Machine Learning Classifiers comparison for Wine Data

Srishti Mouli
Department of Computer Science
University of Saskatchewan
Saskatoon
srm700@usask.ca

Jenia Afrin Jeba
Department of Computer Science
University of Saskatchewan
Saskatoon
jaj212@uasak.ca

***Abstract*—Classification of any given data set has been considered as a very important basic of machine learning approach. In this project work we are motivated to learn about the working mechanism of different classifiers used in Machine Learning to evaluate the data set and identify classes it comprises of; initially we are going to evaluate two classifiers namely - Multilayer Perceptron Network (MLP) and Radial basis function network (RBFN) classifier. Here we are working on classifying a publicly available wine data set into three classes based on thirteen features of their chemical compounds composition. We will also check the accuracy, specificity and sensitivity of these two classifiers using K-fold cross validation.**

***Index Terms*—Machine learning, neural network, RBFN, MLP.**

I. Introduction

Here goes the introduction Classification means the selection of an object into a particular group or class. Given some training data points, a classifiers job is to predict the given test data set into some particular discrete classes called targets or categories or labels. The task of approximating any mapping function from an input to an output variable is termed as classification predictive modeling. A classifier has some kind of memory in order to learn the objects [7]. A set of training and testing data are provided to the classifier, training set involves identification of the classes or groups and the testing set is the data to be worked upon in case of supervised learning a label is present to test the data swiftly.

In this work we have decided to apply Multilayer Perceptron Network (MLP) and Radial basis function network (RBFN) as classifiers. Both of these classifiers are very robust and are commonly used.

Initially, we have determined to use classifiers namely Radial Basis Function (RBF) networks and Multilayer Perceptron (MLP) networks for classifying the wine dataset into three classes. These two neural-network based classifiers are based on supervised training and involves a training set of acknowledged input-output pairs [3].

MLP classifier transmits the signals from input to output in single direction via networks and one neuron output doesnt affect another neurons output [8]. The strong point of MLP is in the activation of nonlinear functions. The input layer transmits the signal to the higher layers. There can be more than one hidden layer conditional upon the problem to be solved. The activation functions are contained in the hidden layers and they map the weighted input to the output. MLP learns by updating the weights

after individual iteration where the update is contingent upon the difference of error between the input and output values.

On the other hand, RBFN classifier is a two-layer network where each hidden layer network comprises of a radial activated function. The input in RBFN networks is nonlinear whereas the output is linear. The output layer is a weighted sum of the input from the hidden layers [1]. RBFN is also a supervised learning method. The goal of these networks is to predict the target value by applying a combination of radial kernels, such as Gaussian functions. The output of a RBF network consists typically of centers and weights of these functions which are connected in a way to provide function prediction to learning. The uniqueness of RBFN comes from the fact that of its capacity control, that any set of input data can be fitted precisely by allowing the data points to be centers and approximation functions being applied [20]. The basis of learning in RBFN networks is cost updating which is the mean square error of the mappings from input to output.

The advantage of RBFN over MLP is that an RBF network trains faster than the MLP network due to presence of a smaller number of hidden layers and the interpretation of hidden layers is much easier in RBFN as the network output is decided by local receptive fields whereas in MLP it is calculated globally [12].

The objective of this paper is that, we are going to classify wines in three integer categories (namely, 1, 2 and 3) using these two classifiers. These data are the consequences of a chemical analysis of wines grown-up in the similar region in Italy but derived from three different cultivars. The analysis decided the amounts of 13 ingredients found in each of the three types of wines. The attributes are [21]:

1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids

8) Nonflavanoid phenols
9) Proanthocyanins
10)Color intensity
11)Hue
12)OD280/OD315 of diluted wines
13)Proline

We will feed 178 samples into classifiers to analyze and predict the results. This study focuses on the attributions characteristics of wines and predicts the final result depending upon them. After that we will calculate the accuracy, specificity and sensitivity is for the K-fold cross validation of the results obtained.

II. Literature Review

Several interesting classification works have been performed using MLP and RBFN. Some of them are discussed below.

In [1] based on an optimization layer by layer (OLL) procedure, authors have presented a learning algorithm. From previously reported OLL-type learning algorithms, this algorithm mainly differs in the way that the weights between the output layer and the last hidden layer are determined through optimization of a piece-wise linear objective function. For training an MLP to be a pattern classifier, this function is subject to specifically designed constraints.

