Machine Learning in Table Tennis Match Prediction

Sophie Chiang
The Perse School

***Abstract*—Machine learning (ML) has shown to be useful in many different fields of classification and prediction. In this paper, these techniques are applied specifically to predicting the outcome of a table tennis match, and a similar idea has been applied to different sports for arranging effective training for players. Player and match statistics from data on historical matches were to predict the outcome, and 12 features were extracted and derived in an attempt to capture the quality of players and their opponents. A multi-layer perceptron neural network model was shown to be have the best overall performance.**

I. Introduction

The application of ML techniques to sport result prediction has become increasingly more popular recently in sport analytics. Such methods have been frequently used in tennis [8] and football [10] by coaches to arrange for suitable and effective training in order to improve player performance for future matches. The unpredictable nature of sport also makes predicting results an interesting challenge to general sports fans.

There are only two possible outcomes of a match in table tennis for a respective player; a win or a loss, and unlike team sports, a combination of players with each of varying individual ability and skill level do not need to be considered in the predictive outcome. Due to this, the likelihood of player substitutions nor offensive and defensive combinations do not need to be analysed. Despite table tennis prediction being somewhat a novelty due to its lack of popularity, the basic machine learning principles still apply.

In this paper, multiple classification algorithms from machine learning are used to model both men and women's professional singles matches. Player statistics collected from historical matches are predominantly used in predicting the outcome, with newly derived information calculated from combining player statistics also being used. The overall performance of each model will be compared and evaluated towards the end of the paper.

II. Related Work

A. 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 [15], pattern recognition [22], computer vision [11] and bioinformatics [16]. The idea is to identify patterns and relationships in raw data that has been inputted into an ML model, known as a feature vector, which can then be turned into an output. In classification, the aim is to predict a target variable (class) from a training dataset, the data in which the model is fitted on, to predict the value of that class in testing instances, where the values of predictor features are known but the value of the class label is not [14]. This type of data processing is known as supervised learning, as instances are given with the corresponding correct outputs (labels), as opposed to unsupervised learning, where instances are unlabelled [13].

B. Sport Prediction

There are a very limited number of works in table tennis prediction, however extensive research has been done in tennis. Both are ideal sports to apply hierarchical probability models to; a table tennis match consists of a sequence of sets, which consists of a sequence of points, therefore due to certain similarities between the two sports, certain concepts can be applied from works that have been conducted in tennis.

For example, Knottenbelt [12] proposed a common opponent model that yielded a pre-play estimate of the probability of each player winning a professional singles tennis match. This was achieved by analysing match statistics for opponents that both players have encountered in the past, which provided for a fair basis comparison. Subsequently, the model was able to compute the probability of each player winning a point on their serve, and hence the match.

Barnett [2] uses historical data from past matches to predict the probability of a player winning a single point and Clarke [7] used a years worth of tournament results to predict the outcome of a tennis match using player rating points.

A number of computer vision based approaches have been applied to table tennis. Voeikov *et al.* [21] proposed a network that allowed for real-time processing of high-resolution table tennis videos. This was able to extract temporal and spatial data such as ball detection and in game events, and is potentially capable of substituting manual data collection by sport scouts, in addition to assisting with referee decision making.

Zhang [23] is able to compute the 3D coordinates of a table tennis ball by its image coordinates. The flying trajectory of the ball can be predicted, and hence the balls landing and striking point can be calculated.

III. Dataset

All table tennis match data was retrieved from OSAI [1] on 18/01/2022. This includes match and player statistics from Tischtennis-Bundesliga, the top professional German table tennis league, as well as men and women's singles matches from the Tokyo 2020 Olympics. Many potential features such as player age, rank, match duration as well as in-match statistics

[1] https://osai.ai/

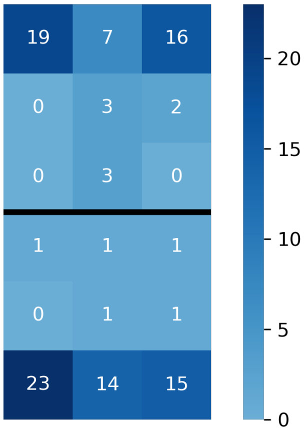


Fig. 1. Distribution of ball placement on a winning stroke of a match

such as percentage of points won on serve and receive, stroke types and error types were available. Interactive maps that demonstrated the ball position of each shot on the table, as well as the stroke type were also accessible.

