Quantifying NBA Shot Quality: A Deep Network Approach

Archit Ramanasai Kambhamettu
architk@umd.edu
University of Maryland, College Park
College Park, Maryland, USA

Abhinav Shrivastava
abhinav@cs.umd.edu
University of Maryland, College Park
College Park, Maryland, USA

Matthew Gwilliam
mgwillia@umd.edu
University of Maryland, College Park
College Park, Maryland, USA

Abstract

Since the introduction of player positional tracking data to the NBA in 2013, the field of basketball analytics has been steadily developing. As such, more and more teams utilize data-driven approaches to maximize the potential for their team to score a basket. In this paper, we explore leveraging recurrent deep-learning architectures for the quantification of the quality of a given basketball possession. To do so, we curate a dataset consisting of player positional and statistical data for basketball shots from the 2015-2016 NBA season, dividing the data into subsets of Mid-Range, 3 Pointer, Paint (Non-Restricted), Restricted Area, Backcourt, and All shots. We then explore the efficacy of three recurrent deep learning architectures: the vanilla Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), and Gated Recurrent Units (GRU) in classifying whether a shot will be made or missed, leveraging the probability of the classification to quantify the quality of the shot. Each of these models incorporates relative player distances with respect to the shooter and the basket, as well as statistical information for both the offensive and defensive players on the court. Our models achieve state-of-the-art accuracy on this task with scores of 81.7%, 81.9%, 82.1%, 81.0%, 96.3%, and 81.8% for the aforementioned data subsets respectively. Furthermore, we validate that our model's probability score is an accurate measure of shot quality by comparing our predictions with experts from the field.

CCS Concepts

• **Applied computing**; • **Computing methodologies** → **Neural networks**;

Keywords

recurrent neural networks, sports analytics, basketball analytics

ACM Reference Format:
Archit Ramanasai Kambhamettu, Abhinav Shrivastava, and Matthew Gwilliam. 2024. Quantifying NBA Shot Quality: A Deep Network Approach. In *Proceedings of the 7th ACM International Workshop on Multimedia Content Analysis in Sports (MMSports '24), October 28-November 1, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3689061.3689068

MMSports '24, October 28-November 1, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-1198-5/24/10
https://doi.org/10.1145/3689061.3689068

1 Introduction

Perhaps the single most significant element of a given basketball possession is its culmination: a shot. While the offense pours its energy into finding ways to finish their possession with a scored basket, the defense works diligently to prevent the offense from doing so. Over the course of a basketball game, hundreds of shots are put up from various positions, with varying levels of contest, yet basketball players, coaches, and analysts all seem to have differing opinions on how "good" a shot is. As there is no universal statistic that can be calculated to quantify this measure, we first define a "good" shot as one that results in a made basket, and thus reformulate the problem of measuring a shot's "goodness" as a binary classification task between a made and missed shot, with the probability that the shot is made representing the shot quality.

Measuring the quality of basketball shots requires adequate context, and we aim to provide such context by creating a dataset consisting of information from player positioning data, player statistics, and detailed shot descriptions for each shot from the 2015-2016 NBA season. Each of these shots can be directly referenced as broadcast video via the box score data from nba.com. [1].

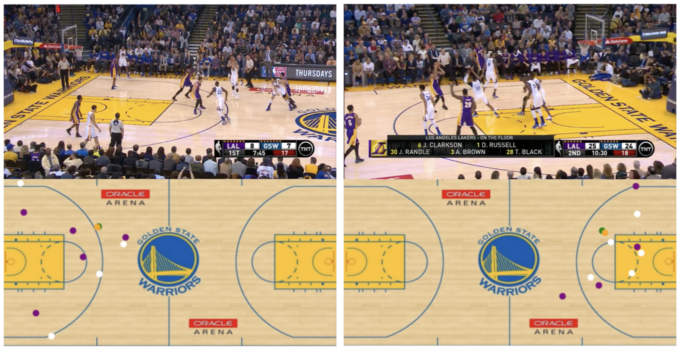
Current work done for this task of predicting shot quality relies predominately on traditional machine-learning models such as XG-Boost and Support Vector Machines [10] [14] or complex statistical modeling [1] [2]. However, considering that a basketball possession is a sequence of events, and we want to predict the last event (whether the shot is made or missed), we propose to use recurrent neural networks, which are well-suited for predicting events in sequences [7].

