Augmenting Tennis Point Stochastic Modeling Utilizing Spatiotemporal Shot Data

Corey Emery[1] Peter Freeman[2]

[1]Carnegie Mellon University, Department of Statistics and Data Science

[2]Thesis Advisor

Abstract

Tennis is an extremely dynamic sport, with court positioning between opposing players and the type, speed, and direction of their shots constantly changing and affecting the outcome of a point. Much of this information is not yet captured in existing tennis analytics, due to the proprietary nature and limited access of high-resolution spatiotemporal data currently collected at most major tennis events. Using Hawk-Eye player-tracking data collected from Men's and Women's Singles matches at the US Open between 2015 and 2018, our paper aims to build on preliminary research in this area, specifically focusing on the "Estimated Shot Value" measurement described by Floyd et al. (2019). We propose amending Floyd's stochastic model to incorporate information regarding shot characteristics such as speed, spin, and type of shot. This allows us to better exploit the capabilities of the Hawk-Eye system for meaningful analysis without sacrificing the simplicity of the structure of the original model. We calculate our models using all player data, rather than only data for a specific player, which allows us to focus on broader trends such as the relative strengths of certain patterns of movement around the court.

Keywords

Sports Analytics, Tennis, Point Modeling, Markov Processes

1 Introduction

The sport of tennis is a complex mixture of physical, mental, technical, and tactical aspects as players on both sides of the net aim to gain control over the pace of the ball, their opponent, and the momentum of the match. Historically not much attention has been devoted to understanding these interactions at a quantitative level. Much of the previous analysis has been based on higher-level information that generally summarizes gameplay at a match or set level. These include number of points won, serving percentage, winner and error counts, etc. While much simpler to collect and understand, these statistics also overlook the richness of the data produced over the course of a tennis match, and thus limit the insights we can create about match play, players' abilities, and the underlying structure of the game.

One of the main reasons that little focus was given to collecting data at a more refined level was the complexity of the task. Points in a tennis match are extremely fast-paced, and having a human attempt to record information at a shot level rather than a point level would likely lead to an unacceptable level of missing and/or inaccurate data.

However, in recent decades the Hawk-Eye system has been introduced. According to the Hawk-Eye website, *"Hawk-Eye has developed the most sophisticated vision processing technology in sport which enables us to not only track balls to mm accuracy but also players..."* It accomplishes this by placing a network of six to ten cameras at various locations around the tennis court, and using the different perspectives of these videos to track the players' locations at a rate of 25 frames per second and to calculate the trajectory of the ball for each of its "arcs" (meaning its path of travel between bounces or hits). When the system was first used in professional tennis play in 2006, its main purpose was to check whether a ball was in or out during an on-court challenge, and to this day this remains its most well-known function. But the system has also proved able to track additional data such as shot speed, spin, and outcome (i.e., winner vs. error). With an electronic system in place that is able to record much of what would have been impractical for a human to collect, Hawk-Eye allows us to dig deeper into tennis and begin to understand some of the complex choices players make and their implications.

Even though Hawk-Eye was first used over 15 years ago, its use in tennis research and analytics is still extremely limited for three reasons. First, the system is still not universally implemented, with many clay court tournaments, including the French Open, continuing to check for marks in the clay to determine whether a shot is in or out. This limits both the amount of data available to work with and the types of insights we can gain (any work cannot be generalized to all court surfaces or compare player behavior on clay, for example). Second, Hawk-Eye is a proprietary system operated under an agreement at each tournament, so much of the data recorded are available exclusively to that tournament's organization, which also limits the sample size that any one group is able to work with. Finally, the high cost of the system means that researchers or smaller tournaments are not able to experiment and collect data independently. Although Mora and Knottenbelt (2016) demonstrated a way to collect similar spatiotemporal using a lower-cost system of cameras and computer vision, this has not been widely implemented in a professional setting and does not contain the same richness as the Hawk-Eye datasets.

This paper aims to build on existing research in tennis analytics to better incorporate the information captured by Hawk-Eye. Specifically, we choose to focus on modeling individual tennis points and the value of a player's strategy and shot choice as captured by Floyd's (2019) "Expected Shot Value" (discussed further in section 2.1).

In Section 2, we layout the background of existing research in this subset of tennis analytics. We describe our data source and processing in Section 3 and general methodolgy and extensions in Section 4. We explore our various models in Section 5, before exploring our results and potential for future work in Sections 6 and 7.

2 Background

Despite the limited availability of Hawk-Eye data, there is existing research that focuses on tennis analysis on a point-by-point level and has a large potential to benefit from additional information about shots in those points. This is exemplified in Bevc (2015), who enhances a Hierarchical Markov chain model in order to more accurately predict the outcome of a tennis match by incorporating data from that match rather than relying solely on historical data. Bevc remarks that while they are unable to obtain Hawk-Eye data, it could allow them to expand their model to a sub-point level and include factors such as court positioning and shot power to further improve predictions.

There are also some papers that were able to gain access to Hawk-Eye data samples, predominantly from the Australian Open between 2012 and 2014, to use in their studies. Wei et al. (2013b) utilize a Bayesian Network to understand player behavior and what conditions best allow a given player to hit a winner (as they put it, that player's "sweet spot"). The same authors (2013a) also employ a similar framework and online model adaptation method to explore player behavior and predict shot locations. Wei et al. (2015) use a latent factor model to represent the serving "style" of a player in order to predict their most likely serve given the context of the match. Finally, Wei et al. (2016) use a similar method as their 2015 paper to predict shot and point outcomes, and include "style" to improve these predictions for specific players. These papers demonstrate the versatility of the Hawk-Eye data to help shed light and various different aspects of tennis strategy.

2.1 Floyd and Expected Shot Value

One paper that uses Hawk-Eye data for sub-point analysis that we specifically focus on in this paper comes from Floyd first in his thesis work at the Rochester Institute of Technology and then published in the *Journal of Quantitative Analysis in Sports* (Floyd et al. 2019). Floyd et al. focus on modeling a tennis point as a Markov chain of independent shots to calculate the expected value of points a player would gain given their and their opponent's court position. This was inspired by previous work done on basketball, where the expected value of different passes, player possessions, and court locations has been analyzed (Floyd et al. and references therein).

The model proposed by Floyd et al. only incorporates player court locations as states in the Markov model, but does not take advantage of additional information that Hawk-Eye provides that could enhance these predictions. For example, incorporating the type of spin on the ball, the speed of the shot, how long the point has been played, and other physicality information could strengthen this model and provide greater insight into the evolution of a point and player strategy.

2.2 The Physicality Metrics of Tennis

In order to inform which aspects of the Hawk-Eye data might be most useful as part of this model, we also investigate the physical impact of a tennis match on players. Although all players will likely experience general physical fatigue, it is important to understand how this develops over the course of a match. Rowland (2014) writes about average point length, number of shots, distance ran, etc., which could help us establish baselines to gauge physical exertion during a specific point. Reid and Duffield (2014) also explore the physiological profile and movement characteristics of tennis matches and relate these to the development of fatigue. Finally, Fernandez-Fernandez et al. (2009) review many physical and physiological responses during match play at a deeper level, including temperature, court surface, intensity, heart rate, oxygen intake, and perceived exertion. These measurements are not taken during professional game play however, so we should be cautious when considering this in the context of our given data.

3 Data

The Hawk-Eye database structure can be split into three major sections, each of which focuses on a different aspect of a tennis point. First, the locations of each player are recorded at a rate of 25 frames per second and stored as an (x, y)-coordinate pair measured in meters as `.prj` files. Second, we have information about the location of the ball at key points of each arc and the complete trajectory calculations used in line-call challenges as `.trj` files. Third, aspects about player shots including speed, spin, outcome, and duration are contained as `.xml` files.

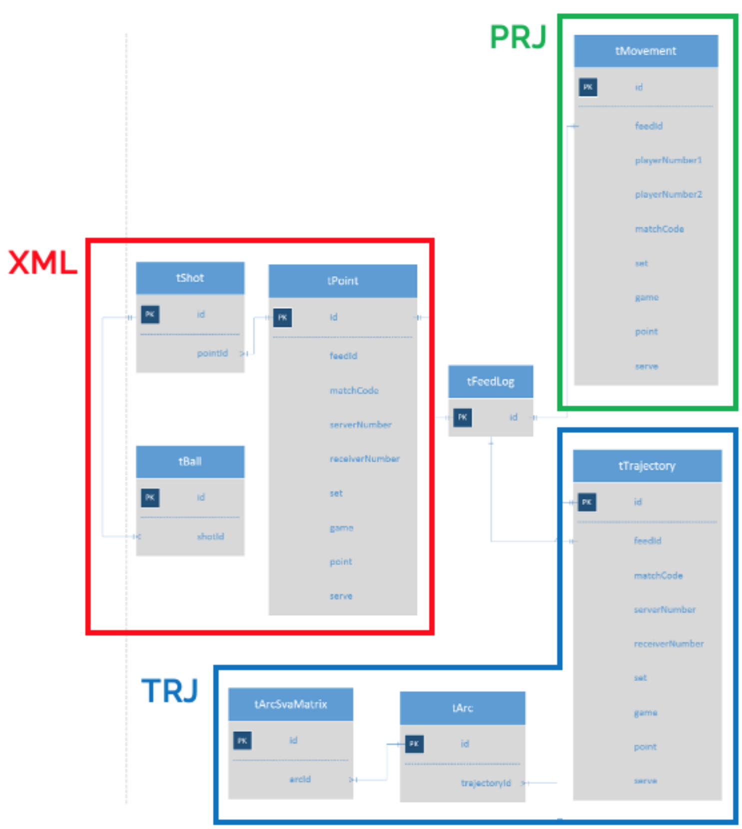

Figure 1: Hawk-Eye Database Structure

The data we use come from the USTA and contain anonymized Hawk-Eye data recorded during the US Open between 2015 and 2018. For our research, we use the raw database files that are updated over the course of the matches rather than the above listed file types. This necessitates some additional processing on our end initially to order the data and more easily group and compare certain attributes.