For example, Figure 1 shows the ball placement of a winning shot for both players by region if each side of the table were to be split into nine equal parts. The heatmap is plotted as if the table were being viewed from above and each value indicates the number of balls landing in it's respective region. Any samples that had missing data entries were also removed from the dataset.

IV. Feature Engineering & Selection

Irrelevant or redundant variables can increase the computation time, as well as decrease a model's performance, therefore choosing appropriate features to input into a classification model is very important.

A. Match Representation

In supervised machine learning, a set of labelled data is required for the model to train on. In the context of table tennis prediction, each match corresponded to two instances of data, one from the perspective of each player respectively, where every sample is composed of two elements:

- A vector of input features (x) consisting of player and match statistics
- The target variable (y), indicating the result of the match that corresponds to its respective sample

The outcome of the match for player i is defined as follows:

$$y = \begin{cases} 1, & \text{if } player_i \text{ wins} \\ -1, & \text{if } player_i \text{ loses} \end{cases}$$

As any incomplete matches were removed from the dataset, combined with the inability to draw in table tennis, there is no other possible outcome.

B. Feature Engineering

In addition to the features extracted from the dataset in Section III, players statistics were combined to form new features [2]. Using pre-existing knowledge on the sport, adding combinations of player statistics as features may improve our predictive model. These features were calculated as differences between different player statistics as this considers the characteristics of *both* players participating in a match. Both Sipko [19] and Cornman [8] use features calculated as differences to predict the outcome of a tennis match. Ultimately, the final feature set with abbreviated feature names and explanations is shown in Table I:

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

TABLE II
Raw Extracted Data

Feature name	Abbreviation
Serve Percentage	SP
Receive Percentage	RP
Long Rally Percentage	LRP
Short Rally Percentage	SRP
Forehand Percentage	FHP
Backhand Percentage	BHP
player ranking	RANK

TABLE III
Newly Derived Features

Feature name	Abbreviation
Rank Difference	RANKDIFF
Serve Advantage	SA
Short Rally Advantage	SRA
Forehand Percentage	FHA
Player Balance	BALANCE

The raw data features are shown in Table II and the newly derived features are shown in Table III. For the purposes of this paper, a long rally is defined as a rally of over five shots, and a short rally is any rally of length under five.

The feature *RANKDIFF* was constructed by calculating the difference between rankings of two opponents.

$$RANKDIFF = \begin{cases} RANK_i - RANK_j & \text{for player } i \\ RANK_j - RANK_i & \text{for player } j \end{cases}$$

where $RANK_i$ and $RANK_j$ are the rankings of players i and j respectively at the time of the match. Therefore if a player's rank is higher than their opponent's rank, $RANKDIFF$ will be a negative value, and vice versa. However, for some match instances where the ranking of both players are above 100, the feature $RANKDIFF$ is considered to be 0. This is due to the fact that the lower the rank of a player, the more likely it is that there will be other players of a similar standard where rank doesn't accurately represent the standard of a player. For example, players of rank 2 and rank 7 is much more likely to have an accurate depiction of their standard in comparison to two players of rank 150 and 155, despite the rank difference being the same. Therefore, if both players have a rank of below 100, the expected difference in skill level is considered to have no benefit.

The serve advantage of a player is calculated as the difference between their serve and receive winning percentage. This depicts the contrast as to how likely a player is to win a point if they are serving, compared to 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 therefrom.

An attempt to measure the *completeness* of a player can be calculated by taking the average of serve, short rally and forehand advantage:

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

Players of a higher skill level tend to have fewer weaknesses and are stronger in more aspects of the game, therefore the feature $BALANCE$ indicates the overall well-roundness of a player's ability.

