Supervised Learning for Table Tennis Match Prediction

Journal:	*IEEE Transactions on Artificial Intelligence*
Manuscript ID	TAI-2022-Jun-L-00320
Manuscript Type:	Letter
Date Submitted by the Author:	02-Jun-2022
Complete List of Authors:	Chiang, Sophie; Perse School, DENES, GYORGY; Perse School, Computer Science
Keywords:	Artificial intelligence in games, Supervised learning, Machine learning

JOURNAL OF IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, VOL. 00, NO. 0, MONTH 2022

Supervised Learning for Table Tennis Match Prediction

Yu-hung Sophie Chiang, and Gyorgy Denes

***Abstract*—Machine learning, classification and prediction models have applications across a range of fields. Sport analytics is one such increasingly popular field, but most existing work focuses on automated refereeing and injury prevention in mainstream sports. Research on other sports, such as table tennis, have only recently started gaining more traction. In this paper, we aim to predict the outcome of table tennis matches; we valuate a range of existing supervised machine learning models to find a reliable predictor, trained on historical player and match statistics. We also derive 12 features and demonstrate their utility in an ablation study. We found that logistic regression performed best on the test data, although differences between models were not always significant. Our results match the accuracy of state-of-the-art in comparable sports, such as tennis.**

***Impact Statement*—The use of supervised learning models has been proposed for outcome prediction in a number of sports; however, there has been little research done for table tennis. This paper extends the application field in this direction, rigorously evaluating existing models. Furthermore, we also present a set of additional features that increase model accuracy, as validated in the ablation study. The results can serve as a baseline for future table tennis prediction models, and can feed back to prediction research in similar ball sports.**

I. Introduction

Table tennis is a quick and highly technical sport, requiring players to respond to an incoming ball trajectory within milliseconds. Rallies are intense, with ball speeds of 60–70mph and rotational speeds of 9000rpm. The proximity between players is a lot lower compared to similar sports such as tennis and badminton. The outcome of a game can be influenced by subtle factors, which can be hard for a human to recognize.

Machine learning methods have been used frequently in other sports, such as tennis and football. Proposed applications involve improved training efficiency as well as result prediction. Specifically, result prediction is of high concern to sport fans, but little has been done in table tennis prediction.

While manually-collected datasets alongside some analysis have been available in the past [1], it is only recent developments in multi-class event spotting and small object tracking that made accurate, detailed in-game data attainable. In this paper, we propose using some of this freshly available data to train and evaluate state-of-the-art classification algorithms on both men and women's professional singles matches.

The paper is organized as follows: we first review relevant publications on sports prediction and describe the OSAI dataset on which our work is built. Then, we describe the proposed feature set. Finally, we evaluate the performance of these state-of-the-art models and perform a feature ablation.

Manuscript submitted for review on June 2nd 2022
Yu-hung Sophie Chiang and Gyorgy Denes are with the Perse School, Hills Rd, Cambridge CB2 8QF, UK (correspondence email: gdenes@perse.co.uk)

II. Background

Table tennis is played by hundreds of millions of people world-wide, with almost 40 000 professionals registered with the International Table Tennis Federation (ITTF). Games are fast-paced and highly technical, but there are factors that make it attractive for mathematical modelling: matches have only two possible outcomes (there are no draws), and as an individual sport, there is no need to consider team line-ups.

A table tennis match consists of a sequence of sets; in a professional singles match, the first player to win best of seven sets wins the match. This paper will be looking at modelling professional singles matches only. In a set, the first player that earns at least eleven points and at least two more than their opponent wins the set. Each player serves twice before alternating, however, if the score reaches at least 10-10, each player serves only once before alternating. The full set of rules are published by the ITTF [2].

III. Related Work

A. Machine Learning

Machine learning (ML) is a branch of artificial intelligence that has been successfully applied to many areas of industry and science, including disease diagnosis in medicine [3], pattern recognition [4], computer vision [5] and bioinformatics [6]. The problem of predicting a table tennis match can be thought of as a supervised binary classification problem, with unambiguous ground-truth match outcome labels widely available.

