INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Is the performance of a cricket team really unpredictable? A case study on Pakistan team using machine learning

Waqar Ahmed[1,2], **Mahwish Amjad**[3], **Khurum Nazir Junejo**[4], **Tariq Mahmood**[5], **Ayaz H Khan**[6]*

1 PhD Student, Pattern Analysis and Computer Vision Department, Instituto Italiano di Tecnologia, Genova, Italy
2 PhD Student, Universita degli Studi di Genova – Dipartimento di Ingegneria Navale, Elettrica, Elettronica e delle Telecomunicazioni, Genova, Italy
3 PhD Student, Department of Computer Science, Institute of Business Administration, Karachi, Pakistan
4 Head of Research & Development, Love for Data, Karachi, Pakistan
5 Associate Professor, Department of Computer Science, Institute of Business Administration, Karachi, Pakistan
6 Assistant Professor, Computer Science Department, Habib University, Karachi, Pakistan. Tel.: +92-331-847-0261

Abstract

Cricket is the second most popular game around the globe, particularly it breeds a high level of enthusiasm in Asia, Australia and UK. However, it is generally known and globally mentioned that Pakistan is an "unpredictable" cricket team, which leads to extreme reactions from the citizens in case of a loss, e.g., verbal anger, breaking of television sets and burning of players' effigies. **Objectives**: In this study, we leverage machine learning techniques to demonstrate that the use of the "unpredictable" tag for Pakistan's cricket performance is unjustified as the match outcome can be predicted with a pretty high confidence. **Method**: We produce a new dataset by scrapping latest statistics from cricinfo.com, the most reliable online source. Also, we propose a novel feature "consecutive wins" that incorporates recent performance trend of the team. With extensive experimental setup, state-of-the-art machine learning methodology was employed to prove effectiveness of proposed tool. **Findings**: Pakistan's cricket performance can be predicted with 82% accuracy, i.e., it is possible to understand the patterns (in advance) which may lead to a winning or losing situation. Hence, using pre-match analysis, it is possible to avoid any prejudiced opinion or potentially dangerous reactions. **Novelty**: We employ state-of-the-art machine learning methodology based on application of various algorithms, feature selection and data splitting methods. Eventually, state-of-the-art prediction accuracy is achieved by exploiting all potential avenues in a structured way.

Keywords: Cricket; one day international; Pakistan; machine learning; performance; unpredictable

OPEN ACCESS

Received: 07-06-2020
Accepted: 27-07-2020
Published: 24-09-2020

Editor: Dr. Natarajan Gajendran

Citation: Ahmed W, Amjad M, Junejo KN, Mahmood T, Khan AH (2020) Is the performance of a cricket team really unpredictable? A case study on Pakistan team using machine learning. Indian Journal of Science and Technology 13(34): 3586-3599. https://doi.org/10.17485/IJST/v13i34.813

*__Corresponding author__.

Tel: +92-331-847-0261
ayaz.hassan@sse.habib.edu.pk

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

1 Introduction

Cricket is the second most popular sport after soccer with a global following of 3.5 billion fans[1,2]. Nowadays, the major cricket-playing countries are England, Australia, India, New Zealand, South Africa, Pakistan, West Indies, Sri Lanka and Bangladesh[3,4]. Cricket is played between two teams of eleven (11) players each in a cricket field. The actual game is played on the pitch, a rectangular 22-yard area at the center of field (Figure 1). Batting and bowling are two primary activities during a cricket match; Figure 2 shows the bat and ball used for batting and bowling and Figures 3 and 4 show a batsman batting and a bowler bowling respectively. In an One Day International (ODI) cricket format, teams play each other over the period of a whole day. In an ODI, a toss of a coin decides which team will bat initially. Assume Team A and Team B are the teams and Team A wins the toss and decides to bat first. During batting, Team A attempts to put up as large a score as possible (runs) on the scorecard with Team B bowling. Team A is given 50 overs (600 balls) to make a score and Team B attempts to dismiss ten batsmen of Team A as soon as possible to minimize the runs; each bowler bowls a maximum of ten overs. On dismissal, it is said that the batsman is out (i.e., he cannot play anymore). The four ways of getting out are clean bowl, leg before wicket, catch-out and run-out (for details, see[3]). After Team A finishes batting then Team B bats to overhaul Team A's score with Team A bowling. If this happens, Team B will win; otherwise, Team A wins if Team B has a lesser score in 50 overs, and the game is drawn if scores of both teams are the same after 50 overs. In the event of rain, the match is curtailed to less than 50 overs and the runs (to be made to win the game) are determined by the Duckworth-Lewis system[5].

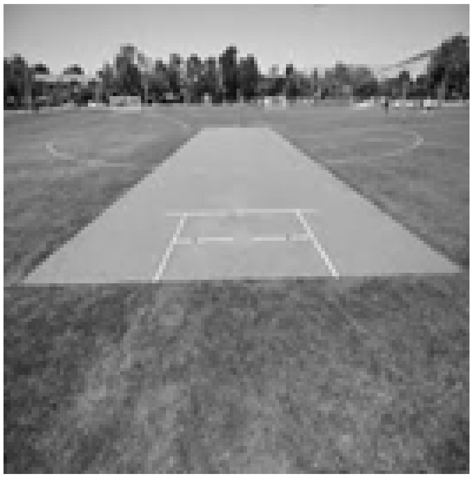

Fig 1. Cricket Field and Pitch

Fig 2. Bat and Ball

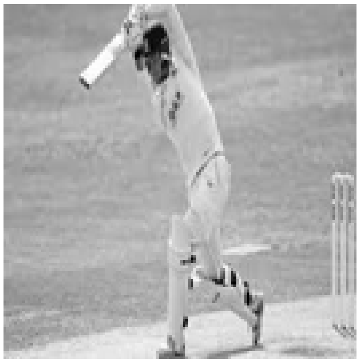

Fig 3. A Batsman Batting

Fig 4. A Bowler Bowling

1.1 Problem statement

In Pakistani society, cricket is considered the favorite sport even though the national sport is hockey. There are numerous media reports and surveys to prove that an impending ODI cricket match breeds a high level of enthusiasm (and sometimes even craze) in Pakistani people[6–10]. However, it is generally known that Pakistan is an "unpredictable" team which has displayed an inconsistent behavior over the last several decades in all types of cricket tournaments[11,12]. If Pakistan loses a match, the "unpredictable" tag leads to extreme reactions from the society. For instance, Fans invaded the pitch during the match in 2001[13], breaking of television sets after losing against West Indies in 2015[14] and against India in 2016[15], verbal anger and severe backlash after losing against India in 2019[16–18], road blockages, burning of players' effigies, and even attacks on cricketers' houses[8,19]. These are extremist reactions and it is important to determine to what extent they are justified. For this, we need to determine the extent to which the "unpredictable" tag is justified.

