IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

The Classification of Data: A Novel Artificial Neural Network (ANN) Approach through Exhaustive Validation and Weight Initialization

Kshitij Tripathi[1], Rajendra G. Vyas[2], Anil K. Gupta[3]

[1]Department of Computer Applications, The Maharaja Sayajirao University of Baroda,Vadodara, India
[2]Department of Mathematics, The Maharaja Sayajirao University of Baroda, Vadodara, India
[3]Department of Computer Science and Applications, Barkatullah University, Bhopal, India

**Corresponding Author: kshitij.tripathi-compapp@msubaroda.ac.in*

Available online at: www.ijcseonline.org

18/May/2018, Published: 31/May/2018

Abstract: The Artificial Neural Networks (ANNs) has proved their significance to perform well in the fields of data-mining and machine learning like classification, pattern recognition, forecasting and prediction to have a few of them. This paper explores a novel approach for classification of data on four benchmark datasets from the perspective of ANNs and its intricacies. The proposed approach is successful in overcoming the drawback of over-fitting of data exists in the classification domain. Further, the proposed methodology reflects very improved and consistent results in comparison to existing techniques available in the ANNs as well as non-ANN domain.

Keywords: Classification, Artificial Neural Network, Machine Learning, N-Fold Cross Validation, Transfer function.

I. INTRODUCTION

The classification of data is a data mining or machine learning technique in which the algorithm or the method tries to establish the relationship between input feature vectors and output variables, generally categorical, thereby adapt (learn) or build a model for prediction. There are two types of learning techniques, supervised and unsupervised, employed in the classification of data. The difference between each of these is that in supervised learning the classes are pre-assigned in the form of output variables or labels, to the instances of data, thereby building a model or establish a relationship between input and output variables is called as supervised learning. In the unsupervised learning, the output variables or labels are not initially given and the method tries to explore the relationship between input vectors (instances) and finally giving labels to these thereby classifying them into different classes or assign labels, called as clustering. This article is focussed on supervised learning and the aim is to increase the classification accuracy of the model by analysing the factors which make a sufficient impact on the accuracy of the results obtained through ANNs [6]. Rest of the paper is organized as follows. Chapter II gives the brief introduction to the back-propagation algorithm; Chapter III is about the datasets which we choose to perform the experiment. Reviewing the literature, few related works are cited in chapter IV, chapter V is about proposed k-fold training-validation-test approach (TVT). Chapter VI is about transfer functions used for the proposed k-fold TVT approach, the statistic of the experimental results is mention in chapter VII. The experimental results we obtain, its advantages and disadvantages are discussed in Chapter VIII and chapter IX respectively. Comparisons of our results with the recently available results in the literature are mention in chapter X. Finally, the conclusion is drawn in chapter XI.

II. ARTIFICIAL NEURAL NETWORKS

An Artificial Neural Networks (ANNs) [3, 18, 19, 20] can be viewed as information processing system which resembles the biological nervous system. In other words, artificial neural network functions in a way similar to the human brain. One of the main functions of ANNs is to produce an output pattern when presented with an input pattern. The ANNs consists of nodes connected by adaptable weights that store experimental knowledge from the examples of tasks through a learning process [19]. The neural network architecture is motivated by the model of human brain and nerve cells. A neuron is the fundamental unit of the brain [19].

In 1943, for the first time McCulloch and Pitts [25] introduced ANN, in which they present neuron as input-output processing unit based on binary threshold function which simulates the behaviour of the biological neuron [19] as shown in Figure1(a and b). The next major development in artificial neural networks came with the publication of the book by Hebbs [15]. Through the perceptron convergence theorem, Rosenblatt [7] introduced the new approach. After those various developments ([2, 6, 11, 22]) are noteworthy in the field of ANNs. One of the major contributions of Rumelhart, Hinton and McClelland [5] for discovering the back-propagation algorithm is considered to be the milestone in the ANN domain.

