

Feeling the Pressure: A Unified Framework for Automating Pass Rushing Statistics in NFL Games

Sungmin Hong^{1*}, Laura Kulowski^{1*}, Dan Volk^{1*}, Henry Wang^{1*}, Keegan Abdoo²,
Conor McQuiston², Jonathan Jung², Mike Band², Diego Socolinsky¹
¹ AWS Generative AI Innovation Center, ² NFL Next Gen Stats (NGS)

1. Introduction

In professional football, the pass rush has become an increasingly important aspect of the game, with pass rushers being some of the top paid defensive players in the league. In spite of the importance of the pass rush, pass rushing statistics only include the final outcomes of a play, e.g., sack and pass-made. They do not capture the dynamics of the pass rush or fine-grained insights throughout a play on how much pressure a rusher generates during the rush [1-7]. This lack of in-play insight prevents a post-game analysis from taking into account an individual player's contribution on defense and fails to quantify pressures generated by each pass rusher. Even if a player does not record a sack, they can still generate pressure and impact the play.

There are a few challenges that need to be tackled to enable such a fine-grained pressure estimation throughout a play. First, blockers and rushers who create a pocket must be identified from among all the players on the field. This is crucial for ensuring that players are only evaluated when they are in a rusher or blocker role. Second, the match-up between blockers and rushers needs to be known to take into account how much resistance a rusher gets when they rush toward a quarterback. Finally, the pressure scores of each rusher and a team needs to be estimated throughout a play.

In this paper, we propose a unified framework that tackles these challenges by leveraging machine learning (ML) models and the National Football League (NFL) Next Gen Stats (NGS) data with positional and kinematic information from player sensors [1].

For the blocker/rusher identification and matchup estimation, we propose two approaches: sequential AutoGluon and graph neural network (GNN) [8-11]. For the sequential AutoGluon approach, we built two separate Amazon AutoGluon models that construct ensembles of multiple models [8, 11]. For the GNN approach, a novel GNN architecture parameterized by players as nodes and their matchups as edges is used to solve both tasks simultaneously [9, 10]. The pressure probability is modeled as the likelihood of the event of sack, hit, and hurry over the course of a play. We built a tree-based model to predict the pressure probability for individual rushers using custom engineered features [12]. Our proposed framework is used to build fine-grained pass rushing statistics indicating, how fast and for how long pressure is applied and how well rushers perform against certain blockers. It can also be used to better understand how quarterbacks perform when faced with different levels of pressure. This will enable teams to better understand player performance and allow broadcasters to tell a more compelling data driven story. We will present one example use case of the elaborated football analytic measures derived from our framework

* Equal Contribution

later in the paper.

Both qualitative and quantitative results of our framework showed the robustness and feasibility for downstream real-world use cases. Our framework achieved 0.99 overall accuracy for the blocker-rusher identification task and 0.97 average precision, 0.9 precision and 0.91 recall for the blocking matchup identification task on a test set of about 8,800 plays collected from 2022 season. Our model shows the balanced performance and feasibility for real-world use cases. The pressure probability module showed 0.96 accuracy for individual players' pressure when it was compared to the outcomes of all passing plays in the 2022 season.

There are three main contributions of our paper that we would like to highlight:

- We propose a unified framework that enables the estimation of individual pressure scores throughout a play.
- Our framework showed high accuracy and performance for rusher and blocker identification, rusher-blocker match-up and pressure score estimation that is feasible to be translated to real-world downstream use cases
- We present the real-world applications of our framework including the enriched analytics and an entertainment game.

The rest of the paper is organized as follows. In Section 2, we review previous works. Section 3 describes our framework for blocker and rusher identification and their match-ups and Section 4 explains the pressure score estimation. Finally, Section 5 presents real-world applications of our methodology followed by conclusion in Section 6.

2. Related Work

Machine learning (ML) has been widely used for analytics for a variety of sports, including football, soccer, hockey, etc. [13-27]. The advance of ML-based sports analytics was accelerated by the growing collection of sports data [1, 19, 20]. National Football League (NFL) launched the Big Data Bowl challenge to encourage the development of data-driven sports analytics. Open-sourcing player positional and kinematic data collected using sensors attached to players enabled studies such as our own [1].

2.1. Blocker-Rusher Identification and Match-up

Identifying blockers/rushers and their blocking assignment accurately has been a challenge for the league and broadcasters. In 2018, Pass Block Win Rate (PBWR) and Pass Rush Win Rate (PRWR) were shown on ESPN, where raw positions, orientations and pairwise distances were used to calculate a score that determines who is blocking whom [28]. Follow-up works were also done to predict the likelihood of a defensive player being a pass rusher pre-snap [29, 30]. As part of submissions to NFL Big Data Bowl 2023 [1], these works focus on applications such as live coaching and scouting, and only leverage features available before the snap of the ball. Backtracking was also used to recursively identify the optimal blocking assignment minimizing the number of unblocked defenders and distance from each lineman to their defense [31].

2. 2. Pressure Probability

Previous studies have found correlations between offensive linemen win rates and successful pass completion rates [23]. A follow-up study went further by actually tracking individual players and used survival analysis to model completion percentage as a function of lineman performance and time-in-pocket [22]. However, these works did not have the scale of data available today to perform a comprehensive analysis. More recently, the 2023 NFL Big Data Bowl [1] also produced several submissions aiming to quantify the pressure generated by individual pass rushers as well as the entire defensive unit. One paper, inspired by strain rate in materials sciences introduced the STRAIN metric, that uses the distance between a rusher and the passer along with the rate of change for those signals to quantify the pressure generated throughout the play [18]. Another approach presented in [15] introduces a new metric called the Instantaneous Disruption Probability Increase (IDPI) to evaluate the performance of defensive linemen in the NFL. The IDPI metric measures how much an individual lineman increases their team's probability of disrupting a pass play compared to the average player in a similar situation. It uses an LSTM to calculate disruption probabilities with and without each lineman [32]. This allows the metric to identify effective pass rushers even if they do not record many sacks or hurries themselves. Our approach builds off these methods by quantifying the pressure generated on an individual level, while introducing novel feature engineering approaches to explicitly incorporate the positions of blockers when predicting pressure.

3. Data Overview

We leverage the NFL NextGenStats (NGS) data for blocker-rusher identification, blocking matchup and pressure probability estimation [1]. Tracking data is measured for each of the 22 players on the field at a rate of 10Hz. The NGS data provides players coordinates (x, y), their speed and acceleration (s, a, sX, sY), their direction (dir), their relative position to the quarterback (relX, relY, relDist), and the quarterback's tracking data (qbX, qbY, qbSx, qbSy). The data is normalized so that all plays are moving in the same direction (offense on left, defense on right). Additionally, individual variables are normalized to reduce unnecessary variability across the play. The x coordinates (x, qbX) are centered on the line of scrimmage with negative numbers indicating the offensive side of the line. The y coordinates (y, qbY) are scaled between the lateral boundaries of the field with 0 indicating the near side and 1 the far side. Speed and acceleration features (s, a, sX, sY, qbSx, qbSy) are scaled using standard scaling based on a sample of speed metrics. The detailed description of the features is summarized in Table A.1 in the appendix.

In addition to the input features, we have several target features that are provided by the football analytics organization Pro Football Focus (PFF). PFF uses subject matter experts to manually label each play, indicating which players are blocking and which are rushing, who performed what block type on which player, and whether the rusher generated pressure. For each play, blockers and rushers are identified with a Boolean flag while matchups are denoted by a mapping between unique player IDs on the offensive and defensive teams. The target variables are summarized in Table A.2 in the appendix.

For the pressure probability estimation, we will use the PFF pressure indicators. Pressure is broken down into three types: hurry, hit, and sack. Sack indicates that the rusher reaches and tackles the passer behind the line. Hit indicates the rusher was able to hit the passer after the ball

was thrown. Hurry indicates that a player generated enough pressure to affect the passers actions by forcing them out of the pocket or passing quickly. It's important to note that sack and to a lesser extent hit, are fairly easy to determine when watching a play. Hurry on the other hand is a more subjective concept and different labelers may not assign a hurry the same way on the same play.

3. Blocker-Rusher Identification and Matching

3. 1. Blocker-Rusher Identification

The blocker-rusher identification problem entails identifying defensive blockers and offensive rushers within the first few seconds of play. Since certain positions are generally associated with rushing and blocking, we start by constructing a rule-based baseline model using player positions. For each position, we calculate the probability of assuming the roles pass rush, pass block, pass route, coverage, or other using approximately 8,800 plays from the 2018 season. We assign each position to the role with the highest probability using the following rules,

$$position = \begin{cases} pass_rush, & \text{if } \max(p_i) = p_{pass_rush} \\ pass_block, & \text{if } \max(p_i) = p_{pass_block} \\ other, & \text{elsewhere} \end{cases}$$

where p_i is the probability of a pass rush, pass block, pass route, coverage, or other. Several example positions, probabilities, and assigned roles are shown in the table below. We evaluate the rule-based model on test data from 2022 season. As shown in Fig. 1, the model achieves an F1-score of 94% and accuracies above 95% for 12 of the 19 identified positions. The model has the lowest accuracy for the outside linebacker (OLB, 62%), linebacker (LB, 70%), fullback (FB, 75%), and running back (RB, 85%).