Authors in [2] have introduced a supervised learning algorithm. Without forgetting what has been learned in earlier training sessions, neural network classification algorithms get the capability of incremental learning from new data. Using different distributions of the training data, this algorithm is able to generate a number of hypotheses and also combine these hypotheses using a weighted majority voting. Even when new classes are introduced, this scheme allows the classifier to learn from new data when the original data is no longer available which was previously trained with a training database.

Authors in [3] has considered three training algorithms namely Scaled Conjugate Gradient (SCG), Levernbergh-Marquardt (LM) and Resilient Backpropagation (RP) Neural Network for investigating the performance of the Multilayer Perceptron (MLP) classifier with different number of hidden neurons which is in the ranged of 1 to 10. They have discriminated the qualities of agarwood oil significant compounds (dataset were obtained at Forest Research Institute Malaysia (FRIM) and University Malaysia Pahang (UMP)) by diverse qualities and simulated it in Matlab version 2013a. They considered the areas (abundances,

Both the Bayesian detector and Multilayer Perceptron classifiers are well-known for the problem of detection and classification of one of M orthogonal signals in a Gaussian noise environment, because it can provide a measure for the performance evaluation of the neural networks. For the detection part, they have observed that for the signal-known-exactly case (M = 1) the performance of the neural detector converges to the performance of the ideal Bayesian decision processor. The probability of error of the neural network can be numerically calculated; it can be compared to the minimum Bayesian error for the classification case.

Authors in [5] have proposed an ATR classifier. A fast and parallel processing is enabled by its modular structure. A superior recognition performance is the result of the cascaded structure. It consists of arrays of VQs and MLPs. By applying variable-size aspect windows they have reduced noise and dimensionality. To enhance inter-target discriminability and to capture intra-target similarities, the K-

means and the modified LVQ algorithm have been used. Additional feature extraction and information integration have been performed by the WMLPs and the TMLP.

Using PfSGA(Parameter-free Species Genetic Algorithm), a structural learning method of MLP classifiers have been proposed by authors in [6] and later they applied it to Korean sign language. for selecting an appropriate architecture the method is a superior tool for given high dimensions. There was no sacrifice in classifier accuracy and the method could be used efficiently. The results on a practical data, finger spelling etc. were accurate.

In classifying QK, the authors have discussed the use of the MLP in [7] this paper. Results proved that, for differentiating correct and incorrect Qalqalah Kubra pronounced, MLP trained with MFCC features performed correctly.

For Mongolian character recognition problem, authors in [8] have investigated MLP and performed simulation on single size and single font. They conducted testing on noisy Mongolian character set. They have found very high success rate with moderate noise level using MLP classifier for Mongolian character recognition.

Based on the EEG signals related to music, authors in [9] have developed an off-line emotion classification algorithm. The results of MLP showed much higher classification accuracy compared to the chance probability of 25

For medical decision making on chronic kidney disease using neural network classifier, the effect of class imbalance in training data is explored by authors in [10]. To recognize different patterns, back propagation networks are a popular type of neural networks. For the prediction of chronic kidney disease, based on multilayer perceptron a comparative study of some sampling algorithms was performed using different learning rate values.

Authors in [11] have proposed a fast face detection algorithm based on skin color model and radial basis function network. At first, in this algorithm, input image is made to carry out RGB and YCbCr color space conversion, after that to achieve the coarse positioning of face region, relevant skin model is established, finally, the radial basis function network is combined to train input image. Thus, according to the training results whether it is the skin color is determined and on face, the detection is done.

Authors in [12] have proposed an Active Disturbance Rejection Control based on Radial Basis Function (RBF) Neural Network. By RBF Neural Network, nonlinear functions can be approached accurately, and the TD can be improved. Based on Radial Basis Function (RBF), the Active Disturbance

Rejection Control performed better in case of Neural Networks robustness.

Authors in [13] have presented a face detection system using Radial Basis Function Neural Networks With Variance Spread Value. For identification, the face region is localized and extracted from the background for feeding into the face recognition system. For normalizing the image general pre-processing approach was used. To distinguish between face and non-face images, a Radial Basis Function (RBF) Neural Network was used. Compared to other neural network architecture RBF Neural Networks offer several advantages such as the network possesses the property of best approximation and using fast two stages training algorithm, they can be trained.

Radial basis function networks (RBF) that can be seen as a special case of regularization networks, have a rich selection of learning algorithms. There is difference between practical suitability of regularization derived networks (RN) and the theoretical results of regularization theory. Authors in [14] study a relationship between RN and RBF. They have showed that for RN theoretical estimates hold for a concrete RBF which have been applied on real-world data.

