



# Predicting Tennis Match Outcomes with Network Analysis and Machine Learning

Firas Bayram<sup>(✉)</sup> , Davide Garbarino, and Annalisa Barla 

DIBRIS, Università di Genova, Genova, Italy  
S4712149@studenti.unige.it, davide.garbarino@edu.unige.it,  
annalisa.barla@unige.it

**Abstract.** Singles tennis is one of the most popular individual sports in the world. Many researchers have embarked on a wide range of approaches to model a tennis match, using probabilistic modeling, or applying machine learning models to predict the outcome of matches. In this paper, we propose a novel approach based on network analysis to infer a surface-specific and time-varying score for professional tennis players and use it in addition to players' statistics of previous matches to represent tennis match data. Using the resulting features, we apply advanced machine learning paradigms such as Multi-Output Regression and Learning Using Privileged Information, and compare the results with standard machine learning approaches. The models are trained and tested on more than 83,000 men's singles tennis matches between the years 1991 and 2020. Evaluating the results shows the proposed methods provide more accurate predictions of tennis match outcome than classical approaches and outperform the existing methods in the literature and the current state-of-the-art models in tennis.

**Keywords:** Machine learning · Network analysis · Learning Using Privileged Information · Multi-Output Regression · Tennis outcome prediction

## 1 Introduction

### 1.1 Motivation

The sports industry is one of the most growing business sectors in the world. According to the Business Research Company, the global sports market reached a value of nearly \$488 billion in 2018, having grown at an annual growth rate of more than 4% since 2014, and is expected to reach almost \$614 billion by 2022. As we live in the age of data and analytics, this steady growth rate for the sports market size has motivated many researchers to conduct studies on sports data analytics where in sport competitions a result can convey a great deal on different aspects involved in sports like the volume of fans retention, television contracts or sponsorship deals. Essentially, sports data analytics is exploited by either the sports teams directly or by sports gambling stocks. One primary

technique of predictive analytics is to build machine learning models to generate predictions for upcoming events and matches using historical player data and statistics.

Several leading male tennis professionals like Roger Federer, Novak Djokovic and Andy Murray have realized the importance of data analytics in tennis, so they introduced data analytics specialists into their teams to help them better prepare for tournaments. They scrutinize and analyze their opponents' key skills and tactics to make use of those insights to avail themselves of the opportunity to boost their chances of winning matches. Accurate prediction of the outcome of tennis matches has an impact on advising players of their odds so they can adjust their plans according to the forecast of the match.

## 1.2 Related Work

Complex network techniques have been applied to represent the network of tennis matches [9]. The majority of approaches were to rank the players in tennis history taking into account a global view of the player's performance throughout his career and compare it to the existing system that ATP is currently following, which is to rank tennis players based on the immediate past 52 weeks. The most notable work on tennis network modeling was done by Radicchi [21] where the author determined the best players on specific playing surface and proposed a ranking algorithm *Prestige Score*, that is analogous to PageRank score [6], to quantify the importance of tennis players and concluded that the prestige score is more accurate and has higher predictive power than ranking schemes adopted in professional tennis. Michieli [17] applied multiple ranking algorithms to see how active tennis players have improved their overall prestige over the recent years and compared the results of the ranking methods used with the ATP Ranking and identified *Jimmy Connors* as the best player in history up to 2017. Breznik [5] identified the best left and right-handed players in tennis history applying network analytic methods and the PageRank algorithm.