Model Name	Static Features	Dynamic Features	F1-Score	Percent Change Relative to Baseline (%)
AG_InitialPositions	[x0_norm, y0_norm]		0.94	+1.0
AG_Static	[x0_norm, y0_norm, position, offenseTeamPlayer]		0.95	+2.0
AG_Dynamic_1sec	[position, offenseTeamPlayer]	[x_norm, y_norm, sX, sY, aX, aY]	0.98	+4.6
AG_Dynamic_2sec	[position, offenseTeamPlayer]	[x_norm, y_norm, sX, sY, aX, aY]	0.99	+5.5
AG_Dynamic_3sec	[position, offenseTeamPlayer]	[x_norm, y_norm, sX, sY, aX, aY]	0.99	+5.6

Table 1. F1-scores of different AutoGluon models for blocker-rusher identification with static and dynamic features. Static features include the initial x- and y-positions of the player relative to the ball at the start of play (x0_norm, y0_norm), the player position (position), and offense/defense roles (offenseTeamPlayer). Dynamic features include the position (x_norm, y_norm), velocity (sX, sY) and acceleration (aX, aY) components during the first few seconds of play. The percent change of the F1-score relative to the baseline model is indicated.

Performance of AutoGluon Models for Blocker/Rusher Identification

To improve the results of the baseline model, we used Amazon AutoGluon to train a collection of models for the multiclass rusher/blocker classification problem [8, 11]. As a starting point, we used static features like the positions of the players (e.g., OLB, LB, etc.), whether they are on offense or defense, and their locations relative to the ball at the start of play. Then, we added dynamic features such as the players' locations, velocities, and accelerations during the first one to three seconds of play. The performance of each AutoGluon model relative to the baseline model is shown in Table 1. All of the AutoGluon models have higher F1-scores compared to the baseline model. The AutoGluon models that include location, velocity, and acceleration trajectories during the first two to three seconds of play show the highest improvement, with F1-scores about 5.5 percent above the baseline, at values of about 0.99.

We then examine the performance of the AutoGluon models for different player positions. We compute the percent change in accuracy relative to the baseline model for each position as

$$percent_change_i = \frac{accuracy_i - baseline_accuracy_i}{baseline_accuracy_i}$$

where $accuracy_i$ is the model accuracy and $baseline_accuracy_i$ is the baseline model accuracy for position i . Fig. 2 shows the percent change in the accuracy scores for the AutoGluon models for each position. We observe that all of the AutoGluon models improve the accuracy for the OLB and LB by over 20%. The AutoGluon models that include location, velocity, and acceleration trajectories lasting two and three seconds provide the largest lift in accuracy over most positions. These models increase the accuracy by about 56% for the OLB, 40% for the LB, 20% for the FB, 10% for the ILB, TE, and MLB, and a few percentage points for the remaining positions.

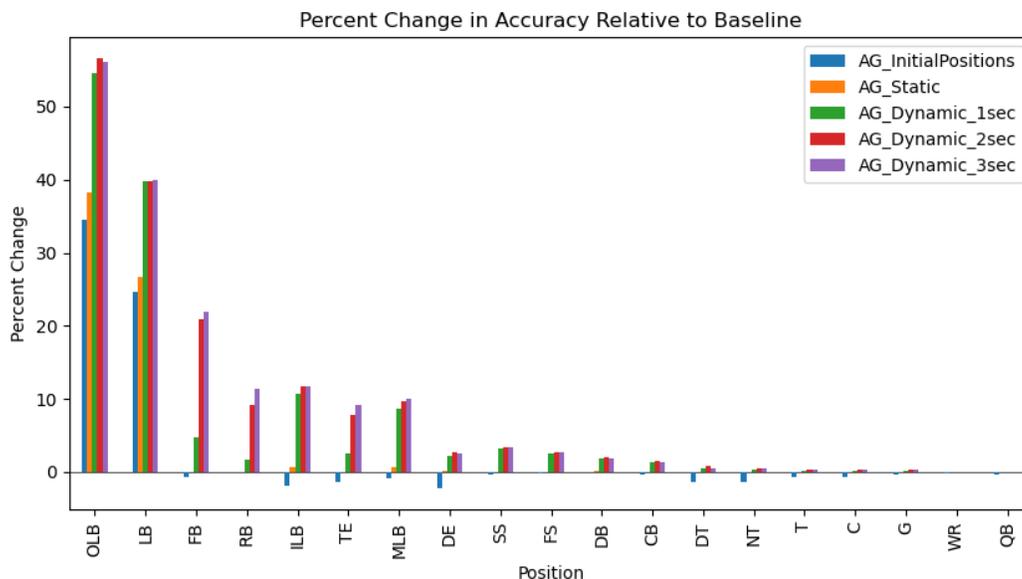


Fig. 2. Percent change in accuracy for the AutoGluon models relative to the rule-based baseline model for different player positions.

3. 2. Blocking Matchup

Matchups were historically determined through a manual process, where human experts review the play footage and assign defensive players to offensive players. Sometimes, an offensive player can have multiple defensive players assigned. The goal of the “Blocking Matchup” model is to perform the assignment automatically after a play ends. There is rich information to leverage for building a model, including contextual information before the play and spatial-temporal information during the play.

Similar to blocker-rusher identification, we adopted an iterative mechanism for feature engineering and model selection, starting from static features and simpler models to dynamic features and more sophisticated model architecture. During each iteration, we incorporated additional domain knowledge and improvements addressing shortcomings of previous versions.

As a baseline, we calculated the distances at snap of the play for all possible pairs of offensive and defensive players (11 players from each side results in 121 pairs). We also one-hot encoded positional information, such as defensive tackles and defensive ends. Intuitively, we knew that defensive players had to get close to offensive players if they were matchups, so calculating the minimum distances during the play would likely be an important indicator. Fig. A.1 in Appendix shows that the distributions for these distance features are indeed different between the matching pairs and non-matching pairs.

We build logistic regressors and random forests for the classification task. To ensure these models can generalize, we train the model on around 8,800 individual plays from the 2018 season and test the model on similar number of plays from the 2022 season. In further iterations, additional features are incorporated. For example, to better capture the relative distances between players, we calculate the percentage of times a given player pair was closest to each other. We also sample spatial temporal features such as the normalized X and Y coordinates of the player, and their speed differences 20 times during the play. These features capture more granular information and help

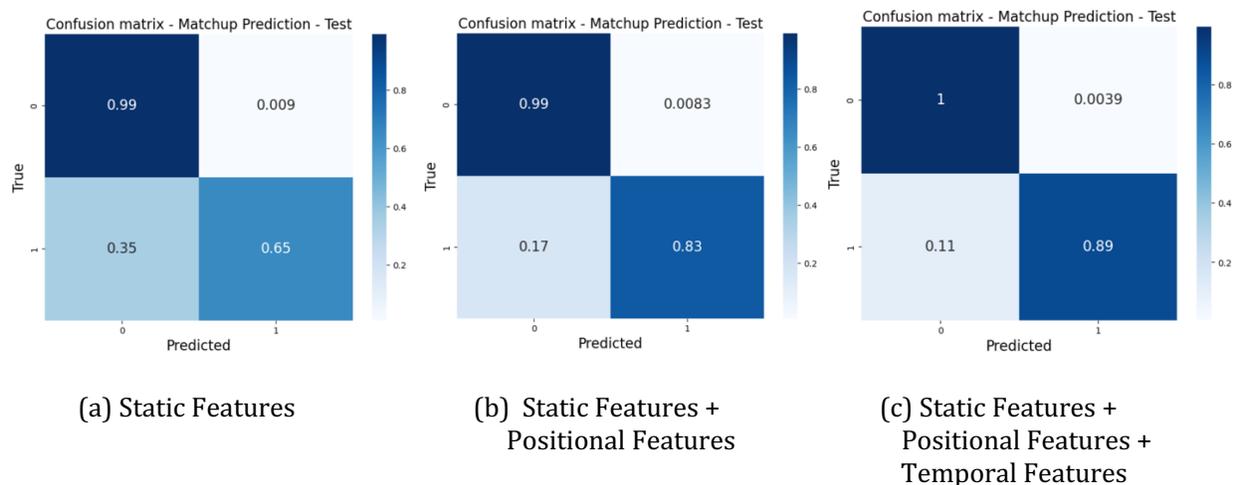


Fig. 3. Confusion matrices of models on the matchup prediction task, as we incorporate different features. Static features include scalar values such as distance at snap, positional features include positional encodings, and temporal features include samples of features that capture the dynamics of the play.

improve cases where the model makes mistakes. We further boost the classification performance by leveraging the AutoGluon package [8, 11]. Fig. 3 shows the model performance progression with increasing number of features, from using only static features to using all static, positional, and temporal features.

3. 3. Unified Approach

While developing the Blocker-Rusher Identification model and Blocking Matchup model, we saw the opportunity to solve both tasks with a single model. Given the shared features for both models and the spatial-temporal nature of the problem, we considered tackling these tasks using graph neural networks. Players represent nodes in the graph, and their matchup relationships represent the edges connecting the nodes. Given all of the players on the field for a given play, we train a unified graph neural network model to accomplish two tasks simultaneously in a single forward pass: (1) classify each player as either a blocker/rusher or a non-blocker/rusher, and (2) classify each possible pairing of offensive and defensive players on the field as a true blocking matchup or not [9].