Authors in [15] presented an intelligent radar signal sorting system with a robust radial basis function (RBF). The random overlapped radar signal stream can automatically be sorted and the input pulse stream can be separated to individual radar pulse sequence. The traditional method produces some non-expectation in learning process. As its basis function, the proposed RBF has better learning properties and function approximation capabilities; it uses Log-Sigmoid function, to eliminate any risk of instabilities. This algorithm better performs than the traditional algorithm.

III. Methodology

The preliminary step is data collection, which has been gathered from the website. The proposed pathway is shown in Fig 1. The obtained data was fed into the algorithm which involved shuffling of data as a pre-processing step.All the data collected were passed into the next phase of pre-processing where shuffling of data was done. In machine learning procedures the dataset is divided into training and testing data. Data Shuffling is performed so as to remove bias patterns in the dataset before training is performed. Another advantage of shuffling data is that our neural network becomes more generalized i.e., for every set of input values there is a set of output values depending upon them. In our study shuffling of data is done in Matlab by using a random function. This random function algorithm takes the entire set of data and randomizes it by exchanging the rows and columns. Shuffling of data was done to exempt monotony of data. The sample was then bifurcated into training and testing data as per the model requirement. In this study we have used two classifiers: MLP and RBFN. Classification means the selection of an object into a particular group or class. A classifier has some kind of memory in order to learn the objects. A set of training and testing data are provided to the classifier, training set involves identification of the classes or groups and the testing set is the data to be worked upon in case of supervised learning a label is present to test the data swiftly.The final step involved model evaluation by cross validation; performance of each of the classifiers would be recorded which would be expressed in terms of specificity,sensitivity and accuracy.

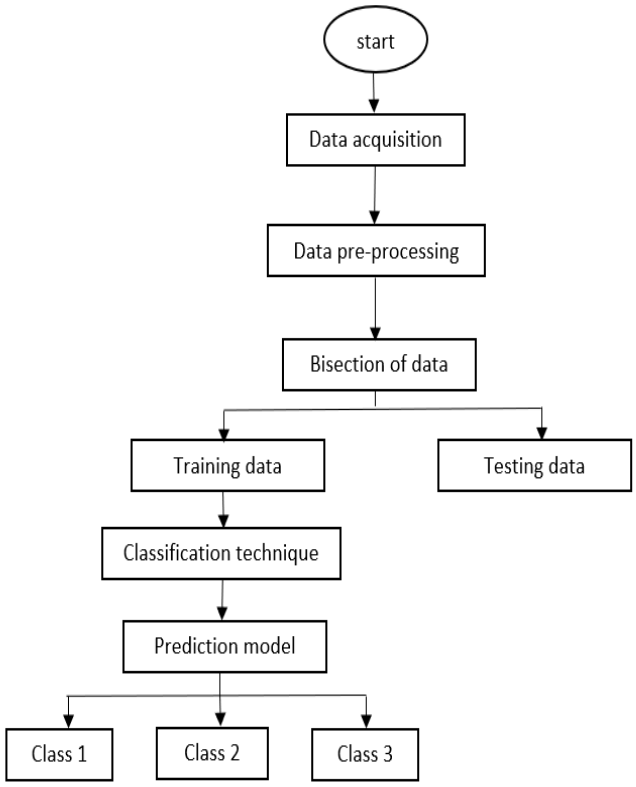
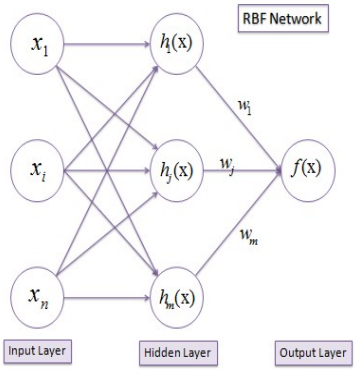


Fig. 1. Schema of proposed methodology

A. *Proposed approaches:*

I. Algorithm of Radial basis function network (RBFN) classifier:

There are three layers in RBF networks namely: input layer, hidden layer and output layer. Each neuron from the input layer will link with a predictor variable. In regard to categorical variables, there are n-1 neurons where n indicates the quantity of categories. There are variable number of neurons in hidden layer. One neuron entails a radial basis function positioned on a center point having the similar magnitudes as the predictor variables. To form the network outputs, the output layer contains a weighted sum of outputs collected from the hidden layer.



$$f(x) = \sum_{j=1}^{m} w_j h_j(x)$$

$$h(x) = \exp\left(-\frac{(x-c)^2}{r^2}\right)$$

Fig. 2. A schematic diagram of Radial basis function network (RBFN) classifier

RBFN is comprised in a two layer network and each hidden layer network consists of a radial activated function. The input in RBFN networks is nonlinear whereas the output is linear. The output layer is a weighted sum of the input from the hidden layers [19]. RBFN is also a supervised learning method. The goal of these networks is to predict the target value by applying a combination of radial kernels, such as a Gaussian functions. The output of a RBF network consists typically of centers and weights of these functions which are connected in a way to provide function prediction to learning. The uniqueness of RBFN comes from the fact that of its capacity control. This can be characterized from the fact that any set of input data can be fitted precisely by allowing the data points to be centers and approximation functions being applied [20]. The over fitting data needs to be regularized and kernel machines such as SVM are used. The basis of learning in RBFN networks is cost updating which is the mean square error of the mappings from input to output.