C. *Feature Scaling*

Different features tend to have a varying range of values, therefore it is normal to scale features as part of data preprocessing prior to learning. *Standardization* is a scaling technique to centre values around the mean with a unit standard deviation [4], and was performed across features.

V. Method

A number of different machine learning algorithms were used, and the *accuracy* for each model was calculated as an estimate for it's overall performance. In cross validation, the dataset is split into k random subsets, known as folds, and one is selected as a test set for the model to test on, while the others are used as a training set for the model to train on. This is repeated k times where each fold is used once as a test set, and the overall performance of the model is calculated as the average of accuracy scores for each iteration [3]. Accuracy is defined to be the proportion of correctly predicted instances to the total number of predictions:

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

where tp and tn are true positives and true negatives, and fp and fn are false positives and false negatives respectively [20].

Other methods can also be used to evaluate a model's performance. For example $f1$ score takes into account precision and recall.

A. *Logistic Regression*

Logistic regression was applied on the dataset using Scikit-Learn's implementation [18]. The logistic function $\sigma(t)$ is defined as follows:

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

The logistic functions maps any input value x where $x = \{x \in \mathbb{R}\}$ to a value between 0 and 1, allowing the output to be interpreted as a probability. If this probability is considered to be over 0.5, it can be classified as true, where the player is predicted to win the match, otherwise the result is classified as false. The model gives the best reproduction of match outcomes for the training set by minimising the *logistic loss* function:

$$L(p) = -\frac{1}{n}\sum_{i=1}^{n} p_i \log(y_i) + (1 - p_i)\log(1 - y_i)$$

$$n = \text{number of matches}$$
$$p_i = \text{predicted probability of a player winning match } i$$
$$y_i = \text{outcome of match } i$$

where a loss function measures the disparity between observations and their estimated fits [9].

B. *Random Forest*

A random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(\mathbf{x}, \theta_k),\ k = 1, ...\}$ where the $\{\theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\mathbf{x}$. For the kth tree, a random vector θ_k is generated and a tree is grown using θ_k and the training set, to produce a classifier $h(\mathbf{x}, \theta_k)$ [5]. After a large number of trees are produced, the output of a random forest model is the most popular output of all trees. Decision trees tend to be simple to interpret and ”quick” to learn, making it a popular ML technique.

C. Support Vector Machines (SVM)

SVMs were used in predicting match outcome [18]. The idea is that SVMs map an input vector in a feature space of n dimensions, where n is the number of features. The optimal hyperplane is identified which separates data points into two classes. This is known as the decision boundary, and the marginal distance between this boundary and the instances closest to the boundary is maximized. The existence of a decision boundary can allow for any detection of miss-classification.

D. Multilayer Perceptron Neural Networks (MLP)

An MLP neural network is a mathematical model that consists of one or more hidden layers in between it's input and output layers. The network consists of mutually connected artificial neurons, where neurons are organised in layers, and connections are directed from lower layers to upper layers. Neurons from the same layer are not interconnected with each other [17].

Each connection between two neurons has an associated weight, and in the process of learning and training an MLP model, these weights are adjusted such that there is a minimal difference between the model output and the desired output.

VI. Experimental Results

As explained in Section V, 10-fold cross validation was used to calculate the accuracy of each model. The results are shown in Table IV:

TABLE IV
Model Accuracy

Model	Accuracy (%)
Logistic Regression	69.6
MLP-Neural Network	70.4
Support Vector Machine	
→ Linear	69.3
→ RBF	62.7
→ Polynomial	52.7
→ Sigmoid	60.9
Random Forest	69.7

SVM algorithms use a set of mathematical functions that are defined as *kernels*, different SVM algorithms use different type of kernel functions. Different kernels including linear, polynomial, sigmoid and radial basis function (RBF) were used for the purposes of this study.

VII. Discussion

For each model, the *hyperparameters*, which are parameters that are not optimised by the training algorithm, were tuned manually by using a grid search to test different values. A grid search uses brute force to search the entire space for different hyperparameter configurations.