1.2 Proposed framework

In this paper, we address this problem by employing a comprehensive machine learning methodology[20] to predict the performance of Pakistan's ODI cricket team. Our motive is that, if the performance can be predicted accurately, we can construe that extreme reactions are not justified and the citizens needs to have more knowledge of the situation to understand why the loss or win occurred (rather than reacting solely on the team's unpredictable nature). Our study is the first of its kind for Pakistan cricket while being unique with respect to existing work (Section 3) in three ways: 1) we experiment with a set of standard feature selection techniques to determine most decisive attributes, 2) we ablate on a set of standard algorithms to determine the best classifier for the task of interest, rather than selecting some of our choice in a biased way, 3) we exhaustively examine data split strategy in order to determine the policy (e.g., manual or automated) that augments classifier's performance. The results help us in answering our research question: Is it justified for the general public to claim that Pakistan team's performance is unpredictable in ODI cricket format?

2 Related work

Regarding statistical modeling of cricket data, the work done in [21] shows that a negative binomial and Poisson distribution can both model certain player movements and their performances. In [22], the authors show that the batting average of batsmen can be estimated through both non-parametric and parametric approaches. Finally, the study conducted in [23] showed that a modification of the Duckworth-Lewis system can be used to quantify the magnitude of victory in ODI which could assist in breaking tournament ties. Over recent years, the related work has focused more on predicting cricket match outcomes due to advances in machine learning [20]. New predictions about the outcome can be generated dynamically during the match as the situation changes (e.g., in a ball-by-ball fashion) [24]. Due to the unstable nature of these predictions, in this paper, we focus on predicting the match outcome once, before the match starts and based on historical data.

In [25], the authors use the dynamic programming optimization approach to estimate the optimal scoring rate at any stage during batting and the total number of runs at the end. The work done in [26] investigates the effect of Duckworth-Lewis method to predict the winner of an ODI and showed that this method does not have sufficient information to predict the match outcome. Although the authors statistically studied several other potential factors, these have not been experimentally evaluated to predict the match outcome before the match is played. In [27], the authors show that winning the toss at the outset of the ODI match provides no competitive advantage but playing on one's home field does provide an edge. On a related note, the study in [28] established that home teams generally enjoy a significant advantage (for test cricket matches). The authors applied logistic regression to explore how the relative batting and bowling strengths of teams, home advantage, winning of the toss, and a lead of runs while batting first time affect outcomes of test matches. They also concluded that generally no winning advantage is gained as a result of winning the toss. A collection of artificial neural networks is used in [29] for predicting the outcome of an ODI cricket tournament. To predict the tournament outcome, relevant data is run through each network for each separate team; the winner is the team having the highest aggregate score. Authors achieve an accuracy of 86%. Our analysis shows that the authors considered insufficient data set to train the model due to infrequent tournaments played every year. Also, the authors don't consider critical features related to pitch, season, location of match and opponents.

In [30], the authors implement a simulation for predicting the runs scored in an on-going ODI cricket. They use both historical and instantaneous data and develop separate models for matches played at home-ground and other-grounds. Nearest neighbor and bagging algorithms are used for prediction with latter being the winner with maximum 70% accuracy, in both home-ground and other-ground scenarios. This work is based on 125 matches played between a limited time frame, i.e., January 2011 and July 2012. We believe that the accuracy could have been enhanced had the authors either considered a more comprehensive real-time situation, or had skipped it completely; the authors themselves mention that the selection of attributes for prediction is a non-trivial task and this is exactly what needs to be addressed to improve the accuracy. In [31], the authors use Bayesian classification to predict the outcome of an ODI based on home-ground, day/night, winning of toss, and batting first features. The accuracy achieved is 51% which is deficient and occurs, in our opinion, due to the selection of the simplistic Bayesian approach which cannot model the complicated dynamics of modeling a cricket match. Vistro et. al. [32] utilizes the SEMMA process to predict the IPL match winner before the match starts. For model building, different machine learning algorithms including Random Forest, SVM, Naive Bayes, Logistic Regression, and Decision Tree were employed. However, the dataset used comprises of the records from the season 2008 to 2017 of the T20 match format which has entirely different dynamics than the ODI cricket format.

Specifically, only four papers in related work [25–27,31] focus on ODI match outcome predictions with highest accuracy of 70%. There is much research gap exists here as the number of applications is severely limited with no application of state-of-the-art machine learning algorithms and processes. With a rapid increase in the frequency of these algorithms, there is a need to conduct a comprehensive comparison of these algorithms using state-of-the-art machine learning execution pipelines [20,30,33]. Our study fills this evident gap with exhaustive feature and model selection procedure that brings up state of the art ODI cricket match outcome prediction with 82% accuracy. Besides various tools being recently utilized during live match such as online score and winner predictor, our proposed framework can be used to conduct pre-match analysis and regulate the possible public reaction. Also, our proposed tool can help team's coach to form better strategy about following match i.e., if the predictor infers the possible outcome as loss, team can proceed with alternate plans.

3 Background Knowledge

3.1 Machine learning algorithms

Our particular machine learning application is a binary classification task, i.e., predicting a binary label for unseen or test data after an algorithm has been trained on the seen training data whose labels are known. Particularly, we want to predict whether

Pakistan team will lose or win an upcoming ODI match, based on data of historical ODI data over a set of identified features (described in Section 4). Recent advances in machine learning has generated a large set of potential binary classifiers. At an abstract level, these can be split into two categories: ensemble and non-ensemble classifiers[(20)]. In both categories, we have selected the representative algorithms which have seen success in both academic research and industry in the last two decades (for proof, refer to[(34–38)]). We avoid selecting a particular ML algorithm of our own choice and experience, because there is no standardized guarantee about the performance of any ML algorithm. The standard practice is to experiment with a set of selected algorithms to determine the one which gives the best results. For implementation, we use Python's Sci-Kit Learn API which is "currently the most widely used open source library for Machine Learning applications" and has a larger community and usage for small training data sizes like our cricket dataset[(39,40)].

