A COMPARISON BETWEEN DIFFERENT CLASSIFIERS FOR TENNIS MATCH RESULT PREDICTION

Soumadip Ghosh[1], Shayak Sadhu[2], Sushanta Biswas[3], Debasree Sarkar[3], Partha Pratim Sarkar[3]

[1, 2] Academy of Technology, Aedconagar, Hooghly-712121, West Bengal, INDIA.

[3]Department of Engineering and Technological Studies, Kalyani, Nadia-741245, West Bengal, INDIA.

Email: soumadip.ghosh@gmail.com; shayakchemistry@gmail.com; biswas.su@gmail.com

DOI: https://doi.org/10.22452/mjcs.vol32no2.2

ABSTRACT

Tennis is a very popular sport in the world. Many researchers have worked in the fields of forecasting the outcome of tennis matches using past statistical data records. This paper mainly investigates the comparison between three different classifiers namely decision tree, learning vector quantization and support vector machine. The research study aims to predict the result of tennis singles matches using eight UCI databases of grand slam tennis tournaments and evaluate the classification performance using various measures such as the root-mean-square error, accuracy, false positive rate, true positive rate, kappa statistic, recall, precision, and f-measure. All these performance measures confirm the supremacy of the decision tree classification algorithm compared to the others.

Keywords: ***Classification algorithm, CART, LVQ, Support vector machine, Kappa statistic, Confusion matrix.***

1.0 INTRODUCTION

Tennis is one of the popular games both played and watched worldwide. It is a sport played either individually (singles) or in a team of two (doubles). There are four main grand slam tennis competitions occurring in every year which are namely Australian Open, French Open, Wimbledon and US Open. These four grand slam tournaments are the most famous tennis tournaments all over the world. Needless to say, the court surfaces of these mega tennis events are also different. Australian and US Open are to be played on hard courts, French Open is to be played on clay courts and Wimbledon is to be played on grass courts. Every court surface has its own features and creates variations in bounce and speed of the ball. Clay courts produce gentler paced ball and an equally accurate bounce with extra spin. Hard courts produce faster paced ball and very accurate bounces. Grass courts produce faster ball movements with added unpredictable types of bounces. Furthermore, the scoring systems of men's and women's singles matches in grand slam tournaments are also different. Typically, in men's matches, a player who wins three sets out of five sets wins the match. Whereas, in the women's matches, the first player winning two sets out of three sets wins the match.

Due to the growth of technology, predictions are widely used in tennis matches, especially by coaching staffs, news agencies and spectators. The tennis prediction model is developed to evaluate the chance of winning matches that the players will face. When a game is played, the result depends on many factors including the playing environment, player's skill and past match results. Many approaches such as statistical data evaluation have been used so far. But predicting the theoretical outcome of tennis matches is a challenging task and has been a keen interest for many researchers. The scope of this research area is more than sufficient for making significant improvement in the quality of prediction and the interpretation of results. The present research study basically aims to predict the outcome of a tennis singles match using past match records of the grand slam tournaments.

Data mining [1, 2] is a computational technique used for discovering useful knowledge from large data reservoirs. It is an essential step towards the discovery of valuable knowledge. In the field of research, various techniques of data analysis, including machine learning, artificial intelligence and other statistical analysis methods [3, 4] have been used. Artificial intelligence in combination with statistical analysis gives way to machine learning algorithms. The field of machine learning usually deals with classification algorithms [5, 6] that have the ability to learn from raw data. Classification is a significant data mining technique used for extracting useful information from a large-scale real-time database that matched a given pattern and assign items to a collection of target classes or categories. Classification is used in our work to predict tennis matches results.

Current research works employ some approaches to data classification such as decision tree learning, learning vector quantization and support vector machine. Artificial neural network (ANN) [7, 8, 9] is a computational model inspired by human central nervous systems used to estimate or approximate functions that depends on a large number of unknown input data. Fundamentally, Learning Vector Quantization (LVQ) [10, 11] is a popular neural network model that combines competitive learning with supervision. The present work uses this network model for tennis match result prediction. Support vector machine (SVM) [12] is another supervised learning model that can analyze data and recognize patterns used for classification and regression analysis. We use this powerful classification technique to perform prediction on match result. Decision tree (DT) [13, 14] learning approach considers decision tree as a prediction model that is a tree with internal nodes as each decision and leaf nodes as the result of the decisions made. This approach can be used to analyze our result as each path from the root to leaf node represents a solution for our problem.

In our present study, we consider three types of classifier namely decision tree, learning vector quantization and support vector machine. We apply these classifiers to eight benchmark UCI data sets to investigate the performances of these classifiers. We use different measures such as root-mean-square error, accuracy, false positive rate, true positive rate, kappa statistic, recall, precision, and f-measure for evaluating individual classifier performance.

We arrange this research study as follows: Section 2.0 covers the related works done in this area. Section 3.0 gives the description of the dataset being used while Section 4.0 explains the different classification techniques used in this research study. Section 5.0 describes the detailed procedure of classification. Section 6.0 presents the results of performance analysis, and Section 7.0 specifies the conclusion and future scope of extending our work.

2.0 RELATED WORKS

Predicting the outcome of a tennis match has been an interesting field to many researchers. The researchers so far have used various methods such as building statistical models, use of time series, classification, regression analysis, etc. Some of the relevant works done in this field of research are discussed in brief.

Tennis match prediction using independent and identical distributed Markov Chain and Revised Markov Chain is a significant work in this field proposed by T. Barnett, A. Brown, and S. R, Clarke [15]. The proposed model took each set as independent and identical distribution and set up a Markov Chain with any two players, such as A and B with the current score of (a, b), where $a \geq 0$, $b \geq 0$. Using this initial score, the procedure tried to find the probability of wining either A or B. As Markov chain is memory-less so the next state depends on the current state.

Time series analysis is also helpful in extracting useful statistical data that can be used for prediction. Use of time series history to extract useful pattern and predicting the result has been used by A. Somboonphokkaphan, S. Phimoltares, and C. Lursinsap [16]. They derived attributes based on the time series history of the data and used Multilayer Perceptron (MLP) based method to predict the winner of the tennis match. In this method, they used data warehousing to analyze the data for classification. As the environmental conditions also affect the match so they used both environmental and statistical data for estimation of the match results.