Thus, our contributions are two-fold – first, we curate a dataset that is suitable to quantify the quality of an NBA shot, consisting of player position detail relative to the shooter and the basket, relevant offensive/defensive statistics for each of the 10 players on the court, and detailed descriptions of each shot from the 2015-2016 NBA Season. Second, we select and tune appropriate recurrent neural network architectures for our proposed task of quantifying the quality of a basketball shot. We will release our dataset and all corresponding code upon the acceptance of our paper.

2 Related Work

The SportVu Dataset [2], which contains player identification mapped to (x,y) court positioning, has paved the way for spatiotemporal analysis for basketball-related tasks. In [15], the authors leverage the LSTM recurrent architecture coupled with the SportVU dataset to accurately classify the different play types for the Toronto Raptors. In [9], the authors utilize the locations of the players and

[1] https://www.nba.com/stats/players/boxscores
[2] https://github.com/linouk23/NBA-Player-Movements



Good Shot! (Made) Bad Shot! (Missed)

Figure 1: We assemble a dataset for training models to predict whether a basketball shot will be made or missed, based on player data and frame-by-frame coordinates. We give an example of a make and a miss, with the corresponding broadcast angles for reference.

balls for action recognition. Furthermore, with the SportVu Dataset, researchers began to take a holistic approach to quantifying the quality of a basketball possession. In [13], the authors explore the different measures that would be significant in classifying missed and made three-pointers, emphasizing the importance of including the closest defender's distance to the shooter. Although the author utilizes a gradient-boosting machine regression algorithm for his make/miss model, he emphasizes that neural architectures are an area worth exploring for this task.

In [2] and [1], the authors attempt to quantify the quality of a shot by calculating the Expected Value Possession [EVP], calculated as the expected value of points scored given player locations at timestamp t. Although effective in continuously measuring the changes in the EVP for a basketball possession in general cases, the model does not consider player statistics and thus overfits to outlier samples. Furthermore, training the complex statistical model requires several hours across hundreds of processors and over 4 terabytes of memory, which we overcome with our curated dataset and recurrent neural architectures.

In [11], the authors aim to leverage LSTM coupled with basketball trajectories from the SportVu dataset to classify whether a three-point shot would be made or missed. Their model performs well when localizing on trajectories from 2ft (0.93 AUC) to 8ft (0.843 AUC) from the basket; however, the model does not incorporate the development of the play itself, nor the abilities of the players in making predictions. Furthermore, their work is limited to 3-point shots, which we expand to all types of shots.

Two works, [10] [14], leverage traditional machine learning architectures to classify a shot as made or missed. [10] utilizes play-by-play data and limited statistics (including the closest defender, and the number of dribbles), for the classification of the shot, resulting in a maximum accuracy score of 68% with the XGBoost Model. In [14], the authors combine the SportVU dataset with a play-by-play dataset as data for their model, resulting in a maximum AUC of 0.65 with their multi-layer perception.

Apart from the research world, companies including SportPerform [3] and SecondSpectrum [4] have proprietary models that can analyze the quality of shots; however, both their data and their models are restricted to the public. Thus, to promote the development of this line of research, we will release our dataset incorporating fine-grain shot annotations, player positioning, and corresponding player statistics. Furthermore, each of these sequences can be directly referenced in video, which we take advantage of for validating our model. Additionally, as none of the aforementioned approaches incorporate the time sequence nature of player positions during a basketball possession in a deep learning context, we explore and fine-tune recurrent architectures that incorporate this into the predictions and establish baselines for this approach to the task.

3 Dataset Curation

Table 1: Features that we use for shot predictions.

l p0.34

Player Type	Features
Shooter	Offensive Box Plus-Minus, Offensive Win Shares, Player Efficiency Rating, Field Goal% (from dist), True Shooting %, Usage %, % Field Goals Attempted (from dist), Distance from Basket
Other Offense	Distance from Basket, Distance from Shooter
Defense	Blocks, Defensive Box Plus-Minus, Defensive Win Shares, Steals, Value Over Replacement, Distance from Shooter, Distance from Basket

Table 2: Sequence types along with makes, misses, total, and duration (in seconds) for each.

Shot Type	Made	Missed	Num Seq.	Seq. Len
Mid-Range	8459	12821	21280	10
Restricted Area	16060	11209	27269	10
Paint (Non-Restricted)	4978	7501	12479	10
Three Point	8287	15246	23533	10
Backcourt	7	138	145	10
Total	37791	46915	84706	10

The data collection process for our model was extensive and spanned multiple datasets, but it resulted in a large, clean, and easily testable dataset that maintained its stability over multiple iterations of the neural architecture.