B. ML in Sports

In the past, manual data collection methods for sports have typically proven time-consuming and prone to human error and bias. Recent improvements in data capture has sparked interest in automatic data collection and analysis for a range of sports. Xing et al. [7] proposed a dual-mode two-way Bayesian inference approach to track multiple highly dynamic and interactive players from videos in team sports such as basketball, football and hockey. Claudino et al. [8] used different ML methods, such as neural networks and decision tree classifiers, to investigate injury risk and performance in football, basketball, handball and volleyball. Davoodi and Khanteymoori [9] used neural networks for horse racing prediction, where eight features were used as input nodes to each neural network. This included information such as horse weight and race distance, to predict the eventual finishing time and rank of every horse in a race.

Applications of ML in sports can help players and performance analysts in identifying critical factors that contribute to winning. Appropriate tactics can be identified in maximising player performance. Aside from formulating strategies to win matches, using machine learning methods for sport result prediction has become popular due to the expanding domain in betting [10], which necessitates high predictive accuracy. Other applications include automated scouting and recruitment [10] and umpiring assistance [11].

C. Prediction in Tennis

A number of data-driven models are available for *tennis*. Clarke and Dyte [12] predicted the outcome of professional tennis matches with 61–69% accuracy using player rating points. Mapping player ability to a single rank can fail to capture complex factors, especially when comparing lower-rated players. In our paper we consider more complex features.

Barnett and Clarke [13] use rich historical data to predict the probability of a player winning a single point, building up a Markov chain to predict the winner of a match. The authors' approach is compelling, but there is no published data on the accuracy of their model on a larger dataset.

Knottenbelt et al. [14] proposed a common opponent model to find a pre-play estimate of winning a match. This was achieved by analysing match statistics for opponents that both players encountered in the past. The model computed the probability of each player winning a point on their serve, and hence the match. The authors found a 59–77% accuracy, with an estimated return on investment of 6.85% when put into the betting market for over four major tennis tournaments in 2011.

D. ML in Table tennis

In table tennis, ML research has focused so far on computer vision and automated data collection. Voeikov et al. [15] proposed a neural network (TTNet) that allowed for real-time processing of high-resolution table tennis videos. They extract temporal and spatial data, such as ball detection and in-game events, potentially replacing manual data collection by sport scouts. The model can also assist referees. More recently, Zhang et al. [16] used computer vision to allow a robot to play table tennis. They computed the 3D coordinates of a table tennis ball from a pair of video feeds to estimate the trajectory, the landing and striking point.

We build on these existing works by utilising data by Voeikov et al. [15], and apply it to the yet unexplored problem of supervised table tennis match prediction.

IV. Dataset

Our primary source of data are automatic captures from TTNet [15], released by OSAI [17]. We use Tokyo 2020 Olympics and Tischtennis-Bundesliga (German table tennis league) data, which include men and women's singles matches. Potential features include player rank and in-match statistics

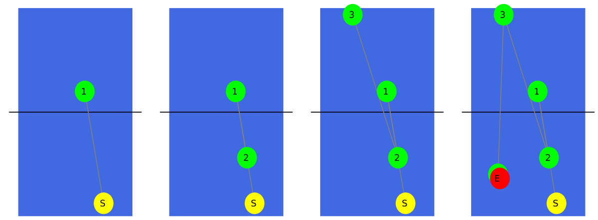

Fig. 1. Progression of a rally demonstrating the landing point of each ball bounce. Yellow indicates service which starts a rally and red indicates an error ending the rally. Green indicates all other ball bounces. Based on data from OSAI [17].

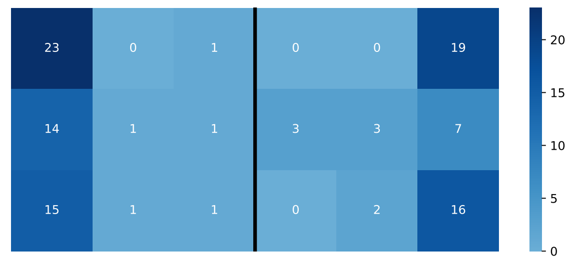

Fig. 2. Location of the last bounce of the winning ball, summed over an example match. Each side of the table is split into nine equal parts.

such as percentage of points won on serve and receive, stroke and error types. Match progression can be plotted for each set, recording the location of each ball bounce (see Fig. 1).