Our Graph Neural Network employs a combined loss function comprising Binary Cross-Entropy (BCE) loss for node with a weighting factor for link prediction [33]. The overall loss function for our GNN model is a weighted sum of these two components:

$$BCE\ Loss = -\frac{1}{N} \sum_{i=1}^N w_i [y_i \cdot \log(\sigma(x_i)) + (1 - y_i) \cdot \log(1 - \sigma(x_i))]$$

$$Total\ Loss = BCE\ Loss\ on\ Nodes + \lambda \cdot (BCE\ Loss\ on\ Links)$$

where λ is a weighting factor. We put a heavier weight on link prediction loss, as links are sparse among all possible pairs. We use PyTorch Geometric to encode each play in our dataset as a graph, where the nodes represent the 22 players on the field and edges between those nodes are created when pairs of players meet certain conditions (e.g., connect each pair of players who reach a minimum distance during the play of less than three yards) [34]. We leverage the GATv2 convolutional operation from PyTorch Geometric along with two branches of neural decoding to train the model to estimate the probabilities of the classes mentioned above for each of the 22 players and each of the 121 possible matchups that constitute the play [10]. We used the Adam optimizer and trained the models for 10,000 epochs [35].

Fig. 4 shows the problem set up with GNN, and the confusion matrices of prediction results of GNN for rusher and blocker identification and match up prediction. The quantitative performance of GNN matched that from AutoGluon model. While GNN's performance on matchup prediction is slightly worse and tended to over predict (0.954 Average Precision for GNN compared to 0.969 for AutoGluon). The GNN model over predicts the existence of links between nodes due to the sparsity of actual matchups between all pairs of players. Future experimentations are needed to account for the imbalance of matchup and non-matchup pairs, such as further tweaking on the loss functions or modifying model architectures that validates only links between predicted rusher and blocker nodes.

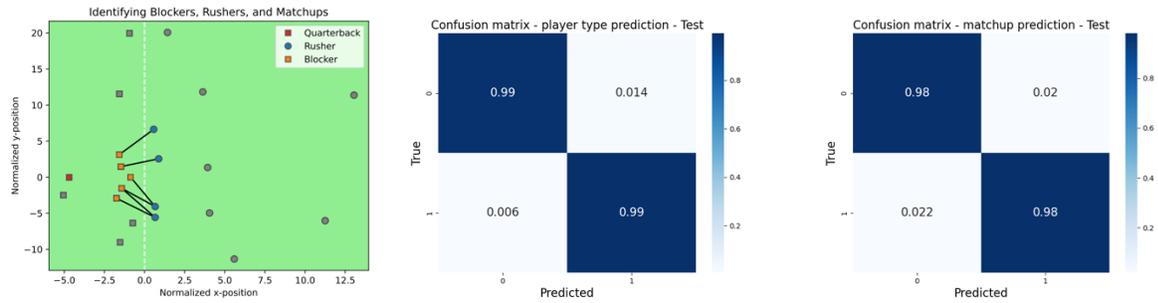


Fig. 4. (left) Problem set up with GNN. Players on the field represent nodes and matching pairs have “links” between them. (middle) Performance of GNN on classifying the players. (right) Performance of GNN on predicting the matchups.

4. Pressure Probability

The goal of the pressure probability estimation is to quantify the amount of pressure generated by an individual rusher, or all pass rushers, at any point in time during the play. We use a classification approach where we predict if a player will generate pressure during the play [12]. It is important for the final model to be well calibrated so that the model prediction can be directly translated into a reliable probability indicating the likelihood of pressure over time [36, 37]. In this way, we can move from the paradigm of pressure flags for each player on each play to a continuous pressure score estimation. When combined with the results of the rusher-blocker identification and the matchup models, this can be used to produce a fine-grained understanding of both rusher and blocker contributions during the pass rush.

4.1 Data and Feature Engineering

Blocker Interference

There are several ways to approach modeling this problem which we discuss below, but an important consideration is whether the defensive players are analyzed in isolation or simultaneously with the positional data for all other players. Analyzing the entire state of play with all 22 players is considerably more complex and forces the model to learn the significance of the positions of offensive and defensive players relative to one another. If focusing only on a single player’s tracking data, such as their speed, position, and distance to the quarterback, the model lacks visibility into the presence of potential blockers. This is an important factor, because there is a big difference between a player being two yards away from the passer, but facing a double team and a player being the same distance with no blockers in his way. In order to model defensive players in isolation, we need to engineer features that provide insight into the presence of offensive players.

To accomplish this, we developed a metric that calculates the presence of blockers between the rusher and the quarterback, which we call blocker-interference. For each rusher we calculate the angle between the rusher and the passer, we then calculate the angle between the rusher and all remaining offensive players. We can then calculate whether a blocker is between the passer using cosine similarities and orthogonal distance. If we calculate the cosine similarity between the rusher-quarterback vector and the rusher-blocker vector we can remove any cases where the value

is less than 0 and therefore the offensive player is behind the defender. Similarly, we can remove any offensive players behind the passer. Finally, we can calculate the orthogonal distance and add a weighting approach to indicate how close the defender is to blocking the path of the rusher to the quarterback. If a player is within 1 yard of the direct path, they are assigned a value of 1 otherwise their influence decreases exponentially. This is visualized in Fig. A.2 in the appendix.

Action Filtering

The average play length in the NFL data set was roughly ten seconds. However, each play also included three seconds prior to the snap (pre-snap) and sometimes activity after the pass rush (pass catch and run, fumble, etc.). Since the pressure labels are evaluated by PFF based on the activity that occurred prior to the ball being thrown, we needed to filter any activity that happens after that action. This prevents the model from being trained on non-pass rush relevant data and ensures that probabilities are assigned only during the active pass rush. The NGS data provides time-stamped events including categorical actions such as ball snap, pass forward, run, etc. The beginning of the play is easy to identify as there is always one ball snap event and can remove any data before it. For the end of the play, the logic is more complicated. If there is a pass or a quarterback sack, then that is the defined end of the play. If those events are not present, we use other events such as a run or a handoff. In this way, we can allow for trick plays where the quarterback may hand the ball off to a receiver prior to a forward pass. If the quarterback drops back to pass, but instead decides to run the ball, the start of the run event will be used as the end of the play. We discuss this further in the model results section.

Due to the presence of trick plays, we also check to see if the quarterback and the primary passer are the same player. NGS provides a single passer's ID for each play. If this is different than the quarterback, the new passer ID is set and all passer-specific metrics (qbX, qbY, qbSx, qbSy, relX, relY, relDist) are recalculated to reflect the primary passer on the play.

Target Metric

We use two separate definitions of the target variable of pressure. First, we use the pressure labels provided by PFF. As previously mentioned, these include Boolean flags for hurry, hit and sack. Since the PFF labels can be inconsistent, especially in regards to hurries, we add a heuristically defined target as well. For this heuristic, we define pressure as any play where a rusher gets within 1 yard of the passer during the pass rush or within 1.5 yards of the passer at the time of the pass. The heuristic target is combined with the PFF labels to create a single Boolean flag for pressure. Each of these pressure labels are defined for each defensive player over the entirety of the play rather than at a specific timestamp.

After this preprocessing, we are left with a snapshot of the rusher location, passer location, relative parameters, and blocker interference. Our target metric is Boolean and fixed at the play level rather than changing temporally. As a result, players are assigned a pressure flag even at the snap of the ball which allows the model to learn a rough estimate for the possibility of pressure given the pre-snap location.

4. 2 Estimation Framework

As mentioned above, there are two primary ways to approach this problem. The first is to try to model the rusher's pressure in isolation and the second is to model pressure while including the positional feature of other players. The second approach adds significant complexity as the model must learn how to determine the importance of positional data of 22 players simultaneously instead of just the players of interest. Additionally, the order in which player features are provided to the model can impact the output. As a result, we choose to model the relationship between the rusher and the passer and explicitly calculate the presence of relevant players through the previously mentioned blocker-interference metric.

We also choose to model each timestamp rather than using temporally aware models such as transformers or LSTMs. We are able to do this because the data already includes speed, acceleration, and directional features, which already incorporate temporal information. This allows us to reduce the complexity of the model and reduce latency at inference. Additionally, this approach yields instantaneous predictions of pressure for a provided snapshot without any influence from prior information. It essentially asks the question "given this rusher's current configuration (speed, direction, distance to passer, nearby blockers, etc.), how likely are they to apply pressure at some point during the play?"

To train the model we use a randomly sampled subset of 6,000 pass plays from the 2018 season, holding out later seasons for evaluation. We train on each timestamp of the pass rush for each player within those 6,000 plays.

We explored several approaches for the estimation framework, including AutoGluon, a feed forward neural network and a random forest. The neural network tended not to make strong probability estimates and instead biased towards no pressure since that is the dominant class. The AutoGluon models rely heavily on boosted trees which gave inconsistent and noisy probability estimates. Ultimately, the random forest model handled the class imbalance well and provided cleanly calibrated and consistent probability estimates. The final random forest model uses 400 estimators. Since probability estimates in a random-forest are done through voting of individual estimators we needed to include a large number of estimators so that the final probability estimates could be granular (in this case 0.25%). The model was trained with a 20% subset of the 2018 season data which was enough for the model to converge on a consistent output. Seasons 2019-2022 were used for evaluation. Final probability estimates were smoothed using a 3-step centered moving average to make the probabilities more consistent over time and limit the presence of spikes.