For tennis match prediction, most existing approaches to tennis prediction apply statistical models to tennis matches, starting from the hierarchical structure of the sport's scoring system under the assumption that points in tennis are independently and identically distributed or i.i.d, Klaassen and Magnus [11] show that the assumption is false but they find that deviations from i.i.d. are small and hence the i.i.d. assumption provides a reasonable approximation. O'Malley [18] and, Klaassen and Magnus [12], are illustrative examples of such models. Knottenbelt et al. [13] improved the hierarchical model that is based on the probability of winning an individual point by exploiting statistics from matches played against common opponents. In recent years, machine learning models have been utilized to predict the winner of a tennis match by representing the match and the player by a set of features instead of a single value, player's features are derived from historical match statistics. Ma et al. [16] applied logistic regression model on 16 variables representing player skills and performance, player characteristics and match characteristics. Sipko and Knottenbelt [22] have extracted more detailed set of features and applied logistic regression and artificial neural

network to predict the outcome of a tennis match, their best model ANN resulted in a log loss of 0.6111. Peters and Murray [20] addressed the effects of surface type and the variation of player skills through time by using free parameters to represent the skills of players and the characteristics of court surfaces.

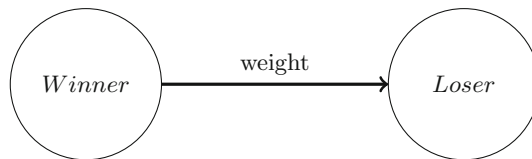
In this paper, we define a new method based on network analysis to extract a new feature that represent the player’s skill on each surface considering the variation of his performance over time which is believed to have a big effect on the match outcome. We also make use of the match statistics directly in the prediction through applying advanced ML paradigms instead of only following the historical averaging process that was done in the literature. This project uses data obtained from the ATP official website which is the main resource for historical data of tennis matches since 1968. Each match is represented by 49 features including, for instance, player’s age and ranking, the number of aces, double faults and 1<sup>st</sup> serve percentage.

## 2 Network Modeling and Surface-Specific Score

In this section, we describe the method followed to extract the surface-specific score.

### 2.1 Network of Tennis Matches

We mapped tennis matches into a weighted and directed graph: edges are directed from winner to loser and they are weighted according to the stage and type of tournament as shown in Fig. 1. The ATP has four tiers of events— Grand Slams, Masters 1000, ATP 500 and ATP 250. With the four Grand Slams awarding the most points, 2000. The numbers in each tournament category represent how many points the winner receives. For clarity, we simplified the numbers that we used to assign weights to the edges to hold the proportion between the values as follows: ATP 250/500 : 1, Masters 1000: 2, ATP Finals: 3, Grand Slams: 4. In case of multiple links of the same direction exist between the players, the weights are summed up.



**Fig. 1.** A single tennis match represented in a directed graph representation

## 2.2 Surface-Specific Score

It is evident in tennis that players' performances are affected by the court surface, Barnet and Pollard [1] showed that the type of surfaces favors those players who are best suited to this particular surface. The dataset in hand does not include any information about players' skills on each surface. Therefore, in order to quantify the latter, we subset the graph based on surface and all prior tennis matches, we compute several centrality measures for each player and, by means of PCA, we evaluate a surface-specific score. Centrality measures are a widely used analysis mechanism to reveal important elements of complex networks [8]. Note that all the used measures are normalized.

**CM1 Out-In-degree-difference Centrality:** The in-degree of a player  $v$  ( $d_{in}(v)$ ) is the sum of the edge weights of his losing matches, while out-degree ( $d_{out}(v)$ ) is the sum of the edge weights of his winning matches. The Out-In-degree-difference centrality measure (CM1) of each node is computed as the difference between its in-degree and out-degree.

**CM2 Hubs Centrality:** The hub score of a node estimates how many highly authoritative nodes this node is pointing to; in the tennis players network, Micheli [17] demonstrated that good hubs are often associated with successful players because they have won against a wide range of players while the authorities are modest players with long careers.