The dataset is then cleaned to filter out doubles matches (player position tracking is not currently supported when there are multiple players on one side of the court) and older matches that are not bound by a service-level agreement between Hawk-Eye and the USTA. Matches that are not contained in all three sections of the database are also removed to ensure a complete set of data for each match we used in our analysis.

Furthermore, cleaning on a point level is performed to remove data that shows the same player hitting multiple shots in a row, since this is not a valid scenario in tennis. Our analysis is dependent on points being internally consistent in order to create meaningful Markov chains over players' shots. We additionally filter out extraneous shots (for example, shots recorded after a shot had already gone out of bounds, thus ending the point) and points where the outcome (winner vs. error) did not align with who scored the point, as this also signals loss of data. Overall, this affected approximately 20% of the points in our data and 27% of the shots, and thus removing these points does not pose a significant hurdle to our work.

After this cleaning, our dataset contains 701 unique matches. Since the dataset is anonymized, it is currently not differentiated between men's and women's matches, although this could be achieved by analyzing the number of sets in each match. The matches contain 144,695 points and 473,401 shots. The data also encompass a total 366 of unique players in 690 unique matchups. Due to the larger nature of our dataset, we might be able to overcome the issue of model complexity mentioned in other papers that previously used Hawk-Eye data: since their match samples are limited, the level of complexity they can incorporate into their models is also limited.

In order to perform our exploratory analyses and construct our model, we perform additional computations with the Hawk-Eye data. These include calculating the distance a player ran during a point from the position (`.prj`) data to understand how fatigue in a point might affect the outcome. We used player and ball location (`.xml`) data to establish which zone of the court the player was in when each shot was hit, as setup in the Floyd et al. model.

In order to replicate and expand upon the Floyd et al. methodology, we organize our data into three tables, one focusing on player position-tracking, one on shot-level information, and the last on point-level information.

The position-tracking table is used to establish the zones of each shot and ultimately the strike and return states, which are based on the relative positioning of players on the court. This also lets us calculate the transition probabilities between these states.

The point-level table is used to group the shots and contains information regarding who won each point. This allows us to assign strike and return categories. This also provides information regarding whether a serve was in or out, which helps us filter out invalid data.

The shot-level table provides us with information to order shots within a point, which also helps establish strike and return categories. It also includes shot type, spin, and speed, which we use to augment the Floyd et al. model and adjust category weights to more accurately represent the values of various shots in tennis.

4 Methodology

4.1 Markov Chain Models

Markov chains are an effective way to model a sequence of events or states (the status of the point after each shot) over the course of a sequence (the point itself). By definition, this process is stochastic, meaning that there are probabilities associated with moving from the current state to any other potential state. This type of model also makes the assumption that these transition probabilities rely only on the present state, and are independent of any past information in the process (this is called the Markov Property). An example of a Markov chain for a tennis point can be seen in Figure 2.

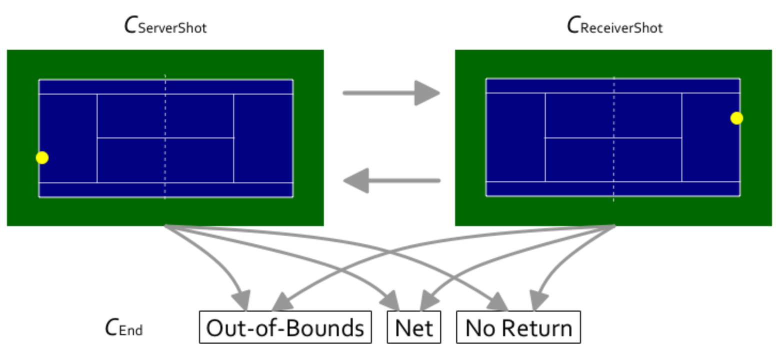

Figure 2: Markov chain model of a tennis point

This final assumption holds reasonably well in tennis, with some modifications and additional assumptions. Shots in tennis do not occur at equal intervals, but rather vary based on the type, speed, spin, and location that the ball is hit to. Because of this, we cannot fully assume this process will have the properties of a continuous-time Markov chain. Rather, we assume that the process is Semi-Markovian, and that the process is memoryless only when there is a jump to a new state (when the next player hits the ball).

This limits our ability to analyze the point while the ball is moving between the two players (and our chain becomes closer to a discrete-time chain), but still allows us to gain insight into the players' dynamic and interactions in terms of their shot choices and movements.

4.2 Floyd's Original Approach

In his publications, Floyd sets up a Markov chain with coarsened states representing the server's shot, the receiver's shot, and the end of the point. He defines the point's end as either the ball going out-of-bounds, into the net, or not being returned by the opponent. This structure both satisfies the assumptions necessary in order to use a Markov chain model, and makes interpretation much clearer and more meaningful. Due to these qualities, our model maintains this same general structure.

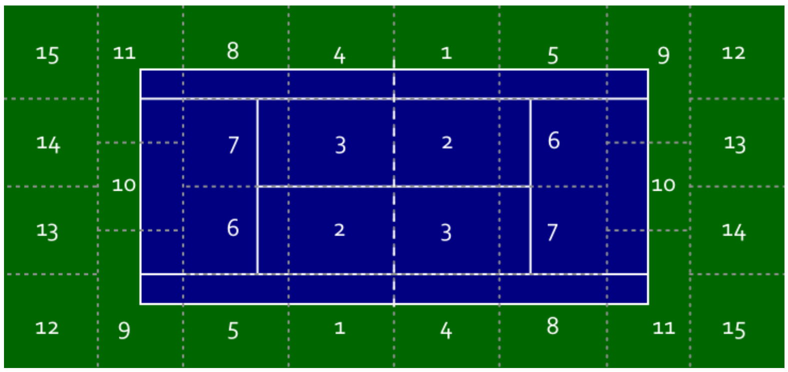

Figure 3: Floyd et al. Court Zoning System

The next step is to establish states to describe each shot from each player's perspective. Floyd et al. focus on relative player positioning, and thus discretize player court position into a zone system. In his thesis he uses 18 zones, and in the publication he describes 15 zones (as shown in Figure 3), removing some level of detail from the area surrounding the baseline. All possible states for groundstrokes are thus described by the combination of the zones for the "striker" (the player hitting the current shot) and the "returner" (the player receiving the current shot), with separate states describing first and second serves and returns as these shots are expected to have different qualities and a different distribution from groundstrokes.

Represented mathematically, we have that the state space of shots that end the point are:

$$\mathtt{C}_{\mathtt{end}} = \{\mathtt{out\text{-}of\text{-}bounds},\ \mathtt{net},\ \mathtt{no\ return}\}$$

with associated point values for strikers ($\mathtt{PV}_{\mathtt{S}}$) and returners ($\mathtt{PV}_{\mathtt{R}}$):

$$\begin{aligned}
&\mathtt{PV}_{\mathtt{S}}(\mathtt{out\text{-}of\text{-}bounds}) = 0 \qquad && \mathtt{PV}_{\mathtt{R}}(\mathtt{out\text{-}of\text{-}bounds}) = 1 \\
&\mathtt{PV}_{\mathtt{S}}(\mathtt{net}) = 0 && \mathtt{PV}_{\mathtt{R}}(\mathtt{net}) = 1 \\
&\mathtt{PV}_{\mathtt{S}}(\mathtt{no\ return}) = 1 && \mathtt{PV}_{\mathtt{R}}(\mathtt{no\ return}) = 0
\end{aligned}$$

The remainder of the state space falls into $\mathtt{C}_{\mathtt{shot}} = \{\mathtt{C}_{\mathtt{ServerShot}} \cup \mathtt{C}_{\mathtt{ReceiverShot}}\}$, which represents states where the point is still in play. This includes all of the zone combinations and first and second serve states as described in the previous paragraph.

Next, Floyd et al. describe a set of possible categories that each shot falls into from both the striker and returner's perspective, including things such as pure winners (shots that lead directly to a point for the striker), shots that continue play, set-up shots that lead to pure winners or errors, etc.