Use of player's characteristics can be considered as another useful way-out to predict the outcome of the match. The research study by J. D. Corral, and J. Prieto-Rodr´ıguez [17] demonstrated the use of physical and mental attributes of a player to calculate the chance of winning using the Probit model. The model is a type of regression in which the dependent variable can only take one of the two values (either yes or no).

It is possible to predict match result solely based on player's characteristics as the input parameter. This has been shown by A. Panjan, N. Šarabon, and A. Filipčič [18], where the dataset contains the various player physical attributes such as body weight. These parameters are taken into consideration to produce the result by using classification and machine learning procedures.

Spatio-temporal data are useful in predicting shots in Tennis as well as in forecasting match result. Predicting the outcome of other games has also been attempted X. Wei, P. Lucey, S. Morgan, and S. Sridharan [19]. Soccer prediction algorithm as designed by A. S. Timmaraju, A. Palnitkar, and V. Khanna [20] used KPP (k-past performance) for estimation of football match result. D. Buursma [21] used past statistical records to predict results of various other games while keeping the game constraints in mind.

3.0 ABOUT THE DATASET

The research study uses the benchmark Tennis Match Statistics dataset [22] that we consider in our work have been provided by UCI machine learning repository. In a year, there are four major tennis tournaments to be held and they are namely Australian Open, French Open, Wimbledon, and US Open. In total, there are eight datasets considering all men's and women's grand slam tournaments. All of the datasets have a common format and consists of 42 attributes and 127 tuples. The database is detailed in details in Table 1.

Table 1: Dataset attribute list along with their description

Sl. No.	Attribute name	Value Type	Attribute Description
1	Player 1	String	Name of Player 1
2	Player 2	String	Name of Player 2
3	Result of the match	0/1	Referenced on Player 1 is Result = 1 if Player 1 wins (FNL.1>FNL.2)
4	FSP.1	Real Number	First Serve Percentage for player 1
5	FSW.1	Real Number	First Serve Won by player 1
6	SSP.1	Real Number	Second Serve Percentage for player 1
7	SSW.1	Real Number	Second Serve Won by player 1
8	ACE.1	Integer Number	Aces won by player 1
9	DBF.1	Integer Number	Double Faults committed by player 1
10	WNR.1	Number	Winners earned by player 1
11	UFE.1	Number	Unforced Errors committed by player 1
12	BPC.1	Number	Break Points Created by player 1
13	BPW.1	Number	Break Points Won by player 1
14	NPA.1	Number	Net Points Attempted by player 1
15	NPW.1	Number	Net Points Won by player 1
16	TPW.1	Number	Total Points Won by player 1
17	ST1.1	Integer Number	Set 1 result for Player 1
18	ST2.1	Integer Number	Set 2 Result for Player 1
19	ST3.1	Integer Number	Set 3 Result for Player 1
20	ST4.1	Integer Number	Set 4 Result for Player 1
21	ST5.1	Integer Number	Set 5 Result for Player 1
22	FNL.1	Integer Number	Final Number of Games Won by Player 1
23	FSP.2	Real Number	First Serve Percentage for player 2
24	FSW.2	Real Number	First Serve Won by player 2
25	SSP.2	Real Number	Second Serve Percentage for player 2
26	SSW.2	Real Number	Second Serve Won by player 2
27	ACE.2	Integer Number	Aces won by player 2
28	DBF.2	Integer Number	Double Faults committed by player 2
29	WNR.2	Number	Winners earned by player 2
30	UFE.2	Number	Unforced Errors committed by player 2
31	BPC.2	Number	Break Points Created by player 2
32	BPW.2	Number	Break Points Won by player 2
33	NPA.2	Number	Net Points Attempted by player 2
34	NPW.2	Number	Net Points Won by player 2
35	TPW.2	Number	Total Points Won by player 2
36	ST1.2	Integer Number	Set 1 result for Player 2
37	ST2.2	Integer Number	Set 2 Result for Player 2
38	ST3.2	Integer Number	Set 3 Result for Player 2
39	ST4.2	Integer Number	Set 4 Result for Player 2
40	ST5.2	Integer Number	Set 5 Result for Player 2
41	FNL.2	Integer Number	Final Number of Games Won by Player 2
42	T_Round	Integer Number	Round of the tournament at which game is played

All the attributes listed in the given database denote their usual meanings related with the game of tennis. Among the 42 attributes considered, attribute number 3 is the class attribute that indicates the result of the match. The dataset designates that the result of the singles matches as either 1 or 0 with respect to player 1. It is taken to be 1 if player 1 wins or 0 otherwise. Therefore, the study considers two classes in this dataset namely class 1 and class 0.

The remaining 41 attributes are the input attributes. The first two input attributes denote the names of player 1 and player 2 respectively. The attributes with serial numbers 4 to 22 are referenced on player 1. The fourth attribute named FSP.1 indicates the first serve percentage for player 1 and the fifth attribute termed FSW.1 denotes the first serve won by player 1. The sixth attribute named SSP.1 specify the second serve percentage for player 1 and the seventh attribute termed SSW.1 means the second serve won by player 1. The eighth attribute termed ACE.1 denotes the aces won by player 1 and the ninth attribute called DBF.1 designates the double faults committed by player 1. The tenth attribute WNR.1 means winners earned by player 1 and the eleventh attribute termed UFE.1 indicates the unforced errors committed by player 1. The twelfth attribute termed BPC.1 denotes the break points created by player 1 and the thirteenth attribute called BPW.1 indicates break points won by player 1. The fourteenth attribute named NPA.1 denotes the net points attempted by player 1 and the fifteenth attribute termed NPW.1 indicates the net points won by player 1. The attribute number 16 is termed as TPW.1 and it denotes the total points won by player 1. The attributes with serial numbers 17 to 21 uses the common variable format STX.1 where X is the set number. The term STX.1 denotes the result of set X for player 1 with X = 1, 2, 3, 4, 5. The attribute number 22 is named FNL.1 and it indicates the final number of games won by player 1. The attributes with serial numbers 23 to 41 are referenced on player 2 and they denote the same sequence of properties as followed by attributes 4 to 22. The last attribute named T_Round indicates the round of the tournament at which the current game is played.