**CM3 PageRank Centrality:** PageRank centrality [19] is a spectral centrality measure. The algorithm assigns a centrality score based on its neighbors score. PageRank acknowledges that not all wins are equal, wins over strong opponents weigh more than beating mediocre players. PageRank score of player  $i$  is computed as following:

$$P_i = \frac{d}{N} + (1 - d) \sum_j P_j \left( \frac{w_{ji}}{k_j} + \frac{\delta(k_j)}{N} \right) \quad (1)$$

Where  $d = 0.15$  is the damping factor,  $N$  is the number of players,  $w_{ji}$  is the edge weight between nodes  $i$  and  $j$ ,  $k_j$  and  $P_j$  are the out-degree and PageRank value of the node  $j$ , respectively and  $\delta$  is a function to correct the sinks (nodes with outdegree zero).

Finally, to estimate the surface-specific score, we followed Algorithm 1, based on the Principal Component Analysis (PCA) technique. We refer to the resulting first principal component scores as *surface-score*. These values can be interpreted as time-varying surface-specific scores. Table 1 and Table 2 show the top 5 scores on Clay and Hard surfaces respectively, as of June 2020.

## 3 Tennis Match Representation

In this section, we discuss how we processed the raw data to generate the features that, in addition to the surface-specific score, are used as input to the ML models.

---

**Algorithm 1:** Surface-specific score extraction algorithm

---

**Input:** Graph of matches on specific surface  $G^{(s)}$ , Centrality measure functions  $C$

**Output:** Vector of surface-specific scores  $y^{(s)}$

```

for  $i = 1$  to  $N$  do                               /* For each node in graph  $y^{(s)}$  */
    for  $c = 1$  to  $M$  do                             /* For each function  $C$  */
        |  $CM(c) = C(i)$                           /* Calculate CM vector for node  $P(i)$  */
    end
    |  $y^{(s)}(i) = PCA(CM)$                         /* Apply PCA to vector  $CM$  */
end
return  $(y^{(s)})$ 

```

---

### 3.1 Player’s Features

There are two types of features in our dataset with respect to their availability time. Unlike some features which are available before the start of the match, such as the age and rank of the players, the statistics are only available after the end of the match. Therefore, player’s skills are estimated by taking the average of the statistics of his historical matches for an upcoming match.

### 3.2 Labelling - Experimental Design

The raw data classifies the players as a winner and loser, whereas before the match takes place, we only have players labelled as Player 1 and Player 2. Thus, we randomly sampled the data and assigned Player 1 to be a *winner* or *loser*. The match outcome for match  $n$  can be defined as following:

$$y_n = \begin{cases} 1, & \text{if Player 1 is the winner.} \\ 0, & \text{if Player 1 is the loser.} \end{cases} \tag{2}$$

### 3.3 Symmetric Representation

Inspired by Sipko and Knottenbelt [22], and O’Malley [2], to achieve the symmetry of model’s prediction outcome regardless of the random labeling discussed

**Table 1.** Top 5 scores on clay, as of June 2020      **Table 2.** Top 5 scores on hard, as of June 2020

Player	Surface-score
Rafael Nadal	0.539565
Novak Djokovic	0.262964
Roger Federer	0.252947
David Ferrer	0.206311
Guillermo Vilas	0.150491

Player	Surface-score
Roger Federer	0.627906
Novak Djokovic	0.566555
Rafael Nadal	0.375329
Andy Murray	0.320519
Andre Agassi	0.23004

in the previous section. We took the difference between the players' features of the same characteristic in order to obtain identical results even if we swap the labelling of Player 1 and Player 2. Also, in this way, even supposing that we reverse the labels of Player 1 and Player 2 we are going to get the same feature values but with different sign, and different target class. This would also help us avoid any bias to the same feature of both players due to assigning different weight of a feature for Player 1 than Player 2 by the model. For example, the model might give a higher weight for Player 1 rank than to Player 2 rank. Using the difference of variables to extract the features halves the number of dimensions of the dataset. The difference of the variables is calculated based on the labeling criterion defined in Equ. 2 as follows:

$$FEATURE_i = STAT_{i,p1} - STAT_{i,p2} \quad (3)$$

## 4 Machine Learning Methods

We think of a tennis match as a vector  $x_i$  composed of  $P$  input features. The corresponding match outcome  $y_i$  may be a *win* 1 or a *loss* 0.