Inference

During inference, data preprocessing and feature engineering are performed at runtime, including the blocker-interference metric, and can be configured with simple flags. Once feature engineering is complete, the dataset generator produces a single Pandas data frame. The player data frame dimensions are $11t \times n$, where t is the number of timesteps and n is the number of features. Inference is very fast with the random forest model. A full season's worth of data can be predicted in less than a minute. Prediction outputs will be a probability on a scale of 0 to 1. The model is designed to provide estimates for all players and positions. As will be discussed later, inconsistencies between the NGS events and the labeler interpretations of the play activity were the

primary source of outlier errors for the model. The fidelity of this model is reliant on the accuracy of upstream NGS event calculations. This is discussed further in the evaluation section.

4. 4. Pressure Probability Results

We use three main approaches for evaluating the pressure probability metric: model calibration, classification performance, and outlier analysis. Calibration evaluates how reliable the probability estimates are compared to the actual performance. For instance, if a model predicts 20% probability of pressure, then pressure should be present 20% of the time. This is the primary metric of evaluation, as the reliability of the predicted probability is necessary for calculating downstream metrics. Classification performance is evaluated using standard classification metrics with a focus on model precision. These classification results are used to validate the model's performance, but are secondary to the calibration results. Finally, we use corner cases to identify potential model issues, and human review to ensure that the predictions stand up to expert scrutiny.

For each of these metrics, we choose to evaluate based on the maximum probability as this is most similar to how the PFF labels were determined. A human labeler will assign a pressure/no-pressure label based on the point in the play where the maximum pressure on the passer is being applied. Therefore, we choose that maximum pressure as the point in the play most relevant to the assigned target value.

The evaluation detailed in this section is for a full season of pass plays from the 2022 NFL season. Similar evaluation was performed across seasons 2019-2021 with negligible differences between the seasons. A subset of pass plays from the 2018 season was used for training.

Calibration

We assess model calibration by binning the predictions at different probability intervals and then evaluating the frequency of a positive label within that bin. For example, if the model is predicting 40-45% we want the actual frequency of the label for those predictions to be within that range. For this model, when it predicts 40% in reality approximately 39% of the time the label is actually pressure, so the model is fairly well calibrated at this point. On the other hand, when we predict 80%, the actual results are closer to 83%. The model is a little under-confident as we get to higher probabilities. The expected calibration error (ECE) indicates that we can expect the predicted probability to be within 0.6% of the truth and the maximum calibration error (MCE) says that the maximum error expected is about 7.6% [36, 37]. The lower these metrics are the better calibrated the model. A plot of the calibration performance can be seen in Fig. 5.

Classification

The classification results are based on the maximum pressure achieved over the course of the play. In this way we can evaluate the classification at the play level, similarly to our target metric. This helps avoid confusion as the model should always predict low pressure probabilities at the beginning of the play. We also separate evaluation by all-defenders vs only-rushers, as the probability for rushers is considerably higher than for players who are in coverage. Simply by choosing a dominant class strategy (always predicting no-pressure) this would result in 95.2%

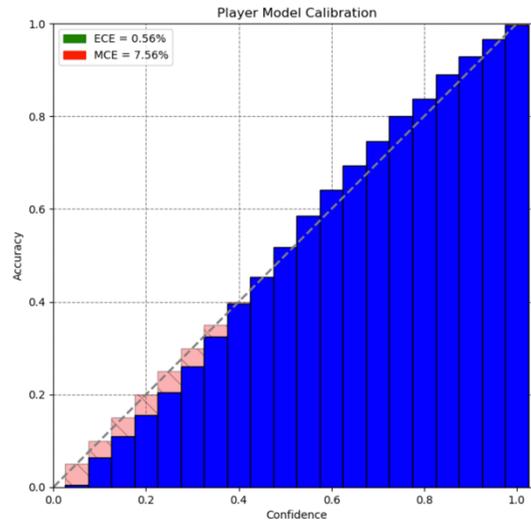


Fig. 5. The pressure probability model calibration shows a well calibrated mode with low ECE (0.56%) and MCE (7.56%). The model does show some under confidence at higher probabilities and overconfidence at lower probabilities, but the model outputs are reliable enough to provide a good estimate of applied pressure.

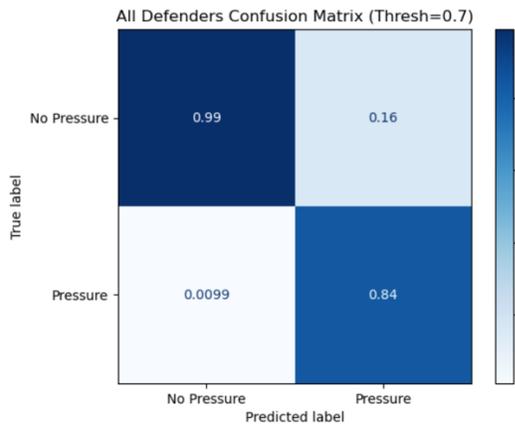


Fig. 6. All-defender confusion matrix.

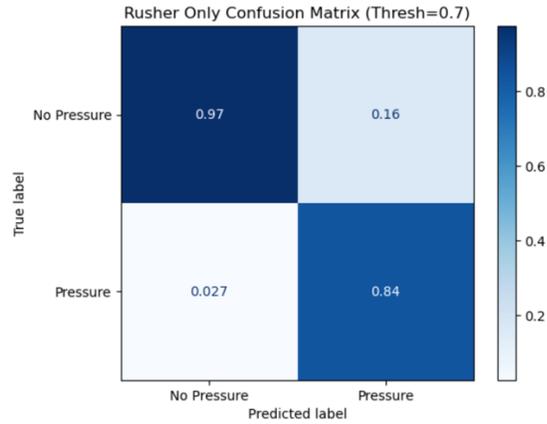


Fig. 7. Rusher-only confusion matrix

accuracy for all defenders and 87.8% accuracy for rushers. Our model achieves 98.3% accuracy when looking at all defenders and 95.8% accuracy when looking only at rushers, cutting the remaining uncertainty by more than half. The confusion matrices in Figs. 6 and 7 show the precision for each class with 99% precision for no-pressure and 84% precision for pressure. This consistent with our observation that PFF labels were subjective and in particular Hurry was sometimes arbitrarily assigned.

Examples and Outlier Analysis

Figs. 8-10 shows three examples of different type of plays predicted by the model. The first, shown in Fig. 8, is a random play where three players get pressure on the quarterback throughout the play. The video of the tracking data is on the left and the estimated probabilities are on the right. The video shows that three different players create pressure at different times throughout the play, the dynamics are reflected in the chart on the right with two players getting pressure early and a third getting pressure late. The video is clipped for just the duration of the pass rush.