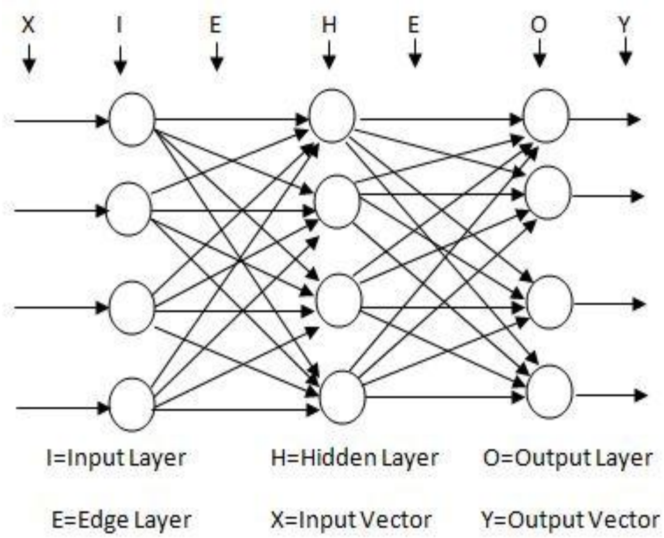


Figure 1(a). Artificial Neural Networks

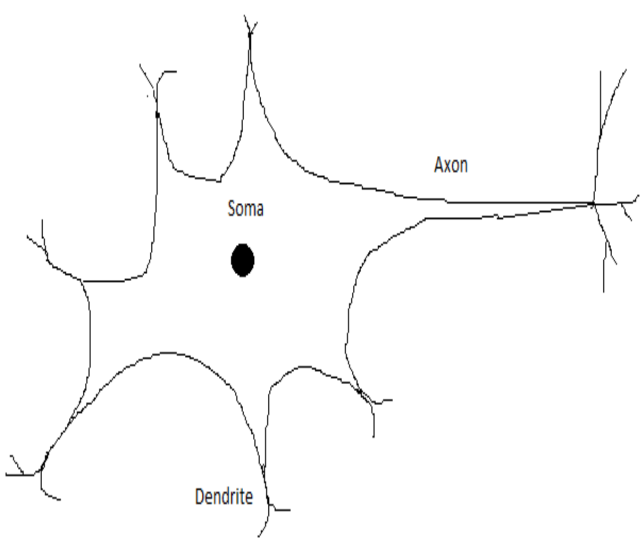


Figure **1**(b). Biological Neuron

Figure 1 describes the architecture of ANNs having one input layer one hidden layer and one output layer apart from edge layers in which each edge is associated with weights. Each circle represents the neuron associated with its transfer function except the input layer which has no transfer function. The configuration and training of ANNs are based on back-propagation algorithm [5] which have had proved its significance in almost entire ANNs culture. All the algorithms or functions used for training ANNs are variants of a back-propagation algorithm whose general working is as follows:

Step1. First, each edge layer is assigned the weights according to randomized technique or function.
Step2. Inputs are applied to the input layer.
Step3. The weighted sum of the inputs is determined for each neuron.
i.e. $(w_1x_1+w_2x_2+\dots w_nx_n)$ *where* w_i= *weights of corresponding edges* x_i=*corresponding inputs.*
Step4. The weighted sum acts as an input for the transfer function.
Step5. The output of each neuron is calculated up to the output layer by applying transfer function.
Step6. The error is determined at the output layer which is the difference between computed output and given output. The weights are then readjusted to minimize the error according to the back-propagation training algorithm.

The step 2 to step 6 above have been iterated with all instances of given data set containing input-output vector pairs and each iteration is called an epoch. Training is stopped when the network gives the optimized performance. The stopping criteria's of training are:

a) The maximum number of epochs is reached.

b) The minimum gradient is reached.

c) Best validation performance is achieved.

d) The goal is reached (mean square error is minimized or zero).

The main derivations [20] applied in the back- propagation algorithm is:

$$y_{ik} = \sum_j z_j w_{jk} + b$$

Where y_{ik} is the net input to k^{th} output neuron and z_j is the input to k^{th} neuron and w_{jk} is the weight associated with it.

$y_k = f(y_{ik})$

y_k is the output of k^{th} output neuron, f is transfer function applied to the weighted sum of inputs and weights assigned to edges and b is the bias associated with the particular neuron.
The error function to be minimized is

$$\varepsilon = \frac{1}{2} \sum (t_k - y_k)^2$$

Here t_k is the target output and y_k is the computed output.
The weight update equations are:

$$w_{hj}^{k+1} = w_{hj}^{k} + \Delta w_{hj}^{k} \text{ (For hidden to output layer weights)}$$

$$= w_{hj}^{k} + \eta\left(-\frac{\partial \varepsilon_k}{\partial w_{hj}^{k}} \right)$$

$$= w_{hj}^{k} + \eta \; \delta_j^k \, \varepsilon((z_h^k)$$

$$w_{ih}^{k+1} = w_{ih}^{k} + \Delta w_{ih}^{k} \text{ (For input to hidden layer weights)}$$

$$= w_{ih}^{k} + \eta\left(-\frac{\partial \varepsilon_k}{\partial w_{ih}^{k}} \right)$$

$$= w_{ih}^{k} + \eta \; \delta_h^k \, \varepsilon(x_i^k)$$

Note: η is the learning rate and δ_h^k represents an error and signal slope product that is error scaled by the signal slope.