Strike Category	Symbol (λ_{SC})	Weight (w_{SC})	Definition
Pure Winner	$\lambda_{\texttt{PW}}$	1.00	In-bounds strike which is not returned
Set-Up of Pure Winner	$\lambda_{\texttt{SUPW}}$	0.75	Strike by winning player prior to $\lambda_{\texttt{PW}}$
Forced Losing Strike	$\lambda_{\texttt{FLS}}$	0.75	Strike by winning player prior to $\lambda_{\texttt{LS}}$
Non-Impactful	$\lambda_{\texttt{NI}}$	0.50	Strike which does not fall into other categorizations
Set-Up of Opponent's Pure Winner	$\lambda_{\texttt{SUOPW}}$	0.25	Strike by losing player prior to $\lambda_{\texttt{PW}}$
Losing Strike	$\lambda_{\texttt{LS}}$	0.00	Strike which goes out-of-bounds or into the net

Table 1: Original Floyd et al. Strike Categories and Weights

Return Category	Symbol (λ_{RC})	Weight (w_{RC})	Definition
Returned Pure Winner	$\lambda_{\texttt{RPW}}$	1.00	Returned $\lambda_{\texttt{PW}}$
Returned Set-Up of Pure Winner	$\lambda_{\texttt{RSUPW}}$	0.75	Returned $\lambda_{\texttt{SUPW}}$
Returned Forced Losing Strike	$\lambda_{\texttt{RFLS}}$	0.75	Returned $\lambda_{\texttt{FLS}}$
Returned Non-Impactful Strike	$\lambda_{\texttt{R}}$	0.50	Returned $\lambda_{\texttt{NI}}$
Returned Set-Up of Opponent's Pure Winner	$\lambda_{\texttt{RSUOPW}}$	0.25	Returned $\lambda_{\texttt{SUOPW}}$
Returned Losing Strike	$\lambda_{\texttt{RLS}}$	0.00	Returned $\lambda_{\texttt{LS}}$
Not Returned	$\lambda_{\texttt{NR}}$	0.00	Unable to return

Table 2: Original Floyd et al. Return Categories and Weights

They assign a weight to each category to represent the expected point value of hitting that shot, and these weights are ultimately critical to the Expected Shot Value calculation. This is because the authors are able to represent this Markov chain as an equation rather than constructing the actual model.

$$\texttt{ESV}(\texttt{c}_\texttt{u}) = \begin{cases} \displaystyle\sum_{\texttt{c}_\texttt{w} \in \texttt{C}_\texttt{shot}} \texttt{RV}(\texttt{c}_\texttt{w}) P_{\texttt{uw}} + \sum_{\texttt{c}_\texttt{w} \in \texttt{C}_\texttt{end}} \texttt{PV}_\texttt{S}(\texttt{c}_\texttt{w}) P_{\texttt{uw}} & \text{if the player is the striker} \\ \displaystyle\sum_{\texttt{c}_\texttt{w} \in \texttt{C}_\texttt{shot}} \texttt{SV}(\texttt{c}_\texttt{w}) P_{\texttt{uw}} + \sum_{\texttt{c}_\texttt{w} \in \texttt{C}_\texttt{end}} \texttt{PV}_\texttt{R}(\texttt{c}_\texttt{w}) P_{\texttt{uw}} & \text{if the player is the returner} \end{cases}$$

In these equations, the Serve Value ($\mathtt{SV}$) and Return Value ($\mathtt{RV}$) are the weighted averages of the weights for the types of shots hit from a current strike state ($\beta^{\mathtt{S}}$) or return state ($\beta^{\mathtt{R}}$).

$$\mathtt{SV}(\beta^{\mathtt{S}}) = \sum_{\lambda_{\mathtt{SC}} \in \Lambda_{\mathtt{SC}}} w_{\lambda_{\mathtt{SC}}} \frac{|\mathrm{S}_{\beta^{\mathtt{S}}, \lambda_{\mathtt{SC}}}|}{|\mathrm{S}_{\beta^{\mathtt{S}}}|}$$

$$\mathtt{RV}(\beta^{\mathtt{S}}) = \sum_{\lambda_{\mathtt{RC}} \in \Lambda_{\mathtt{RC}}} w_{\lambda_{\mathtt{RC}}} \frac{|\mathrm{R}_{\beta^{\mathtt{R}}, \lambda_{\mathtt{RC}}}|}{|\mathrm{R}_{\beta^{\mathtt{R}}}|}$$

and the transition probabilities ($P_{\mathtt{uw}}$) are defined as the observed transition frequency in the data of going from state $\mathtt{u}$ to $\mathtt{w}$. In the original model, this is conditioned on a certain player being the striker (since the striker is in control of the ball at this point), but since we are not calculating ESVs for specific players this is omitted here.

$$P_{\mathtt{uw}} = \frac{N_{\mathtt{uw}}}{\sum_{\mathtt{w}'} N_{\mathtt{uw}'}}$$

For strikers, the ESV equation can be interpreted as the point outcome for each end state weighted by how often that state is reached from the current state, plus the anticipated point outcome given that the point continues and the striker now becomes the returner. This anticipated value is calculated as the weighted average of all the category weights (accounting for the frequency that each type of shot is hit by that player in the given state) as defined in Table 2 above. The process is the same for returners, except we calculate the anticipated point outcome using the strike values instead of the return values.

4.3 Our Proposed Modifications

We feel that the most potential for improvement in this model lies in the strike/return categories and weights that Floyd et al. define. The Hawk-Eye data provide us with much more information than player positioning. We can analyze how deep into a match the players are (in terms of how many points, games, etc. have already been played) in order to factor in the idea of physicality. We can also break down nuance in shot choice by incorporating metrics about speed and spin. On a serve, a faster shot is likely harder to return, and shots that have topspin are often more aggressive than those with backspin.

We incorporate these attributes by creating new categories and adjusting their associated weights to more accurately reflect how much these shots contribute to a won or lost point. As with the original model, there is still much gray area in terms of determining the appropriate weights for each type of shot. We attempt numerous different combinations of variables and compare the Expected Shot Values produced by each for similar points in order to assess their performance.

Since this statistic is not exactly equivalent to the probability of winning a point, we do not have an objective metric against which to measure accuracy. Alternatively, we rely on the validation method used by Floyd et al. by examining the correlation between ESV and win rate to ensure that our models do not present any significant performance drop compared to the Floyd et al. model. Instead of using Leave-One-Out-Cross-Validation (LOOCV) however, we choose to calculate the correlation coefficient based on the training data. Not only does this save us time in terms of computation, but we also do not expect it to affect the nature (i.e., linear vs. curved) of the relationship, which is what we are interested in for our validation.

5 Analysis

5.1 Establishing a Baseline Model

The first step of our process is to recreate the original Floyd et al. model using our data. This allows us to effectively examine the impact of our proposed changes on individual points, and identify any improvements in the correlation that suggest our models better fit the underlying strategies of a point in tennis.

Our methodology splits the model building into five main steps. We start by feeding in our three main data tables (position-tracking, shot-level, and point-level information), and then classify the court position for each player at each shot based off the original paper's zoning scheme, and combine this information with serving data to assign the strike and return states for each shot.

Second, we assign each shot a category based on that shots outcome as shown in Tables 1 and 2. This is done by isolating individual points and determining how far from the end of the point each shot is and which player ultimately wins the point. The weights for each strike and return category are also fed into our methodology to be used in our calculations.

Third, with the states assigned to each shot we calculate the overall transition probability matrices. Unlike Floyd et al., we create two matrices: the probability of going from a strike state to the next strike state and the probability of going from a strike state to the next return state. We choose to do this because the striker is always in control of the shot, but depending on whether we are utilizing the strike value or return value in calculating the Expected Shot Value influences the state of the next shot we consider.

Additionally, instead of calculating these matrices for a specific player, we do it for our data overall. We do this so that as we create more strike and return categories in our proposed models, we still have a sufficient amount of data in each bin to perform our calculations. While this does not focus on individual players like the Floyd et al. structure, it does let us test our proposed changes. This structure also requires only small changes to be able to analyze individual player performance.

Finally, we calculate the strike and return values as the weighted average of the strike and return categories, and for every shot calculate the Expected Shot Value for both players utilizing the summations described by Floyd et al.

5.2 Generalizing the Process

We next establish generalized functions that assign strike and return categories, and their respective weights, that can be fed into this methodology. This allows us to quickly make changes and test new models. This is where we feel we can make the most effective use of the Hawk-Eye data currently. By factoring in data regarding shot type, spin, and speed, we might be able to more accurately assign weights and improve the models.

5.3 Adding Spin

The first factor we consider incorporating is the spin of the shot. In the Hawk-Eye data, the spin of the shot is encoded as both a numeric variable (in RPM) and a nominal variable with categories `TOPSPIN`, `BACKSPIN`, `FLAT`, and `NOSPIN`. In order to maintain a reasonably small space of strike/return categories and weights, we choose to incorporate the categorical variable into our model.