To reduce the dimensionality of the problem, a rally can be represented as the location of the winning shot. Furthermore, each half of the table can be split into nine equal sections, and the location of winning shots can be grouped (Fig. 2). Further grouping can involve the number of forehands and the number of backhands used to win a point, or whether it was a 'short' or 'long' rally (Fig. 3). Samples with missing data entries were removed from the dataset.

A. Match Representation

We represent each match (i) from a participating player's (P) perspective as follows:

- a feature vector ($\underline{x}_i$) consisting of player and match statistics,
- the target variable (y), indicating the result of the match:

$$y_i = \begin{cases} 1, & \text{if } P \text{ wins} \\ -1, & \text{if } P \text{ loses} \end{cases}. \tag{1}$$

With incomplete matches removed from the dataset, there are no other possible outcomes (there are no draws in table tennis). Each match actually maps to two ($\underline{x}_i$,y_i) pairs, from the perspective of the two participating players.

B. Feature Engineering

We followed the approaches taken in *tennis* [13], [18], [19] to form new features for table tennis. These features are player-focused, as unlike in team sports, we do not need to consider line-ups, collective team ability or substitutions.

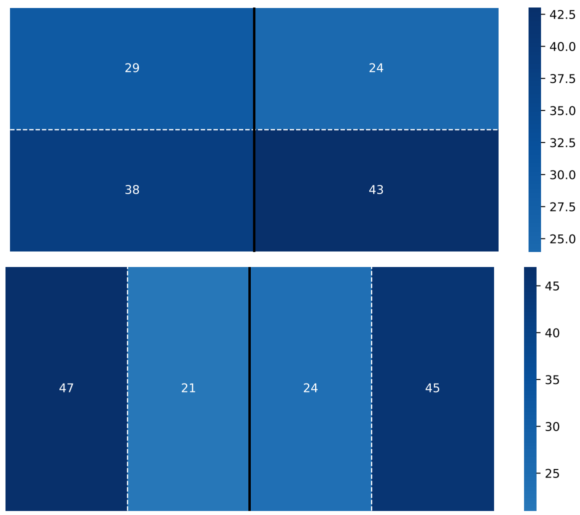

Fig. 3. Number of points won by forehand vs. backhand (top); by a short vs. long rally (bottom) in an example match [17].

Table I shows the final set of features. Newly derived features are indicated with *. More detailed explanations can be found in the subsections below.

SP: Serve Percentage: The proportion of points won on serve by a player. If the serve and error are made on opposing sides of the table, it was won by the serving player.

RP: Receive Percentage: The proportion of points won by the receiving player. E.g. in Fig. 1, the serve and error is made by the same player, so the point is won by the receiver.

LRP: Long Rally Percentage: The proportion of points won on a long rally by a player to the total number of rallies won. In this paper we define a long rally as a rally of at least five shots. E.g. Fig. 3 shows that one player won 47 points on a long rally, while the other won 45.

SRP: Short Rally Percentage: Compared to LRP, this is the proportion of points won on a *short rally* by a player. E.g. Fig. 3 shows that one player won 21 points on a short rally, while the other won 24 in the entire match.

FHP: Forehand Percentage: The proportion of points won on a forehand by a player, determined by the type of stroke used on the winning shot of a rally. Fig. 3 shows one player won 24 points on a forehand, and the other winning 38.

BHP: Backhand Percentage: The proportion of points won on a backhand by a player. Fig. 3 shows one player won 43 points on a backhand, and the other winning 29.