Here, in the following section, Gaussian activation function is used denoted as h(x); the parameters r is the radius and c is denoted as the center of the input plane defined discretely at individual RBF unit. The classifier learns by adjusting the parameters and then produce a set of input-output. Mainly, three types of parameters are used: the width of the unit r, the center of each neuron of the hidden layer c and the weight between the hidden layer and the output layer w. We have chosen K-means clustering algorithms for determining the RBF unit centers. The number of input variables in the input layer determine a set of clusters each having r-dimensional centers.The centers of the RBF units are nothing but the cluster centers. The number of nodes in the hidden layer is determined by the number of clusters, H. The procedure of K-means algorithm is: the center of each cluster is initialized to an arbitrarily chosen training pattern. the Euclidean distances between the training patterns and the cluster centers is calculated;then each training pattern is assigned to the nearest cluster. the average position for each cluster center is calculated after all training patterns are assigned. Hence, they are called new cluster centers.

- the number of hidden neurons is defined as "K".
- the positions of RBF centers is set using K-means clustering algorithm.
- σ is calculated as follows:

$$\sigma = \frac{d}{\sqrt{2M}}$$

here M = the amount of hidden neurons, d = the highest gap between two hidden neurons

- actions of RBF node is calculated as:

$$\varphi(X) = e^{\frac{-\|X-\mu\|^2}{\sigma^2}}$$

μ = center of the neuron and

φ(X) = response of the neuron related to input X.

- Train the output using equation $W=(G^TG)^{-1}G^TT$
  where G = a matrix representing the output of a neuron for an input vector
  G^T = the transpose of matrix G.
  W = weights connecting an output neuron to a hidden neuron

Fig. 3. Algorithm parameters for RBFN classifier

II. Algorithm of Multilayer Perceptron (MLP) classifier:

In MLP the signals are transmitted from input to output via networks in single direction. The unique feature is that one neuron output doesnt affect other [18]. A typical MLP architecture has three layers: Input layer, Hidden Layer and output layer. The strength of MLP is in non linear activation functions and with the exception of polynomial functions any non-linear function can be applied. The function of the input layer is to transmit the signal to the upper layers. The hidden layer contain the activation functions, there can be more than one hidden layer depending upon the problem in hand. The activation function maps the weighted input to the output. Learning in MLP is achieved by updating the weights after each iteration and the update depends upon the error difference between the input and output values. As we know the output beforehand in a MLP network, they fall in the category of supervised learning where the Input and output are known prior to start of learning.

As already mentioned that the architecture of MLP classifiers learning algorithm takes on three layers: input, hidden and output layer. Except the input layer, rest of the two layers are neurons which utilize non-linear activation function and back-propagation technique as a part of supervised learning. There are weights between the input and the hidden layer (can be denoted by W1) and also between the hidden and the output layer (can be denoted by W2).

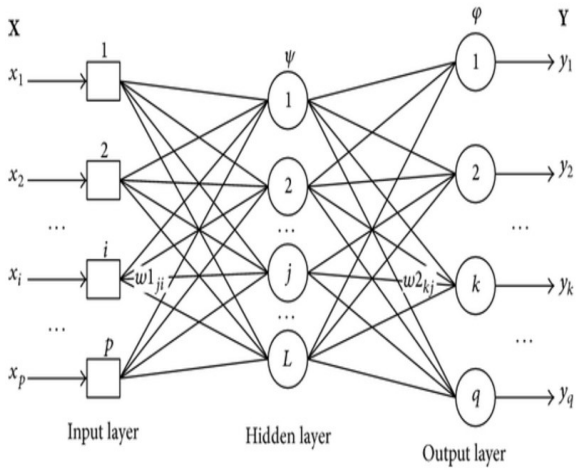


Fig. 4. A schematic diagram of Multilayer Perceptron (MLP) classifier

1. At first, the training data set which we want the network to learn will be taken
2. Next, in layer N, the network with n input units, N-1 hidden layers of nonlinear n hidden units, and n output unitswill be set up. Initial random weights would be generated; using connection weights, each layer (n) will be fully connected to the previous layer (n-1)
3. Suitable error function and learning rate will be selected
4. For each training pattern,the weight update will be applied to each weight (when a set of updates are generated for all the weights regarding all the training patterns, its termed as an epoch of training).
5. Till the network error function continues to decrease to a small value, step 4 would be repeated.