We theorized that in most cases, topspin shots are a more aggressive, offensive shot, and thus should have a higher weighting overall. Backspin is usually a more defensive shot, and thus should have a lower weighting overall. Flat shots lie somewhere between these two: in cases where it forces an error or sets up a pure winner, we weight it similarly as topspin shots since a flat shot is used to change pace and try to speed the ball past an opponent. But in cases where it set up an opponent's pure winner, we weight it similarly as backspin shots, as it is likely a mishit or more defensive shot. `NOSPIN` shots are left as the default weighting. Specific weights for this model can be found in Table A.1.

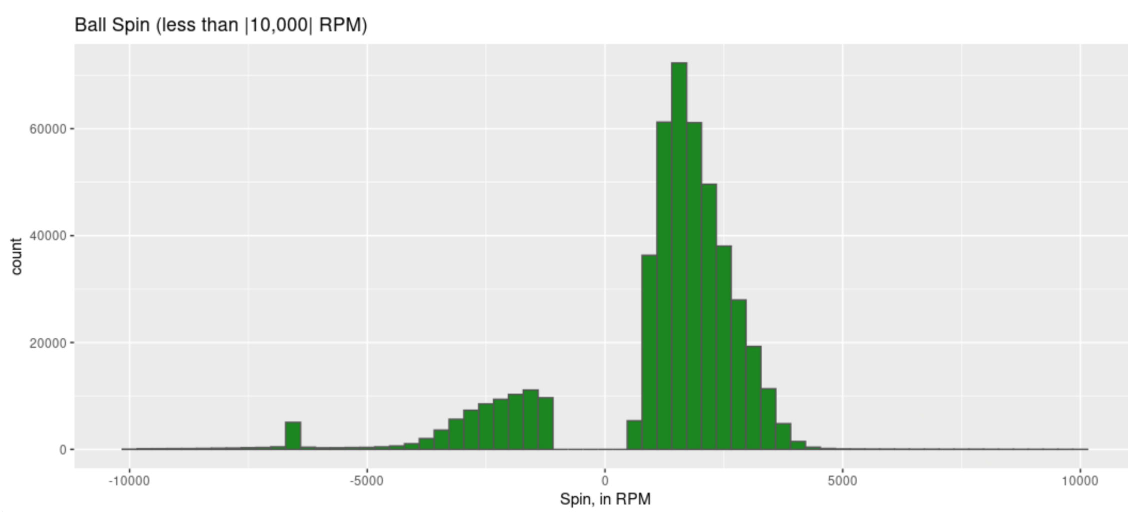

Figure 4: Histogram of shot spin, in RPM

Looking at the spin in RPM, there is a large spike of `NOSPIN` values that can be accounted for by serves (Hawk-Eye does not attach a specific spin type to serves). This is removed in Figure 4.

Additionally, we notice a small spike of very negative shot RPMs (near -6000 RPM). This anomaly could be an error in measurement from Hawk-Eye, or it could represent a different type of shot: a drop shot. We test this idea by creating a separate model and adding a drop shot category to the existing spin categories. We weight this category the same as topspin shot, since it is also an offensive shot. Specific weights for this model can be found in Table A.2. Examining this grouping of points further to uncover whether these are in error or to explain how this grouping is so close together is an avenue for future work as well.

5.4 Factoring in Speed

Next, we generate a model that factors in the speed of a shot. We break this down between first serves, second serves, and groundstrokes. Each of these shot types is unique, so to accurately represent the tennis point it makes sense to assign different values to different speeds based on the shot type. In order to keep a manageable category space, we also discretize the speeds instead of considering them as having continuous values.

From analyzing distributions of our data and online sources, we split serves into fast and slow categories, and other shots into fast, slow, and average categories. We split first serves at 110 mph and second serves at 90 mph. Other strokes are defined as fast if they are greater than 85 mph and slow if they are less than 50 mph. Specific weights for this model can be found in Table A.3.

However, we realize that groundstrokes also have a lot of nuance in terms of speed. Sometimes shots are intentionally hit slower in order to disrupt the pace of the point, and these types of shots should not be given a lower weight. Serves, on the other hand, do not exhibit this same variation (except in rare cases), so assigning more value based on higher speed makes more sense.

Thus, we also examine a model that removes our speed weightings for non-serves and instead assigns these shots the default weighting. Specific weights for this model can be found in Table A.4.

5.5 Combining Features

In order to examine the interactions between speed and spin, we set up two additional weighting systems. For the first model, we attempt to isolate what we think might be the most meaningful interactions, so as to not expand the number of categories excessively. This leads us to focus on serve speed (as described in the previous section), fast topspin shots as offensive shots (with a higher value attached to them), and slow backspin shots as defensive shots (and a lower value). Specific weights for this model can be found in Table A.5.

The second considers only non-serves (since serves do not have spin recorded), but considers more combinations of speed and spin. This follows our general theory that topspin shots and faster shots should have higher weights, and thus orders shot types from highest value to lowest value: fast topspin, fast backspin, slow topspin, slow backspin. Again, all other shots are assigned the original default values from the Floyd et al. paper. Specific weights for this model can be found in Table A.6.

5.6 Considering Shot Type

When considering forehand vs. backhand, this choice is most meaningful in the context of other features such as spin. For example, while a backhand slice is a common shot (and in many cases more defensive), a forehand chip is much less common (and could be indicative of a drop shot or a shot intended to change the pace of the point). In this sense, we set up a model that weights both topspin and forehand backspin shots with higher values, and backhand backspin shots with lower values. Specific weights for this model can be found in Table A.7.

One additional idea for future work is to examine shots like inside-out forehand, where the player hits a forehand on the side of the court that normally corresponds to their backhand. However, knowing that requires knowing whether a particular player is left-handed or right-handed, which we do not have in our current anonymized dataset. If the players are known however, this information would be easy to collect and incorporate into a model.

Alternatively, we can attempt to predict handedness by examining the types of shots that a particular player hits from each zone of the court. For right-handed players, the majority of forehands should come from zones 6, 9, 12, and 13, and backhands from zones 7, 11, 14, and 15. We see an example of this in Table 3, and we can infer that `dGe.Player309` is right-handed.

	Zone							
	Court Left-Side				Court Right-Side			
Shot Type	7	11	14	15	6	9	12	13
Forehand	2903	9374	6369	141	3520	18312	370	21326
Backhand	3518	33900	23943	893	667	1872	91	2952

Table 3: Shots by Zone for `dGe.Player309`

5.7 Estimating the Weight of an Non-Impactful Shot

Floyd et al. define a non-impactful shot as a default category, meaning that it does not lead to the end of the point or set up a winner for either player. In essence, these shots merely continue the point, and thus are assigned a pure neutral weight of 0.5.

However, by prolonging the point, a player increases the amount of fatigue they experience over time, which can make it harder to hit winning shots. Additionally, by continuing the point a player gives their opponent another opportunity to hit a winner against them. Both of these factors suggest that the weight for this type of shot could be less than 0.5.

Our data also support this. For the strike category `NI`, the proportion of these shots that the striker eventually won is 0.4333, and for the return category `R`, the proportion of these shots that the returner eventually won is 0.4262. Using these values as a more accurate weighting estimate, we substitute them into the original Floyd et al. weight set and reconstruct the model. The full set of weights for this model can be found in Tables A.8 and A.9.

6 Results

6.1 Validation and Model Strength

We validate our models by examining the correlation coefficient between Expected Shot Value and the win rate. Since every point is either won or lost (and thus has a win rate of either 1 or 0), we calculate win rate by binning the shots based on their ESV value and treating the win rate as the percentage of shots for which the striker won (when looking at Strike ESVs) or lost (when looking at the Return ESVs). We define 11 bins, with breaks at 0.05, 0.15, 0.25, etc.

In the original Floyd et al. methodology, the authors use one plot to demonstrate the correlation between ESV and win rate. However, we feel it makes more sense to examine Strike and Return ESVs separately. This is because the underlying calculations of these two are different: one relies on Return Value and the other Strike Value, and also the win rate for one is calculated on the striker winning the point whereas the other is based on the striker losing the point. For these reasons, we create separate plots to examine the correlations, as seen in Figure 5.

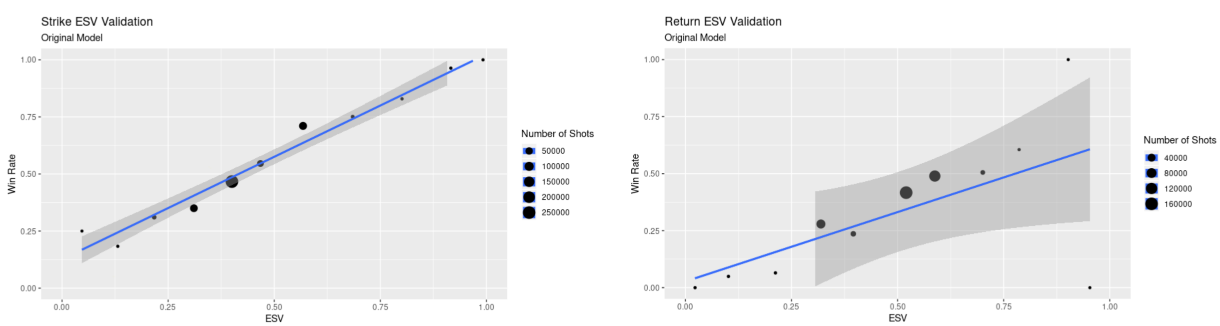

Figure 5: Floyd et al. Model Validation, with 95% confidence bars shown

Because we are treating each bin as an individual data point when calculating the correlation coefficient, this method is highly influenced by shots with very high or low ESVs. Because there are very few of these points, the win rate might not truly reflect what the actual win rate for points in that bin is, and this can affect our assessment of model strength and accuracy.