It is observed that some of the input attributes have missing values. The attributes namely ST3.1, ST4.1, ST5.1, ST3.2, ST4.2 and ST5.2 may contain N/A values. In men's singles tournament datasets, the attributes ST4.1, ST5.1, ST4.2 and ST5.2 may assume N/A values when fourth and fifth sets are not played. But, these attributes are not considered as valid attributes in case of women's tournament datasets. Similarly, the women's singles match sometimes may ignore playing the third set; so the value of the attribute ST3.1 and ST3.2 may denote N/A values. But, for the game of tennis the correctness of the dataset should be maintained even if the missing values are present. These missing values create a problem to the classification step as they are treated as non-numeric values and can alter the overall prediction result of the classifier model. So, the work should modify the dataset in such a way that the N/A values are replaced by appropriate values such that the output of the tuples is not altered and the consistency of the dataset is maintained in accordance to the rules of tennis.

4.0 CLASSIFICATION TECHNIQUES USED

Data classification takes place in two steps. The first step is the learning step also known as the training phase, where classification model is built upon the rules based on the training data. Each tuple in the training data is mapped to predetermined class. This helps the classifier to build rules based on these tuples. This learning helps the classifier to predict the type of class based on the input. There can be two types of learning model. Supervised learning is the first type where class label for each tuple in the training dataset is given. It is called supervised because each training tuple is mapped to a given class label. This first step of classification can also be described as the learning of a mapping or a function $y = f(x)$, where the class label y is mapped with the tuple x. Using these rules a classifier model is built. In the second step, the model used for classification is used to predict the class labels for a set of tuples that is taken as the testing data. This is used to estimate the predictive accuracy of the classifier. The accuracy of a classifier on a given set of test inputs can be given by the percentage of test inputs that has been correctly classified by the classifier.

In this work, we have carefully selected three different classification models to deliver best performance for this type of dataset. In the present work, we have selected the following classifiers:

- Decision Tree (DT)
- Learning Vector Quantization (LVQ)
- Support Vector Machine (SVM)

These three are very popularly used models and provides us the most possible accurate result along with minimizing the error occurring during training the model. The configuration related descriptions for each of the models are given below.

4.1 Decision Tree (DT)

Decision tree [13, 14] is a classification model in which a decision tree learns from the tuples in the training dataset. A decision tree appears like a flowchart in a tree like structure, where each internal node denotes condition testing on an attribute, each branch resulting from that node denotes the outcome from the test. The leaf node in the decision tree holds a class label. There are three commonly known variations of decision tree algorithms. They are namely ID3 (Iterative Dichotomiser), C4.5 and CART (Classification and Regression Trees). These algorithms employ a greedy approach. Most of these methods follow a top-down approach, in which the tree starts with a set of tuples and their corresponding classes. An attribute selection needs to be done for obtaining the best splitting criterion that can correctly and accurately separate the tuples into classes. This attribute selection method (also called splitting rules) determines how the tuples are to be divided. The attribute having the best score is selected as the splitting attribute.

In the present work, we consider **Classification and Regression Trees (CART)** algorithm which uses *Gini index* for selection of attribute. The gini index is based upon the amount of impurity contained in a set of tuples D (i.e., a database). This can be represented as:

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2 \tag{1}$$

where p_i is the probability that a tuple that belong to D falls under an arbitrary class C_i. This is done for a range of m classes. Binary splitting is done for each attribute in Gini index. For splitting on the attribute A, we consider two partitions D_1 and D_2 from D, the *Gini index* of the partitioning can be given as:

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \tag{2}$$

All possible combination of partitions is considered for each attribute A in the given data. The subset that gives the minimum Gini index is selected as the splitting subset. After this selection we need to calculate the amount of impurity present after the occurrence of binary splitting. This can be calculated as:

$$\Delta Gini(A) = Gini(D) - Gini_A(D) \tag{3}$$

The attribute which has the maximum reduction of impurity after splitting is selected. This attribute is designated as the splitting attribute in CART. After building the tree, there exists some unnecessary branching of the tree which reflects the irregularity of the training data due to noise. Tree pruning is done to solve this problem of overfitting of data. This method uses the help of statistics to remove the least reliable branches. A pruned tree is simple, faster and performs classification faster than the unpruned one.

There are two general approaches to tree pruning: pre-pruning and post-pruning. As already specified, the research study employs CART for selection of splitting attribute and building the initial decision tree model. After tree construction, *minimal cost complexity pruning* algorithm is used which is a post-pruning approach. This algorithm produces a decision tree classifier with minimum cost complexity. After the classifier is built, the testing data is fed to it to generate the classified labels. Using the given UCI datasets, we have seen that the result is given whether player 1 wins or not. Thus, binary classification that is taken in CART is suitable to classify the test dataset. Table 2 below provides the list of configuration parameters used in the given CART model.

Table 2: Configuration parameters of the CART model

Parameter	**Value**
Attribute selection measure	Gini index
Minimal number of instances at terminal nodes	2
Pruning approach used	Post-pruning approach
Pruning algorithm name	Minimal cost complexity pruning
Number of folds used	5
random seed number	1

4.2 Learning Vector Quantization (LVQ)

Learning Vector Quantization [10, 11] is one of the most commonly used classification model. The LVQ is a supervised version of vector quantization technique that is used for labeled input data. This learning technique uses the class information to relocate the Voronoi vectors slightly for improving the quality of classifier decision regions. It is a two stage process– a self-organizing map (SOM) followed by LVQ. Essentially, the first step is the feature selection indicating an unsupervised recognition of a realistically small set of features in which the vital information content of the input data is intensified. The second step is the classification phase where the feature domains are allocated to separate classes. Basically, the LVQ is made up of two layers (excluding the input layer) – competitive layer and linear layer. The first layer includes a competitive sub-network in which each neuron is allocated to a class. Different neurons in the first layer can be allotted to the same class. Each of those classes is then allocated to one neuron in the second layer. The number of neurons present in the first layer, Q, will thus always be at least as large as the number of neurons present in the second layer, M. After building the model, the test data is then provided to the classifier to predict the output class.

The value of M is same as the number of classes present in the data set. The selection of the number of neurons available in the hidden layers, denoted by, Q, is an important constraint. Basically, a thorough investigation helps us in selecting the number of PEs present in the hidden layer [23]. The number of hidden layer PEs is given by the following equation as

$$Q = \left(number\ of\ inputs + number\ of\ outputs\right) * \frac{2}{3} \tag{4}$$

The idea is to apply LVQ1 learning rule followed by LVQ2.1 learning rule. Table 3 below provides the list of configuration parameters used in the given LVQ model.