3.1 SportVU Dataset

A large focus of our recurrent architectures is to understand the importance of player positions in shot making, so most of the data collection process revolves around the aforementioned SportVu Basketball Dataset, collected from a GitHub Repository. The SportVu dataset is collected from a real-time camera-based tracking system installed in every basketball arena in the NBA league, which we

[3] https://www.statsperform.com/
[4] https://www.secondspectrum.com/index.html

download for the 2015-2016 NBA season and re-purpose for our model. The dataset holds important metadata about every player on the court such as their name and unique player ID, along with their x-y position on the court. This information is repeated every time an updated position is tracked, which occurs every four-hundredths of a second; however, upon further analysis, we find that the 60-gigabyte dataset is extremely redundant. Thus, to remove this redundancy, we opt to only include data from time points closest to the nearest whole second, significantly reducing the size and removing any redundancy.

As the SportVu dataset contains x and y locations in an unknown coordinate system, which would not necessarily correspond to higher or lower signals in neural architecture, we instead represent the player locations with relative distance metrics. Thus, we represent the shooter's location as his distance to the basket and represent the other 9 players' locations as the distance from the basket and the distance to the shooter. As the SportVu dataset does not have the basket location as part of the dataset, we empirically estimate the coordinates of the hoop in the SportVu coordinate system, which remains constant for all games, by calculating the average position of the ball following a made shot (given by the play-by-play dataset in Section 3.2).

3.2 NBA Play by Play

Although the SportVU dataset gives us access to player movement data for every second of an NBA game, we are only interested in the movement of players leading up to a shot attempt. However, because the SportVu dataset does not contain any gameplay annotations, we merge our SportVU dataset with a play-by-play shot-detail dataset from the official NBA stats website, collected by a third party on Kaggle [5]. These descriptions include the shot type, distance from the basket, and whether or not the shot was a success. It also contains detailed descriptions of the nature of the shot (eg. Turnaround Fadeaway, Layup). Although not used for the scope of this paper, we believe including this information could foster further analysis of the different types of shots. In total, there are four main types of shots: Mid-Range, 3-pointer, Paint (Non-Restricted), Restricted Area, and Backcourt. We focus both on quantifying the quality of a shot within each subset and between all shots combined.

In order to maintain the sequential nature of the dataset, where every ten timestamps correspond to one sequence, we duplicate the information from the play-by-play dataset for every timestamp and merge it with the aforementioned information extracted from the SportVu dataset. Thus, we now have a dataset that incorporates the players' spatial information on the court corresponding with detailed shot descriptions.

3.3 NBA Player Stats

While player movements alone can be quite revealing when considering the success of a play, they do not consider what kinds of players are on the court at the time. Thus, we enhance our model's ability to learn much more nuanced information by introducing additional player-specific statistics for both the shooter and every defensive player on the court. We collect offensive [6] and defensive [7] player statistics and merge them with our player movement dataset by utilizing an ID conversion database [8] that ensures parity between players on official NBA datasets and basketball-reference datasets. Table 1 details the statistics included for each of the types of players on the court. As noted, both the Field Goal % and %Field Goals Attempted are specific to the distance that the player is shooting from, as players shoot better from different ranges. Including these statistics provides the model with additional insight into the unique skill set of each player for a particular shot, which is crucial for the proper evaluation of the quality of a given possession.

[5]https://www.kaggle.com/datasets/brains14482/nba-playbyplay-and-shotdetails-data-19962021

Table 3: Prediction accuracy by model for each shot type.

Shot Type	League Average	RNN	LSTM	GRU
3PT Shot	0.354	0.786	0.816	0.819
Mid Range	0.398	0.787	0.809	0.817
Restricted Area	0.603	0.809	0.810	0.809
Paint (Non Restricted)	0.400	0.806	0.805	0.821
Backcourt	0.027	0.897	0.963	0.931
All	0.452	0.811	0.815	0.818

3.4 Final Dataset

Our dataset curation process culminates in a complete set of 84,706 unique NBA events, each containing a 10-second time series of distance data and offensive/defensive stats for a total of 53 features every second. Table 2 further summarizes the data distribution among the shot types. We strongly believe the creation of this dataset can lead to additional insight into the development of successful NBA plays. Along with the player statistics and positional embedding, we also map each available play to the corresponding video feed, which we utilize in our model validation.