An example of this from our data is in the validation of Return ESVs in the > 0.95 bin in Figure 5 above, where a player lost a point during which they had a very favorable shot condition. Since this is the only shot in the bin, the win rate is 0, and this significantly weakens the correlation, as seen in Table 4. Moving forward, we might consider using a weighted R^2 to decrease the influence of outliers by giving data points more influence if they contain a greater number of shots.

Examining these plots, even if we ignore the outlier data point, the validation plot for Return

ESV appears to follow a curved pattern. However, the relationship is strongly linear for Strike ESV. We theorize this difference might be due to the fact that for any given shot, the striker is in control of the point, whereas the returner is at their mercy, and this might strengthen the association between ESV and win rate for strikers and weaken it for returners. We thus choose to focus on Strike ESV R^2 in comparing model performance.

Model	Strike ESV R^2	Return ESV R^2
Shot Type + Spin	0.9820	0.3633
Spin + Dropshot	0.9815	0.3709
Spin	0.9814	0.3706
Estimated Neutral Shot Weight	0.9762	0.9486[1]
Floyd et al. (Baseline)	0.9758	0.3864
Serve Speed	0.9758	0.3864
Full Speed	0.9756	0.3837
Full Speed + Spin	0.9756	0.3850
Non-Serve Speed + Spin	0.9755	0.3841

Table 4: Model Correlations, ordered from highest to lowest Strike ESV

We see that the relationship between Strike ESV and win rate for each of these models is strong, and by examining the validation plots in Appendix A.2 we see that the plots are all strongly linear. We see a jump in R^2 when looking at models that factor in spin, which might be partially due to the fact that these models move data points from the lowest bin to the second lowest bin.

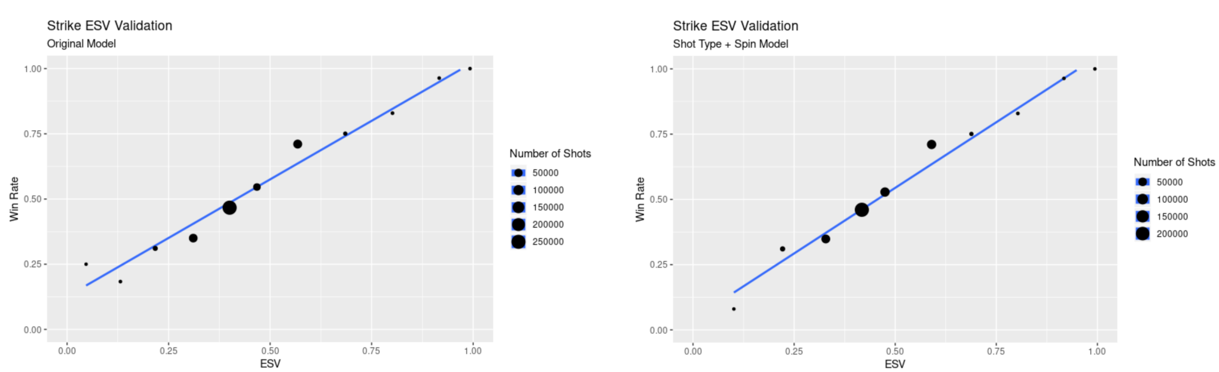

Figure 6: Floyd et al. Model vs. Shot Type + Spin Model Validation

Since the win rate of this lowest bin originally had a higher win rate, this suggests that we might have uncovered a pattern that more accurately models the mechanics of a tennis point. However, we still want to check this change using a weighted R^2 to ensure that this improvement is not purely due to outliers.

Examining the Strike ESV plots in Appendix A.2, we notice they are all extremely similar, and

[1]In this model, the erroneous example that has a high Return ESV and zero win rate is pushed to the next lowest bin, thus significantly increasing R^2 as this shot is no longer in a bin by itself.

there is one data point with an ESV roughly 0.6 that consistently has a higher win rate than what the linear model would predict. This suggests that there might be an underlying structure in these shots that our models do not capture. We examine this further in the next section.

6.2 Examining Win Rate Residual

In all of our models, the bin of shots with Strike ESVs between 0.55 and 0.65 have a higher average win rate than what would be expected from our linear model validation. Digging deeper into these shots to understand why this is the case, we examine potential patterns that could be targeted in future iterations of this model.

We start by examining what types of shots we are looking at. Since our model averages over all data, the ESVs do not differ for shots in the same state. The states, number of shots, and Strike ESVs are shown in Table 5.

Location State	Strike ESV	Number of Shots
1S	0.5657	104056
2-5	0.5958	3
2-9	0.5719	102
2-11	0.6141	293
2-13	0.6339	150
3-2	0.5730	84
3-4	0.6375	5
3-9	0.6144	202
3-11	0.5888	240
6-2	0.5770	98
6-13	0.5860	1049
6-14	0.5946	1005
7-12	0.6253	104
7-13	0.5768	1141
7-14	0.5826	2141
9-1	0.5685	4
9-4	0.6344	6
11-1	0.5737	6
12-12	0.5651	3

Table 5: Floyd et al. Model Strike ESVs $[0.55, 0.65]$

We find that this bin includes all of the first serves in our data, and since there are so many shots from this state, it dominates all of the other shots in this bin (of the 110,692 shots, only 6,636 are not serves).

When we remove the first serves, the average Strike ESV for the bin rises from 0.5670 to 0.5881, and the win rate drops from 0.7106 to 0.6698, which brings the data in the bin more in line with our linear model (see Figure 7).

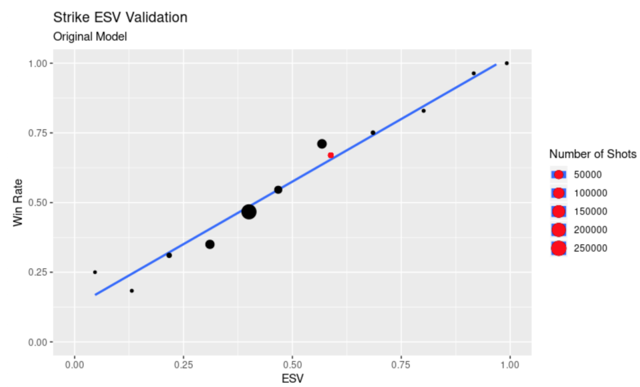

Figure 7: Floyd et al. Model Validation, with the bin that has First Serves excluded shown in red

This suggests that there is still an underlying structure around serves that future models could address. Since we notice this bin is above the trend line in every model, we note that none of our existing solutions effectively address the issue. This could be because of missing data associated with serves (Hawk-Eye does not measure spin on serves), or because our current models only give above average weight to serves that produce a pure winner, set one up, or force an opponent's error. It is possible that a good serve sets up an advantageous position four or five shots into the point, which is beyond the scope of our current model. A good first serve also gives the server a level of control throughout the point, a concept that a future model can try to encode.

6.3 Potential to Incorporate and Analyze Physical and Mental Factors

In addition to information related to the actual shots, Hawk-Eye also provides physical information about the players that can be useful in determining how good or bad of a shot they hit under the current match circumstances. We explore possible extensions below.

Some physical information is already encoded in our models, and thus do not need to be explicitly represented. One example of this is distance and direction ran between shots. Since the states of our Markov chain model are based off of court positioning, the transition between states effectively accounts for the distance a player moves to hit a shot. To understand the effect of distance and direction ran on a player's hitting ability, we can filter points based on the previous return zone and the current strike zone and examine the distribution of categories these shots fall into.

Table 6 demonstrates an example of this. The first two rows demonstrate how charging the net (forward movement) appears to have an average shot weight above 0.5 and is thus an advantageous choice under the right circumstances. However, we also see how increasing the distance the player has to move decreases the advantage. We also see the negative impact of being forced to retreat from the net, as well as being dragged across the baseline or backcourt.