3.1.1 Ensemble classifiers

The ensemble algorithms work by creating randomized samples of the same training data and the combined result (either sum or average or majority vote) of these samples is used for predicting test data labels. The decision tree algorithm is the building block of ensemble methods. It generates a tree with features as nodes and labels as leaves. Tree construction starts at the top from the root node. At each node, a single feature is selected which provides the most information about distinguishing between the different labels. This is determined through some statistical measure such as entropy or Gini index. Each feature selection partitions training data until we are left with classified labels (leaves). The randomized sampling reduces the variance in decision tree's performance procedure. We selected four ensembles: Random Forest, Bagging Classifier and AdaBoost. Random Forest is a well-known and highly successful ensemble technique that learns a large collection of decision trees by sampling training data, and predicts labels on testing data which have majority votes from these sampled trees. It samples both the training instances as well as features. Assume S defines the training data with 5 features (F_1-F_5) with values from A-E:

$$S = \begin{bmatrix} F_1 & F_2 & F_3 & F_4 & F_5 \\ A_1 & B_1 & C_1 & D_1 & E_1 \\ . & . & . & . & . \\ \dot{A}_n & \dot{B}_n & \dot{C}_n & \dot{D}_n & \dot{E}_n \end{bmatrix} \tag{1}$$

Random Forest can then create two random samples like S1 and S2 as shown below:

$$S_1 = \begin{bmatrix} F_1 & F_2 & F_3 \\ A_4 & B_4 & C_4 \\ . & . & . \\ . & . & . \\ A_9 & B_9 & C_9 \end{bmatrix} \ldots S_m = \begin{bmatrix} F_3 & F_4 & F_5 \\ A_{22} & B_{22} & C_{22} \\ . & . & . \\ . & . & . \\ A_n & B_n & C_n \end{bmatrix} \tag{2}$$

A tree is learnt with each sample and the predicted label is one which has majority votes from all samples. For instance, if we three samples S_1, S_2 and S_3 and respective predicted labels are C_1, C_1 and C_2, then the predicted output is C_1. The Bagging Classifier works similar to Random Forest. The difference is that it samples only training rows but doesn't sample the features. So, the whole feature set whenever a node needs to be added to the tree. From above example, the Bagging Classifier might generate the following matrices:

$$S_1 = \begin{bmatrix} F_1 & F_2 & F_3 & F_4 & F_5 \\ A_4 & B_4 & C_4 & D_4 & E_4 \\ . & . & . & . & . \\ . & . & . & . & . \\ A_9 & B_9 & C_9 & D_9 & E_9 \end{bmatrix} \ldots S_m = \begin{bmatrix} F_1 & F_2 & F_3 & F_4 & F_5 \\ A_{44} & B_{44} & C_{44} & D_{44} & E_{44} \\ . & . & . & . & . \\ . & . & . & . & . \\ A_{79} & B_{79} & C_{79} & D_{79} & E_{79} \end{bmatrix} \tag{3}$$

Adaboost works for only binary classification problems and based on Python's Boosting Classifier, which creates a sequence of bags (samples) of training data, where each bag gives more weight to instances that were wrongly classified in the previous bag. The initial bag of samples is created similar to Bagging Classifier and is evaluated using weighted average. The misclassified instances are given more weight in next bag of samples so that they are learnt better. The process continues until the overall accuracy does not improve. Each training instance X_i is initially assigned a weight according to,

$$Weight\,(X_i) = \frac{1}{n} \tag{4}$$

where n represents the total instances. The misclassification rate is based on the difference between the total and correctly classified instances,

$$E_t = \frac{(Correct - N)}{N} \tag{5}$$

The error rate E$_t$ is the sum of average error of each training instance multiplied by the weight assigned to it,

$$\sum_{i=0}^{i=N} \frac{(W_i \times E_i)}{W} \tag{6}$$

This is used to compute the stage value, $stage = \frac{ln(1-E)}{E}$, which updates the weights of a training instance W$_i$ as,

$$W_i = W_i \times e^{stage+error} \tag{7}$$

Here, error is 1 for misclassified instance and 0 otherwise, i.e., weight will remain same for correctly classified instances.

3.1.2 Non-Ensemble classifiers

We selected five non-ensembles: Logistic Regression (LR), Naive Bayes (NB), K- Nearest Neighbor (K-NN), Multi-Layer Perceptron (MLP) and Support Vector Machines (SVM). LR measures relationship between the categorical label and predictors by estimating a probability distribution over the label values. We first acquire a linear combination of weights and features through

$$x = w_0 + w_1 * f_1 + w_2 * f_2 + \ldots. + w_n * f_n \tag{8}$$

and then calculate the probability distribution through

$$\sigma^x = e^x / (1 + e^x) \tag{9}$$

We update the weights using gradient descent to optimize the following cost function:

$$\sum_{i=1}^{n} y_i log(h(x_i) - (1 - y_i) log(1 - h(x_i))) \tag{10}$$

The NB algorithm is based on Bayes theorem with the naive independence assumption that all features are conditionally independent given the class label. Even though this is usually incorrect (since features are usually dependent), the resulting model is easy to fit and works remarkably well in many applications, including text classification, medical diagnosis, and systems performance management. The conditional probability for NB algorithm is calculates as:

$$P(C_i \mid X) = \frac{P(X \mid C_i) P(C_i)}{P(X)} \tag{11}$$

Where P(C$_i$|X) represents the posterior probability, P(C$_i$) represents the prior probability, P(X|C$_i$) represents the likelihood and P(X) represents probability of predictors.

