performance of Offensive Linemen in the NFL.” Accessed: Dec. 01, 2024. [Online]. Available: http://dynresmanagement.com/uploads/3/5/2/7/35274584/nfl_offensive_linemen.pdf

[2] M. Lopez, T. Bliss, A. Blake, P. Mooney, and A. Howard, “NFL Big Data Bowl 2025,” @kaggle, Oct. 10, 2024. https://www.kaggle.com/competitions/nfl-big-data-bowl-2025/overview/ (accessed Dec. 01, 2024).

[3] T. C. Nokeri, Data Science Solutions with Python. Berkeley, CA: Apress, 2022. doi: https://doi.org/10.1007/978-1-4842-7762-1.

[4] Schad, “Why do the Dolphins allow hundreds of quarterback pressures?,” TCA Regional News, Nov. 2021, Accessed: Nov. 30, 2024. [Online]. Available: https://www.proquest.com/docview/2595203614?pq-origsite=primo&accountid=6143&sourcetype=Wire%20Feeds