Movement and Zones	PW	SUPW	FLS	NI	SUOPW	LS	Mean Weight
Baseline to Net: $(9-11) \rightarrow (1-8)$	2072	489	900	785	592	1045	0.6212
Backcourt to Net: $(12-15) \rightarrow (1-8)$	335	94	204	226	246	329	0.5112
Retreat: $(1-4) \rightarrow (9-15)$	13	7	19	85	36	77	0.3544
Cross Court: $9 \rightarrow 11$ or $11 \rightarrow 9$ or $12 \rightarrow 15$ or $15 \rightarrow 12$	188	92	280	896	177	488	0.4523

Table 6: Shot Distributions for Different Types of Court Movement

Additionally, our models currently do not incorporate the notion of fatigue. Hawk-Eye provides us with information about how long a match is underway, how many shots are in the current point, and how far a player runs over the course of a match. In addition to this, in a non-anonymous context we can factor in information regarding temperature and a specific player's conditioning.

All of these factors can impact the weighting systems used in our models. We feel that while using constant weights is good for understanding the overall value of a particular shot, it is more realistic that these weights change over the course of a match to reflect the players' physical condition. For example, hitting a weaker shot should have a lower weight when a player is fatigued as it will be even harder in this state to retain a hold on the point when giving the opponent an easier setup. Conversely, if the opponent is in a fatigued state, hitting a weaker shot might not have as low a weight as it will take less energy to produce a winning shot. By incorporating physicality data, a future model could develop a methodology to estimate how much these weights should change over the course of a match. By combining this with player physicality profiles, such as that developed by Spence and Kovalchik (2019), this could also be adjusted to each player personally.

Finally, although the mental aspects of the game are much harder to measure quantitatively, future work could attempt to incorporate the idea of momentum and control by further adjusting weights based off of streaks of points won or above average shots hit (in effect, trying to determine if a player is "in the zone"). While these ideas add significant complexity to the existing models and thus are beyond our current capacity to analyze efficiently, future work with greater computational resources could consider these directions to tackle more nuanced aspects of tennis.

6.4 Adjusting Player Comparison to Mean Performance

In their original paper, Floyd et al. compare player performance to the average as one application of Expected Shot Value. They do this by selecting a specific zone for an opponent to strike from, and calculating the Return ESV for every possible zone for our target player. They do this as well for all other players, and compare the average of these ESVs to our target player to assess where their strengths and weaknesses lie.

Our models have combined the data in a new way, aggregating across players and thus defining a different concept of the "mean" that might highlight different features than the original method. By combining our process with the individualized one utilized by Floyd et al., our work possesses the potential to augment existing player analysis tools and make recommendations on which court positions and shot types are the strongest for a particular player and which can be focused on improving.

7 Conclusion

In Section 6, we demonstrate the validity of numerous extensions to the original Expected Shot Value estimator proposed by Floyd et al. By leveraging a larger pool of data and aggregating over all players, we achieve a stronger linear correlation between Expected Shot Value and win rate, as well as a relationship that is closer to 1:1 than Floyd et al. achieve. Not only does this validate the model, it shows how at scale we might be able to improve performance.

We also uncover additional extensions and underlying patterns that could be avenues for future research. These include improvements to how serves are modeled (Section 6.2) and how physicality over the course of a point and match are factored into the ESV calculations (Section 6.3). We finally show how we can use our ESVs to define the average player in a different way than Floyd et al., and thus add an additional perspective to help identify player strengths and weaknesses and guide player development strategies.

There are many aspects of the original Floyd et al. methodology that we were not able to focus on in this paper, and that are still open questions beyond those that we have posed above. The weighting systems we use remain arbitrary (except for that referenced in Section 5.7), both in terms of the default values and the size of the changes between various combinations of shot speed and spin. Future work could devote resources to better estimating these weights, and perhaps interesting findings lie in the magnitude of the differences between these weights, providing a quantitative value to the question of how much better a certain shot type is compared to another.

We recognize that this paper largely relies on existing research in its methodology and analysis, but we feel that the exploration of these modifications can yield important improvements to our understanding of tennis. We cannot change the overarching mechanics of how the game is played, but by honing in more on nuance we can recommend small changes that will maximize impact. The sport is built on a complex interaction between not only court positioning, but also shot choice, physical endurance, and mental stamina. While all of these factors cannot currently be modeled, with increasingly rich data we are starting to capture more and more detailed aspects of these interactions. We hope that our work provides a basis for how this data might be better utilized and leveraged toward this end in the future.

A Appendix

A.1 Model Weighting Systems

When reading these tables, the original strike and return categories established by Floyd et al. will comprise each column, with our modifications (and the corresponding option code) listed in each row. The weights in each cell represent the weight for the combination of these two factors. All direct winners will always have weight 1 and all direct losers will always have weight 0, and thus are excluded from these tables.

Similar to Floyd et al., we also assign our weights in a subjective fashion. Using the original set of weights as our baseline, we adjust 0.05 up or down depending on whether we judge a type of shot to be better or worse than average. The only model this does not hold for is the Non-Serve Speed + Spin Model, where we establish four different levels of shot performance.

A.1.1 Spin Model

	$\lambda_{\texttt{SUPW}}$/ $\lambda_{\texttt{RSUPW}}$	$\lambda_{\texttt{FLS}}$/ $\lambda_{\texttt{RFLS}}$	$\lambda_{\texttt{NI}}$/ $\lambda_{\texttt{R}}$	$\lambda_{\texttt{SUOPW}}$/ $\lambda_{\texttt{RSUOPW}}$
Topspin Shot (`-T`)	0.8	0.8	0.55	0.3
Backspin Shot (`-B`)	0.8	0.8	0.55	0.3
Flat Shot (`-F`)	0.7	0.7	0.45	0.2
No Spin (`-N`)	0.75	0.75	0.5	0.25

Table A.1: Strike/Return Categories and Weights

A.1.2 Spin + Dropshot Model

	$\lambda_{\texttt{SUPW}}$/ $\lambda_{\texttt{RSUPW}}$	$\lambda_{\texttt{FLS}}$/ $\lambda_{\texttt{RFLS}}$	$\lambda_{\texttt{NI}}$/ $\lambda_{\texttt{R}}$	$\lambda_{\texttt{SUOPW}}$/ $\lambda_{\texttt{RSUOPW}}$
Topspin Shot (`-T`)	0.8	0.8	0.55	0.3
Backspin Shot (`-B`)	0.8	0.8	0.55	0.3
Flat Shot (`-F`)	0.7	0.7	0.45	0.2
Drop Shot (`-D`)	0.8	0.8	0.5	0.2
No Spin (`-N`)	0.75	0.75	0.5	0.25

Table A.2: Strike/Return Categories and Weights

A.1.3 Full Speed Model

	$\lambda_{\texttt{SUPW}}$/ $\lambda_{\texttt{RSUPW}}$	$\lambda_{\texttt{FLS}}$/ $\lambda_{\texttt{RFLS}}$	$\lambda_{\texttt{NI}}$/ $\lambda_{\texttt{R}}$	$\lambda_{\texttt{SUOPW}}$/ $\lambda_{\texttt{RSUOPW}}$
First Serve, $>$ 110 MPH (-1SF)	0.8	0.8	0.55	0.3
First Serve, $<$ 110 MPH (-1SS)	0.7	0.7	0.45	0.2
Second Serve, $>$ 90 MPH (-2SF)	0.8	0.8	0.55	0.3
Second Serve, $<$ 90 MPH (-2SS)	0.7	0.7	0.45	0.2
Other Shot, $>$ 85 MPH (-F)	0.8	0.8	0.55	0.3
Other Shot, $<$ 50 MPH (-S)	0.7	0.7	0.45	0.2
Other Shot, $[50, 85]$ MPH (-A)	0.75	0.75	0.5	0.25

Table A.3: Strike/Return Categories and Weights

A.1.4 Serve Speed Model

	$\lambda_{\texttt{SUPW}}$/ $\lambda_{\texttt{RSUPW}}$	$\lambda_{\texttt{FLS}}$/ $\lambda_{\texttt{RFLS}}$	$\lambda_{\texttt{NI}}$/ $\lambda_{\texttt{R}}$	$\lambda_{\texttt{SUOPW}}$/ $\lambda_{\texttt{RSUOPW}}$
First Serve, $>$ 110 MPH (-1SF)	0.8	0.8	0.55	0.3
First Serve, $<$ 110 MPH (-1SS)	0.7	0.7	0.45	0.2
Second Serve, $>$ 90 MPH (-2SF)	0.8	0.8	0.55	0.3
Second Serve, $<$ 90 MPH (-2SS)	0.7	0.7	0.45	0.2
Other Shot (-A)	0.75	0.75	0.5	0.25

Table A.4: Strike/Return Categories and Weights

A.1.5 Full Speed + Spin Model

	$\lambda_{\texttt{SUPW}}$/ $\lambda_{\texttt{RSUPW}}$	$\lambda_{\texttt{FLS}}$/ $\lambda_{\texttt{RFLS}}$	$\lambda_{\texttt{NI}}$/ $\lambda_{\texttt{R}}$	$\lambda_{\texttt{SUOPW}}$/ $\lambda_{\texttt{RSUOPW}}$
First Serve, $>$ 110 MPH (-1SF)	0.8	0.8	0.55	0.3
First Serve, $<$ 110 MPH (-1SS)	0.7	0.7	0.45	0.2
Second Serve, $>$ 90 MPH (-2SF)	0.8	0.8	0.55	0.3
Second Serve, $<$ 90 MPH (-2SS)	0.7	0.7	0.45	0.2
Topspin Shot, $>$ 85 MPH (-TF)	0.8	0.8	0.55	0.3
Backspin Shot, $<$ 50 MPH (-BS)	0.7	0.7	0.45	0.2
Other Shot (-A)	0.75	0.75	0.5	0.25