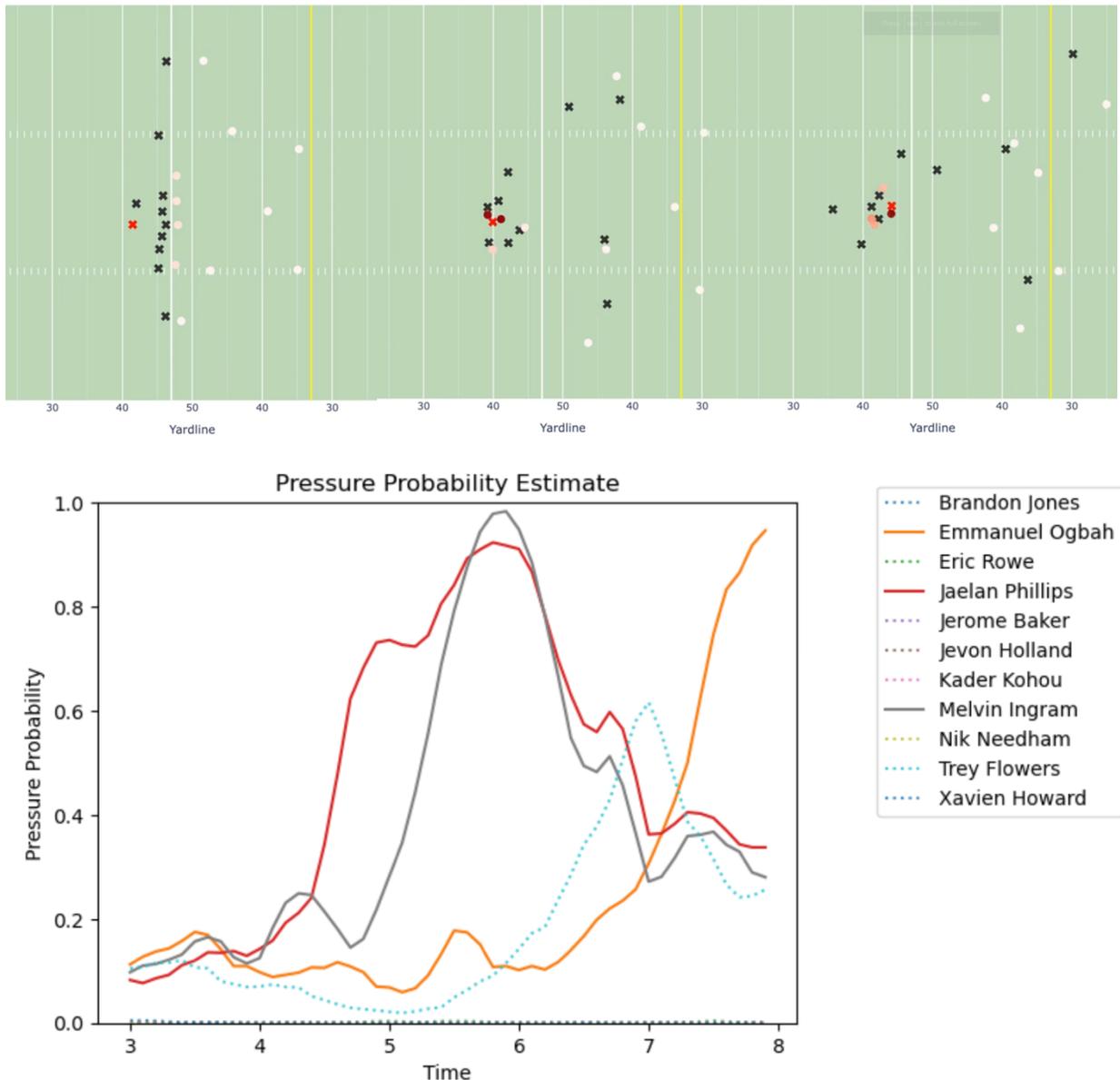


Fig. 8. Randomly sampled play where pressure is generated by 3 separate players. Top images show snapshots of the play, while the bottom plots the predicted pressure over time (ball snap at 3s). At snap (top left) all players have low predicted pressure. (Top center) 2.7s after snap, two players converge on the passer generating high pressure. 4.7s after snap, the passer evaded previous pressure and then pressure is generated by a third player.

Fig. 9 shows a specific type of extreme model error due to inconsistencies between the PFF labels and the events data. The NGS events shows that the quarterback decided to run at time step 59 (2.9 seconds into the play), which ends the pass rush event according to the logic provided by the NFL. As he runs, he is tackled for a loss and the PFF label gives the tackler a sack. Due to the pass rush filtering the run and tackle are removed and we only get predicted probabilities for the first 2.9s of the play. This behavior is expected and acceptable as the NGS event definitions will be treated as the truth.

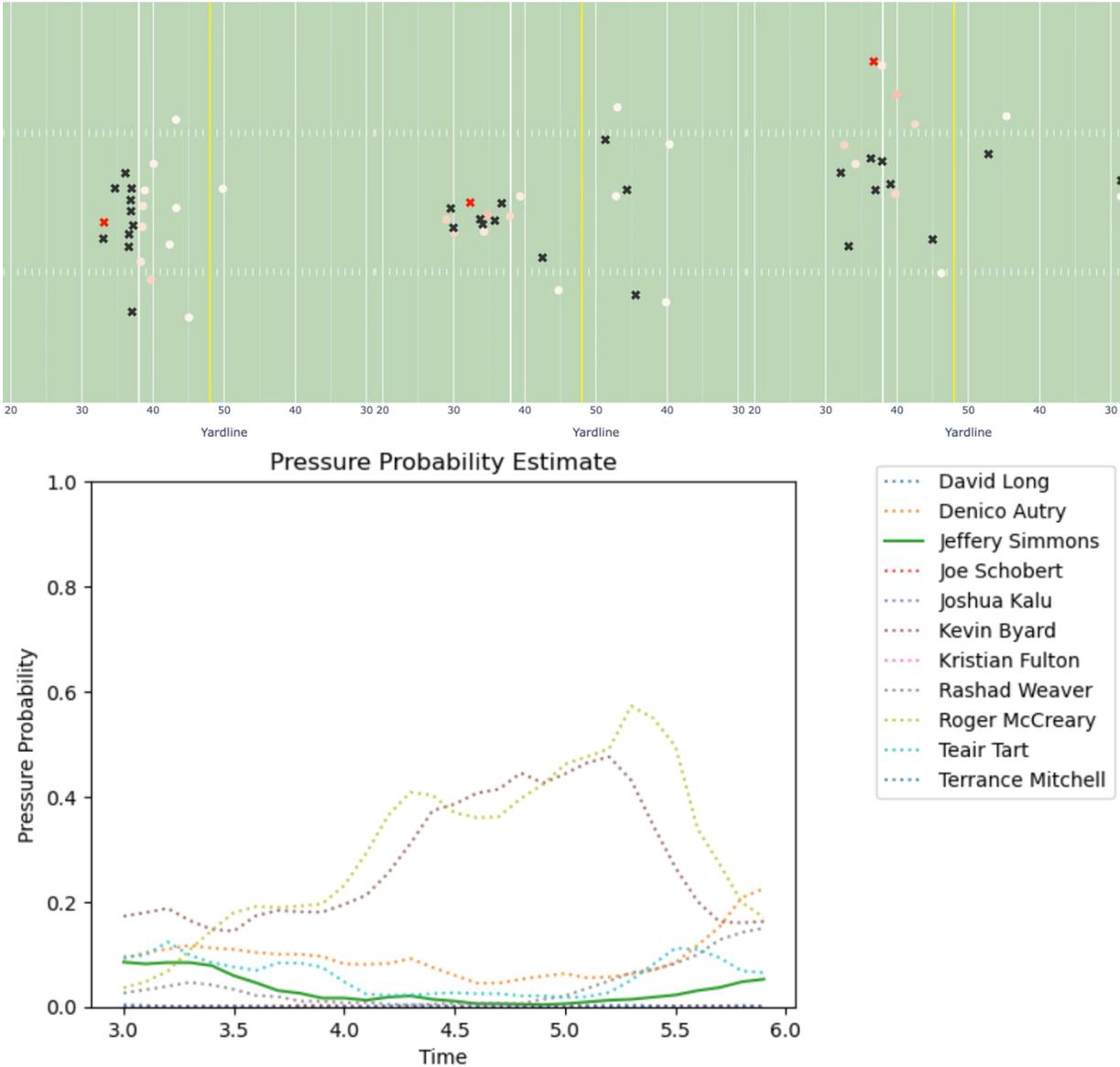


Fig. 9. Illustration of a play with label inconsistencies. Top images show snapshots of the play, while the bottom plots the predicted pressure over time (ball snap at 3s). At snap (top left) all players have low predicted pressure. (Top center) 3s after snap, pocket collapses and quarterback begins to scramble toward the sideline. This is where NGS classifies the pass rush as having ended due to the scramble. 6s after snap, the quarterback is under pressure and tackled. PFF labels this as a sack.

The final play example, shown in Fig. 10, is a circumstance where one player applies pressure and rushes the quarterback but is not assigned a flag by PFF. This could arguably be called a hurry since the quarterback's play is impacted, but the labelers do not assign the label. They do give credit to the player who arrives later, but also generates high pressure according to the model. Many of the instances of high predicted pressure with no positive label were in instances that would be classified as hurries instead of sacks or hits. This makes sense, as the judgement for what is or is not a hurry is more subjective than hits and sacks.