The ANNs have performed very well in classification domain. The major factors which are contributing to giving good results or give high classification rate or minimized misclassification rate are the initialization of weights, transfer functions employed, number of hidden layers, number of neurons in each of these hidden layers and the methodology applied (like N-fold cross-validation or the inclusion of validation set[10]). In this article, the experiments are performed by considering above mentioned factors for obtaining optimal results.

III. DATASETS

The data sets chosen for performing the experiments are Wine, Iris, Breast-cancer (Wisconsin) and Red wine.Table1 summarizes the details of datasets. These datasets can be accessed from UCI machine learning repository. These datasets are ready to go with, that is there are no missing values and also no instance is left out because it is present in the experimental software itself except the Red wine dataset.

III.I. Brief introduction of Datasets

Wine dataset: This dataset contains the results of chemical analysis of wines grown in the same region of Italy but derived from three different cultivators (3 class output).The input data contains 13 constitutes found in these three types of wines.

Cancer dataset (Breast cancer Wisconsin): This is a dataset of clinical cases where inputs are symptoms present or not on 1 to 10 scale and the output is the class (2 for benign and 4 for malignant).

Iris dataset: This dataset contains output as three classes of Iris plant and input is the physical characteristics of flowers of Iris. The classes are Setosa, Versicolour and Virginica.

Red wine dataset: This dataset is based on physiochemical tests of Portuguese wine for modelling taste preferences (9 classes).

Table.1 the characteristics of dataset

Name of dataset	Number of instances	Number of Input attributes	Number of output attributes	Name of dataset
Wine dataset	178	13	3	Wine dataset
Cancer dataset (Wisconsin)	699	9	2	Cancer dataset (Wisconsin)
Iris dataset	150	4	3	Iris dataset
Red wine dataset	1599	11	6	Red wine dataset

IV. LITERATURE REVIEW

There are various techniques currently employed in the classification of data [18] in which eminent of them are k-nearest neighbour (k-NN), decision trees, support vector machines and artificial neural networks.

In last decay, various researchers have contributed in the field of classification of data. In 2009, P. Cortez, A. Cerderia, F. Almeida, T. Matos and J. Reis [16] proposed regression method through SVM technique for modelling wine preferences. In 2012, P. Piro, R. Nock, F. Nielsen and M. Barlaud [17] proposed k-NN technique and describe a solution to some problem by universal nearest neighbour algorithm. In 2014, Zhun-ga, Quan Pan and Jean Dezertb [26] introduced 'c x k' neighbour classifier based on evidence theory for data classification.

Apart from classification domain, various researchers employed ANNs in different other tasks like deep belief networks through initialization of weights [13] and use ANN for loss minimization control of a PMSM with core Resistance Assessment [21].

It has been observed that the essential factors for obtaining the optimized state (goal state) of ANNs are the initialization of weights, transfer function employed on various layers, number of neurons in hidden layers and the number of hidden layers [18, 20]. Apart from that, the other concept which is contributing to the result is the inclusion of validation set. The reason for the same is, the ANNs are prone to over-fitting that is giving good results in the training phase, but not as good as in testing phase, so to avoid over-fitting we divide the dataset (all instances) in to three parts that is training set, validation set and the test set. During the training phase, we first train the ANN with the training set and also test it with validation set simultaneously to avoid over-fitting. Finally, the ANN which is giving best result with the validation set is applied for testing. Hardly any researcher has addressed this factor in the ANN domain. When we apply the proposed technique by employing benchmark ANN training algorithm (back-propagation algorithm or variants of it) and observed that the results obtained are very consistent and improved no matter which dataset is used.

V. CURRENT METHODOLOGIES AND THE PROPOSED METHODOLOGY

The common methodologies applied for classifying the data in the ANNs domain are:

V.I. Training and testing approach

In this approach (Figure 2) the dataset is divided into two parts according to the specific ratio, called as training set and test set. The ANN train itself by applying the back-propagation algorithm and after the ANN gets trained, it is tested on the test for the purpose of evaluation. Generally, the mean square error is chosen to evaluate the performance and finally, the confusion matrix is generated for observing the accuracy of classification.