Table A.5: Strike/Return Categories and Weights

A.1.6 Non-Serve Speed + Spin Model

	$\lambda_{\texttt{SUPW}}$/ $\lambda_{\texttt{RSUPW}}$	$\lambda_{\texttt{FLS}}$/ $\lambda_{\texttt{RFLS}}$	$\lambda_{\texttt{NI}}$/ $\lambda_{\texttt{R}}$	$\lambda_{\texttt{SUOPW}}$/ $\lambda_{\texttt{RSUOPW}}$
Topspin Shot, $>$ 85 MPH (`-TF`)	0.82	0.82	0.57	0.32
Backspin Shot, $>$ 85 MPH (`-BF`)	0.78	0.78	0.53	0.28
Topspin Shot, $<$ 50 MPH (`-TS`)	0.72	0.72	0.47	0.22
Backspin Shot, $<$ 50 MPH (`-BS`)	0.68	0.68	0.43	0.18
Other Shot (`-A`)	0.75	0.75	0.5	0.25

Table A.6: Strike/Return Categories and Weights

A.1.7 Shot Type + Spin Model

	$\lambda_{\texttt{SUPW}}$/ $\lambda_{\texttt{RSUPW}}$	$\lambda_{\texttt{FLS}}$/ $\lambda_{\texttt{RFLS}}$	$\lambda_{\texttt{NI}}$/ $\lambda_{\texttt{R}}$	$\lambda_{\texttt{SUOPW}}$/ $\lambda_{\texttt{RSUOPW}}$
Topspin Shot (`-T`)	0.8	0.8	0.55	0.3
Forehand Backspin Shot (`-C`)	0.8	0.8	0.55	0.3
Backhand Backspin Shot (`-S`)	0.7	0.7	0.45	0.2
Other Shot (`-D`)	0.75	0.75	0.5	0.25

Table A.7: Strike/Return Categories and Weights

A.1.8 Estimated Neutral Shot Weight Model

$\lambda_{\texttt{SUPW}}$	$\lambda_{\texttt{FLS}}$	$\lambda_{\texttt{NI}}$	$\lambda_{\texttt{SUOPW}}$
0.75	0.75	0.4333	0.25

Table A.8: Strike Categories and Weights

$\lambda_{\texttt{RSUPW}}$	$\lambda_{\texttt{RFLS}}$	$\lambda_{\texttt{R}}$	$\lambda_{\texttt{RSUOPW}}$
0.75	0.75	0.4262	0.25