As for the supervised classification methods, we resorted to four methods. Here we briefly describe the approaches and how we used them to predict the tennis match outcome.

**RF Random Forests.** Random Forests is an ensemble technique that combines bagging and random feature sub-spacing. Random Forests algorithm constructs a large collection of classification or regression ensembles of independent decision trees (forests) and aggregates their predictions by averaging [2].

**LR Logistic Regression.** Logistic regression exploits the logistic sigmoid function as loss function to estimate the probability of the sample being assigned to one of the two possible classes [10].

**LUPI Learning Using Privileged Information.** Learning Using Privileged Information, or Support Vector Machine using Privileged Information (SVM+), has been first proposed by Vapnik and Vashist [24]. LUPI is an advanced learning paradigm that uses additional (privileged) information that is available only for the training examples and not available for test examples. This additional information (prior knowledge) can be exploited to build better models and improve the results. In tennis, and sports in general, the match statistics are only available in the training examples as they are collected during the match and the final stats sheet is published after the end of the match. On the other hand, for the testing examples, we only have the match characteristics and players' profiles, thus we utilized LUPI paradigm to leverage the match statistics that are considered to be as additional information we have for the training examples to improve the performance of the learning method. In LUPI framework, we are given the training triplets:

$$\{(x_1, x_1^*, y_1), \dots, (x_n, x_n^*, y_n)\}, x_i \in X, x_i^* \in X^*, y_i \in \{0, 1\}, i = 1, 2, \dots, n$$

where  $X$  is the space of match features vector  $x$ ,  $x^*$  represents the match statistics vector that belongs to the correcting space  $X^*$  (the space of privileged information) and  $y_i$  is the labels vector. Supporting the learning process by incorporating the match statistics in the model can capture the complexity of the training examples by discovering hidden patterns between the players that cannot be discovered in the original features space  $X$ . There are many examples in the tennis world where Player 1 has better estimates than Player 2- this is going to be reflected by positive match features and SVM might map the match data point  $x$  to the *winning* side of the margin in space  $X$ . But still, if Player 1 struggles against Player 2 under specific conditions or due to his play style, this is going to be evident in the statistics of the matches between the two players, hence the additional information  $x^*$  that belongs to the same point  $x$  might fall in the *losing* side of the margin in space  $X^*$ . Therefore, the match statistics can facilitate the learning process by tightening or relaxing the SVM constraints to improve the predictions by including the privileged information that we have about the training matches.

**MTR Multi-Target Regression.** Multi-target regression, also known as multi-output regression [3], is an advanced machine learning paradigm that involves predicting simultaneously two or more numerical output variables given the same input features. We utilized MTR approach to predict the statistics of the players in the match and then consequently predict the winner of the match applying classification model. Figure 2 shows the workflow diagram of implementing the MTR paradigm to predict the winner of the tennis match. The workflow diagram shows that the MTR phase works as a ‘black box’ to the match outcome prediction since it will only help the classification model better predict the winner of the match and the MTR prediction accuracy in itself is irrelevant to the final evaluation of our learning classification task. The most common approach to deal with multi-target regression problems is *problem transformation* by transforming the multi-target problem into multiple independent single-target problems each one solved by fitting a regressor to make a single-output regression for each target. The other approach is *algorithm adaptation* methods that modify a single-output method to simultaneously support multi-output problems, this is usually done by modeling the dependencies among these targets.