Table 3: Configuration parameters of the LVQ model

Parameter	**Value**
Number of hidden layers	One
Learning function used	LVQ2.1 weight learning function
Training epochs	150
Learning rate	0.1
Distance function	Euclidean distance

4.3 Support Vector Machine (SVM)

Support Vector Machine [12] is a promising model for classification of both linear and non-linear data. SVM uses non-linear mapping to transform the linear dataset into a higher dimension. The model searches for the separating hyperplanes between classes. An individual hyperplane is a decision boundary to separate two classes. Support vectors are the essential training tuples from the training dataset. With a sufficiently high dimension and appropriate non-linear mapping, two classes can be separated using support vectors and margins defined by these support vectors. Training of SVM is extremely slow, but is very accurate due to their ability to model non-linear decision boundaries. This is why SVM has been selected to perform prediction on tennis match result.

In our present work, the optimal configuration for developing an SVM classifier is described here. Several possible combinations like the number of folds used, value of random seed, and different kernel based techniques are investigated in simulation. Finally, an SVM model with a *Gaussian radial-basis function (RBF) kernel* is selected for match result prediction. A non-linear version of SVM can be represented by using a kernel function K as:

$$K\left(x_i, x_j\right) = \varphi\left(x_i\right).\varphi\left(x_j\right) \tag{5}$$

Here $\varphi\left(x\right)$ is the non-linear mapping function employed to map the training instances. An SVM model with a Gaussian RBF kernel is defined as:

$$K\left(x_i \,.\, x_j\right) = e^{\frac{-\left\|x_i - x_j\right\|^2}{2\sigma^2}} \tag{6}$$

Table 4 below provides the list of configuration parameters used in the given SVM model. All these parameters have their usual meanings.

Table 4: Configuration parameters of the SVM model

Parameter	**Value**
Type of kernel used	Non-linear
Kernel name	Gaussian radial-basis function (RBF)
Cache size	250007
Value of σ	0.01
Complexity parameter	1.0
Number of folds used	-1
Random seed value	1
Epsilon value for round-off error	1.0e-12
Tolerance Parameter	0.001

5.0 DETAILED PROCEDURE

Basically, we have to consider four datasets for each of the men's and women's grand slam tennis tournaments. Thus, in total we have to consider eight datasets in a calendar year. The detailed procedure is divided into two major steps- *data preprocessing* followed by *data classification*. Initially, the researchers apply some data preprocessing techniques to the original data. The preprocessing procedure involves different techniques such as data cleaning and data transformation. After preprocessing, one should build a model that will serve the purpose of predicting the output labels (as a win or loss in this case) from the given data. Using well-known performance evaluation statistics, we try to compare among the classifiers namely DT, LVQ and SVM.

5.1 Data preprocessing

Initially, the following *data preprocessing techniques* are applied to the dataset before the classification task —

***Data cleaning*:** Data cleaning is one of the most important steps to be considered while considering classification of the dataset. Data cleaning makes an attempt to fill in missing values, smoothening of the noise present in the dataset and also correcting the inconsistency present in the dataset. For this dataset, the study considers two main preprocessing filters: replacing missing values and replacing N/A elements with suitable values without violating the rules of a tennis match. In this dataset, the attributes labeled NPA.1, NPW.1, NPA.2 and NPA.2 which represents the net points attempted by player 1 and 2 respectively are the attributes with missing values. The reason for this is that the particular player has not attempted for any net point in the tennis match. A missing value is normally substituted by the arithmetic mean for that attribute based on statistics. The dataset attributes ST3.1, ST4.1, ST5.1, ST3.2, ST4.2 and ST5.2 also contain N/A values. These attributes represent the set results for each player. They can be denoted as N/A if that set has not been played by the particular set of players and the match result have been already decided. The work replaces them by appropriate values so that it does not conflict with the final result of the game being played.

***Data transformation*:** The procedure normalizes the datasets as because ANN based techniques require distance measurements in the training phase. It converts attribute values to small-scale ranges like 0.0 to 1.0 or -1.0 to +1.0.

5.2 Data classification

Afterwards, the tennis match dataset is distributed into two disjoint sub-sets, namely the training set and the test set. Basically we employ two different techniques for distributing the training and test datasets separately. They are namely *10-fold cross-validation* and the *70%-30% distribution* among the training and test datasets. In the present work, we employ three well-known classification techniques namely Decision Tree (DT), Learning Vector Quantization (LVQ) and Support Vector Machine (SVM) for training and testing purposes using the benchmark eight tennis match UCI databases. Finally, we compare the results generated by individual classifiers for quantitative analysis. The major steps of the detailed procedure are depicted below in Fig. 1.