K-NN is considered to be the most fundamental and simple classification algorithm and should be the first choice when there is little or no prior knowledge about the distribution of the data. It has seen successful applications in complex domains such as content retrieval, gene expression, protein-protein interaction, 3D structure prediction. K-NN works on similarity measure principle: if an instance is similar to the k nearest points in the training data set, then this instance will be assigned the majority class label of these k points. With total n features, every instance xi can be represented as a coordinate: $xi = (F_1, F_2, F_3, \ldots., F_n)$. The similarity between two instances x_i and x_j is calculated using Euclidean matric as

$$dist(x_i, x_j) = \sqrt{\sum_{i=1}^{n} (x_i - x_j)^2} \tag{12}$$

The MLP algorithm is the most applied type of artificial neural network on small- sized datasets. It was the first algorithm to give successful results in complicated image-based tasks such as object recognition and character and digit recognition. In this study, we used 2-layer MLP consisting 200 neurons in each layers with ReLU as activation function. The MLP is trained for 200 epochs using ADAM optimizer considering weight decay regularizer strength equals 0.0001. The model can be summed up as

$y = w*x + b$, where w and b are the weight and bias vectors which are the parameters learnt by the MLP, x is training data vector and y is the label. On each misclassification, the weight and bias vectors are updated using

$$w = w + lr * (expected - predicted) * x \tag{13}$$

and

$$b = b + lr * (expected - predicted) * x \tag{14}$$

where lr controls the rate of learning; its value is set to 0.001 initially which is scaled down by a factor of 10 after 100 epochs. In principle, every neuron of MLP is connected to every other neuron. Thus, number of parameters in the MLP used equals 40K which is described as,

$$(number\ of\ neurons\ layer1 * number\ of\ neurons\ layer2) + 1 \tag{15}$$

The SVM is a supervised learning algorithm that classifies the instances using a hyper-plane. It finds the optimal hyper-plane by maximizing the margins, i.e., the maximum distance between two instances, with each instance belonging to a different class. If the training data is $S = ((x_1, y_1), \ldots, (x_n, y_n)\}$ with feature vector $xi \in R^d$ and y_i as class label, the hyper-plane for binary classification problem can be defined as

$$f(x) = sign(w*x + b) \tag{16}$$

where w and b define the orientation and offset of the hyper- plane. SVM finds the optimal hyper-plane, i.e., to minimize (w^2) subject to

$$y_i(w*x_i + b) - 1 > 0,\ i = 1, 2, \ldots, n \tag{17}$$

For more details on all algorithms discussed in this section, please refer to [20,33].

3.2 Feature selection techniques

It is a common machine learning trend that all the features in the training data are not relevant, i.e., one or more features might not be correlated strongly with the label to influence the predictions. Feature selection techniques reduce feature cardinality by selecting the most relevant attributes. They can also potentially avoid over-fitting with more efficient model learning. We selected three state-of-the-art feature selection techniques which are also implemented in Python [20,33]: Mutual Information Feature Selection (MIFS), Random Forest Feature Selection (RFFS) and Recursive Feature Elimination (RFE). MIFS measures the amount of information acquired by considering two features X and Y together as:

$$I(X,Y) = \sum_{i=1}^{n} \sum_{j=1}^{n} P(x(i), y(j)) \leftrightarrow log \frac{P(x(i), y(j))}{P(x(i) \leftrightarrow P(y(j)))} \tag{18}$$

where i and j represent iterate over all column values of X and Y, and P(x(i), y(j)) represents the probability of a particular value pair of X and Y occurring together.

RFFS is based on the Random Forest algorithm (described in previous section). It estimates an importance score for each feature by considering the different feature sets available in the sampled trees. The importance is a measure of the purity of the feature set in a sample: a feature set is purer if the features are correlated more with each other than with the feature sets available in other samples. Finally, RFE is a greedy optimization technique based on backward selection technique. Initially, it builds model with all predictors. After then, it continuously eliminates the predictors from the dataset and estimates the performance with each new feature set. The output is the feature set with the best performance.

4 Materials and Methods

Our proposed methodology for predicting the performance of Pakistan's ODI cricket team is shown in Figure 5 which we describe in the following sections.

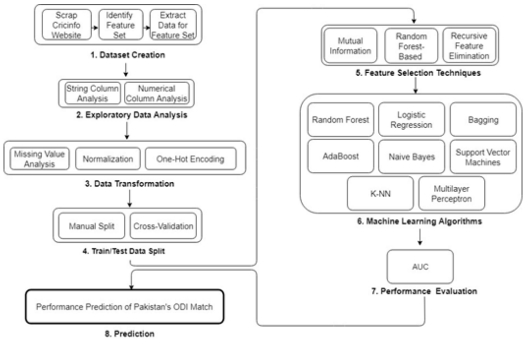

Fig 5. Proposed methodology

4.1 Dataset creation

The first task was to gather data related to Pakistan's ODI games. We did a thorough analysis of the following available online repositories: Cricsheet (cricsheet.org), ESPN's Cricinfo (www.espncricinfo.com), Cricbuzz (www.cricbuzz.com), NDTV Sports (sports.ndtv.com), FirstPost (www.firstpost.com). We found ESPN's Cricinfo website to be the best repository providing all historical match records with most comprehensive set of available attributes. We collected data for 19 years (January 2000-September 2019), a larger time period compared to any of related works. As per practice, we also wanted to gather as much data as possible for good learning and better predictions[20]. In all, we had 200 ODI matches. The description of each match is shown on a separate page on Cricinfo. We scrapped these pages using Python's Beautiful Soup API and used it to identify and extract data for nine categorical features (shown in Table 1). We also included a numerical feature Consecutive Wins (histogram shown in Figure 6).

4.2 Exploratory data analysis

Let us briefly discuss the features. Pitch Type represents the condition of the pitch, which we discretized into green, slow, bouncy, and dry; these types can have an impact on players' performance and match outcomes. For instance, it is easier to bat on a dry pitch than on a bouncy one, and a green pitch favors fast bowlers as compared to spin bowlers. Pakistan played most ODIs on dry pitches (90), twice as many on bouncy (45) and green (45) and four times as played on slow ones (20). The feature Season represents the weather conditions (temperature, humidity, wind etc.) which have impacted Pakistan team's performance to an extent that it can cause a slide of fallen wickets while Pakistan is batting (increasing chances of a loss) or bowling (increasing chances of a win). We discretized Season into winter, spring, summer and autumn and used the World Season Calendar[41] to determine the season at a given venue (each season is of thirteen weeks (shown in Table 2). From Table 1, we see that Pakistan played almost an equal number of matches in Autumn (40) and Summer (40), a little more in Winter (50) and most matches in Spring (70).