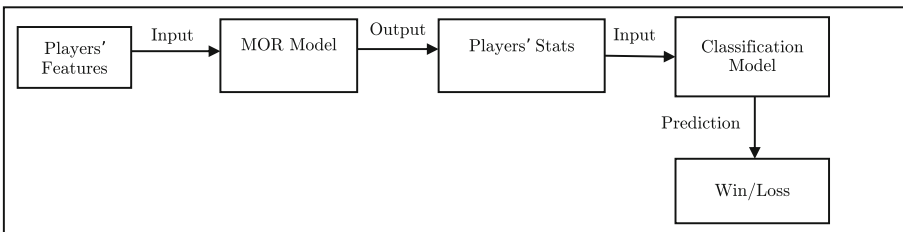
We briefly describe four different approaches to MTR that we adopted in our classification pipeline:

- MTR Multi Single-Target Regressors:** we decompose the problem to  $d$  single-target regression problems by fitting a random forest regressor to independently predict each target [3].
- MTR-RC Regressor Chains:** RC [23] is a *problem transfer* method and is built on the idea of linking chains of regressors and stacking predictions to other models of the chain as additional features. The training procedure of RC involves selecting a chain that is represented by an ordered set of target variables  $C = \{y_1, y_2 \dots y_d\}$ . To predict the first

target value  $y_1$ , we trained a random forest regressor on the original input vector  $X$  of the training dataset. Then, for the subsequent targets  $y_j$  where  $j \in \{2, \dots, d\}$  we perform transformation on the training dataset to consist of the union of the original input vector  $X$  and the actual value of the previous targets in the chain  $y_{k < j}$  in the original training dataset.

**MTR-RF Multi-Target Random Forest Regressor:** this method [4, 14] is an extension of the single-target random forest regressor; the only difference is the modification of the calculation of the impurity measure of a node as the sum of squared error over the multi-target value.

**MTR-TSF Multi-Target Regression Via Target Specific Features:** this method, proposed by Wang et al. [25], deals with the multi-target regression (MTR) tasks by learning target specific features (TSF). The method assigns a cluster index to each match features vector  $X_i$  using hierarchical clustering algorithm. The index is then added to expand the feature space  $X_{exp} = X \cup X_{index}$ . Target specific features are learned by querying a corresponding dependent similarity matrix, generated by a classification and regression tree boosting method (CART-boosting)[7]. The transformed training dataset for the  $j$ th match statistics target  $Y_j$  is constructed by finding the union  $\hat{D}_J = X \cup X_{index} \cup X_{TSF}$ .



**Fig. 2.** Multi-output regression workflow diagram

## 5 Experiments

The experiments were conducted on a node hosted on DLTM<sup>1</sup>, the node is equipped with an Intel Xeon CPU with  $27 \times 2.3$  GHz, 64 GB RAM.

<sup>1</sup> <https://www.dlrm.it/>.



## 5.1 Dataset Splitting

For the standard ML models, the training dataset consists of 62,141 tennis matches between the years 1991 and 2011, and 21,083 matches between 2012 and 2020 for testing, that makes up approximately 75:25 ratio. For the advanced ML paradigms and because of their complexity, we reduced the dataset size to consist of 39,033 tennis matches between the years 2001 and 2014 for training dataset, and 13,011 matches between 2015 and 2020 for testing, that also makes up approximately 75:25 ratio. For each dataset, we used the tennis matches played in the last three years of the training set for validation.

This way of dividing the tennis dataset by complete years is widely adopted in the literature. The main reason is to maintain the temporal order of tennis matches. Shuffling the data randomly would result in testing the model on older matches than the ones used for training, which is not legitimate. For example, it would be futile to predict the winner of a tennis match played in 2006 using machine learning model trained on data that include tennis matches played in 2016. Moreover, ATP has a fixed calendar of certain tournaments played in specific weeks of the season. Splitting the matches by complete years allows us to have match samples of each tournament over the different dataset splits. In other words, we have match examples of each tournament distributed in the training, validation and test sets, which makes the learning process more feasible.

