Explainable Injury Prediction Onset of NBA Players

Alexandre Santos
Universidade de Coimbra
alexandres@student.dei.uc.pt

Pedro Sá
Universidade de Coimbra
pedronuno@student.dei.uc.pt

***Abstract*—Notwithstanding the recent advances in physical therapy and conditioning, a significant portion of athletes still end up suffering from injuries, which largely impacts in-game audiences and the franchises themselves [2]. Machine Learning (ML) is a driving force for predicting injuries, as traditional methods fail to capture high-order interactions between the several factors that contribute to injury.**

We provide an approach to injury prediction modelling with a special focus on interpretable and explainable models, considering that knowing which factors have more predictive power can be key to prevent athletes from suffering injuries on the short and long-term.

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Despite all the advances in physical therapy and conditioning of athletes in the NBA, there is still a rather frequent onset of injuries that affect the players' performance and consequently the teams' performance. These injuries to relevant players on their team have a big impact on the size of in-game audiences, and therefore on the franchises' revenue. For context, in the 2013/2014 season, the number of games missed by players due to injury cost the NBA a total of $357.9 million [2]. This cost is highly related with the NBA becoming a player's league back in the 80s and 90s with the appearance of Michael Jordan [5], therefore player conditioning is vital.

A. Problem Statement

Current methods (such as player load monitoring) of injury risk assessment are not meaningful, as they fail to represent the complexity of interactions across several factors, and only analyze part of the problem at hand [3]. With the recent growth and widespread use of analytics in basketball, Artificial Intelligence (AI) and Machine Learning (ML) play a vital role in forecasting potential injuries. This modern approach provides significant insight for decision-makers, as this information can be leveraged by NBA organizations to optimize training regimes, load management strategies, and other business-related tasks – such as player performance projection and growth [1].

B. Goals

The objectives of this work are twofold: (i) to present an efficient method for predicting injury based on historical records and game statistics and (ii) to provide interpretability characteristics to this predictions. We believe these to be key factors to understand which factors have more predictive power and have stronger contributions to the risk of sustaining injury. Furthermore, it may motivate teams organizations and corresponding health staff to share more data and bring the full potential of the prediction of ML models.

As such, we make the following contributions:

- We experiment different feature engineering techniques to improve results;
- We benchmark a set of baseline algorithms and a well-known state-of-the-art method – XGBoost;
- We present interpretable predictions with SHAP;
- We provide an analysis of relevant factors for sustaining injury.

II. Related Work

[7], [9] present a comprehensive review on state-of-the-art approaches to injury prediction in sports, stating the common usage of Decision Trees, Ensembles and Support Vector Machines (SVM) for predictive models and suggesting injury history, distance covered and days between consecutive games as strong injury predictors.

In [4], the authors propose a time series modelling of the training load of 1 and 3 weeks prior to injury, for which a XGBoost model was able to predict a sizable portion of injuries, suggesting that activity in the week prior to an injury has the highest predictive power.

Moreover, in [3], the authors follow a deep learning approach by proposing Multi-layer bidirectional Encoder Transformers for Injury Classification (METIC). Results suggest superior predictive capability over baseline classifiers, whereas a comparison against state-of-the-art models is lacking.

III. Methodology

The following section presents the methodology adopted for tackling the aforementioned problem. We will describe the data used and approach we followed.

A. Data

The data collected was organized in the following categories:

- **Inactive list** - Contains the NBA names players that are not eligible to play a particular game due to a recent event. A player could be inactive for the whole season, or a certain match, depending on the decision of the coaching staff and medical team;
- **Biographical** - player's age, weight in pounds and height in inches;

- **Statistical** - Rebounds, blocks, average speed, distance travelled (in miles), contested shots, contested 2-point shots, contested 3-point shots, deflections, charges drawn, box-outs, games played and minutes played. Some of these stats, in addition to the full game stat, have the defensive or offensive focused stat.