4 Experiments

4.1 Training Details

As our dataset consists of time-series sequences, we opt to utilize the RNN [12], LSTM [5], and GRU [3] architectures. Due to the imbalanced nature of our dataset, we leverage a weighted random sampler to provide higher sampling weights to the minority class. This technique increases the probability of sampling instances from the minority class for each batch to ensure a better representation of the sample space for the model to train on. Each of the models leverages the adaptive momentum optimizer (ADAM) [6] during training, and to further counteract the imbalance in the dataset, we use a Weighted Binary Cross Entropy loss function [4] [8]:

$$\text{WBCE}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left[w_0 y_i \log(\hat{y}_i) + (1-w_0)(1-y_i)\log(1-\hat{y}_i)\right] \tag{1}$$

(where $w_0 = \frac{1}{\text{\# shot made class}}$). This method assigns a higher penalty for misclassifying instances from the minority class, improving the model's ability to accurately predict these underrepresented

[6]https://www.basketball-reference.com/leagues/NBA_2016_shooting.html
[7]https://www.basketball-reference.com/leagues/NBA_2016_advanced.html
[8]https://github.com/djblechn-su/nba-player-team-ids

User Prediction:	Made Shot (3/5)	Made Shot (4/5)	Made Shot (3/5)
Model Prediction:	Missed Shot (0.2166)	Made Shot (0.5354)	Missed Shot (0.4231)
Ground Truth:	Missed Shot	Missed Shot	Made Shot

User Prediction:	Made Shot (5/5)	Missed Shot (0/5)	Made Shot (4/5)
Model Prediction:	Made Shot (0.9261)	Missed Shot (0.0873)	Made Shot (0.9653)
Ground Truth:	Made Shot	Missed Shot	Made Shot

Figure 2: Ground Truth vs. Deep model predictions vs. basketball enthusiast predictions, with broadcast views for reference.

categories. We use a training-validation split of 75-25 when training the deep learning architectures, with a batch size of 64, and as aforementioned, we opt to utilize a sequential length of 10 for each of the models, corresponding to 10 seconds before the shot was put up.

Our vanilla RNN model comprises three layers with a hidden size of 16: two tanh layers, and one sigmoid to detail the probability the shot is made, which we train using a learning rate of 0.001 over 100 epochs. Our LSTM model incorporates an LSTM layer with a hidden state followed by one fully connected ReLU layer and one fully connected sigmoid layer, both of size 8, which we train with a learning rate of 0.0012 over 100 epochs. Our GRU model comprises a GRU layer with a size of 16 followed by a fully connected softmax layer, which we train with a learning rate of 0.0003 also over 100 epochs.

All of the models were trained on 4 RTX A4000 GPUs for approximately two hours in total, and the architectures remained consistent across all the varying shot types.

4.2 Results

Depicted in Table 3 is a detailed breakdown of the accuracy levels for each of the aforementioned tasks in comparison to League Average Percentages [9] [10]. The recurrent model can classify whether the shots will be made or missed with approximately 81% accuracy, which is a significant improvement over any baseline.

As the motivation is more than to perform a simple binary classification of sequential data, we feel that a simple accuracy metric does not encapsulate the complexity of the task. Thus, we conduct a survey to evaluate the efficacy of our model in comparison to the human-standard baseline. To do so, we randomly select 11 separate 10-second broadcast footage of NBA gameplay leading up to a shot and ask five basketball experts, who currently work in the basketball industry, to determine whether the given shot will be made or missed. Of the 11 clips, our shot-zone independent model corresponds with the users on 7 of them and corresponds with the ground truth on 8 of the clips. The users correspond with the ground truth on 8 of the 11 clips as well. Thus, the model performs well in comparison to both the user predictions and the ground truth. Of the 11 clips, we extract the last frame of 6 such clips and display them in Figure 1, along with the corresponding User Prediction, Model Prediction, and Ground Truth. The rightmost image of the first row of Figure 1, demonstrates an example that our model understands the complexity of the shot and can output a probability that has a proper meaning for whether the quality of the shot is high or low, as the model prediction is high for a negative class.

Furthermore, shots that are nearly guaranteed to be missed according to the users (shown in the center image of the second row of Figure 1) also have a very low probability of being made according to the model, and likewise, shots with a are nearly guaranteed to be made according to the users (shown in the leftmost image of the second-row Figure 1) also have a very high probability of being made according to the model.