## 5.2 Classical Machine Learning Models Results

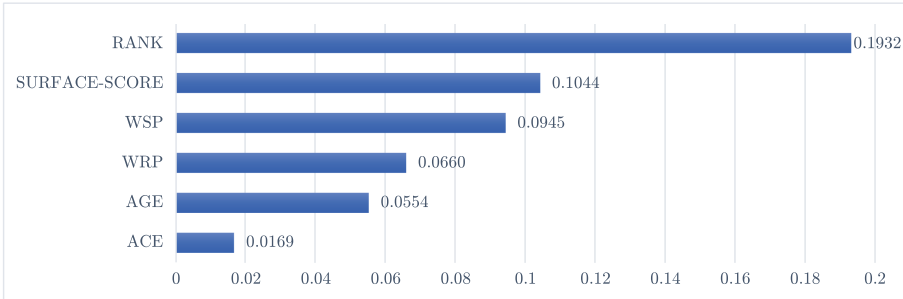
Before fitting our model to make the final predictions, we perform feature selection step using sequential backward selection method to select the best features of the dataset. Since the approach requires setting the number of features to be selected a priori, we tuned this parameter and evaluated the performance of the selected subset of the features using the validation set. For RF model, the optimal number of features is 6, while for LR model, the feature selection approach resulted in no improvement in the accuracy while evaluating on the validation set. Table 3 compares the results of RF and LR models when evaluated on the testing set using accuracy and log loss classification as metrics. Random forests algorithm outperformed logistic regression when implemented with feature selection step and without. When comparing the prediction results between the two algorithms using various classification metrics, random forest

**Table 3.** Random forests with and without feature selection (RF with FS and RF without FS respectively) and logistic regression results in terms of log-loss and accuracy.

Model	Log-loss	Accuracy
RF without FS	0.6095	0.6681
RF with FS	0.5996*	0.6729*
LR	0.6110	0.6663

with feature selection resulted in higher prediction accuracy, and also improved the uncertainty of the predictions measured by the log-loss.

To inspect the impact of the surface-score feature on the prediction results, we used the Shapley Value [15] as shown in Fig. 3. To calculate Shapley Values of each feature in the set, a model is trained with that feature present, and another model is trained with the feature withheld. Then, predictions from the two models are compared on the current input. We see that the **RANK** variable is the most important feature in making the predictions, then **SURFACE-SCORE** feature that we inferred based on network analysis.



**Fig. 3.** Average impacts (in absolute terms) of features on model output magnitude, WSP is winning on serve percentage, WRP is winning on return percentage

### 5.3 SVM and SVM+ Results

Table 4 provides a summary of the accuracy results of applying SVM+ using different kernel function combinations in decision and correcting spaces. We also report the accuracy results of the classical SVM algorithm using the corresponding kernel functions. Since we have used different years range as discussed in Sect. 5.1, and for the sake of comparing, we applied the classical machine learning models on the same dataset. The standard ML models, RF and LR, are trained on the original features of the dataset without using the match statistics, named *RF-OF* and *LR-OF* respectively. We can see the improvement in the models' performance when leveraging the match statistics as privileged information to correct the decision function. Also, considering a RBF kernel has provided the highest accuracy percentage using both SVM and SVM+ models. Furthermore, the models in LUPI framework have superior predicting performance when compared to RF and LR. SVM+ improved the classification accuracy results of both models.

### 5.4 Multi-Output Regression Results

To predict the statistics of the match, we fitted several multi-output regression models and used the mean squared error metric to evaluate the performance of

**Table 4.** Accuracy results of SVM+ with different kernels. SVM, random forest and logistic regression are applied on the original features

Decision space kernel	Correcting space kernel			Original features		
	Linear	RBF	Sigmoid	SVM	RF-OF	LR-OF
RBF	0.66159	0.6612	0.6622*	0.655		
Linear	0.64822	0.64927	0.6561	0.64768	0.657	0.654
Sigmoid	0.6526	0.6503	0.6607	0.6305		

each model. Table 5 reports the predictive results for both regression and classification phases for the different models. We can notice the big impact of the output of the regression phase on the final classification predictions. Improving the mean squared error of the MTR models by  $10^{-3}$  leads to obtain higher accuracy results by 1.5%. Comparing the performance of the different MTR approaches, we can see that the models which consider the dependency between the features have better results than the ones that neglect it. This supports the hypothesis that tennis match statistics are correlated and modeling this relationship between the features is a powerful technique that should be followed to achieve higher prediction accuracy. Specifically, multi-target regression via specific target features (MTR-TSF) has had the best regression performance. Consequently as a result, the classifier that was built on the MTR-TSF predictions as an input has the highest accuracy results compared with the other MTR models.