In this way, these two algorithms perform their respective operations.The advantage of RBFN over MLP is that an RBFN network trains faster than the MLP network due to presence of less number of hidden layers and the interpretation of hidden layers is much easier in RBFN as the network output is decided by local receptive fields whereas in MLP it is calculated globally[21].

B. Data Set Description

We are going to classify wines in three integer categories (namely, 1, 2 and 3) using two classifiers. The dataset which are being used in this project is found from this link [21]: https://archive.ics.uci.edu/ml/datasets/wine
Data Set Information:
These data are the consequences of a chemical analysis of wines grown-up in the similar region in Italy but derived from three different cultivators. The analysis decided the amounts of 13 ingredients found in each of the three types of wines.

Data Set Characteristics:We have 178 data samples. The nature of our data set (number of features, number of categories are summarized below in the table:

TABLE I
WINE DATA SET ATTRIBUTE INFORMATION METADATA

Data Set Characteristics:	Multivariate	Number of Instances:	178
Attribute Characteristics:	Integer, Real	Number of Attributes:	13
Associated Tasks:	Classification	Missing Values?	No

The attributes are already listed in the introduction section.

Attribute Information: All attributes are continuous;1st attributes class identifier (1-3). Figure 4, figure 5 and figure 6 display, the histograms for the features of wine which are: Alcohol, Malic Acid and Ash respectively.