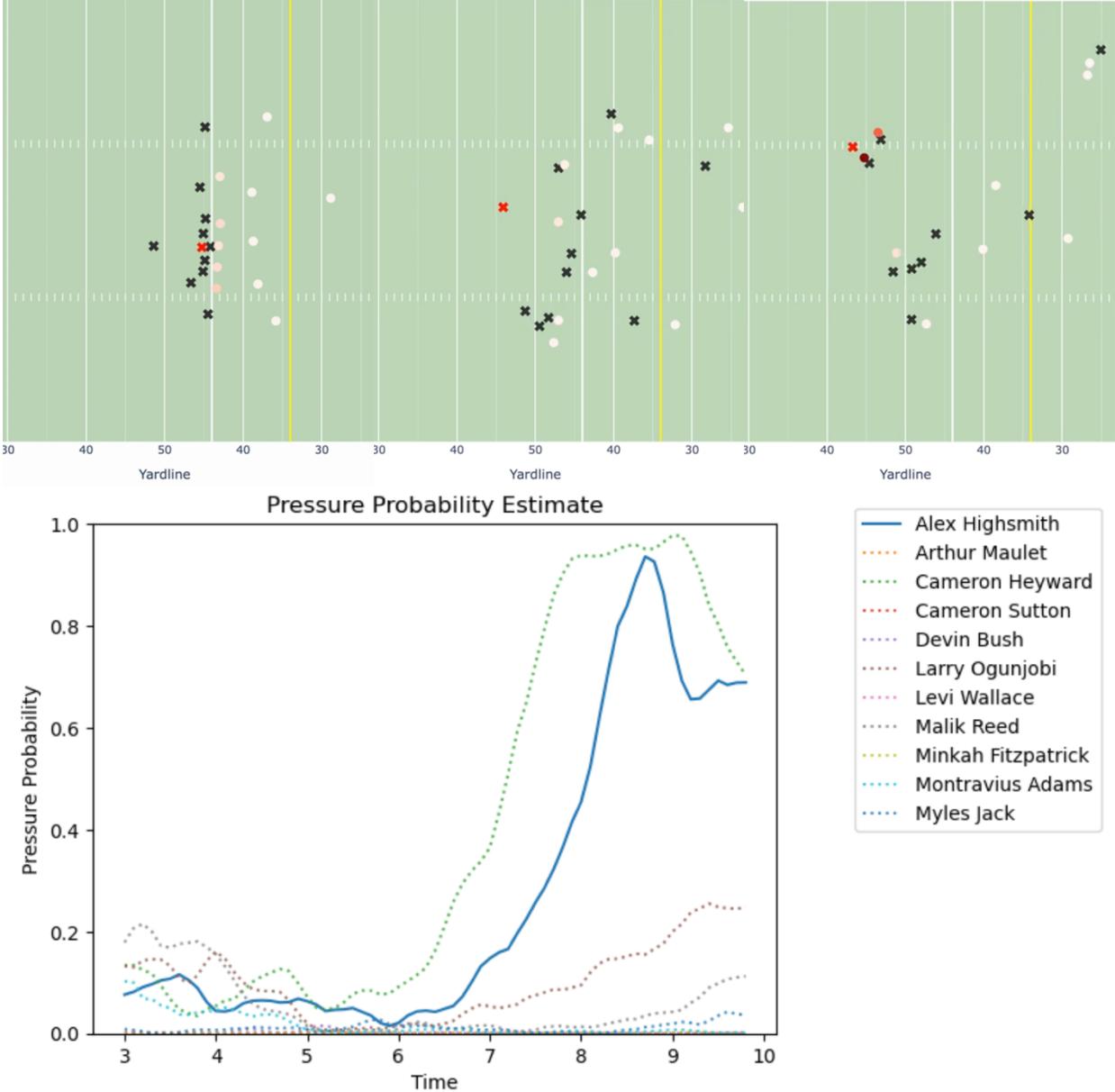


Fig. 10. Illustration of a play with unlabeled pressure. Top images show snapshots of the play, while the bottom plots the predicted pressure over time (ball snap at 3s). At snap (top left) all players have low predicted pressure. (Top center) 3s after snap, no player has significant pressure and quarterback begins to roll out to the left. 4.9s after snap, Cameron Heyward is generating pressure due to his proximity to the passer and lack of blockers, but is not given credit by the PFF labeling team.

After human review of these outliers were determined to be primarily due to inconsistencies in labeling or in the definition of what constitutes the end of a pass rush event. These outliers were not deemed significant, but will be monitored over time to determine if future adjustments to the model need to be made.

5. Practical Applications to Pass Rush Analysis

Pressure Metrics

Our models enable a fine-grained understanding of what takes place throughout the pass rush, including who was involved, who engaged who, and how pressure developed throughout the play. The continuous nature of the pressure probability predictions allows subject matter experts at the NFL to break down the model output 13 distinct metrics including the time of the pressure, the peak pressure, quick pressures, pressure rate over expectation, etc. These metrics can easily be analyzed for a single play or aggregated over an entire season for comparison between players [38]. These are visualized in Fig. 11 and further broken out in Table A.3 in the appendix.

The Anatomy of a Pressure

A breakdown of metrics derived from new Next Gen Stats Pressure Probability Model

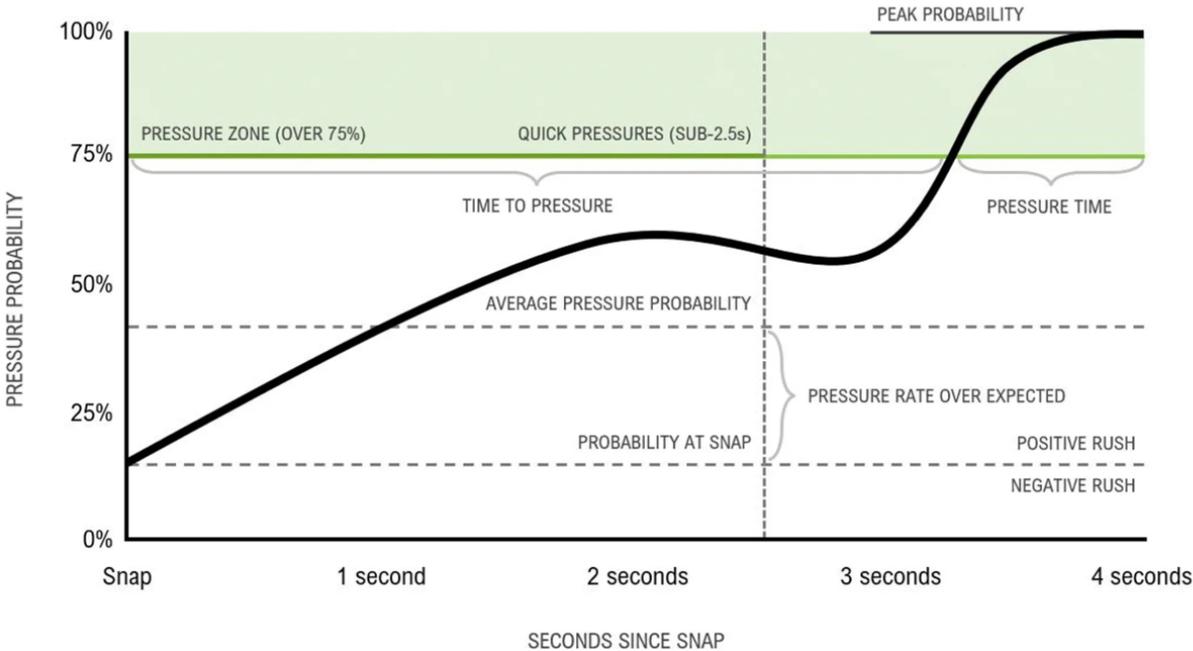


Fig. 11. The football analytic metrics derived from the pressure probability estimated by the proposed framework [38]

Real-World Applications

To see how this type of analysis can be applied within the game, we included the example of a specific play in Fig. 12 [38]. Consider Micah Parsons' sack in the Week 2 matchup between the Jets and Cowboys. Late in the game, with the Cowboys leading 30-10, Parsons sacked Zach Wilson for a 5-yard loss.

For this play, Micah Parsons is able to quickly get off the line at the snap. Given his advantageous alignment from a wide-nine technique off the right edge against Duane Brown, the model predicts pressure at snap of 35.4 percent. Parsons beat his block to set up an unimpeded path to Wilson within two seconds of the snap qualifying this a quick pressure. Parsons maintains an average pressure probability of 74.8 percent over the 4.8-second dropback (39.4 percentage points over expectation). Parsons' peak pressure probability converged to 100 percent within three seconds, with a full 2.9 seconds spent pressuring Wilson over the course of the play.

We can also use these metrics in aggregate to compare player performances. Through the first two games of the 2023 season, Parsons is tied with the Titans' Arden Key for most pressures league-wide (15 each), and ranks first in quick pressures (8), pressure time (17.2 seconds), positive rushes (42), net positive rushes (+29) and average pressure probability (28.3%, among 292 pass rushers with at least 10 pass-rush snaps).

Cowboys Pass Rushers by Pressure Probability

Micah Parsons sacks Zach Wilson on 1st & 10 with 6:53 left in Q4 (Week 2, 2023)

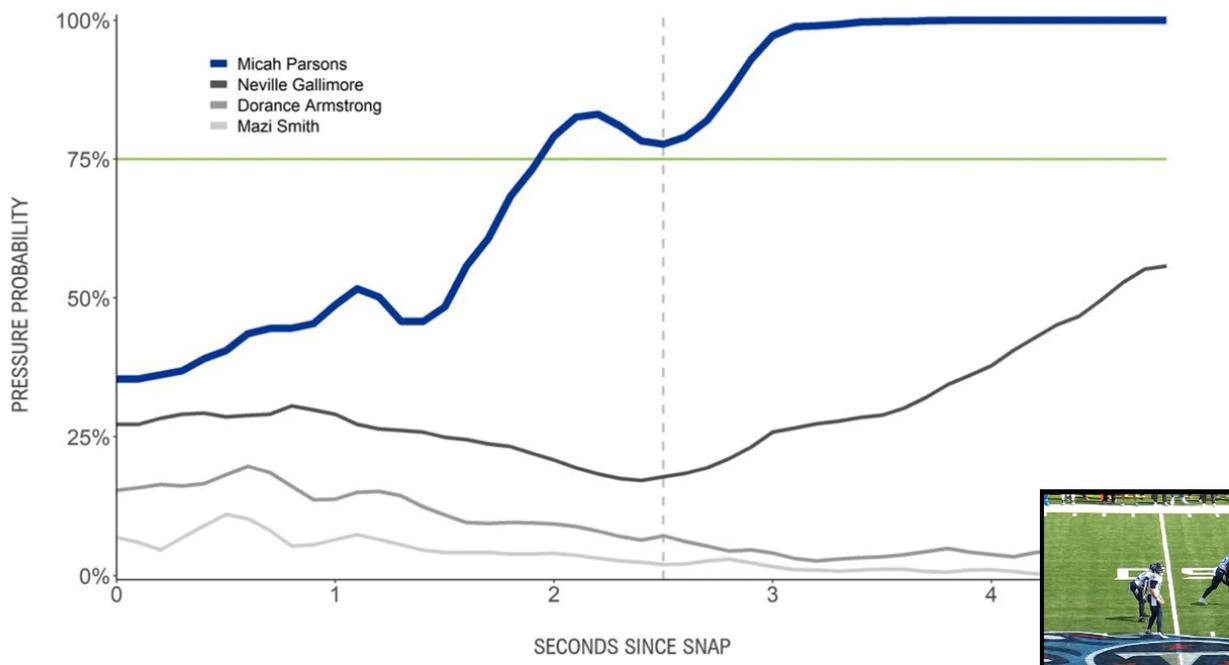


Fig. 12. The real-world example of the application of the analytic metrics derived from the pressure probability. The example play happened in the Week 2 matchup between the Jets and Cowboys in 2023 [38]

Offensive Line Analysis

With the ability to identify passers and rushers and quantifying the pressure in a temporal manner, we can use the matchup model to analyze the prevention of pressure. By combining these models we can identify matchups, attribute sacks to individual blockers, discern double teams and much more. Key metrics are listed in Table A.4 in the appendix. This shows the added benefit of our model as we are not only able to provide more descriptive statistics for pass rushers, but we can also begin to quantify offensive line performance, which until now has had very limited visibility from a statistical perspective.