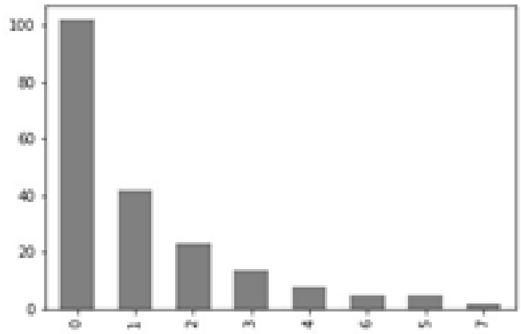

Fig 6. Histogram of feature consecutive wins

Opposition Team is a critical feature because Pakistan's performance has been strongly dependent on its opponent country. For instance, notwithstanding its "unpredictable" tag, Pakistan has found it easy to beat Bangladesh and Sri Lanka as compared to Australia, India and South Africa. From Table 1 we see that from January 2000 - September 2019, Pakistan played most matches against Sri Lanka followed by New Zealand, Australia and South Africa, while matches against England, Bangladesh, India and West Indies were less frequent. Matches against the newer cricketing nations like Afghanistan, United Arab Emirates (UAE) and Hong Kong are relatively infrequent. Home Ground represents whether Pakistan played on home ground (Yes) or not (No). Out of 200 ODIs, Pakistan has only played 2 matches on home ground as international teams stopped touring Pakistan due to security concerns. United Arab Emirates (UAE) has been the substitute venue in these years. From 2018 onwards, international cricket has returned to Pakistan due to enhanced security improvements. It is logical to assume that team wins are more predictable on home ground matches (for any sport). Therefore, Pakistan's handicap in this situation might offer some explanation of its unpredictable performance.

The feature Day/Night represents whether an ODI was played in whole day (No) or in both day and night (Yes). This is an important feature as Pakistan is generally considered losing more day/night matches than day matches. The team actually played more Day/Night matches (125) as compared to Day matches (75). Also, the feature Batting First represents whether Pakistan batted first (Yes) or bowled first (No). This could be an important predictor as batting first always creates a pressure on the opponent team in chasing down the score while batting second. In 200 matches, Pakistan batted first a little less frequently (90) as compared to bowling first (110) (Table 1).

Table 1. Categorical features extracted from cricinfo reports

Features	Frequency distribution
Pitch Type	Green (20), Slow (45), Bouncy (45), Dry (90)
Season	Autumn (40), Summer (40), Winter (50), Spring (70)
Opposition Team	Hong Kong (2), Afghanistan (2), UAE (3), Ireland (3), Others (11), India (12), Bangladesh (12), England (18), Zimbabwe (18), West Indies (20), South Africa (18), Australia (22), New Zealand (25), Sri Lanka (35)
Venue Country	UAE (60), Bangladesh (20), Sri Lanka (20), New Zealand (20), Zimbabwe (15), England (15), Australia (15), South Africa (15), West Indies(15), India(10), USA (2), Ireland (2), Pakistan (2), Others (2)
Home Ground	Yes (2), No (198)
Day/Night	No (75), Yes (125)
Batting First	Yes (90), No (110)
Venue Ground	Perth (1), Bristol (1), Cardiff (1), Delhi (1), Brisbane (1), Queenstown (1), Edinburg (1), Chittagong (1), Chennai (1), Nelson (1), London (1), Manchester (1), Oval (1), Bloemfontein (1), Dunedin (1), Melbourne (1), Leeds (1), Mohali (1), Kandy (1), Nottingham (3), Durban (4), Fatullah (4), Belfast (4), Port Elizabeth (4), Hamilton (4), Johannesburg (3), Bridgetown (4), Southampton (3), Cape Town (3), Chandigarh (3), Birmingham (3), Sydney (4), Napier (4), Kolkata (4), Lahore (4), Dublin (4), Centurion (4), Dambulla (4), Christchurch (4), Auckland (4), Adelaide (4), Pallekele (4) Groslset (5), Wellington (5), Hambantota (6), Providence (6), Harare (8), Colom o (8), Dhaka (17), Dubai (17), Sharjah (18), Abudhabi (25)
Match Won	Yes (90), No (110)

Table 2. The values of feature season based on World season calendar

Time Interval	North Hemisphere Cricket Grounds	Southern Hemisphere Cricket Grounds
21 DEC - 21 MAR	Winter	Summer
22 MAR - 20 JUN	Spring	Autumn
21 JUN - 19 SEP	Summer	Winter
20 SEP - 20 DEC	Autumn	Spring

The attribute Venue Country represents the country where the match was played. Out of 200 matches, 60 were played in UAE and an almost equal number of matches at other venues (Sri Lanka, Bangladesh, New Zealand, Australia etc.) This feature is also important as Pakistan has been known to perform better in Indian and Arabian countries (India, Bangladesh, UAE etc.) As compared to African and Australian ones (Australia, New Zealand, Zimbabwe etc.). We also included the feature Venue

Ground because it gives us more information to learn a better predictive model for Pakistan's performance. Moreover, the ground is characterized by its weather patterns and pitch; in the same country, the performance can vary over different grounds due to differing weather and pitch conditions. The feature Match Won shows weather Pakistan won (Yes) or lost the match (No), and is our label (what we want to predict). There is a balance: of the 200 matches played, Pakistan won 45% of them (90) and lost the remaining 55% (110) (Table 1). In other words, we don't have a label imbalance problem to bias our machine learning executions[(20)]. However, it is noteworthy that Pakistan lost more matches than they won. Finally, the numerical feature Consecutive Wins represents the count of consecutive wins at the time of a new ODI We consider it to represent the "winning rhythm" of the team, and has not been included before in any related work. Figure 6 represents the skewed histogram distribution; notably, from January 2000 - September 2019, Pakistan played 100 matches without any consecutive wins, around 40 matches after winning 2 matches consecutively and around 30 matches after winning 3 matches consecutively. In other words, it has been difficult to keep up the winning pace over a large number of matches.