Through the use of the NBA api provided by swar in GitHub [8], we were able to retrieve the biographical and statistical information of each player from the *nba.com* [6] site. This tool has the capability to let the user define the granularity of the requested data in the parameters of the function. This granularity could define specifically the stretch of the season (regular, post, before all-star break or after all-tar break) and then define the interval analysis (month or last N games, N being a max of 15 games). The other source is the *prosportstransactions* website which contains information of the inactive list in the NBA, although for this source there was no existence of a tool to obtain the data in an usable format. Therefore, a web scrapping method was successfully implemented for that purpose.

B. Approach

Our approach is five-fold: i) to aggregate the maximum available data ii) to clean data as to perform a reliable feature extraction iii) to manipulate the data in order to uncover data representations that effect a model's predictive capabilities iv) to experiment different algorithms as to find methods to provide better results v) to produce explanations about predictions.

a) Data Aggregation: Some statistical data requests returned only one player, which would make the retrieval of information very time-consuming. In addition, some stats were only available in website from a certain point of time. For example, the request of the contested shots information was only available from the 2015/16 season included. Furthermore, some requests had different depth of granularity making the aggregation of the data more difficult. Due to these contraints, the interval of 2015/16 and 2021/22 season was selected as the timeframe for all datasets and the statistical data is by season as the level of granularity. Having the availability of the data checked with the aid of glossary of stats provided by the *nba.com* [6], a better understanding of the features was done. With this step accomplished we move on to the cleaning section.

b) Data Cleaning:

- **Standardized player's names**: the inactive list had some entries in the column name with a alias, nickname, the full or the standard name of a player. In contrast, the statistical data had already the standard name independent of the type of request. Therefore, we formatted the player's names in the inactive list to be the same as the ones retrieved from the api in order to cross the data.
- **Standardized team names**: the teams names retrieved from the API were in the abbreviated form as for the inactive list, but with the mascot's names. Also, some teams along the years had a change of the name, for example in the end of the 2011/12 season the New Jersey Nets changed to Brooklyn Nets.

c) Data Manipulation: Based on prior knowledge, we use a clustering method – Kmeans – on a subset of features to extract a new feature describing the playstyle of a player (more offensive, more defensive). Moreover, we (separately) gaussianize and discretize the feature space with *scikit-learn's* **QuantileTransformer** and **KBinsDiscretizer**.

d) Algorithms: Based on contextual research on the topic, we use the baseline methods of SVM, Decision Tree and Random Forest, and XGBoost which is known for achieving state-of-the-art performance on tabular data.

e) Interpretability: For interpretability, we select the top performing model on all cases and produce explanations with SHAP. The SHAP values of a model describe the contribution to the model output, and help understand how it makes decisions.

IV. Experimental Work

The following section details the experimental work conducted. For every experiment we split the respective dataset into a training and testing set following a $\frac{80}{20}$ ratio.

A. Experiment .1 – Data manipulation

We perform several transformations on the original data and produce 3 additional datasets, represented in table I. The **FE** version is the result of performing feature engineering for extracting the *playstyle* feature and replacing it for the features used for the clustering. The **QT** version is the result of guassianizing the raw dataset, with all features. The **BIN** version consists of the raw dataset with a discretized feature space, with all features.

TABLE I
DATASET VERSIONS

Dataset	Description	# features
Raw	Raw dataset	1
FE	Raw w/ feature engineering	1
QT	Raw w/ gaussianized features	1
BIN	Raw w/ discretized features	1

For every dataset we run a grid-search on the XGBoost algorithm and evaluate in terms of accuracy on the test set. The parameter search space is detailed in Table II and the attained results are in Table III.