6. Conclusion

In this paper, we presented a unified framework leveraging machine learning models and NFL's NextGenStats data that enabled the pressure score estimation throughout a play by addressing three major challenges, 1) blocker and rusher identification, 2) block-rusher match-up estimation, and 3) in-play pressure probability estimation at individual player and team levels.

Pressure probability estimation can be expanded upon by attempting to model an average player's performance in a specific scenario. Explicitly modeling an average player's expectation could allow for a standardization of pass rushing metrics. It will be interesting future work to investigate further to estimate average player performance reliably.

Future research may focus on increasing the granularity of insights on specific blocking and rushing techniques. This could require a multi-modal approach, combining player tracking data with video to classify linemen techniques to better understand how players matchup against one another and how a pass rusher's strategies develop over the course of a game or season.

We believe that our framework can be leveraged in the other sports, e.g., hockey, soccer, etc., to provide more detailed insight that goes beyond the mere outcome of a play to give proper credits to players and a team on how effective their defense is against offense throughout a course of a play. Our work can be a stepping stone for the variety of interesting applications and analytics in sports.

References

- [1] Addison Howard, Ally Blake, Andrew Patton, Michael Lopez, Thompson Bliss, Will Cukierski. (2022). NFL Big Data Bowl 2023. Kaggle. <https://kaggle.com/competitions/nfl-big-data-bowl-2023>
- [2] Inayatali, Hassaan, et al. "Between the Lines: How Do We Measure Pressure?", Kaggle (2023), <https://www.kaggle.com/code/hassaaninayatali/between-the-lines-how-do-we-measure-pressure/notebook>.
- [3] Pardun, Tyler and Pardun, Michael, "Introducing PPI: The Pocket Preservation Index", Kaggle (2023), <https://www.kaggle.com/code/tylerpardun/introducing-ppi-the-pocket-preservation-index>.
- [4] Bachelder, Nick, "IDPI: A Situational Metric for Pass Rushers", Kaggle (2023),

<https://www.kaggle.com/code/nickb1125/idpi-a-situational-metric-for-pass-rushers>.

[5] Karpick, Vincent, "Completions Added Through Suppression of Pressure", Kaggle (2023), <https://www.kaggle.com/code/vincentkarpick/completions-added-through-suppression-of-pressure>.

[6] Yurko, Ronald, et al. "Going deep: models for continuous-time within-play valuation of game outcomes in American football with tracking data." *Journal of Quantitative Analysis in Sports* 16.2 (2020): 163-182.

[7] Nguyen, Quang, et al. "Here Comes the STRAIN: Analyzing Defensive Pass Rush in American Football with Player Tracking Data." *arXiv preprint arXiv:2305.10262* (2023).

[8] Klein, Aaron, et al. "Model-based asynchronous hyperparameter and neural architecture search." *arXiv preprint arXiv:2003.10865* (2020).

[9] Scarselli, Franco, et al. "The graph neural network model." *IEEE transactions on neural networks* 20.1 (2008): 61-80.

[10] Brody, Shaked, Uri Alon, and Eran Yahav. "How attentive are graph attention networks?." *arXiv preprint arXiv:2105.14491* (2021).

[11] Erickson, Nick, et al. "Autogluon-tabular: Robust and accurate automl for structured data." *arXiv preprint arXiv:2003.06505* (2020).

[12] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016.

[13] Inayatoli, Hassaan, et al. "Between the Lines: How Do We Measure Pressure?", Kaggle, Kaggle (2023), <https://www.kaggle.com/code/hassaaninayatoli/between-the-lines-how-do-we-measure-pressure/notebook>.

[14] Pardun, Tyler and Pardun, Michael, "Introducing PPI: The Pocket Preservation Index", Kaggle, Kaggle (2023), <https://www.kaggle.com/code/tylerpardun/introducing-ppi-the-pocket-preservation-index>.

[15] Bachelder, Nick, "IDPI: A Situational Metric for Pass Rushers", Kaggle, Kaggle (2023), <https://www.kaggle.com/code/nickb1125/idpi-a-situational-metric-for-pass-rushers>.

[16] Karpick, Vincent, "Completions Added Through Suppression of Pressure", Kaggle, Kaggle (2023), <https://www.kaggle.com/code/vincentkarpick/completions-added-through-suppression-of-pressure>.

[17] Yurko, Ronald, et al. "Going deep: models for continuous-time within-play valuation of game outcomes in American football with tracking data." *Journal of Quantitative Analysis in Sports* 16.2 (2020): 163-182.

[18] Nguyen, Quang, et al. "Here Comes the STRAIN: Analyzing Defensive Pass Rush in American

Football with Player Tracking Data." *arXiv preprint arXiv:2305.10262* (2023).

[19] Macdonald, Brian. "Recreating the game: using player tracking data to analyze dynamics in basketball and football." *Harvard Data Science Review* 2.4 (2020).

[20] Fujii, Keisuke. "Data-driven analysis for understanding team sports behaviors." *Journal of Robotics and Mechatronics* 33.3 (2021): 505-514.

[21] Alamar, B. C. and J. Weinstein-Gould (2008). Isolating the effect of individual linemen on the passing game in the national football league. *Journal of Quantitative Analysis in Sports* 4 (2).

[22] Alamar, B. and K. Goldner (2011). The blindside project: Measuring the impact of individual offensive linemen. *Chance* 24, 25–29.

[23] Fernandez, Javier, and Luke Bornn. "Wide Open Spaces: A statistical technique for measuring space creation in professional soccer." *Sloan sports analytics conference* (2018).

[24] Goes, F. R., et al. "Unlocking the potential of big data to support tactical performance analysis in professional soccer: A systematic review." *European Journal of Sport Science* 21.4 (2021): 481-496.

[25] Pappalardo, Luca, et al. "PlayeRank: data-driven performance evaluation and player ranking in soccer via a machine learning approach." *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.5 (2019): 1-27.

[26] Herold, Mat, et al. "Off-ball behavior in association football: A data-driven model to measure changes in individual defensive pressure." *Journal of sports sciences* 40.12 (2022): 1412-1425.

[27] Forcher, Leander, et al. "The "Hockey" Assist Makes the Difference—Validation of a Defensive Disruptiveness Model to Evaluate Passing Sequences in Elite Soccer." *Entropy* 23.12 (2021): 1607.

[28] Burke, Brian. "We Created Better Pass-Rusher and Pass-Blocker Stats: How They Work." *ESPN*, ESPN Internet Ventures, www.espn.com/nfl/story/_/id/24892208/creating-better-nfl-pass-blocking-pass-rushing-stats-analytics-explainer-faq-how-work#full. Accessed 29 Nov. 2023.

[29] Averyehorvath. "Pass_rush_predictor_for_blocking_assignments." Kaggle, 10 Jan. 2023, www.kaggle.com/code/averyehorvath/pass-rush-predictor-for-blocking-assignments.

[30] Josephferraiola. "XPassRush: Identifying Pass Rushers Pre-Snap." Kaggle, 8 Jan. 2023, www.kaggle.com/code/josephferraiola/xpassrush-identifying-pass-rushers-pre-snap.

[31] Egle, Michael. "Good Rush, Bad Blocking." *Kaggle*, Kaggle, 8 Jan. 2023, www.kaggle.com/code/michaelegle/good-rush-bad-blocking/notebook.

[32] Graves, Alex, et al. "A novel connectionist system for unconstrained handwriting recognition." *IEEE transactions on pattern analysis and machine intelligence* 31.5 (2008): 855-868.

[33] Good, Irving John. "Rational decisions." *Journal of the Royal Statistical Society: Series B (Methodological)* 14.1 (1952): 107-114.

- [34] Fey, Matthias, and Jan Eric Lenssen. "Fast graph representation learning with PyTorch Geometric." *arXiv preprint arXiv:1903.02428* (2019).
- [35] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).
- [36] Naeini, Mahdi Pakdaman, Gregory Cooper, and Milos Hauskrecht. "Obtaining well calibrated probabilities using bayesian binning." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 29. No. 1. 2015.
- [37] Guo, Chuan, et al. "On calibration of modern neural networks." *International conference on machine learning*. PMLR, 2017.
- [38] NFL Next Gen Stats Team, "Next Gen Stats: Introduction to pressure probability", NFL, <https://www.nfl.com/news/next-gen-stats-introduction-to-pressure-probability>, (2023).