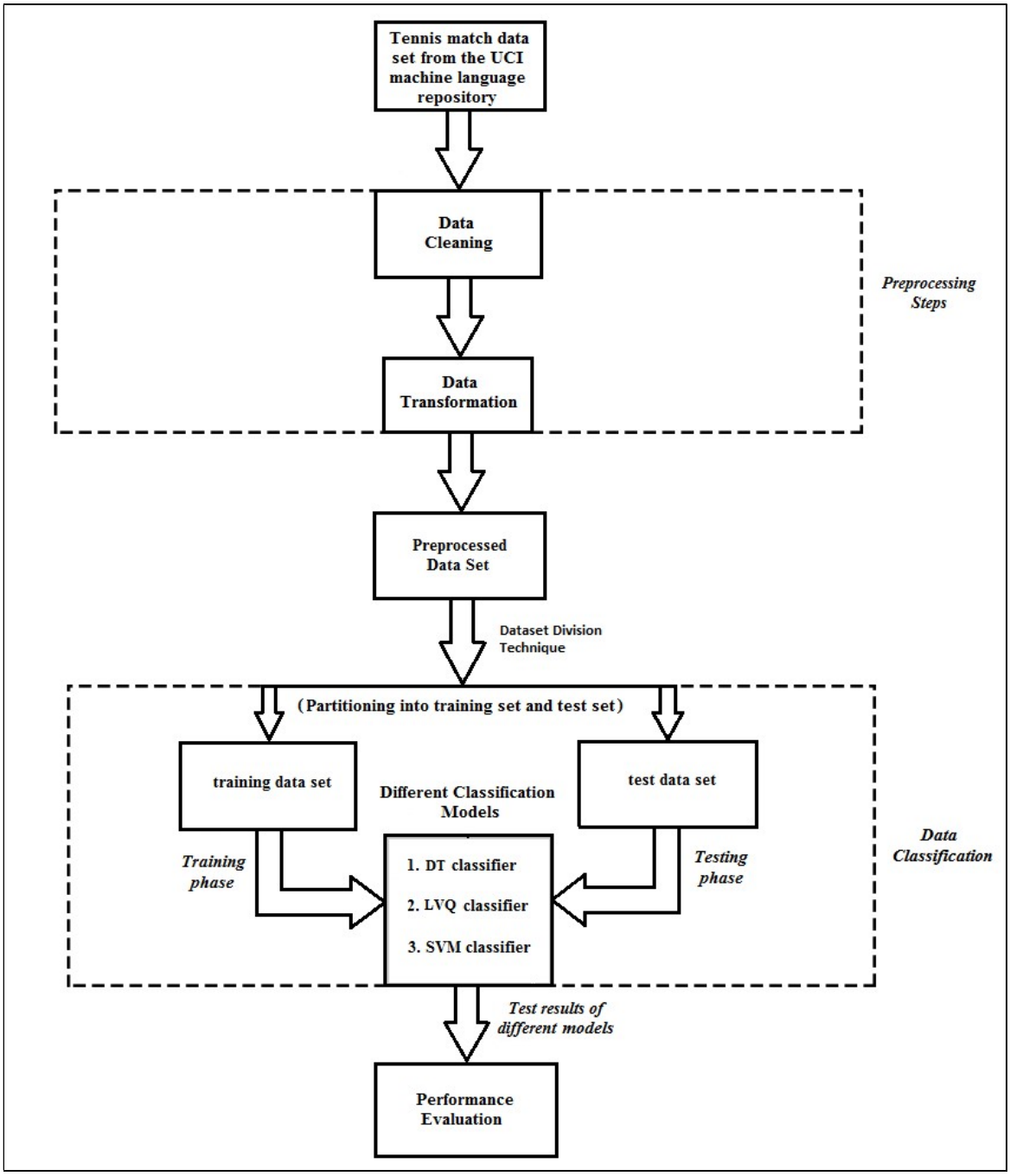

Fig. 1: Major steps of the detailed classification procedure

6.0 RESULTS AND DISCUSSION

The three classification techniques namely Decision Tree (DT), Learning Vector Quantization (LVQ) and Support Vector Machine (SVM) are trained and tested on the benchmark tennis match databases using the MATLAB software (version R2015a). The dataset are divided into Men's and Women's singles match results and are trained and tested to generate different statistics for the accuracy evaluation of each of the aforesaid classifiers.

6.1 Performance Measures

After building the classification model as mentioned in Section 4.0 we test the model using the testing dataset for evaluating performance for each type of classifiers. The research work estimates the performances of these classification models on the basis of different performance measures described below.

6.1.1 Root-mean-square error (RMSE)

RMSE [24] is well-known performance measure measuring dissimilarity between the values predicted by a classifier and the values actually found from the system being modeled. The RMSE of a classifier's estimation with regard to the calculated variable $e_{classifier}$ is the square root of the mean-squared error:

$$RMSE = \frac{\sqrt{\sum_{k=1}^{n}\left(e_{discovered,k} - e_{classifier,k}\right)^2}}{n} \tag{7}$$

where $e_{discovered}$ are the discovered values and $e_{classifier}$ are the predicted values for $\forall k$. Here, n denotes the number of data records present in the database.

6.1.2 Kappa statistic

The Kappa statistic [25], represented by κ, is a well-known performance metric in statistics. It is the measure of reliability among different raters or judges. The following equation estimates the value of κ as:

$$K = \frac{prob(O) - prob(C)}{1 - prob(C)} \tag{8}$$

Here *prob(O)* is the probability of witnessed settlements amongst the raters, and *prob(C)* is the probability of settlements estimated by coincidence. Basically, the magnitude of κ lies between 0 and 1. If $\kappa = 1$, the judges have approved each other's decision. If $\kappa = 0$, then the judges do not agree with each other. There is a set of guidelines to interpret the magnitude of kappa statistic. If the kappa statistic value is less than 0 it indicates **"no agreement"**. A kappa statistic confidence interval of 0–0.20 is denoted as **"slight"**, 0.21–0.40 as **"fair"**, 0.41–0.60 as **"moderate"**, 0.61–0.80 as **"substantial"**, and 0.81–1 as **"almost perfect agreement"**.

The weighted kappa statistic allows us to count disagreements in a different way and is particularly useful when codes are ordered. Basically, three matrices are used, the matrix of the observed scores, the matrix of the expected scores based on chance agreement, and the weight matrix. The weight matrix cells positioned on the diagonal (top-left to bottom-right) represent agreement and thus contain zeros. The off-diagonal cells contain weights indicating the significance of that disagreement. Every so often, cells one off the diagonal are weighted 1, those two off are weighted 2, etc. The equation for weighted κ is:

$$K = 1 - \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} w_{ij}\, x_{ij}}{\sum_{i=1}^{k}\sum_{j=1}^{k} w_{ij}\, m_{ij}} \tag{9}$$

where k = number of codes and w_{ij}, x_{ij}, m_{ij} and are elements in the weight, observed, and expected matrices, respectively. When diagonal cells contain weights of 0 and all off-diagonal cells weights of 1, this formula produces the same value of kappa as the calculation given above.

6.1.3 Confusion matrix

In the soft computing field, the confusion matrix [26, 27] is a specific tabular representation illustrating a classification algorithm's performance. It is a table layout that permits more thorough analysis than accuracy. Each column of the matrix denotes the patterns in a predicted class while each row indicates the patterns in the actual class. Table 3.1 below displays the confusion matrix for a two-class classifier with the following data entries:

- True positive (tp) indicates the number of 'positive' patterns classified as 'positive.'
- False positive (fp) means the number of 'negative' patterns classified as 'positive.'
- False negative (fn) denotes the number of 'positive' patterns classified as 'negative.'
- True negative (tn) implies the number of 'negative' patterns classified as 'negative.'