TABLE II
HYPERPARAMETER SEARCH SPACE FOR XGBOOST ALGORITHM

Parameter	Range	Best
No. estimators	$\{50, 75, 100\}$	50
Max. depth	$[2, 5[$	4
Booster	{gbtree, dart}	gbtree
Learning Rate	$\{0.1, 0.5, 1\}$	0.1

We found that XGBoost actually performs best when dealing with a discretized feature space (**BIN**), but at the expense of lack of expressivity of the model and the data: this is because tree-based models are able to deal with this condition

TABLE III
XGBoost accuracy on validation (+ std.) and test sets for different dataset versions

Dataset	Val	Test
Raw	0.673 ± 0.009	0.653
FE	0.6746 ± 0.009	0.662
QT	0.6726 ± 0.01	0.653
BIN	0.6795 ± 0.006	0.652

fairly well, but the gain in performance (+0.002 of accuracy) doesn't justify the lack of cardinality on the data, which is specially relevant in this context. As such, we consider the **FE** version to be the overall best.

B. Experiment .2 – Benchmark

We select the most suitable dataset version, which is the **FE**, fix the optimal parameters for XGBoost and compare it with other baseline methods. These baseline approaches we're not used in any of the past experiments and we do not perform hyper-parameter searching for them. Results are evaluated in terms of accuracy on the test set, for which the results are displayed in Table IV.

TABLE IV
Benchmark on baseline methods vs. fine-tuned XGBoost

Method	Test Accuracy ↑
Decision Tree	0.553
SVM	0.646
Random Forest	0.649
XGBoost	0.63
XGBoost (fine-tuned)	0.662

We understand there is a considerable competitive advantage for the fine-tuned XGBoost algorithm considering it was fine-tuned and the dataset version chosen was specific to it, not the other methods. Nonetheless, we observe that it produces better results than its baseline counterparts, while the off-the-self XGBoost performs worse than the SVM and Random Forest. Despite that, we argue that gain in performance (≈ 0.03) justifies the time it takes to fine tune the model ($\approx 3min$), instead of resorting to off-the-shelf methods.

C. Interpretability

Finally, we select the top-performing model and couple SHAP explanations. Figure 1 provides an overview on global interpretability of the model, for which the features are sorted in function of importance (more to less important).

The first finding is that a high **GP** (the number of games played) *mostly* drives the model to classify non-injury. This seems counter-intuitive as one would guess that a higher number of games played aggravates the risk of sustaining injuries. Figure 2 compares the effect of **GP** on the model output in the context of minutes played (**MIN**). We observe that a high number of games played does drive the model to classify non-injury, but it also happens to be that the number of minutes played also increase. For an athlete that plays less games, it's actually better if he plays more minutes, and for an athlete that averages $[30, 50]$ games, it's better if he plays less minutes. This suggests the presence of dynamics intrinsic to the practice of the sport itself, which heavily relies on consistency. A player that performs a lot of games is conditioning the body to better sustain physical effort.

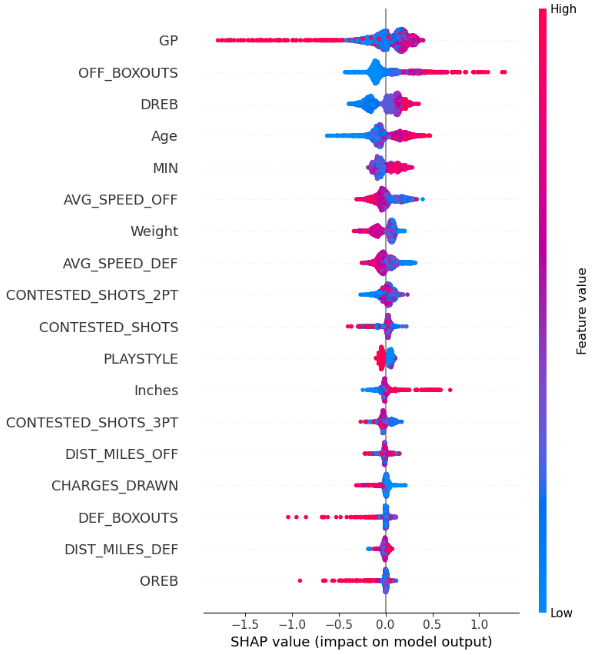

Fig. 1. SHAP summary plot of explanations

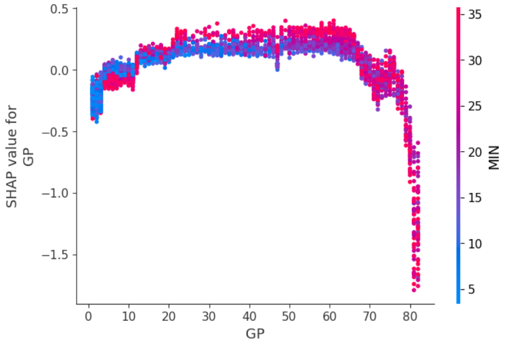

Fig. 2. SHAP dependence plot on GP (games played) and MIN (minutes played)