RANK: The ranking of the player by ITTF

RANKDIFF: Rank Difference:*

$$\text{RANKDIFF} = \begin{cases} \text{RANK}_a - \text{RANK}_b & \text{for player } a \\ \text{RANK}_b - \text{RANK}_a & \text{for player } b, \end{cases} \tag{2}$$

where RANK_a and RANK_b are player rankings for payers a and b at the time of the match. A rank advantage (i.e. lower numerical value than an opponent's), yields a negative RANKDIFF. Rankings are mostly reliable for the top players; for example, players of rank 2 and 7 are more likely to have an accurate depiction of their relative ability than players of rank 150 and 155, despite the difference being identical. To account for this, we apply a simple non-linearity and set RANKDIFF to 0 for matches where both players are ranked over 100.

SA, SRA*, HA*:* Serve advantage is calculated as the difference between their serve and receive winning percentage. This shows how likely a player is to win a point if they are serving rather than if they are on receive. Subsequently, the advantage a respective player has in a short rally over a long rally, as well as the advantage a respective player has in a forehand stroke over a backhand stroke, can be calculated.

BALANCE:* Players of a higher skill level tend to have fewer weaknesses and are stronger in more aspects of the game. We propose measuring the overall well-roundness as:

$$BALANCE = \frac{|SA| + |SRA| + |FHA|}{3}. \tag{3}$$

C. Feature Scaling

To account for the varying numerical range of the input features, and to make features comparable, we standardize each input to a zero mean with a unit standard deviation.

D. Live vs. aggregate data

Each feature vector $\underline{x}_i$ represents a single match i from a P player's perspective with several features observed during that game (SP, RP, LRP, SRP, FHP, VHP, SA*, SRA*, FHA*, BALANCE*). We postulate that the result of a match can be predicted from live in-match data well before the game

TABLE I
FEATURE SUMMARY

Feature	Explanation
SP	percentage of total points won on serve
RP	percentage of total points won on receive
LRP	percentage of total points won on a long rally
SRP	percentage of total points won on a short rally
FHP	percentage of total points won on a forehand
BHP	percentage of total points won on a backhand
RANK	player ranking
RANKDIFF*	difference in rank between opponents
SA*	player serve advantage
SRA*	player short rally advantage
FHA*	player forehand advantage
BALANCE*	measure of how well rounded a player is

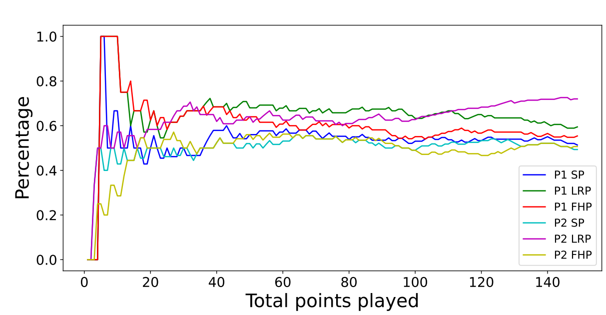

Fig. 4. Illustration of how individual features stabilize after a relatively short time on an example match. P1 and P2 correspond to the perspective of the two participating players.

ends, which is supported by the quick convergence of features as illustrated on an example match in Fig. 4. Hence, we fit our models on $(\underline{x}_i, y_i)$ pairs. However, we also validate the performance of these fitted models on pre-match prediction, when the feature vector $\underline{x}_i$ is averaged over all matches of player p except for the target match.

V. Models

We evaluated four models (logistic regression, random forest, SVMs, MLPs) as implemented in Scikit-Learn [20].

A. Logistic Regression

The logistic function maps the input feature $\underline{x}_i$ to a probability value p_i. Values over 0.5 correspond to the player winning match i. Training minimizes the *logistic loss* function [21]:

$$(p) = -\frac{1}{n}\sum_{i=1}^{n} p_i \log\left(\frac{y_i+1}{2}\right) + (1-p_i)\log\left(\frac{1-y_i}{2}\right), \quad (4)$$

where n is number of matches, p_i is the predicted probability of a player winning match i, and y_i is as defined in Eq. 1.

B. Random Forest

Random forest classifiers consist of an ensemble of simpler decision trees $\{h(\underline{x}, \theta_k),\ k = 1, \ldots\}$. For the kth tree, a random vector θ_k is generated and fitted to produce a classifier $h(\underline{x}, \theta_k)$ [22]. During inference, each tree casts a vote from input $\underline{x}$; the output is decided by a majority vote. Decision trees tend to be simpler to interpret and quicker to train.