**Table 5.** Regression and classification results of multi-target regression via specific target features (MTR-TSF), multi single-target regression (MTSR), multi-target regression via regressor chain (MTR-RC), Multi-target random forest regressor(MTR-RF), random forest on the original features (RF-OF), and Logistic regression on the original feature (LR-OF)

Approach	Regression phase						Classification phase
	Mean squared error for each target						
	M-ACES	M-DFS	M-W1S	M-W2S	M-WSP	M-WSG	Accuracy
MTR-TSF	26.63	8.29	0.0191	0.0281	0.0162	0.0610	0.666*
MTSR	27.60	8.71	0.0204	0.0297	0.0173	0.0647	0.65
MTR-RC	26.39	8.34	0.020	0.0295	0.0173	0.0665	0.65
MTR-RF	27.27	8.54	0.198	0.0291	0.0168	0.0631	0.649
RF-OF	-						0.657
LR-OF	-						0.654

## 6 Conclusions

### 6.1 Contribution

In this paper, we utilized network analysis techniques and applied advanced machine learning paradigms to improve the current state-of-the-art approaches to predict the winner of a tennis match. We developed a novel method by representing the tennis matches as a network to infer a time-varying and surface-specific score that evaluate the player's performance on a specific court surface at a certain time point. We made use of the extracted score to enhance our dataset and added it to the features set that contains estimations of players' qualities based on the historical matches. We demonstrated that the extracted score has a relevant influence on the prediction results and was ranked as the second most important feature in making predictions of our classification task.

We made use of advanced machine learning paradigms (LUPI and MTR) which resulted in more accurate results when compared to the classical machine learning models. By using these advanced methods, we improved the prediction accuracy results of the classification task by 1.5%. We do emphasize that the proposed methods can outperform the current state-of-the-art models, and can be even generalized to other sports that have a similar data structure, i.e., where the match statistics are only available for training and not available for testing. The advanced paradigms leverage the match statistics directly in making predictions instead of solely using statistics estimators (for example historical average) as per usual when applying the classical machine learning models.

### 6.2 Limitations

Implementing the advanced machine learning paradigm is associated with additional cost and complexity. This cost in both the resources and the execution time slows down the optimization and hyperparameters tuning process to select the best model that produces the optimal performance and results. Building the kernel matrices for LUPI, and generating the dependent similarity matrix for MTR-TSF, are the most time-consuming and costly phases of the models.

### 6.3 Future Work

The limitations of the approaches described in the previous section prompt us to think about models that are memory and run-time efficient. Online learning would be a natural candidate. In online learning paradigm, the model is quickly updated to produce the best model as the data arrive in a sequential order. Thus, there is no need to re-train the model whenever a new data point arrives which is too expensive. Online learning is an option worth exploring in predicting tennis match outcomes as it has a rich literature.

Our dataset only includes the totals of winning points on serve and return. In fact, there is a plenty of other aspects in the game of tennis that differentiate

between the players' qualities and skills. Acquiring more statistics, such as winners and unforced errors records, head-to-head results on a specific surface, or success rate in winning points at the net, can improve the models' performance. It would also be useful to model the non-numerical factors and convert them into numbers to include them in the dataset, such as the player's current form or favoring specific events which can play a part in helping predict the winner of the match.