4.3 Data transformation

Data transformation (also called cleaning or wrangling) is a necessary step for machine learning to prepare the data for modeling process. The first activity we considered is normalization, i.e., scaling all numerical attributes within the same range. Since we have just one numerical attribute, we don't normalize. We also didn't see any need to change discretization of any categorical attributes. Then, we dealt with missing values in our dataset; as per rule, less than 5% missing data in a column should be replaced by standard methods[(20)]. Only Pitch Type has a missing value for 13 matches, as this data was not available from Cricinfo for these matches. We decided not to ignore these rows at the risk of information loss for modeling. We determined these types, either from reports of other matches played at same venue, or from reports of neighboring cricket grounds whose pitches are expected to be of similar type[(5)]. Finally, we had to encode all categorical attributes into numerical format, because several machine learning algorithms don't accept categorical data as input (e.g., Multi- Layer Perceptron) and it's also required for better performance[(20,42)]. We considered both label encoding, which assigns a unique number to each unique label in a column, and one-hot encoding, which creates a separate binary column for each unique label which is true whenever that label occurs in a row (with values in all other created binary columns being false in the same row). For instance one-hot will generate Home Ground Yes and Home Ground No columns, while label will replace Yes and No in Home Ground with numbers "1" and "2". We selected one-hot method as label can confuse the learning algorithm in that value "1" is somehow larger than "2", although there is no numerical relationship in this case.

4.4 Train/Test data split

After the transformations, we split the pre-processed data into train and test sets. This split can be accomplished by using cross-validation or the manual split. Cross- validation works by creating multiple train-test samples (stratified sampling with replacement) to ensure that each instance is used for both training and testing. A fold value is specified to indicate the number of samples; we conducted the standard 10-fold cross-validation. The manual split is a one-time split procedure in which we sampled 80% of data for training and 20% for testing. We used both split types because it is difficult to determine in advance which one will give better results for our small-sized temporal data[(20,33)].

4.5 Machine learning algorithm execution

Our machine learning strategy is as follows: For each split type, and for each feature selection type, execute each of our eight selected algorithm and estimate the performance. This will help us determine the split -feature selection -algorithm combination that works best for our dataset, and which can be used as a reference for future re- searchers. To eliminate any effects of random seeding in Python's implementation, we performed each execution of an algorithm four times and averaged the result; Python uses random seeding to select default values of parameters in several algorithms and to sample data in ensemble methods. We considered accuracy and Area under the Curve (AUC) as evaluation metrics. We selected AUC because: 1) it avoids over fitting, 2) it is less biased towards the dataset as compared to accuracy, 3) it is robust to class imbalance (our cricket dataset is currently balanced but the future ODI matches could generate imbalance if we re-train our model in the future), 4) it gives a measure of how much reliable the prediction is with respect to a random behavior (i.e., in our case the unpredictable behavior of Pakistan's ODI team); AUC is hence considered to be more reliable in machine learning community[(4,20,33,43)].

5 Results and Discussion

The averaged AUC scores for Recursive Feature Elimination (RFE), Mutual Information Feature Selection (MIFS) and Random Forest Feature Selection (RFFS) are shown in Tables 3, 4 and 5 respectively. The algorithms' abbreviations are: RF (Random Forest), Bagging (Bagging Classifier), LR (Logistic Regression), NB (Naive Bayes),

Table 3. AUC scores for selected classifiers with Recursive Feature Elimination(RFE)

	Manual Split	Cross-Validation
Algorithms	AUC Score	AUC Score
RF	0.82	0.71
Bagging	0.8	0.71
AdaBoost	0.65	0.65
LR	0.73	0.79
NB	0.54	0.65
SVM	0.61	0.80
K-NN	0.79	0.54
MLP	0.68	0.56

Table 4. AUC scores for selected classifiers with Mutual Information Feature Selection (MIFS)

	Manual Split	Cross-Validation
Algorithms	AUC Score	AUC Score
RF	0.61	0.62
Bagging	0.61	0.62
AdaBoost	0.54	0.62
LR	0.54	0.51
NB	0.54	0.56
SVM	0.52	0.51
K-NN	0.68	0.42
MLP	0.54	0.78

Table 5. AUC scores for selected classifiers with Random Forest Feature Selection (RFFS)

	Manual Split	Cross-Validation
Algorithms	AUC Score	AUC Score
RF	0.66	0.65
Bagging	0.63	0.60
AdaBoost	0.61	0.74
LR	0.63	0.79
NB	0.56	0.74
SVM	0.63	0.815
K-NN	0.58	0.56
MLP	0.68	0.78

SVM (Support Vector Machines), K-NN (K-Nearest Neighbor) and MLP (Multi-layer Perceptron). We obtained the best (highest) AUC score of 82% for RF algorithm with manual split and RFE feature selection. This means that Pakistan's ODI performance can be predicted with 82% confidence as compared to any random prediction by the crowd and its consequent extreme reactions. We have also done better than the best performance in related work (70%), as accuracy values are generally close to AUC values for a given problem[4,43]. We note that an AUC greater than 70% is considered highly acceptable by both research and industry practitioners[20,33]. We can hence answer our research question by saying that the "unpredictable" tag is unjustified and the audience needs to have more knowledge of the situation to understand why the win or loss occurs. We extracted some of this knowledge by sampling three trees generated by RF (with manual split and RFE) and extracting their more useful patterns, listed below:

- When Pakistan has not won any consecutive matches in a given tournament, and it is going to play a day match on any venue country other than USA, against any opposition country besides Australia, New Zealand and Zimbabwe, then Pakistan will win the match.
- When Pakistan has not won any consecutive matches in a given tournament, and it is going to play in any venue country other than USA, against any opposition besides Zimbabwe, then Pakistan will lose the match.
- When Pakistan is going to play against New Zealand and Zimbabwe, in any venue country other than USA, then Pakistan will win the match.
- When Pakistan is going to play against Zimbabwe, in any venue country other than USA, and in any season besides autumn, then Pakistan will win the match.
- When Pakistan is playing in Bristol (England), and it either has not won any consecutive match, or has had one consecutive win, then Pakistan is going to lose the match.
- When Pakistan has either not won any consecutive match, or has had one consecutive win, and it is playing in any venue country besides Sri Lanka, in any season besides Spring, and against any opposition besides UAE, then Pakistan will win the match.