C. Support Vector Machines (SVM)

SVMs have been used for *tennis* match predictions. These models identify the optimal hyperplane in the multi-dimensional feature space that separates data points into the two target classes (win, lose). During training, the marginal distance between this decision boundary and the instances closest to the boundary is maximized. SVMs have a choice of *kernels*, including linear, polynomial, sigmoid or a radial basis function [19].

D. Multilayer Perceptron Neural Networks (MLP)

An MLP is an artificial neural network consisting of an input layer ($\underline{x}$), an output layer (prediction), and one or more hidden layers in-between. Neurons in consecutive layers are connected (no connections within layers) [23]. Each connection has an associated weight. Training an MLP involves adjustments of these weights using backpropagation to minimize the difference between model output and the desired output.

E. Evaluating Models

To compare the performance of different model predictions, we calculated the *accuracy* of each model

$$\text{accuracy} = \frac{tp + tn}{tp + tn + fp + fn}, \quad (5)$$

where tp and tn are true positives and true negatives, and fp and fn are false positives and false negatives respectively. To get a more balanced idea about model performance, we also compute F1 scores as:

$$\text{precision} = \frac{tp}{tp + fp} \qquad \text{recall} = \frac{tp}{tp + fn} \quad (6)$$

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \quad (7)$$

which is effectively an F measure with $\beta = 1$ [24].

To avoid over-fitting, we used 5-fold cross validation. The dataset was split in training:validation:test in a 72:18:10 ratio. 10% of the original dataset was kept as a test set to validate hyperparameter tuning. The remaining 90% of data was split in an 80:20 ratio for the 5-fold training; 80% to train the model, 20% to optimize hyperparameters.

F. Hyperparameter Tuning

We used a brute-force grid search to fine tune parameters of the model that are outside the usual training domain (hyperparameters) e.g. the number of trees in a random forest classifier. The best combination of hyperparameters for a model is determined by whichever has the highest accuracy on the validation set using 5-fold cross validation.

For logistic regression, the type of solver, penalty function and the C terms value were adjusted. C is a regularisation term; the lower the value of C, the stronger the effect of regularisation. We found that $C = 1.0$, and 'liblinear' solver resulted in the best average accuracy. In terms of regularisation, L2 regularisation gave better results than L1 (Fig. 6).

For SVMs, the two main hyperparameters that were adjusted were the kernel type and penalty value C. Using a linear kernel and $C = 0.2$ gave the highest F1 score on the test set compared to other kernels. The learning curve for an SVM model using a linear kernel is shown in Fig. 6.

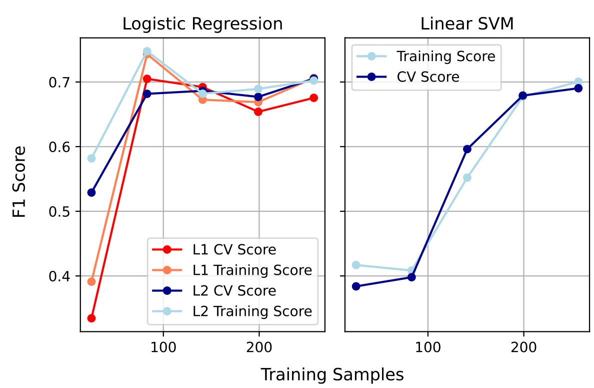

Fig. 6. Logistic regression and linear kernel SVM learning curves. The difference in F1 score for L1 and L2 regularisation is also illustrated.