## References

1. Barnett, T., Pollard, G.: How the tennis court surface affects player performance and injuries. *Med. Sci. Tennis* **12**(1), 34–37 (2007)
2. Biau, G., Scornet, E.: A random forest guided tour. *Test* **25**(2), 197–227 (2016). <https://doi.org/10.1007/s11749-016-0481-7>
3. Borchani, H., Varando, G., Bielza, C., Larrañaga, P.: A survey on multi-output regression. *Wiley Interdisc. Rev. Data Min. Knowl. Discovery* **5**(5), 216–233 (2015)
4. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
5. Breznik, K.: On the gender effects of handedness in professional tennis. *J. Sports Sci. Med.* **12**(2), 346 (2013)
6. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine (1998)
7. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794 (2016)
8. Das, K., Samanta, S., Pal, M.: Study on centrality measures in social networks: a survey. *Soc. Netw. Anal. Min.* **8**(1), 1–11 (2018). <https://doi.org/10.1007/s13278-018-0493-2>
9. Dingle, N., Knottenbelt, W., Spanias, D.: On the (Page) ranking of professional tennis players. In: Tribastone, M., Gilmore, S. (eds.) *EPEW 2012*. LNCS, vol. 7587, pp. 237–247. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-36781-6\\_17](https://doi.org/10.1007/978-3-642-36781-6_17)
10. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York (2009)
11. Klaassen, F.J., Magnus, J.R.: Are points in tennis independent and identically distributed? evidence from a dynamic binary panel data model. *J. Am. Stat. Assoc.* **96**(454), 500–509 (2001)
12. Klaassen, F.J., Magnus, J.R.: Forecasting the winner of a tennis match. *Eur. J. Oper. Res.* **148**(2), 257–267 (2003)
13. Knottenbelt, W.J., Spanias, D., Madurska, A.M.: A common-opponent stochastic model for predicting the outcome of professional tennis matches. *Comput. Math. Appl.* **64**(12), 3820–3827 (2012)
14. Levatić, J., Ceci, M., Kocev, D., Džeroski, S.: Semi-supervised learning for multi-target regression. In: Appice, A., Ceci, M., Loglisci, C., Manco, G., Masciari, E., Ras, Z.W. (eds.) *NFMCP 2014*. LNCS (LNAI), vol. 8983, pp. 3–18. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-17876-9\\_1](https://doi.org/10.1007/978-3-319-17876-9_1)
15. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems*, pp. 4765–4774 (2017)

16. Ma, S.M., Liu, C.C., Tan, Y., Ma, S.C.: Winning matches in grand slam men's singles: an analysis of player performance-related variables from 1991 to 2008. *J. Sports Sci.* **31**(11), 1147–1155 (2013)
17. Michieli, U.: Complex network analysis of men single atp tennis matches. arXiv preprint [arXiv:1804.08138](https://arxiv.org/abs/1804.08138) (2018)
18. O'Malley, A.J.: Probability formulas and statistical analysis in tennis. *J. Quant. Anal. Sports* **4**(2), 1-23 (2008)
19. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report, Stanford InfoLab (1999)
20. Peters, J.: Predicting the outcomes of professional tennis matches (2017)
21. Radicchi, F.: Who is the best player ever? a complex network analysis of the history of professional tennis. *PLoS ONE* **6**(2), e17249 (2011)
22. Sipko, M., Knottenbelt, W.: Machine learning for the prediction of professional tennis matches. MEng computing-final year project, Imperial College London (2015)
23. Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., Vlahavas, I.: Multi-label classification methods for multi-target regression. [arXiv:1211.6581](https://arxiv.org/abs/1211.6581) (2012)
24. Vapnik, V., Vashist, A.: A new learning paradigm: learning using privileged information. *Neural Netw.* **22**(5–6), 544–557 (2009)
25. Wang, J., Chen, Z., Sun, K., Li, H., Deng, X.: Multi-target regression via target specific features. *Knowl. Based Syst.* **170**, 70–78 (2019)