The second finding is that **OFF_BOXOUTS** (offensive box-outs), **DREB** (defensive rebounds), **Age**, **MIN** and **Inches** represent relevant markers for injury prediction. For these features, a higher value drives the model to classify injury. We observe that younger, shorter players, and players that (on average) play less minutes are less likely to incur in injury. Moreover, player that seldom execute offensive box-outs and defensive rebounds are also likely to sustain injury, and vice-versa. The impact these features have on model prediction are also considerate.

The third finding is that **PLAYSTYLE**, the engineered feature, has a negligible effect on the model output. We observe that a low value (defensive playstyle) drives the model to predict injury, whilst a high value (offensive playstyle) drives the model the opposite way. The defensive playstlye is more often than not very physic-centric, accounting for the most part of rebounds and personal fouls. Nonetheless, this feature should be taken with a grain of salt, as the extrapolation of the feature doesn't rely on hardcoded prior knowledge: we only declare the features that we belive describe best a playstyle, and let the clustering method model the data. The method can be (and most likely is) incongruent.

V. Conclusion and Future Work

In this work, we dive into the problem of injury prediction and analysis on NBA players. We use a clustering method for feature engineering data describing an athlete playstyle. We use different out-of-the-box methods and a more robust one – XGBoost – to create a classifier model, for which we were able to achieve a respectful performance. Moreover, we couple explanations to the model predictions and describe some findings we made about the data, the model and the sport itself.

One of the issues with building a solution for the problem at question is the predisposition of most teams organizations to share data. Geospatial activity metrics and biometric evaluations taken regularly by teams would have a positive impact in the prediction power of models positively. Furthermore, they also obscure player injuries if feasible to maintain a competitive advantage.

For future work we mean to have a more thorough collection of the available of data. Specifically, the addition of more interest features like the number of drives, post-ups and contested rebounds in order to add a more comprehensive feature set, which hopefully entails a more clear explainability component. We also intend to explore and analyze the annotations information of the inactive list with the goal of bringing more discoveries to the work and also possible new features. Furthermore, we want to improve the granularity of the dataset by aggregating the players stats by season and month. These latter enhancements could bring more features, motivate a more in-depth analysis, and ultimately improve the predictive power of our model.

References

[1] How data analytics is revolutionizing the NBA.

[2] The 2013-14 NBA Season Injury Review, July 2014.

[3] Alexander Cohan, Jake Schuster, and Jose Fernandez. A deep learning approach to injury forecasting in nba basketball. *Journal of Sports Analytics*, (Preprint):1–12, 2021.

[4] S Sofie Lövdal, Ruud JR Den Hartigh, and George Azzopardi. Injury prediction in competitive runners with machine learning. *International journal of sports physiology and performance*, 16(10):1522–1531, 2021.

[5] Matt Moore Mar 21. Everything you need to know about the nba's rest controversy, including solutions, Mar 2017.

[6] nba. Nba official stats glossary.

[7] Alessio Rossi, Luca Pappalardo, and Paolo Cintia. A narrative review for a machine learning application in sports: an example based on injury forecasting in soccer. *Sports*, 10(1):5, 2021.

[8] Swar. An api client package to access the api for nba.com.

[9] Hans Van Eetvelde, Luciana D Mendonça, Christophe Ley, Romain Seil, and Thomas Tischer. Machine learning methods in sport injury prediction and prevention: a systematic review. *Journal of experimental orthopaedics*, 8(1):1–15, 2021.