VI. Experimental Results

Our main results are reported in Table II and Fig. 5. Both accuracy and F1 score are reported for the validation and test sets. The standard error for each score for the validation set

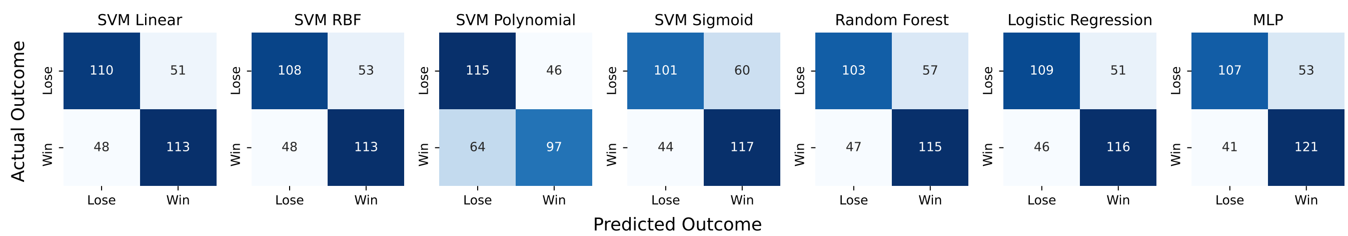

Fig. 5. Confusion matrices comparing the predicted and actual outcomes of test cases for each trained model.

TABLE II
MODEL PERFORMANCE COMPARING VALIDATION AND TEST SETS

Model	Validation set		Test set	
	Acc	F1	Acc	F1
Logistic Regression	0.699±0.024	0.705±0.023	0.722	0.706
Random Forest	0.677±0.032	0.688±0.033	0.667	0.684
Support Vector Machine				
→ Linear	0.696±0.029	0.690±0.035	0.639	0.629
→ RBF	0.700±0.025	0.677±0.034	0.667	0.600
→ Polynomial	0.705±0.021	0.685±0.021	0.611	0.563
→ Sigmoid	0.705±0.017	0.690±0.019	0.694	0.621
MLP Neural Network	0.696±0.019	0.708±0.020	0.694	0.703

TABLE III
MODEL PERFORMANCE WITH AND WITHOUT NEWLY DERIVED FEATURES

Model	With		Without	
	Acc	F1	Acc	F1
Logistic Regression	0.699	0.705	0.631	0.668
Random Forest	0.677	0.688	0.661	0.673
Support Vector Machine				
→ Linear	0.696	0.690	0.556	0.619
→ RBF	0.700	0.677	0.500	0.591
→ Polynomial	0.705	0.685	0.500	0.640
→ Sigmoid	0.705	0.690	0.472	0.642
MLP Neural Network	0.696	0.708	0.639	0.683

is reported as a basis of defining uncertainty. The validation column shows that most models perform comparably with approx. 70% accuracy. This value is also comparable to state-of-the-art metrics in *tennis* match prediction.

F1 scores indicate that MLP Neural Networks (with a *relu* activation) slightly over-perform their competitors, but the difference is not significant. The hidden layer size was set to 2 and the maximum number of iterations the solver iterates was chosen to be 200. The solver for weight optimization is set to 'lbfgs', a quasi-Newton optimizer. The learning rate for scheduling weight updates is set to constant. However, the generic layered structure of a neural network has proven to be time consuming. Additionally, this technique is considered a 'black box' technology, and finding out why a neural network has outstanding or even poor performance is difficult [23].

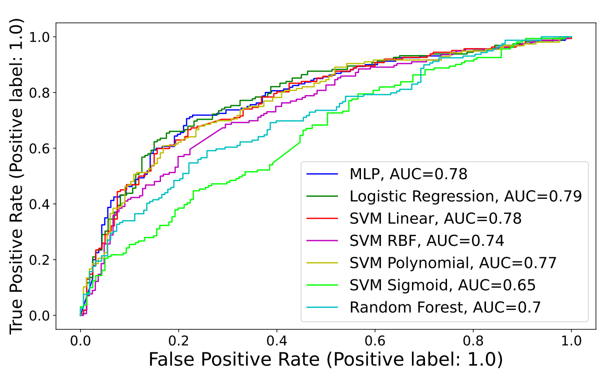

Fig. 7. ROC Learning Curves for the overall performance of each model.

Receiving operating characteristics (ROC, Fig. 7) support the quantitative results. The kernel choice for SVM models makes a noticeable difference; the areas under the ROC curves are otherwise comparable for all other models.

One qualitative advantage of using a random forest classifier is its training speed, which made hyperparameter tuning easier. The maximum number of levels in each decision tree was set to 80, the maximum number of features considered for splitting a node was set to 4, the minimum number of data points allowed in a leaf node was set to 4 and the number of trees that were in the classifier was set to 200.