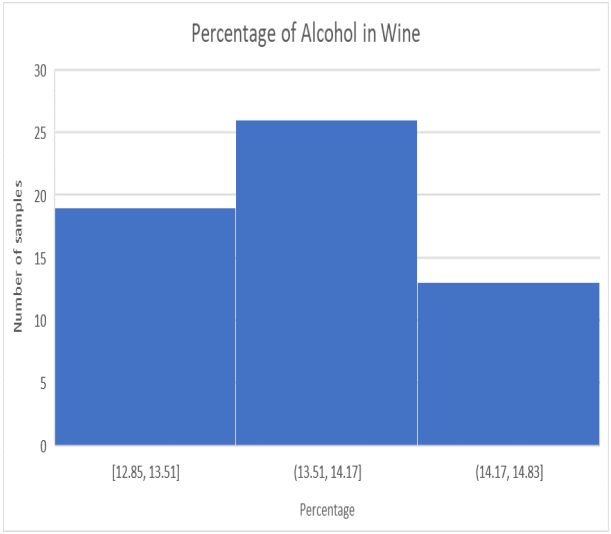


Fig. 5. Percentage of Alcohol in Wine

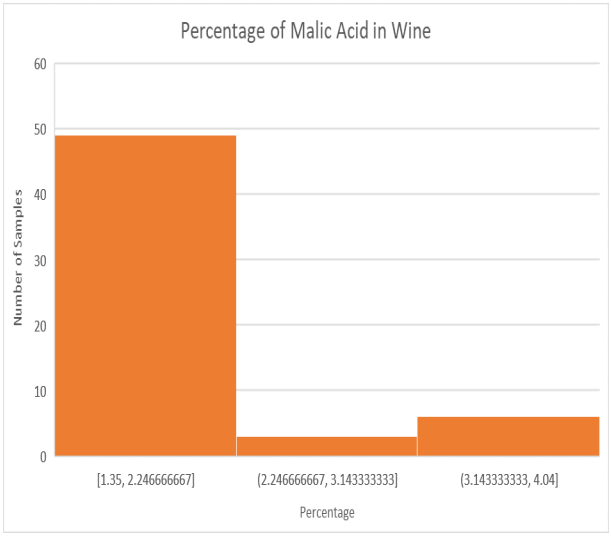


Fig. 6. Percentage of Malic Acid in Wine

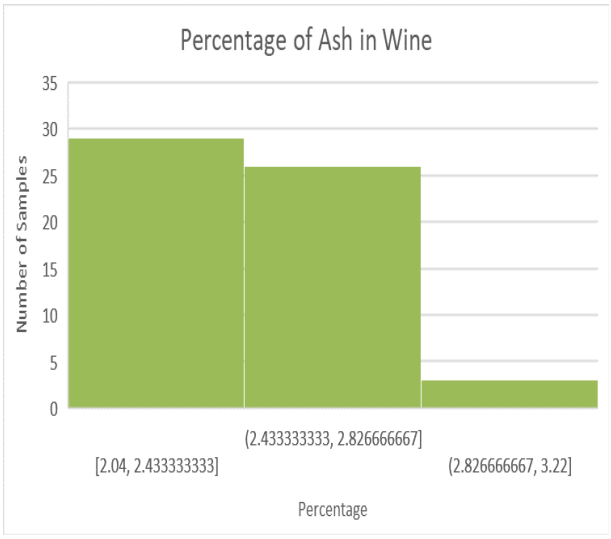


Fig. 7. Percentage of Ash in Wine

IV. Validation

As the classifier is being trained for this dataset it may not achieve accuracy for other dataset. To avoid this cross validation is introduced. This does not improve the accuracy of the classifiers but makes it more robust and generalized. The K-Fold cross validation is applied as it is simple, easy and uses all data for training and validation [22]. K-Fold partition was created for the entire dataset and was run for K times to calculate the accuracy for each subsequent run. Determination of K-Folds is quite challenging because if the K is set too large the bias of the accuracy predictor will be low but the variance will be high and it will be more time consuming, whereas if the K is small the bias of accuracy predictor will be large and time consumed will be less but accuracy may suffer in this case. In this study K is taken as 5 after many hit and trial methods.

V. Performance Analysis

Performance assessment is done to assess the ability of the classifiers to predict the result robustly. The performance metrics: accuracy, specificity and sensitivity are calculated from the confusion matrix as shown below (Table II):

TABLE II
Confusion Matrix for Performance Assessment

Classifier Output	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Here TP, TN, FN, and TN can be defined as: TP (True Positive): Correctly identified. TN (True Negative): Incorrectly identified. FP (False Positive): Correctly rejected. FN (False Negative): Incorrectly rejected. In this study the above mentioned performance metrics are governed by their respective formulae all of them are discussed below:

Accuracy = ((TP + TN) / (TP + TN + FP + FN)) * 100%
Specificity = (TN / (TN + FP)) * 100%
Sensitivity = (TP / (TP + FN)) * 100%

VI. Results

The entire process was manifested on a core I5 computer with 4 GB of ram. The operating environment was Windows 10 and the code execution was done using Matlab 2012.Detection of wines was done using the 13 features discussed above. 178 set of data was taken for 3 categories of wines.Entire data was divided into training and testing data and then were subjected to cross validation.Prediction was done using three performance metrics namely: Specificity, sensitivity and accuracy.To attain the above objective an automated system was developed and different algorithmic runs were done using the above specified configuration and results are discussed in this section. The final predicted result was obtained after going through multiple steps. In the first step which is the preprocessing step and it prepares the data for further processing, a random algorithm is used for this purpose. In the next step the data is divided into training and testing data so as to start the learning process of our model. K- Fold validation is performed in the next step for 5 runs and it is applied both for MLP and RBFN. The accuracy levels of the two classifiers used and RBFN again moves ahead with an accuracy prediction of 97 percent and MLP comes behind with accuracy of 91 percent.The performance metrics obtained by applying sensitivity. Sensitivity measures the fraction of negative that are correctly identified. Table III displays the accuracy levels of the two classifiers used and RBFN again moves ahead with an accuracy prediction of 94% and MLP comes behind with accuracy of 80%.

TABLE III
Classification Accuracy of MLP RBFN over 5-fold

	Fold					
Classifier	1	2	3	4	5	Average Accuracy
MLP	80.00	80.00	80.00	75.00	85.00	80.00
RBFN	90.00	90.00	100.00	95.00	95.00	94.00

Table IV shows the performance metrics obtained by applying sensitivity. Sensitivity measures the fraction of negative that are correctly identified (e.g. percentage of normal persons who are identified as not depressed). From the table it is concluded that average sensitivity of the RBFN algorithm at 96% is much better than MLP at 83.83% after the fivefold process. At each iteration of the k-fold process it is observed that the values of RBFN are higher than MLP. It is also observed that the specificity level of RBFN is quite consistent through all the k-fold iterations.

Table V presents specificity here also RBFN comes at the top with 93.36% and MLP lagging behind at 78.54%. Both of these performance metrics indicate the inclination towards

TABLE IV
SENSITIVITY OF MLP AND RBFN CLASSIFIERS OVER 5-FOLD

	Fold					
Classifier	1	2	3	4	5	Average Sensitivity
MLP	87.50	90.00	66.67	75.00	100.00	83.83
RBFN	80.00	100.00	100.00	100.00	100.00	96.00

TABLE V
SPECIFICITY OF MLP AND RBFN CLASSIFIERS OVER 5-FOLD

	Fold					
Classifier	1	2	3	4	5	Average Specificity
MLP	75.00	70.00	100.00	75.00	72.73	78.54
RBFN	77.78	100.00	85.71	83.71	83.33	93.36

RBFN.