Table 5: A confusion matrix for a two-class classifier

		Predicted Class	
		Positive	*Negative*
Actual Class	*Positive*	tp	fp
	Negative	fn	tn

A two-class confusion matrix defines several standard terms. The accuracy is the sum of the correctly classified examples divided by the total number of examples present. The following equation calculates this as:

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{10}$$

The *precision* is the ratio of the predicted positive examples found to be correct, as designed using the equation:

$$precision = \frac{tp}{tp + fp} \tag{11}$$

The *fp-rate* is the ratio of negative examples incorrectly classified as positive, as determined using the equation:

$$fp - rate = \frac{fp}{fp + tn} \tag{12}$$

The *tp-rate* or *Recall* is the ratio of positive occurrences discovered correctly, as estimated using the equation:

$$recall = tp - rate = \frac{tp}{tp + fn} \tag{13}$$

In some situations, high *precision* may be more relevant while sometimes high *recall* may be more significant. However, in most representations, one should try to improve both values. The combined form of these values is called the *f-measure*, and usually expressed as the harmonic mean of both these values:

$$f - measure = \frac{2 * precision * recall}{precision + recall} \tag{14}$$

6.2 Results and Performance Analysis

Considering the testing phase of DT, LVQ and SVM classifiers, the four testing datasets are applied on each of the individual classifiers and the performance analysis for each of them is described below. The results are divided into two parts: Men and Women. We also consider all four tennis major tournament and place labels such as:

- Australian Open (in mid-January) = 1
- French Open (in May/June) = 2
- Wimbledon (in June/July) = 3

- US Open (in August/September) = 4

6.2.1 Men's Tennis Match Tournaments

The dataset for men and women have the same attribute list as given in Table 1. Each of the three classifiers namely DT, LVQ, and SVM are applied to the four test datasets for classification. The performance comparisons of these classifiers are done on the basis of different evaluation measures like classification accuracy, root-mean-square error, and the weighted kappa statistic as shown below in Table 6. The results suggest that these measures are the averages of individual classifiers corresponding to a grand slam tennis tournament mentioned earlier.

Table 6: Comparisons of the classifiers on the test dataset of Men's Tennis Major Tournament

Classifier	Dataset	70%-30% distribution			10-fold cross-validation		
		Accuracy (%)	RMSE	Kappa statistic	Accuracy (%)	RMSE	Weighted kappa
DT	1	99.32	0.1618	0.9437	98.65	0.1844	0.9439
	2	99.13	0.1629	0.9192	98.61	0.1842	0.9444
	3	98.74	0.1693	0.9425	98.96	0.1906	0.9395
	4	99.37	0.1622	0.9486	97.59	0.1825	0.9354
LVQ	1	93.57	0.2357	0.8827	93.13	0.2523	0.8797
	2	93.84	0.2365	0.9129	93.27	0.2459	0.9012
	3	92.79	0.2516	0.8784	92.35	0.2617	0.8878
	4	92.65	0.2793	0.8281	92.07	0.2892	0.8995
SVM	1	94.17	0.2022	0.9482	93.12	0.2144	0.9444
	2	94.17	0.2024	0.9447	93.11	0.2164	0.9436
	3	94.29	0.2091	0.9364	92.12	0.2126	0.8967
	4	94.31	0.2104	0.9466	92.56	0.2215	0.9396

After testing, we observe that DT classifier is having an average accuracy of **99.14%** in case of 70%-30% distribution and **98.45%** if we use 10-fold cross-validation as referred to Table 6. In comparison to this, the LVQ model is having an average accuracy of **93.21%** using 70%-30% distribution and **92.7%** using 10-fold cross-validation; while SVM is having an average accuracy of **94.23%** for 70%-30% distribution and **92.73%** for 10-fold cross-validation. We have also done performance evaluation based on RMSE and Kappa statistic measures as shown in Table 6. The performance evaluation of each of the classifiers has employed common statistical measures like RMSE which is intended to be kept as low as possible. The result shows that DT has the lowest RMSE value, followed by SVM and LVQ classifiers. To evaluate the accuracy for distinguishing classified data and their validity, we have also used kappa statistics as shown in Table 6. The kappa statistics shows a variation between **0.8-1.0** with DT as the classifier having the highest value and LVQ with the lowest value. The kappa statistic values within the confidence interval of 0.8-1.0 indicate **"almost perfect agreement"**. Based on the statistical measures of different classifiers employed to classify the datasets, DT gives us a moderately better result compared to SVM and LVQ.

Next, the performance evaluation is done based on the confusion matrix of individual classifiers. Different metrics such as TP-Rate/Recall, FP-Rate, Precision, and F-Measure values are calculated in accordance with the generated confusion matrix. The detailed result can be shown in Table 7 below. For a classifier, we expect that it should have higher TP-rate, precession, recall and F-measure values while having lower FP-rate value.

Table 7: Detailed accuracy for different classifiers on Men's Tennis Major Tournament's dataset

Classifier	Data set	70%-30% distribution				10-fold cross-validation			
		TP-Rate /Recall	FP-Rate	Precision	F-Measure	TP-Rate /Recall	FP-Rate	Precision	F-Measure
DT	1	99.27%	1.12%	99.27%	99.27%	98.65%	2.38%	98.65%	98.65%
	2	99.82%	1.85%	99.82%	99.82%	98.61%	2.63%	98.61%	98.61%
	3	98.05%	2.23%	98.05%	98.05%	98.96%	1.12%	98.96%	98.96%
	4	99.37%	1.63%	99.37%	99.37%	97.59%	3.76%	97.59%	97.59%
LVQ	1	93.57%	7.29%	93.57%	93.57%	93.13%	7.84%	93.13%	93.13%
	2	93.84%	7.12%	93.84%	93.84%	93.27%	7.92%	93.27%	93.27%
	3	92.79%	8.18%	92.79%	92.79%	92.35%	7.95%	92.35%	92.35%
	4	92.65%	8.25%	92.65%	92.65%	92.07%	7.85%	92.07%	92.07%
SVM	1	94.17%	7.17%	94.17%	94.17%	93.12%	7.57%	93.12%	93.12%
	2	94.17%	5.24%	94.17%	94.17%	93.11%	5.95%	93.11%	93.11%
	3	94.29%	6.27%	94.29%	94.29%	92.12%	6.87%	92.12%	92.12%
	4	94.31%	6.32%	94.31%	94.31%	92.56%	6.97%	92.56%	92.56%

Table 7 shows that these evaluation measures use the weighted average values corresponding to one of the grand slam tennis tournaments mentioned earlier. It is observed that DT classifier demonstrates a higher precision and lower error rate which is better than SVM and LVQ classifiers. In fact, the average accuracy of DT classifier is more than 4%-5% compared to LVQ and SVM classifiers.