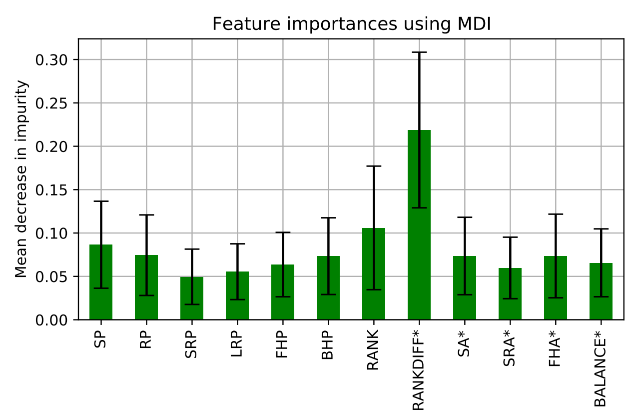

Fig. 8. Importance of features from random forest classifier based on Gini impurity. In our dataset, RANKDIFF appears to be the most important feature.

Another significant advantage of a random forest classifier is that the importance of features can also be extracted and visualized. Fig. 8 shows this as the mean decrease in Gini impurity for features across all trees. The impurity of a node is the probability of a specific feature being classified incorrectly assuming that it is selected randomly [25].

A. Ablation Study

Fig. 8 predicts that the most important feature in a random forest model is RANKDIFF, which justifies the inclusion of hand-crafted features. To reinforce this finding, we computed

TABLE IV
PRE-MATCH PREDICTION MODEL PERFORMANCE

Model	Test set Acc	Test set F1
Logistic Regression	0.639	0.667
Random Forest	0.667	0.714
Support Vector Machine		
→ Linear	0.639	0.667
→ RBF	0.639	0.649
→ Polynomial	0.611	0.632
→ Sigmoid	0.639	0.649
MLP Neural Network	0.667	0.714

the accuracy and F1 score for each model with and without the derived features. All scores are lower for models that do not use newly derived features, and accuracy score is significantly lower in SVMs compared to other models. See Table III.

To understand how well the trained model would perform when used purely as a pre-match predictor, we computed accuracy and F1 scores for aggregate feature vectors $\underline{x_i'}$ which contain the average features from all matches, excluding the target match x_i. Table IV shows similar results to live prediction, with accuracy values of 61–67%.

VII. Conclusion and Future Work

In this paper, we explore how supervised classification models can be used to predict the results of table tennis matches. The original dataset was retrieved from OSAI [17].

We evaluate a number of state-of-the-art classification models (logistic regression, random forest classification, SVMs, multi-layer perceptrons) using 5-fold cross-validation and hyperparameter tuning. We also propose using a handful of engineered features, from which a non-linear rank difference has been proved to be the most salient in our ablation study. We consider the difference of live prediction vs. pre-match prediction and demonstrate that our model performs comparably in both cases.

Our results are comparable to the accuracy of state-of-the-art *tennis* prediction models (approx. 70% accuracy). Following hyperparameter tuning, the difference between models was often modest. Other considerations when picking a model for similar applications could include training time or model transparency (at both of which random forests excel).

Future work could explore combining TTNet with our prediction model to provide live match predictions. It would be also interesting quantifying uncertainty and to test against real betting odds. As automated table tennis analytics are becoming available below professional leagues, the authors are also interested whether the importance of features and the model choice transfers to these matches as well.

Acknowledgements

The authors would also like to thank the OSAI team for granting permission to use their dataset.