Fig 2 presents the computation time for the two classifiers: MLP and RBFN. The time taken by RBFN is nearly half to that of MLP which acknowledges the fact that RBFN is a better classifier than MLP. It is evident from the theory that RBFN has less number of hidden layer and the values of one node does not depend on values of other nodes in the hidden layer.

It is concluded that average sensitivity of the RBFN algorithm at 98.43% is much better than MLP at 93.76% after the five-fold process. At each iteration of the k-fold process it is observed that the values of RBFN are higher than MLP. It is also observed that the specificity level of RBFN is quite consistent through all the k-fold iterations. In Specificity here again, RBFN comes at the top with 96% and MLP lagging behind at 88.76%. Both of these performance metrics indicate the inclination towards RBFN. The time taken by RBFN is nearly half to that of MLP which acknowledges the fact that RBFN is a better classifier than MLP. It is evident from the theory that RBFN has less number of hidden layer and the values of one node does not depend on values of other nodes in the hidden layer.

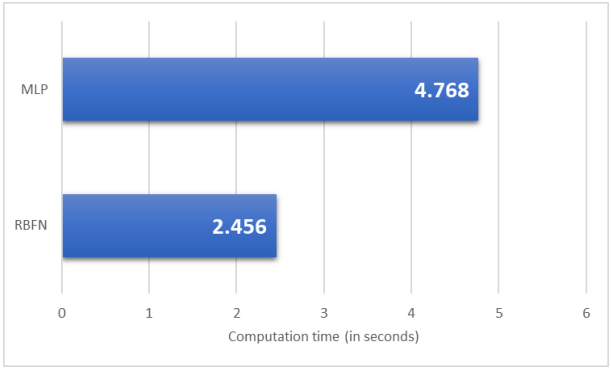


Fig. 8. Variation of computation time for MLP and RBFN

VII. CONCLUSION AND FUTURE WORK

In this study we have tried to focus on learning classification techniques namely RBFN and MLP and applied these methods on a publicly available wine dataset. These methods are very robust in nature. Our aim is to formulate an algorithm which takes less time and is high on accuracy. It is evident from the above results that RBFN is a better classifier as it uses a Gaussian function which is more robust in calculating the results and the values at the hidden layers are can be forwarded individually to the output layer whereas in case of MLP the values at each node is the average of all the values at the subsequent nodes. The and accuracy affirm the claims by our algorithm as they give high values of prediction efficiency. The two classifiers used are the most used classifiers in neural network learning and are easy to interpret. In view of the aforementioned discussion of results it can be concluded that our algorithm can provide a significant assistance in classifying any data set into multiple classes. For future work, we would measure specificity and sensitivity. We would also try to implement another classifier and compare its performance with RBFN and MLP.

REFERENCES

[1] Zhen Zhang ; Weimin Shao ; Hong Zhang, "A learning algorithm for Multilayer Perceptron as classifier", IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339), Year: 1999 , Volume: 3, Pages: 1681 - 1684 vol.3.

[2] R. R. Polikar ; L. Udpa ; S.S. Udpa ; V. Honavar, "LEARN++: An Incremental Learning Algorithm For Multilayer Perceptron Networks" , 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100), Year: 2000 , Volume: 6, Pages: 3414 - 3417 vol.6.

[3] N. S. A. Zubir ; M. A. Abas ; Nurlaila Ismail ; Nor Azah M. Ali ; M. H. F. Rahiman ; N. K. Mun ; N. T. Saiful ; M. N. Taib, "Analysis of Algorithms Variation in Multilayer Perceptron Neural Network for Agarwood Oil Qualities Classification",
2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC), Year: 2017, Pages: 122 - 126.

[4] Z.H. Michalopoulou ; L.W. Nolte ; D. Alexandrou, "Performance Evaluation of Multilayer Perceptrons in Signal Detection and Classification", IEEE Transactions on Neural Networks, Year: 1995 , Volume: 6 , Issue: 2, Pages: 381 386.

[5] L.A. Chan ; N.M. Nasrabadi ; V. Mirelli, "Automatic Target Recognition Using Modularly Cascaded Vector Quantizers And Multilayer Perceptrons",
1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, Year: 1996 , Volume: 6, Pages: 3386 - 3389 vol. 6.