5 Conclusion

In this paper, we present a formidable approach to quantifying the quality of an NBA shot via recurrent neural architectures and also introduce a new dataset that combines various data sources into

[9]https://www.basketball-reference.com/leagues/NBA_stats_per_game.html
[10]https://www.nba.com/stats/teams/shooting

one. Furthermore, we validate our model's exact probability prediction via an evaluation of our model against experts in basketball analytics.

Future Work. Now that we have successfully utilized recurrent architectures for the quantification of the quality of a basketball shot, we find an exciting opportunity to generalize this model to play across all levels. One such avenue would be to translate broadcast basketball game footage coordinates into SportVu coordinates, and coupled with the proper statistics, utilize our model to analyze shot quality on any basketball game. Furthermore, our dataset promotes research involving analyzing the different types of shot postures, whether qualitative measures such as end-of-game pressure can be quantified, and other avenues for advanced basketball analytics.

References

[1] Daniel Cervone, Luke Bornn, and Kirk Goldsberry. 2014. POINTWISE: Predicting Points and Valuing Decisions in Real Time with NBA Optical Tracking Data. https://api.semanticscholar.org/CorpusID:1222592

[2] Daniel Cervone, Alex D'Amour, Luke Bornn, and Kirk Goldsberry. 2016. A Multiresolution Stochastic Process Model for Predicting Basketball Possession Outcomes. *J. Amer. Statist. Assoc.* 111, 514 (April 2016), 585–599. https://doi.org/10.1080/01621459.2016.1141685

[3] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:1412.3555 [cs.NE] https://arxiv.org/abs/1412.3555

[4] Yaoshiang Ho and Samuel Wookey. 2020. The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling. *IEEE Access* 8 (2020), 4806–4813. https://doi.org/10.1109/access.2019.2962617

[5] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. 9, 8 (nov 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[6] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[7] Zachary C. Lipton, John Berkowitz, and Charles Elkan. 2015. A Critical Review of Recurrent Neural Networks for Sequence Learning. arXiv:1506.00019 [cs.LG] https://arxiv.org/abs/1506.00019

[8] Minghao Liu, Haiyi Liu, Sirui Zhao, Fei Ma, Minglei Li, Zonghong Dai, Hao Wang, Tong Xu, and Enhong Chen. 2023. STAN: Spatial-Temporal Awareness Network for Temporal Action Detection. In *Proceedings of the 6th International Workshop on Multimedia Content Analysis in Sports* (Ottawa ON, Canada) *(MMSports '23)*. Association for Computing Machinery, New York, NY, USA, 161–165. https://doi.org/10.1145/3606038.3616169

[9] Akhil Nistala and John V. Guttag. 2019. Using Deep Learning to Understand Patterns of Player Movement in the NBA. https://api.semanticscholar.org/CorpusID:212413773

[10] Maram Shikh Oughali, Mariah Bahloul, and Sahar A. El Rahman. 2019. Analysis of NBA Players and Shot Prediction Using Random Forest and XGBoost Models. In *2019 International Conference on Computer and Information Sciences (ICCIS)*. 1–5. https://doi.org/10.1109/ICCISci.2019.8716412

[11] Rajiv Shah and Rob Romijnders. 2016. Applying Deep Learning to Basketball Trajectories. arXiv:1608.03793 [cs.NE] https://arxiv.org/abs/1608.03793

[12] Alex Sherstinsky. 2020. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena* 404 (March 2020), 132306. https://doi.org/10.1016/j.physd.2019.132306

[13] Bradley A. Sliz. 2017. An Investigation of Three-point Shooting through an Analysis of NBA Player Tracking Data. arXiv:1703.07030 [stat.AP] https://arxiv.org/abs/1703.07030

[14] Lisa Ann Yu Sriraman Madhavan, Ezra Van Negri. 2017. Predicting NBA Basketball Shot Success Using Neural Networks. https://cs229.stanford.edu/proj2017/final-reports/5218669.pdf. CS 229 Machine Learning Final Project, Stanford University.

[15] Richard Zemel, Jonathan Tompson, James Fan, and Steve Perkins. 2014. Classifying NBA Offensive Plays Using Neural Networks. https://www.cs.toronto.edu/~zemel/documents/1536-Classifying-NBA-Offensive-Plays-Using-Neural-Networks.pdf