Table A.9: Return Categories and Weights

A.2 Model Validation Plots

A.2.1 Spin Model

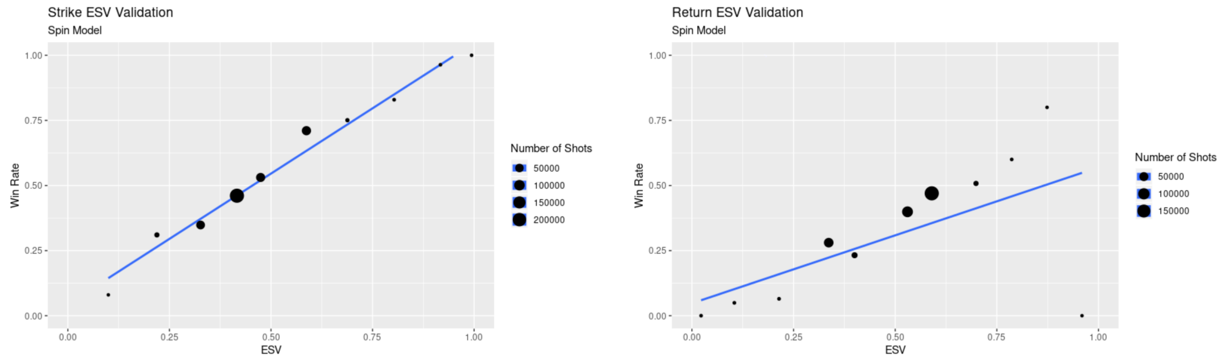

Figure A.1: Spin Model Validation

A.2.2 Spin + Dropshot Model

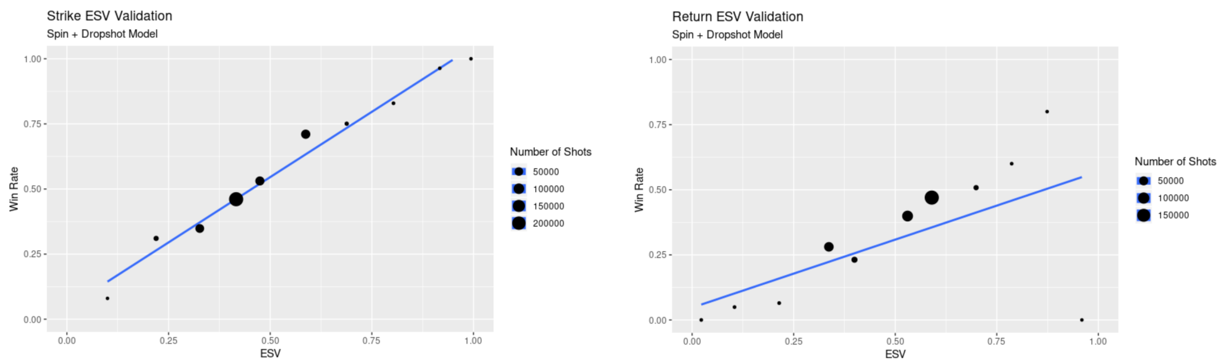

Figure A.2: Spin + Dropshot Model Validation

A.2.3 Full Speed Model

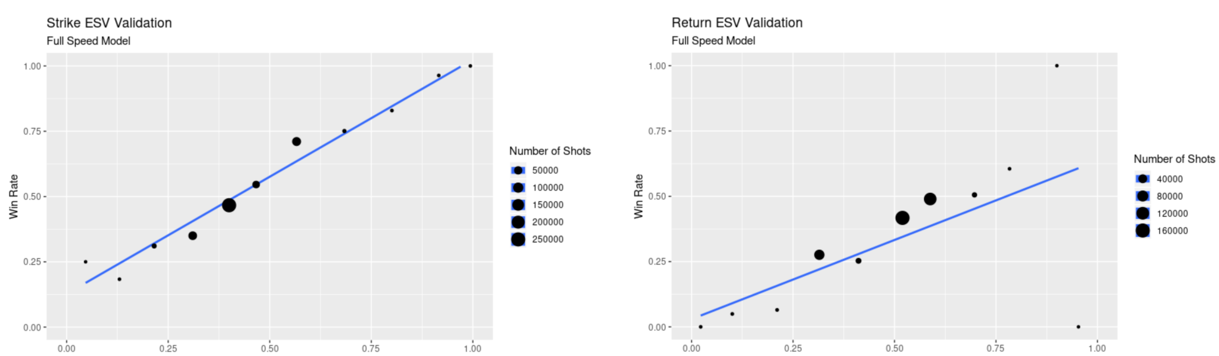

Figure A.3: Full Speed Model Validation

A.2.4 Serve Speed Model

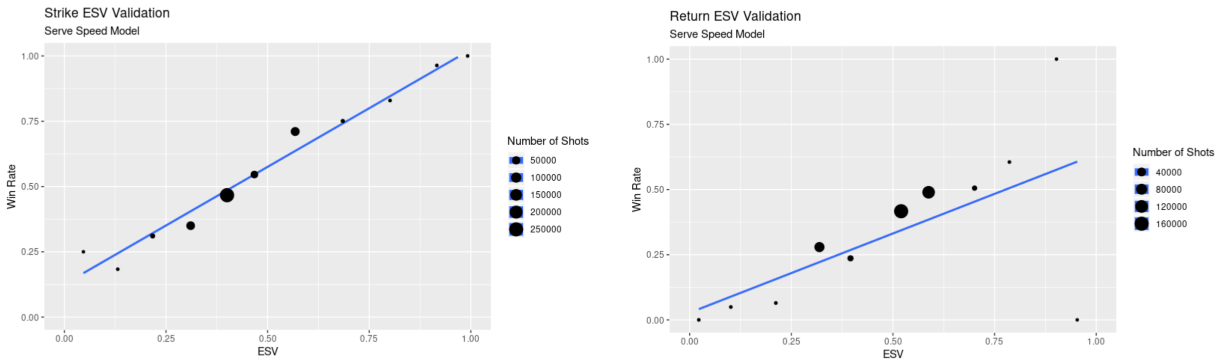

Figure A.4: Serve Speed Model Validation

A.2.5 Full Speed + Spin Model

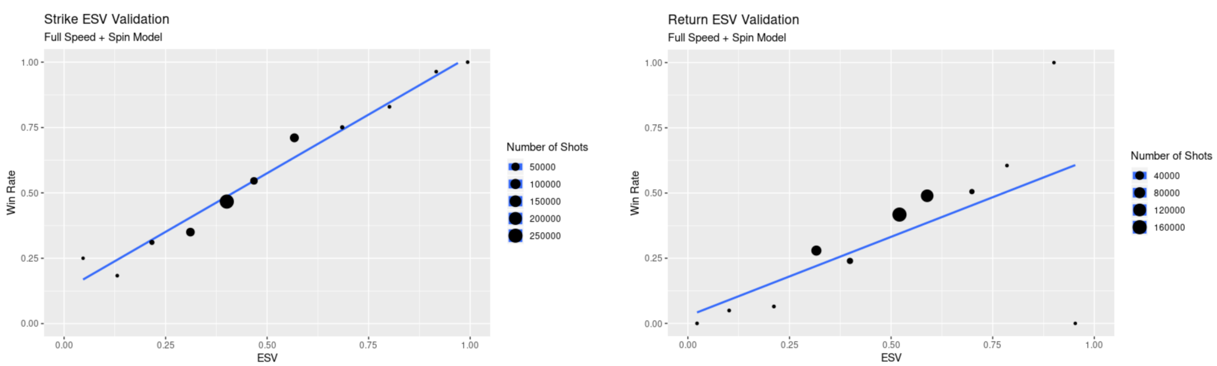

Figure A.5: Full Speed + Spin Model Validation

A.2.6 Non-Serve Speed + Spin Model

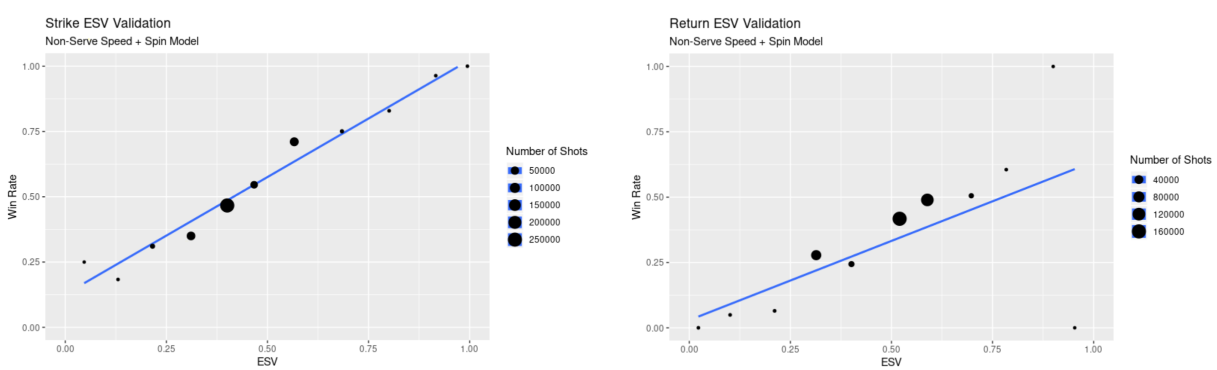

Figure A.6: Non-Serve Speed + Spin Model Validation

A.2.7 Shot Type + Spin Model

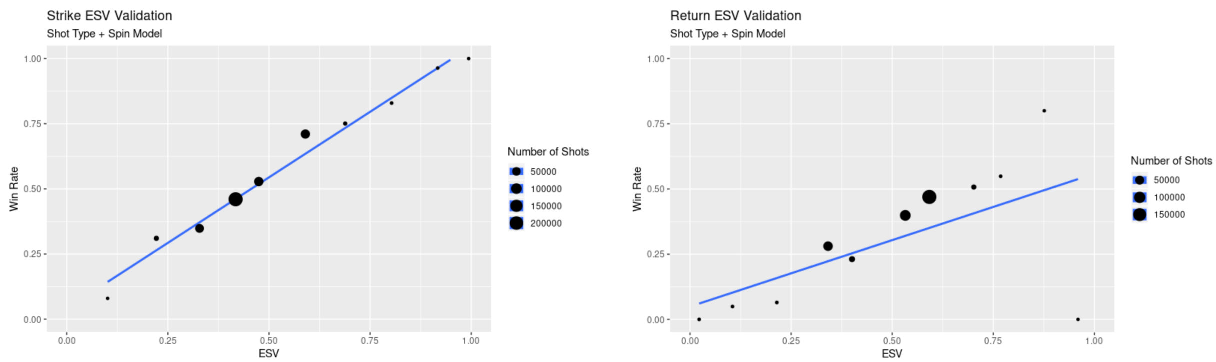

Figure A.7: Shot Type + Spin Model Validation

A.2.8 Estimated Neutral Shot Weight Model

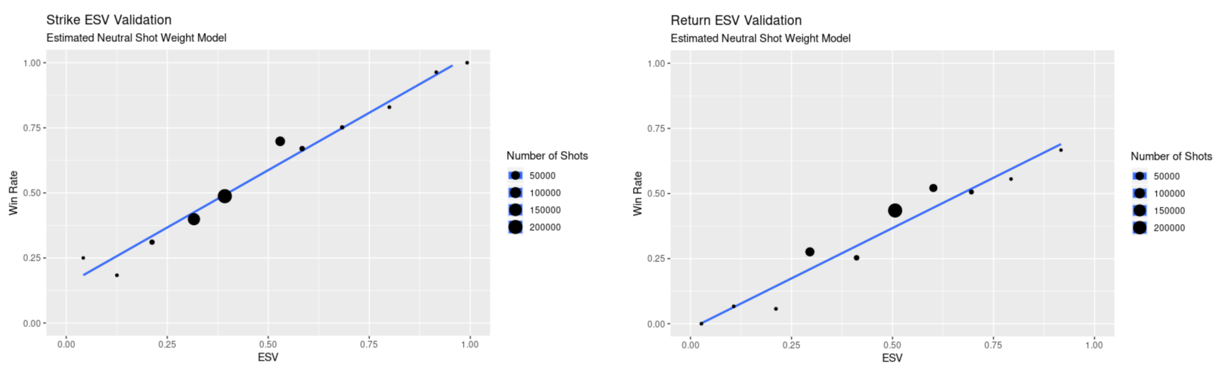

Figure A.8: Estimated Neutral Shot Weight Model Validation

References

Bevc, Martin (2015). "Predicting the Outcome of Tennis Matches From Point-by-Point Data." University of Glasgow, School of Computing Science.

Fernandez-Fernandez, J., D. Sanz-Rivas, and A. Mendez-Villanueva (2009). "A Review of the Activity Profile and Physiological Demands of Tennis Match Play." *Strength and Conditioning Journal, 31*(4), 15–26. doi: 10.1519/ssc.0b013e3181ada1cb

Floyd, Calvin Michael (2017). "Applying Multi-Resolution Stochastic Modeling to Individual Tennis Points." Thesis. Rochester Institute of Technology.

Floyd, C., M. Hoffman and E. Fokoue (2019). Shot-by-shot stochastic modeling of individual tennis points. *Journal of Quantitative Analysis in Sports*, 16(1), 57-71. Retrieved 4 Jan. 2020, from doi:10.1515/jqas-2018-0036

Mora, S. V. and W. J. Knottenbelt (2016). “Spatio-Temporal Analysis of Tennis Matches.” *KDD Workshop on Large-Scale Sports Analytics*, San Francisco, CA, USA.

Reid, M., and R. Duffield (2014). “The development of fatigue during match-play tennis.” *British Journal of Sports Medicine, 48*(Suppl 1), i7–i11. doi: 10.1136/bjsports-2013-093196

Rowland, T. W. (2014). *Tennisology: inside the science of serves, nerves, and on-court dominance*. Champaign, IL: Human Kinetics.

Spence, G., and S. Kovalchik (2019). “How Our Player DNA Scores Work.” Retrieved from https://tennismash.com/wp-content/uploads/2019/01/player_dna_method.pdf

Wei, X., P. Lucey, S. Morgan, and S. Sridharan (2013a). “Predicting Shot Locations in Tennis Using Spatiotemporal Data.” *2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. doi: 10.1109/dicta.2013.6691516

Wei, X., P. Lucey, S. Morgan, and S. Sridharan (2013b). “Sweet-Spot: Using Spatiotemporal Data to Discover and Predict Shots in Tennis.” *7th Annual MIT Sloan Sports Analytics Conference*, Boston, MA, USA.

Wei, X., P. Lucey, S. Morgan, M. Reid, and S. Sridharan (2016). “The Thin Edge of the Wedge: Accurately Predicting Shot Outcomes in Tennis using Style and Context Priors.” *10th Annual MIT Sloan Sports Analytics Conference*, Boston, MA, USA.

Wei, X., P. Lucey, S. Morgan, P. Carr, M. Reid, and S. Sridharan (2015). “Predicting Serves in Tennis using Style Priors.” *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2207–2215.