[6] Seong-Hyo Shin ; Sang-Woon Kim ; Y. Aoki, "A Structural Learning Of Mlp Classifiers Using Pfsga And Its Application To Korean Sign Language Recognition", Proceedings of IEEE. IEEE Region 10 Conference. TENCON 99. 'Multimedia Technology for Asia-Pacific Information Infrastructure' (Cat. No.99CH37030), Year: 1999 , Volume: 1, Pages: 190 - 193 vol.1.

[7] Hasliza Abu Hassan ; Nasharul Heslem Nasrudin ; Muhammad Nazir Muhammed Khalid ; Azlee Zabidi ; Ahmad Ihsan Yassin, "Pattern Classification In Recognizing Qalqalah Kubra Pronuncation Using Multilayer Perceptrons",
2012 International Symposium on Computer Applications and Industrial Electronics (ISCAIE), Year: 2012, Pages: 209 - 212.

[8] O. Batsamhan ; Y.P. Singh, "Mongolian Character Recognition Using Multilayer Perceptron (Mlp)",
Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02. Year: 2002 , Volume: 2, Pages: 621 - 625 vol.2.

[9] S. Yuan-Pin Lin ; Chi-Hong Wang ; Tien-Lin Wu ; Shyh-Kang Jeng ; Jyh-Horng Chen, "Multilayer Perceptron for EEG Signal Classification during Listening to Emotional Music",
TENCON 2007 - 2007 IEEE Region 10 Conference, Year: 2007, Page s: 1 - 3.

[10] Pinar Yildirim, "Chronic Kidney Disease Prediction on Imbalanced Data by Multilayer Perceptron",
2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), Year: 2017 , Volume: 2, Pages: 193 - 198.

[11] Chen Dong ; Xiaoli Wang ; Tao Lin ; Li Mei-cheng, "Face Detection Under Particular Environment Based on Skin Color Model and Radial Basis Function Network",
2015 IEEE Fifth International Conference on Big Data and Cloud Computing, Year: 2015, Pages: 256 259.

[12] Xiangdong Huang ; Qingzhao Ning, "Active Disturbance Rejection Control Based on Radial Basis Function Neural Network",
2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Year: 2018, Pages: 2397 2400.

[13] Khairul Azha A. Aziz ; Ridza Azri Ramlee ; Shahrum Shah Abdullah ; Ahmad Nizam Jahari, "Face Detection Using Radial Basis Function Neural Networks With Variance Spread Value",
2009 International Conference of Soft Computing and Pattern Recognition, Year: 2009, Pages: 399 403.

[14] Roman Neruda ; Petra Vidnerov, "Supervised Learning Errors by Radial Basis Function Neural Networks and Regularization Networks",
2008 Second International Conference on Future Generation Communication and Networking Symposia, Year: 2008 , Volume: 3, Pages: 193 196.

[15] Zhang Yongqiang ; Sun Guozhi, "Application of Radial Basis Function Neural Networks in Complicated Radar Signal Measurement and Sorting",
2007 8th International Conference on Electronic Measurement and Instruments, Year: 2007, Pages: 3-375 - 3-378.

[16] Weihua Wang, "License Plate Recognition Algorithm Based on Radial Basis Function Neural Networks",
2009 International Symposium on Intelligent Ubiquitous Computing and Education, Year: 2009, Pages: 38 41.

[17] Jianchuan Yin ; Fang Dong ; Nini Wang, "Predictive Control Based on Variable Radial Basis Function Network and Its Application to Ship Control",
2006 6th World Congress on Intelligent Control and Automation, Year: 2006 , Volume: 2, Pages: 8602 8605.

[18] J. Dang ; Yangping Wang ; Shuxu Zhao, "Face Recognition Based on Radial Basis Function Neural Networks using Subtractive Clustering Algorithm",
2006 6th World Congress on Intelligent Control and Automation, Year: 2006 , Volume: 2, Pages: 10294 10297.

[19] V. Muneeswaran ; M. Pallikonda Rajasekaran, "Performance Evaluation of Radial Basis Function Networks Based on Tree Seed Algorithm",
2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Year: 2016, Page s: 1 4.

[20] Lijuan Zhou ; Minhua Wu ; Mingsheng Xu ; Haijun Geng ; Luping Duan, "Research of Data Mining Approach Based on Radial Basis Function Neural Networks",
2009 Second International Symposium on Knowledge Acquisition and Modeling, Year: 2009 , Volume: 2, Page s: 57 - 61.

[21] https://archive.ics.uci.edu/ml/datasets/wine