References

[1] J. Wang, K. Zhao, D. Deng, A. Cao, X. Xie, Z. Zhou, H. Zhang, and Y. Wu, "Tac-simur: Tactic-based simulative visual analytics of table tennis," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 407–417, 2019. 1

[2] "Ittf handbook 2022." https://documents.ittf.sport/document/284. Accessed: 2022-05-03. 1

[3] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Computational and structural biotechnology journal*, vol. 13, pp. 8–17, 2015. 1

[4] S. M. Weiss and I. Kapouleas, "An empirical comparison of pattern recognition, neural nets, and machine learning classification methods.," in *IJCAI*, vol. 89, pp. 781–787, Citeseer, 1989. 1

[5] A. I. Khan and S. Al-Habsi, "Machine learning in computer vision," *Procedia Computer Science*, vol. 167, pp. 1444–1451, 2020. 1

[6] P. Larranaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armananzas, G. Santafé, A. Pérez, *et al.*, "Machine learning in bioinformatics," *Briefings in bioinformatics*, vol. 7, no. 1, pp. 86–112, 2006. 1

[7] J. Xing, H. Ai, L. Liu, and S. Lao, "Multiple player tracking in sports video: A dual-mode two-way bayesian inference approach with progressive observation modeling," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1652–1667, 2010. 1

[8] J. G. Claudino, D. d. O. Capanema, T. V. de Souza, J. C. Serrão, A. C. Machado Pereira, and G. P. Nassis, "Current approaches to the use of artificial intelligence for injury risk assessment and performance prediction in team sports: a systematic review," *Sports medicine-open*, vol. 5, no. 1, pp. 1–12, 2019. 1

[9] E. Davoodi and A. R. Khanteymoori, "Horse racing prediction using artificial neural networks," *Recent Advances in Neural Networks, Fuzzy Systems & Evolutionary Computing*, vol. 2010, pp. 155–160, 2010. 1

[10] R. P. Bunker and F. Thabtah, "A machine learning framework for sport result prediction," *Applied computing and informatics*, vol. 15, no. 1, pp. 27–33, 2019. 2

[11] J. Žemgulys, V. Raudonis, R. Maskeliūnas, and R. Damaševičius, "Recognition of basketball referee signals from videos using histogram of oriented gradients (hog) and support vector machine (svm)," *Procedia computer science*, vol. 130, pp. 953–960, 2018. 2

[12] S. R. Clarke and D. Dyte, "Using official ratings to simulate major tennis tournaments," *International transactions in operational research*, vol. 7, no. 6, pp. 585–594, 2000. 2

[13] T. Barnett and S. R. Clarke, "Combining player statistics to predict outcomes of tennis matches," *IMA Journal of Management Mathematics*, vol. 16, no. 2, pp. 113–120, 2005. 2

[14] W. J. Knottenbelt, D. Spanias, and A. M. Madurska, "A common-opponent stochastic model for predicting the outcome of professional tennis matches," *Computers & Mathematics with Applications*, vol. 64, no. 12, pp. 3820–3827, 2012. 2

[15] R. Voeikov, N. Falaleev, and R. Baikulov, "Ttnet: Real-time temporal and spatial video analysis of table tennis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 884–885, 2020. 2

[16] Z. Zhang, D. Xu, and M. Tan, "Visual measurement and prediction of ball trajectory for table tennis robot," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 12, pp. 3195–3205, 2010. 2

[17] "Osai." https://osai.ai/. Accessed: 2022-01-18. 2, 3, 6

[18] M. Sipko and W. Knottenbelt, "Machine learning for the prediction of professional tennis matches," *MEng computing-final year project, Imperial College London*, 2015. 2

[19] A. Cornman, G. Spellman, and D. Wright, "Machine learning for professional tennis match prediction and betting," *Stanford Unverisity*, 2017. 2, 4

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011. 4

[21] E. Hazan, T. Koren, and K. Y. Levy, "Logistic regression: Tight bounds for stochastic and online optimization," in *Conference on Learning Theory*, pp. 197–209, PMLR, 2014. 4

[22] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001. 4

[23] L. Noriega, "Multilayer perceptron tutorial," *School of Computing. Staffordshire University*, 2005. 4, 5

[24] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation," in *Australasian joint conference on artificial intelligence*, pp. 1015–1021, Springer, 2006. 4

[25] A. P. Cassidy and F. A. Deviney, "Calculating feature importance in data streams with concept drift using online random forest," in *2014 IEEE International Conference on Big Data (Big Data)*, pp. 23–28, IEEE, 2014. 5