For logistic regression, the type of solver, penalty function and C value were adjusted. *Regularisation* prevents overfitting of training data by penalising large weights when training a logistic regression predictor, and the parameter C mentioned is used to control the effect of this [1]. The lower the value of C, the stronger the effect of regularisation. The default value is 1, however for the logistic regression model used in this paper, a C value of 0.32 was chosen to give the best accuracy score after hyperparameter tuning.

Fig. 2. Logistic Regression Learning Curve

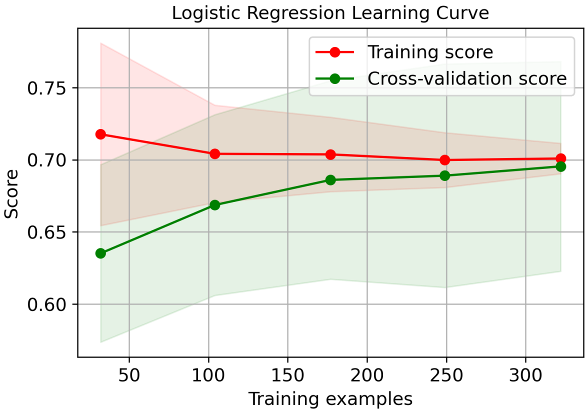
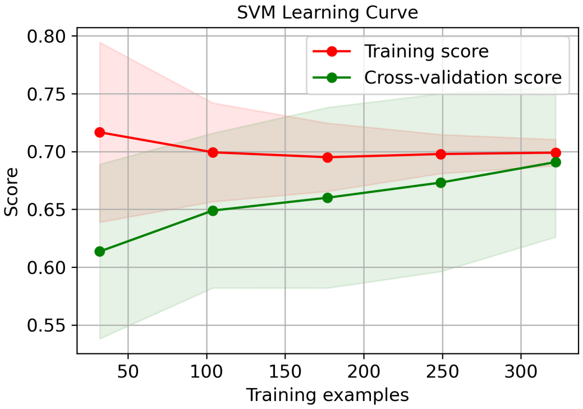


Fig. 3. SVM Learning Curve

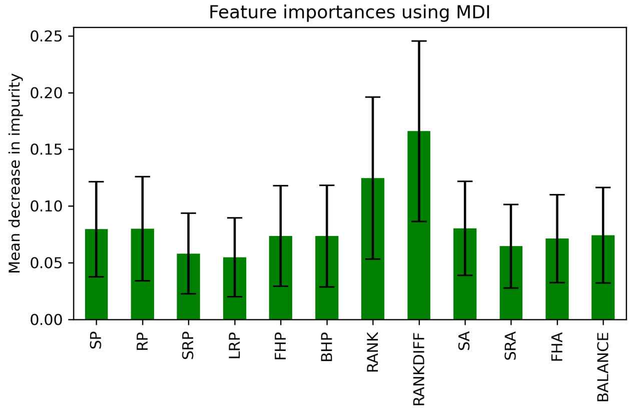


$l1$ regularisation penalises the sum of absolute values of weights, whereas $l2$ regularisation penalises the sum of squares of the weights [1]. $l2$ regularisation was chosen and the learning curves of the logistic regression model, where score refers to accuracy score, can be seen in Figure 2.

For SVMs, the two main hyperparameters that were adjusted were the kernel type and penalty value C. Using a linear kernel was demonstrated to have a better accuracy score than the others, and C was chosen to be 0.30. The learning curve for an SVM model using a linear kernel is shown in Figure 3.

The model that achieved the highest accuracy score were MLP neural networks. The most improvement in accuracy score was shown when tuning hyperparameters of the model such as the number of hidden layers, the maximum number of iterations and the type of solver. However, the generic layered structure of a neural network has proven to be time consuming. Additionally, this technique is considered as a ”black box”

Fig. 4. Feature Importance



technology, and finding out why a neural network has poor performance, or how it performs the classification process, is extremely difficult [17].