Appendix

Additional Tables and Figures

Feature Category	Feature Name	Description
Passer	qbX	x-position
	qbY	y-position
	qbSx	x-velocity
	qbSy	y-velocity
Player	x	x-position
	y	y-position
	s	velocity
	a	acceleration
	relX	x-position relative to passer
	relY	y-position relative to passer
	relDist	distance to passer
	x_dir	x-directional parameter (unit circle)
y_dir	y-directional parameter (unit circle)	
Context	time	Time starting at 3s prior to snap until end of the pass rush event
	position	Official NFL position of a target player
	positionGroup	NFL position group of a target player
	ngsPosition	NGS derived position of a target player
	nflId	Unique ID assigned to each player
	playId	Unique ID assigned to each play
	gameId	Unique ID assigned to each game

Table A.1. Primary features from NFL's NextGenStats player tracking data.

Target Feature	Description
blocker	Flag to indicate that the player is a blocker
rusher	Flag to indicate that the player is a rusher
matchupID	List of player ids for matched up players
hurry	Boolean indicating a hurry
hit	Boolean indicating a hit
sack	Boolean indicating a sack
ngs_pressure	Boolean for if a player gets within 1 yard of the passer during the rush or 1.5 yards at the time of the pass
pressure	Combineati

Table A.2. Description of the target features that were used in the framework from NFL's NextGenStats player tracking data

Distribution of distances for Matching and Non-Matching player pairs

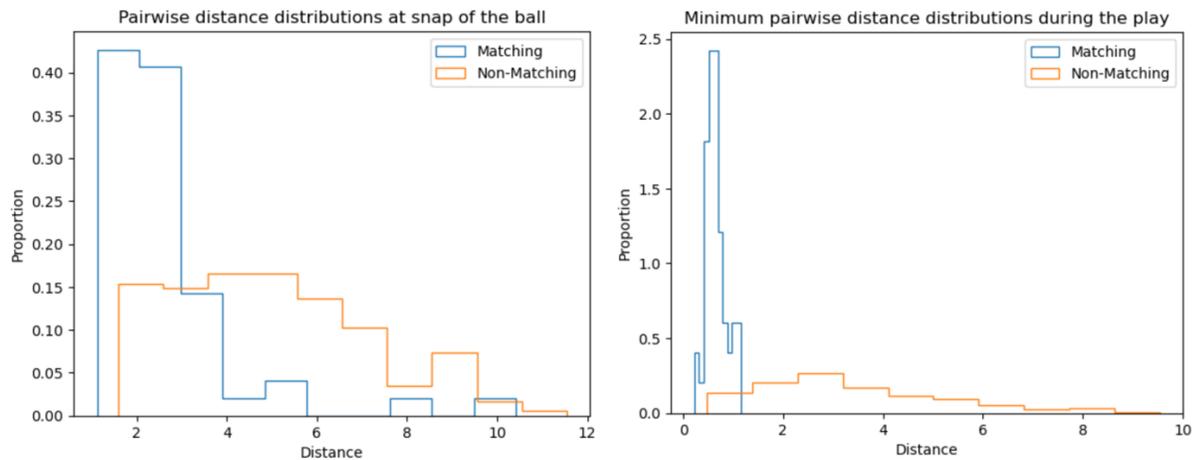


Fig. A.1. Distribution of distances at snap and minimum distances during play for Matching pairs and Non-Matching pairs. It shows that pairs that are matched up during the play have a much smaller average distances compared to pairs that aren't matched up.

Pressure Probability

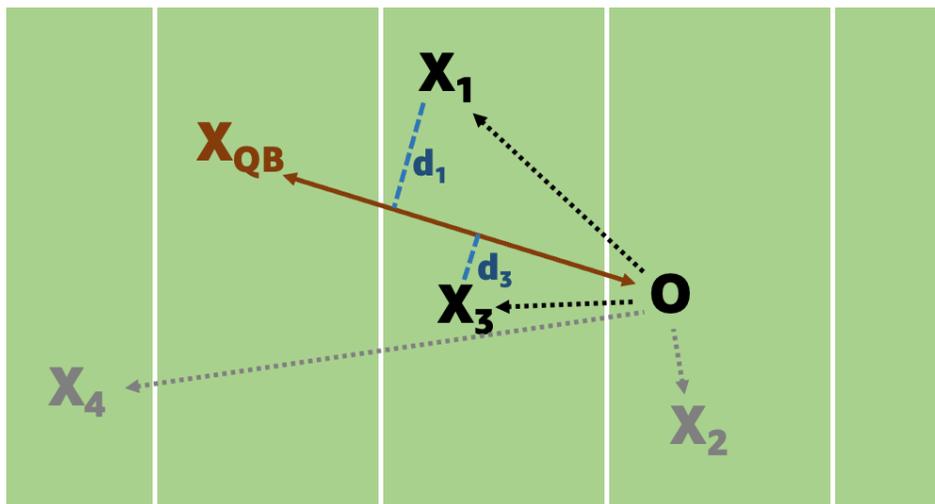


Fig. A.2. Illustration of how the blocker interference metric is calculated. Defenders X_1 and X_3 are eligible since they are between the quarterback and the rusher. The blocker at X_3 is given a stronger weight due to his closer proximity to the rusher's direct path to the quarterback. Blocker interference is the sum of these weighted orthogonal distances d_1 and d_3 .

Fig. A.2. shows a plot of two of the most important features relative distance and blocker-interference. As can be seen, the probability estimates are highest when the player is close to the quarterback (within 1 yard) and when the blocker-interference is very low. This is very intuitive given how pressure is typically based on a player's proximity and the threat the player poses to the passer.

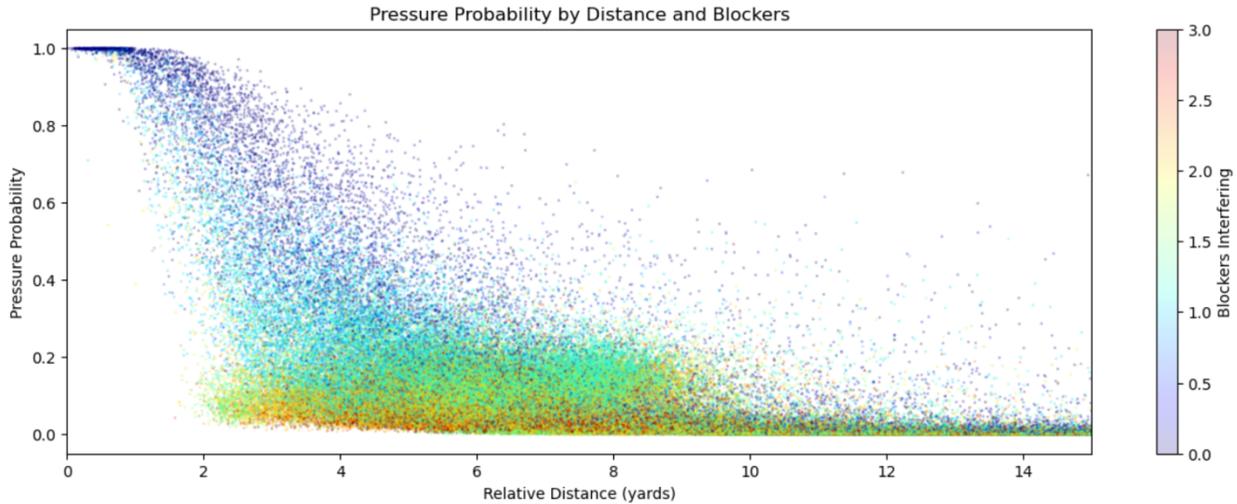


Fig. A.3. Predicted Probability based on relative distance and blocker-interference

Metric	Description
Pressure	A player affects the quarterback as determined by the pressure probability exceeding 75%.
Pressure rate	Total number of pressures over the total number of pass rush snaps
Time to pressure	Time from snap until the rusher's pressure first exceeds 75%
Quick pressures	A pressure that occurs with the first 2.5 seconds after the snap
Pressure time	The total duration of time the player spends above the 75% pressure threshold
Pressure probability at snap	The pressure probability at the time of the snap
Average pressure probability	The average pressure probability during the pass rush
Pressure rate over expected	The difference between the average pressure and the pressure at the snap
Peak pressure probability	The maximum pressure generated by a rusher during the pass rush
Positive rushes	Number of plays where the average probability exceeds the probability at snap
Negative rushes	Number of plays where the probability at snap exceeds the average probability
Positive rush rate	Positive rushes over total rush attempts
Net positive rushes	The difference between the number of positive pass rushes and negative pass rushes

Table A.3. A list of the pass rusher metrics developed by the NFL based on the output of our models [38]

Metric	Description
Matchup frequency	Total number of plays where two players are matched
Pressures allowed	The number of pressures allowed by a lineman
Sacks allowed	The number of sacks allowed by a lineman
Pressure rate allowed	The number of pressures divided by the number of pass blocking snaps
Double teams	A count of times that two or more blockers engage with a pass rusher

Table A.4. A list of the offensive line metrics developed by the NFL based on the output of our models [38]