Considering F-Measure as the best performance evaluation derived from a confusion matrix, DT classifier establishes F-Measure values of **99.12%** using 70%-30% distribution and **98.45%** using 10-fold cross-validation. In comparison to this, the LVQ model is having an average F-Measure value of **93.21%** using 70%-30% distribution and **92.7%** using 10-fold cross-validation; while SVM is having an average F-Measure value of **94.23%** for 70%-30% distribution and **92.73%** for 10-fold cross-validation. The result certainly proves that DT produces superior performance compared to the others.

6.2.2 Women's Tennis Match Tournaments

The women's dataset is similar to the men's dataset having similar attributes that have been described in Table 1. The datasets are pre-processed keeping the final result same and also not violating any rules of the game of tennis. So, the three classification models, namely DT, LVQ, and SVM are applied to the test datasets for classification. We evaluate the performance of these classifiers along the base of different performance measures like classification accuracy, root-mean-square error, and the weighted kappa statistic value as presented below in Table 8.

Table 8: Comparisons of the classifiers using test dataset of Women's Tennis Major Tournament

Classifier	Dataset	70%-30% distribution			10-fold cross-validation		
		Accuracy (%)	RMSE	Kappa statistic	Accuracy (%)	RMSE	Weighted kappa
DT	1	98.53	0.1501	0.9471	97.87	0.1705	0.9797
	2	97.53	0.1601	0.9413	97.29	0.1731	0.9687
	3	98.57	0.1497	0.9425	98.15	0.1714	0.9773
	4	97.65	0.1785	0.9253	97.39	0.1805	0.9636
LVQ	1	93.17	0.2342	0.9217	92.77	0.2675	0.9124
	2	92.47	0.2456	0.9231	91.93	0.2631	0.9017
	3	92.75	0.2617	0.9062	92.35	0.2839	0.8961
	4	93.61	0.2855	0.9123	93.18	0.2943	0.8935
SVM	1	93.43	0.1701	0.9471	92.75	0.1905	0.9351
	2	92.73	0.1979	0.9471	92.35	0.1815	0.9267
	3	93.28	0.1893	0.9364	92.13	0.2014	0.9189
	4	92.65	0.2285	0.9264	92.35	0.2205	0.9203

Table 8 illustrates the different primary evaluation parameters based on the classification results. It can be observed that DT classifier is having an accuracy of **98.07%** if we use 70%-30% distribution and **97.67%** if 10-fold cross-validation is used; while the SVM classifier is having the accuracy value of **93.04%** if 70%-30% distribution is used and the accuracy value is **92.55%** in case of 10-fold cross-validation. This proves that DT classifier has slightly better accuracy than SVM classifier. Also, it has been observed that LVQ classifier produces an accuracy of **93.0%** for 70%-30% distribution and **92.55%** in case of 10-fold cross-validation. Basically, LVQ and SVM are performance wise very close to each other. The results obtained from each of these classifiers are evaluated for obtaining different statistical measures such as RMSE and kappa statistics. The kappa statistic measures of these classifiers lie within the confidence interval of 0.8-1.0. The kappa statistic values lying within this given confidence interval designate **"almost perfect agreement"**. The DT classifier gives the highest kappa statistic value and is performance wise moderately better than LVQ and SVM classifiers. According to Table 8 which shows the performance evaluation results, DT again comes out first compared to the LVQ and SVM classification models.

Next, the performance analysis is made based on the evaluation derived from confusion matrix. The TP-Rate/Recall, FP-Rate, Precision, and F-Measure values are calculated based on the generated confusion matrix. Each of the parameters as illustrated in Table 9 is of immense importance that can draw out the performance evaluation parameters from the result. Considering a classifier, it should have higher TP-Rate, Precision and F-Measure while having lower FP-Rate.

Table 9: Detailed accuracy of the classifiers on Women's Tennis Major Tournament's dataset

Classifier	Data set	70%-30% distribution				10-fold cross-validation			
		TP-Rate /Recall	FP-Rate	Precision	F-Measure	TP-Rate /Recall	FP-Rate	Precision	F-Measure
DT	1	98.51%	2.15%	98.51%	98.51%	97.77%	2.28%	97.23%	97.77%
	2	97.53%	2.65%	97.53%	97.53%	97.24%	2.87%	97.19%	97.24%
	3	98.57%	2.05%	98.57%	98.57%	98.25%	2.17%	97.95%	98.25%
	4	97.65%	2.43%	97.65%	97.65%	97.29%	3.66%	97.19%	97.29%
LVQ	1	93.17%	7.12%	93.17%	93.17%	92.77%	8.23%	92.77%	92.77%
	2	92.47%	8.02%	92.47%	92.47%	91.93%	8.44%	91.93%	91.93%
	3	92.75%	8.19%	92.75%	92.75%	92.35%	8.19%	92.35%	92.35%
	4	93.61%	8.67%	93.61%	93.61%	93.18%	8.02%	93.18%	93.18%
SVM	1	93.43%	6.27%	93.43%	93.43%	92.25%	6.79%	92.25%	92.25%
	2	93.73%	6.14%	93.73%	93.73%	92.37%	7.34%	92.37%	92.37%
	3	93.28%	6.46%	93.28%	93.28%	92.23%	7.25%	92.23%	92.23%
	4	93.65%	6.28%	93.65%	93.65%	92.45%	7.65%	92.45%	92.45%

It is observed from Table 9 that DT classifier establishes a higher precision and lower error rate compared to SVM and LVQ classifiers. The results also suggest that DT classifier demonstrates F-Measure values of **98.08%** for 70%-30% distribution and **97.53%** for 10-fold cross-validation. In comparison to this, the LVQ model is having an average F-Measure value of **93.0%** using 70%-30% distribution and **92.55%** using 10-fold cross-validation; while SVM is having an average F-Measure value of **93.04%** for 70%-30% distribution and **92.55%** for 10-fold cross-validation. These results are certainly better than the average values given by SVM and LVQ classifiers. In fact, the average accuracy of DT classifier is more than 4%-5% compared to LVQ and SVM classification models.

Considering all the evaluation measures used, we have got wonderful results for the DT model compared to LVQ and SVM-based classification models. The CART algorithm based implementation of DT model has the highest values for accuracy, kappa statistic, tp-rate/recall, precision, and f-measure and the lowest values for RMSE and fp-rate. Indeed, DT outperforms LVQ and SVM classifiers in terms of all these performance measures being used. Assuredly, the DT model could predict the outcome of singles grand slam tennis matches with a higher degree of precision as the average accuracy value lies within 97.5% to 99.5%.