These patterns give us some concrete evidence about the different situations leading to a win or loss. Our RF ensemble (with manual split and RFE) consists of 159 trees and possibly hundreds to thousands of patterns which collectively form the RF classifier. For the sake of brevity and confidentiality, we cannot present all patterns in this paper.

We also plotted the AUC values for manual and cross-validation (CV) splits in Figures 7 and 8 respectively. Following are some important insights as guidelines for future machine learning researchers in cricket performance prediction:

- With manual split and RFE, Bagging, LR and K-NN algorithms also give AUC close to 80%
- With manual split, the best algorithm with RFFE is SVM with approximate AUC of 70%
- With manual split, the best algorithm with MIFS is K-NN with approximate AUC of 70%
- With manual split, best average performance (across all algorithms) is given by RFE (0.70), followed by RFFE (0.62) and MIFS (0.57)
- With CV split, the best performance is obtained by RFFE and SVM algorithm with AUC of 81.5%. MLP and LR also score close to 80% with RFFE
- With CV split, the best algorithm with MIFS is MLP with AUC close to 80%
- With CV split, the best algorithms with RFE are MLP and LR with AUC values close to 80%
- With CV split, best average performance (across all algorithms) is given by RFFE (0.71), followed by RFFE (0.68) and MIFS (0.58)
- MIFS is the worst feature selection method, while RFE works best with manual split and RFFE with CV splits
- Ensemble algorithms have performed relatively better across both splits (average AUC of 66%) as compared to non-ensembles (average AUC of 63%), although the difference is less significant in CV split case.

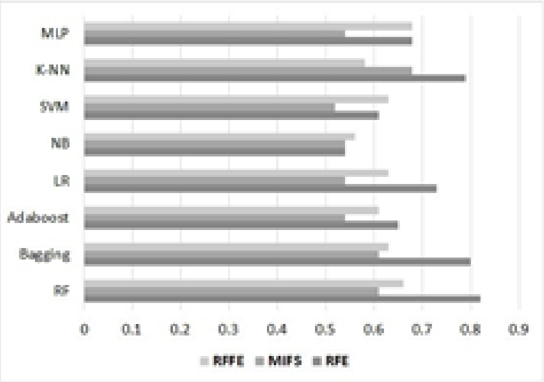

Fig 7. AUC with Manual Split

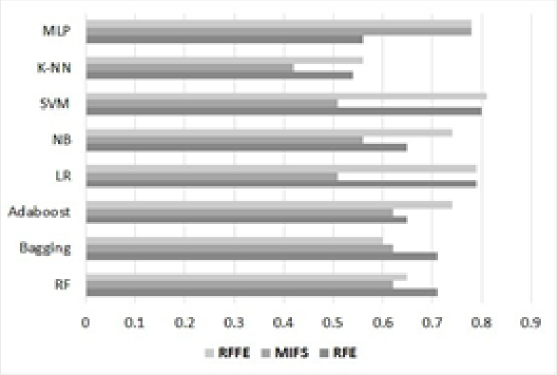

Fig 8. AUC with Cross-Validation

6 Conclusions and Future Work

Pakistan team's performance in ODI cricket has been globally labeled as ”unpredictable” which leads to extreme reactions from the society particularly on a loss. In this study, we have executed a comprehensive machine learning process to prove that the ”unpredictable” tag and its extreme reactions are unjustified and show that Pakistan's ODI performance can be predicted with a high confidence of 82% with random forest algorithm (manual data splits and with recursive feature elimination), i.e., it is possible to understand the patterns leading to a win or lose, and the crowd needs to be knowledgeable on these patterns to avoid their extreme reactions. Our exercise has revealed differing machine learning performance with respect to feature selection, data split method and algorithm selection. In this way, our work has presented a broader machine learning performance analysis in the domain of predicting cricket ODI performance, as compared to related work. However, random forest and other selected ensembles have performed better than their non-ensemble counterparts. Our future work involves the study of all patterns extracted by random forest and to encode them as part of an analytics software for the Pakistan Cricket Board. We will take some steps to further enhance the AUC scores: 1) continue to expand our dataset with data of future ODI matches and re-train our model, 2) add more attributes related to players' performances, e.g., batting and bowling averages and 3) test the performance on upcoming ODI matches of Pakistan etc.

References

1) Bloomberg. One of the world's most popular sports races to save itself . 2018. Available from: https://www.bloomberg.com/news/articles/2018-05-13/cricket-races-to-save-itself.
2) Wood R. World's most popular sports by fans. 2008. Available from: https://www.topendsports.com/world/lists/popular-sport/fans.htm.
3) Countries. 2019. Available from: http://www.cricketarchive.com/Archive/Countries/index.html.
4) Why is auc a better measure of an algorithm's performance than accuracy?. 2018. Available from: https://www.quora.com/Why-is-AUC-a-better-measure-of-an-algorithms-performance-than-accuracy.
5) Eastaway R. Cricket Explained. 1st ed. USA, 1 edition. Martin's Press. 1993.
6) Dawn.com. 'good night Pakistan': Twitter reacts to Pakistan's performance against India, June 2019. 2019.
7) Hasan S. Pakistan-India final brings enjoyment, enthusiasm for karachiites. 2019.
8) Rasheed TA. Another disappointing cricket match. 2015. Available from: https://nation.com.pk/05-May-2015/another-disappointing-cricket-match.
9) India TV News. Ranveer Singh hugs sad Pakistani fan after Ind-Pak match and wins hearts, watch viral video. 2019. Available from: https://www.indiatvnews.com/entertainment/celebrities-ranveer-singh-hugs-sad-pakistani-fan-after-ind-pak-match-and-wins-hearts-watch-viral-video-527922.
10) Verma A. 'Pakistan fans trashing their own team after India's world cup victory is pure gold. 2019.
11) Pakistan cricket team records & stats - espncricinfo. 2019. Available from: http://stats.espncricinfo.com/ci/engine/records/team/seriesresults.html?class=2.
12) Arshad M. Predictable unpredictability is Pakistan's strength. 2017.
13) Massey A. 5 occasions when cricket matches were called off due to violence. 2017.
14) Devastated Pakistan cricket fans smash tv sets after West Indies' defeat. 2016.
15) Samaa Digital. Distressed cricket fans break tvs. 2016.
16) India Today Web Desk. Don't use bad words: Mohammad Amir pleads with Pakistan fans after India defeat. 2019.
17) India Today Web Desk. 'kal raat ye log burger, pizza kha rahe the': Pakistani fan cries over team's loss, blames it on their lack of fitness. 2019.
18) [26]Press Trust of India. World cup 2019: Man files petition to ban Pakistan cricket team after embarrassing defeat to India. 2019.
19) Qazi B. The 7 stages of grief a Pakistani cricket fan experiences. 2013.
20) Marsland S. Machine Learning: An Algorithmic Perspective. and others, editor;USA. Chapman and Hall/CRC. 2014.
21) Reep C, Pollard R, Benjamin B. Skill and Chance in Ball Games. *Journal of the Royal Statistical Society Series A (General)*. 1971;134(4):623–623. Available from: https://dx.doi.org/10.2307/2343657.