One of the main advantages of using random forest is that it is much faster to train. This made tuning hyperparameters easier as training the model several times was less computationally expensive. The maximum number of levels in each decision tree was set to 90, 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 7 and the number of trees that were in the classifier was set to 200.

The importance of features from a random forest classifier can also be evaluated, as shown in Figure 4. This is achieved by calculating the mean decrease in impurity for features across all trees, where *impurity* refers to the Gini impurity index. The impurity of a node is the probability of a specific feature being classified incorrectly assuming that it is selected randomly [6].

We can also compare model accuracy that do not include newly derived features that were defined in Section IV-B:

TABLE V
MODEL ACCURACY WITHOUT NEWLY DERIVED FEATURES

Model	Accuracy (%)
Logistic Regression	61.1
MLP-Neural Network	62.8
Support Vector Machine	
→ Linear	63.1
→ RBF	62.6
→ Polynomial	52.7
→ Sigmoid	55.9
Random Forest	63.6

VIII. CONCLUSION

In this paper the, the concepts of ML were discussed and how the use of supervised ML methods and classification algorithms could be used in table tennis match prediction. From the results displayed in Table IV, it is evident that the model which achieved the highest accuracy score were MLP neural networks. From Figure 4, the feature which is shown to be the most important is $RANKDIFF$, and future works could focus on a selection of the most important features established from the random forest model.

ACKNOWLEDGEMENTS

I would like to express my gratitude towards Dr Denes for his continued support and feedback throughout this project. I would also like to thank the OSAI team for granting permission for using their data for this project.

REFERENCES

[1] H. Ahmadian, J. Mottershead, and M. Friswell. Regularisation methods for finite element model updating. *Mechanical Systems and Signal Processing*, 12(1):47–64, 1998. 4

[2] T. Barnett and S. R. Clarke. Combining player statistics to predict outcomes of tennis matches. *IMA Journal of Management Mathematics*, 16(2):113–120, 2005. 1, 2

[3] D. Berrar. Cross-validation., 2019. 3

[4] D. Bollegala. Dynamic feature scaling for online learning of binary classifiers. *Knowledge-Based Systems*, 129:97–105, 2017. 3

[5] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 3

[6] 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)*, pages 23–28. IEEE, 2014. 5

[7] S. R. Clarke and D. Dyte. Using official ratings to simulate major tennis tournaments. *International transactions in operational research*, 7(6):585–594, 2000. 1

[8] A. Cornman, G. Spellman, and D. Wright. Machine learning for professional tennis match prediction and betting. *Stanford Unverisity*, 2017. 1, 2

[9] E. Hazan, T. Koren, and K. Y. Levy. Logistic regression: Tight bounds for stochastic and online optimization. In *Conference on Learning Theory*, pages 197–209. PMLR, 2014. 3

[10] J. Hucaljuk and A. Rakipović. Predicting football scores using machine learning techniques. In *2011 Proceedings of the 34th International Convention MIPRO*, pages 1623–1627. IEEE, 2011. 1

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

[12] 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*, 64(12):3820–3827, 2012. 1

[13] S. B. Kotsiantis, I. Zaharakis, P. Pintelas, et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007. 1

[14] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190, 2006. 1

[15] 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*, 13:8–17, 2015. 1

[16] 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*, 7(1):86–112, 2006. 1

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

[18] 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*, 12:2825–2830, 2011. 3, 4

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

[20] G. Vanwinckelen and H. Blockeel. On estimating model accuracy with repeated cross-validation. In *BeneLearn 2012: Proceedings of the 21st Belgian-Dutch conference on machine learning*, pages 39–44, 2012. 3

[21] 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*, pages 884–885, 2020. 1

[22] S. M. Weiss and I. Kapouleas. An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. In *IJCAI*, volume 89, pages 781–787. Citeseer, 1989. 1

[23] Z. Zhang, D. Xu, and M. Tan. Visual measurement and prediction of ball trajectory for table tennis robot. *IEEE Transactions on Instrumentation and Measurement*, 59(12):3195–3205, 2010. 1