7.0 CONCLUSION

Prediction of tennis match result is a very challenging research domain. Owing to the advancement of information technology, predictions are extensively used in tennis matches, specifically by coaching staffs, news agencies and

audiences. The tennis prediction model is developed here to evaluate the possibility of winning singles match that the players will face. As a conclusion, we have taken on our objective which is to evaluate and investigate DT, LVQ and SVM classification algorithms using various evaluation measures like classification accuracy, RMSE, weighted kappa statistic, TP-Rate, FP-Rate, Precision, Recall, and F-Measure. The most effective method based on performance evaluation along the eight UCI tennis singles match datasets is the CART algorithm based DT classifier. In fact, its average accuracy is more than 4%-5% compared to the other classifiers.

Decision trees are standard constructs and easy to understand from which rules can be extracted. Considering the benchmark datasets, this classifier also has the lowest RMSE and FP-Rate values and highest F-Measure and weighted kappa statistic values compared to LVQ and SVM classification models. These results suggest that among the three classifiers studied and analyzed, the DT classifier has the capability to improve the conventional classification methods for predicting tennis singles match result. In the future, we should make an attempt to predict the results of tennis doubles matches using past data records of grand slam tennis tournaments. The prediction model will involve men's doubles, women's doubles and mixed doubles matches for all grand slam tournaments. In fact, it might be extended to the different types of ATP tennis completions as well.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami, "Database Mining: A Performance Perspective", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 6, December 1993, pp. 914–925.

[2] M. S. Chen, J. Han, and P. S. Yu, "Data Mining: An Overview from a Database Perspective", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 6, December 1996, pp. 866-883.

[3] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2nd ed., 2009.

[4] I. H. Witten, F. Eibe, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 3rd ed., 2011.

[5] J. Han, and M. Kamber, Data Mining: Concepts and Techniques. Morgan and Kaufmann, 2nd ed., 2006.

[6] A. K. Pujari, Data Mining Techniques, Universities Press (India) Private Limited, 1st ed., 2001.

[7] N. K. Bose, and P. Liang, Neural network fundamentals with graphs, algorithms, and applications. McGraw-Hill, 1996.

[8] S. Haykin, Neural Networks: A Comprehensive Foundation. Prentice Hall, 2nd ed., 1998.

[9] R. Rojas, Neural Networks A Systematic Introduction. Springer-Verlag, Berlin, 1996.

[10] A. Qazi, R. G. Raj, M. Tahir, M. Waheed, S. U. R. Khan, and A. Abraham, "A Preliminary Investigation of User Perception and Behavioral Intention for Different Review Types: Customers and Designers Perspective," *The Scientific World Journal*, Vol. 2014, Article ID 872929, 8 pages, 2014. doi:10.1155/2014/872929.

[11] P. Somervuo and T. Kohonen, Self-Organizing Maps and Learning Vector Quantization for Feature Sequences, Helsinki University of Technology, Neural Networks Research Centre, Finland, 2004.

[12] C. Cortes and V. Vapnik, "Support-vector networks", *Machine Learning*, Vol. 20, No.3, September 1995, pp. 273-297.

[13] J. R. Quinlan, Simplifying decision trees, International Journal of Man-Machine Studies, Vol. 27, No. 3, 1987, pp. 221–234.

[14] L. Breiman, J. H. Freidman, R. A. Olshen, and C.J. Stone, Classification and Regression Trees, Belmont, Wadsworth, 1984.

[15] T. Barnett, A. Brown, and S. R, Clarke, "Developing a tennis model that reflects outcomes of tennis matches", *in proceedings of the 8th Australasian Conference on Mathematics and Computers in Sport*, Coolangatta, Queensland, 2006, pp. 178 - 188.

[16] A. Somboonphokkaphan, S. Phimoltares, and C. Lursinsap, "Tennis Winner Prediction based on Time-Series History with Neural Modeling", *in proceedings of the International MultiConference of Engineers and Computer Scientists IMECS 2009*, Hong Kong, March 18 - 20, 2009.

[17] J. D. Corral, and J. Prieto-Rodr´ıguez, "Are differences in ranks good predictors for Grand Slam tennis matches?", *International Journal of Forecasting*, Vol. 26, No. 1, 2010, pp. 551–563.

[18] A. Panjan, N. Šarabon, and A. Filipčič, "Prediction of the Successfulness of Tennis Players with Machine Learning Methods", *Kinesiology*, Vol. 42, No. 1, 2010, pp. 98-106.

[19] X. Wei, P. Lucey, S. Morgan, and S. Sridharan, ""Sweet-Spot": Using Spatiotemporal Data to Discover and Predict Shots in Tennis", *in proceedings of the 7th Annual MIT Sloan Sports Analytics Conference 2013*, Boston Convention and Research Center, March 1-2, 2013.

[20] A. S. Timmaraju, A. Palnitkar, and V. Khanna, Game ON! Predicting English Premier League Match Outcomes, CS 229 Machine Learning Final Projects, Stanford University, Autumn 2013.

[21] D. Buursma, "Predicting sports events from past results Towards effective betting on football matches", *in proceedings of the 14th Twente Student Conference on IT*, Enschede, University of Twente, The Netherlands, Copyright 2010, January 21, 2011.

[22] Tennis Major Tournament Match Statistics Data Set, UCI Machine Learning Repository, University of California, Irvine, Machine Learning Repository, 2014.

[23] A. Elisseeff, and H. Paugam-Moisy, "Size of multilayer networks for exact learning: analytic approach", *Advances in Neural Information Processing Systems*, Vol. 9, USA: MIT Press, 1997, pp.162-168.

[24] J. S. Armstrong, and F. Collopy, "Error Measures For Generalizing About Forecasting Methods: Empirical Comparisons", *International Journal of Forecasting*, Vol. 8, 1992, pp. 69–80.

[25] J. Carletta, "Assessing agreement on classification tasks: The kappa statistic". Computational Linguistics, *MIT Press Cambridge*, MA, USA, Vol. 22, No.2, 1996, pp. 249–254.

[26] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy". *Remote Sensing of Environment*, Vol. 62, No. 1, 1997, pp.77–89.

[27] L. B. Huang, V. Balakrishnan, R.G. Raj, "Improving the relevancy of document search using the multiterm adjacency keyword-order model", *Malaysian Journal of Computer Science*, Vol. 25, No. 1, 2012, pp. 1-10.