22) Kimber CA, Hansford RA. A Statistical Analysis of Batting in Cricket. *Journal of the Royal Statistical Society Series A (Statistics in Society)*. 1993;156(3):443–443. Available from: https://dx.doi.org/10.2307/2983068.

23) Silva MBD, Pond RG, Swartz BT. Estimation of the magnitude of victory in one-day cricket rmit university, mayo clinic rochester and simon fraser university. *Australian & New Zealand Journal of Statistics*. 2001;43(3):259–268. Available from: https://doi.org/10.1111/1467-842X.00172.

24) Bailey M, Clarke RS. Predicting the match outcome in one day international cricket matches, while the game is in progress. *Journal of sports science & medicine*. 2006;5(4):480–480. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3861745/.

25) Clarke RS. Dynamic programming in one-day cricket-optimal scoring rates. *Journal of the Operational Research Society*. 1988;p. 331–337. Available from: https://doi.org/10.1057/jors.1988.60.

26) Bandulasiri A. Predicting the winner in one day international cricket. *Journal of Mathematical Sciences & Mathematics Education*. 2008;3(1):6–17. Available from: http://w.msme.us/2008-1-2.pdf.

27) Silva BMD, Swartz TB. Winning the coin toss and the home team advantage in one-day international cricket matches. Department of Statistics and Operations Research, Royal Melbourne Institute of Technology. 1998. Available from: https://www.researchgate.net/publication/2259664_Winning_the_Coin_Toss_and_the_Home_Team_Advantage_in_One-Day_International_Cricket_Matches.

28) Allsopp PE, Clarke RS. Rating teams and analysing outcomes in one-day and test cricket. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*. 2004;167(4):657–667. Available from: https://dx.doi.org/10.1111/j.1467-985x.2004.00505.x.

29) Choudhury DR, Bhargava P, Reena P, Kain S. Use of artificial neural networks for predicting the outcome of cricket tournaments. *International Journal of Sports Science and Engineering*. 2007;1(2):87–96. Available from: http://www.worldacademicunion.com/journal/SSCI/SSCIvol01no02paper02.pdf.

30) Sankaranarayanan VV, Sattar J, Lakshmanan LV. Auto- play: A data mining approach to odi cricket simulation and prediction. In: and others, editor. Proceedings of the 2014 SIAM International Conference on Data Mining. 2014;p. 1064–1072. Available from: https://doi.org/10.1137/1.9781611973440.121.

31) Kaluarachchi A, Aparna S, Varde. Cricai: A classification based tool to predict the outcome in odi cricket. In: and others, editor. Information and Automation for Sustainability (ICIAFs), 2010 5th International Conference. IEEE. 2010;p. 250–255. Available from: https://doi.org/10.1109/ICIAFS.2010.5715668.

32) Vistro, Mago D, Rasheed F, David LG. The Cricket Winner Prediction With Application Of Machine Learning And Data Analytics. *International Journal of Scientific & Technology Research*. 2019;8:985–990.

33) elien G´ eron A. Hands-On Machine Learning with Scikit-Learn, Keras, and Tensor-Flow: Concepts, Tools, and Techniques to Build Intelligent Systems. USA. O'Reilly Media. 2019.

34) Dezyre. Top 10 machine learning algorithms. 2016. Available from: https://www.dezyre.com/article/top-10-machine-learning-algorithms/202.

35) Fernandez-Delgado M, Cernadas E, en Barro S, Amorim D. Do we need hundreds of classifiers to solve real world classification problems. *Journal of Machine Learning Research*. 2014;15:3133–3181. Available from: http://jmlr.org/papers/v15/delgado14a.html.

36) Kharkovyna O. Top 10 machine learning algorithms for data science. 2019. Available from: https://towardsdatascience.com/top-10-machine-learning-algorithms-for-data-science-cdb0400a25f9.

37) Natingga D. Data Science Algorithms in a Week: Top 7 algorithms for scientific computing, data analysis, and machine learning. 2nd ed. USA. Packt Publishing. 2018.

38) Wu X, Kumar V. The Top Ten Algorithms in Data Mining. 1st ed. and others, editor;USA. Chapman and Hall/CRC. 2009.

39) Scikit-Learn Developers. Who is using scikit-learn? . 2019. Available from: https://scikit-learn.org/stable/testimonials/testimonials.html.

40) Luz D, Ribiere L. A quick summary and some thoughts on the scikit-learn workshop. 2017. Available from: https://blog.octo.com/en/a-quick-summary-and-some-thoughts-on-the-scikit-learn-workshop/.

41) Asimov I. The Tragedy of the Moon. USA. Dell Pub Co. 1978.

42) Handling character data for machine learning - dzone ai. 2017. Available from: https://dzone.com/articles/handling-character-data-for-machine-learning.

43) . 2014. Available from: https://datascience.stackexchange.com/questions/806/advantages-of-auc-vs-standard-accuracy.