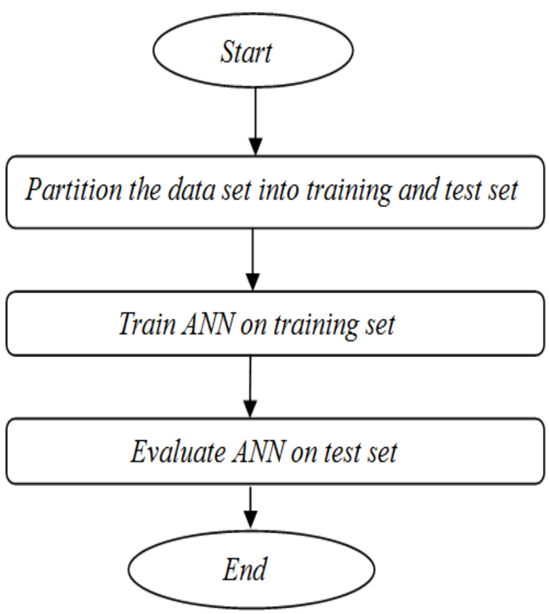


Figure 2. Normal training and test approach

VI.II. Normal k-fold approach or cross-validation approach

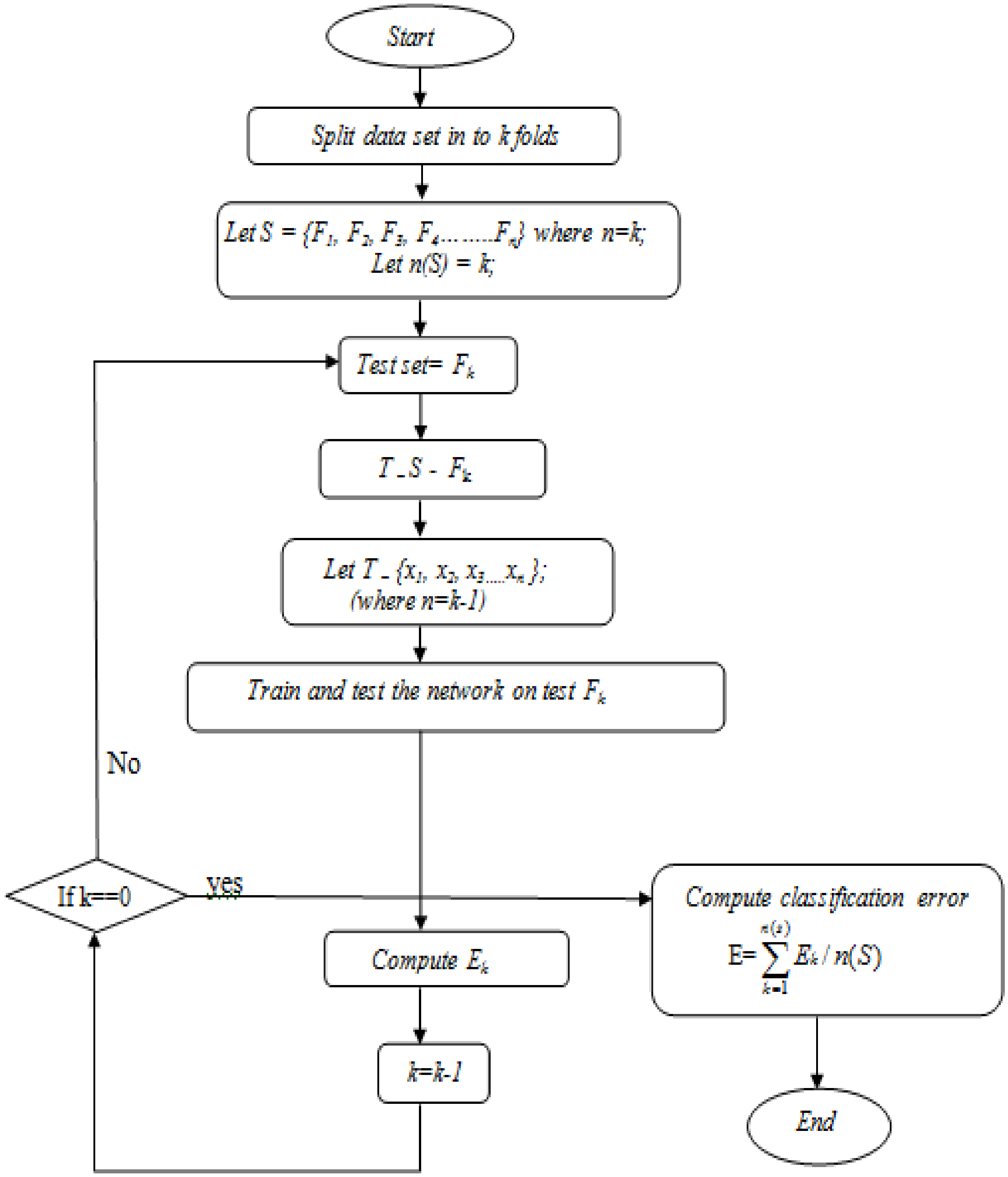
In this approach [23, 14, 9] (Figure 3) the dataset is divided into 'k' partitions or folds. There are generally following possibilities occurring in the cross-validation of data:

a) Holdout: It means splitting the data into two subparts such that one is used for training and the other is used for testing. It is the simplest technique in which the training and testing are done only once and the data is split randomly.

b) **K-fold**: The k-fold strategy (Figure 3) separates the data into subsets called folds where 'k' signifies the number of folds. In this technique, we divide the data set into 'k' folds (mutually distinct sub-parts equal to k). During the training of the network, iteratively, the (k-1) folds are used for training the dataset and the leftout fold (k^{th}fold) is used as a test set. In this way, each fold participates itself as test set only once and remaining (k-1) sets are used for training set until all distinct 'k' folds are employed as a test set. Finally, the mean value of the performances from all 'k' folds is taken for evaluation. The other strategies are leave-1-out and leave-p-out, which are modified form of general k-fold strategy in the way that they are more granular in selecting the number of instances from the given dataset.

VI.III. Normal TVT (Training-Validation-Test) approach

The third approach (Figure 5) starts by dividing the dataset into three parts called as training, validation [10] and test sets according to the specified ratio (generally 70:15:15 percentage ratio). The selected algorithm is applied to the training set and simultaneously validated on validation set [10]. The trained network giving best validation performance is selected for testing the algorithm on the test set. The reason for associating the validation set is to avoid over-fitting of the network. That is ANNs are known for giving good performance on training set but not as good on the test set. The validation set is used to evaluate the trained network before applying it on test data set and network giving best validation performance is chosen to evaluate the performance on the test dataset. The meaning of 'validation' lexeme is different regarding 'validation set' and the 'cross-validation approach'. In the former case 'validation' corresponds to the training-validation-test set approach and in a later case, 'validation' corresponds to k-fold cross-validation approach. The meaning of 'best validation performance' is, after few epochs if validation set is not showing improvement in performance or not giving better results with respect to past than training is stopped.



E_k=*Error for fold k*, *k= Number of folds*, $F_i=i^{th}$ *fold*, *n(S)=cardinality of S*

Figure 3. General k-fold approach

VI.IV. Proposed k-fold TVT approach

In the proposed methodology, we have given emphasis to the following three factors:
(1) Initialization of weights.
(2) Transfer functions on different layers as well as the new modified form of the tan-sigmoid transfer function.
(3) Combination of k-fold (cross-validation) approach and training-validation-test approach.

The proposed methodology (Figure 7) is the combination of training-validation-test approach and k-fold cross-validation approach so we can call it as a k-fold Training-Validation-Testing approach (k-fold TVT) (Figure 7).

In the proposed approach, after partitioning the dataset into k-folds, we assume (k-2) folds as training set and the remaining $(k-1)^{th}$ and k^{th} set as validation set and test set respectively. This method iteratively selects validation dataset in such a way that the test set is fixed and from the remaining folds, the validation set is chosen one by one from (k-1) folds, giving chance to each (k-1)'s fold as a validation set, thereby applying exhaustive approach (Figure 4). In this way we chose the network, giving best validation performance from all the training folds, that is the friendliest validation set to the test set and evaluate it. After that, we change the test set and again apply the same concept until all test sets are evaluated. There is no external factor responsible for over-fitting of the model and the reason is hidden in the data itself. Therefore, this approach just explores the fold which is most friendly in nature to test dataset. Also, the reason for exhaustive validation is single validation set is insufficient for performing the evaluation on the test set. The second characteristic of this approach is that each time during training, we initialize the network with different weights based on the seeds of the random number (approximately 10 seeds) and the best seeded network, that is best initialized (weighing) network is used, which finally gives better result after training.

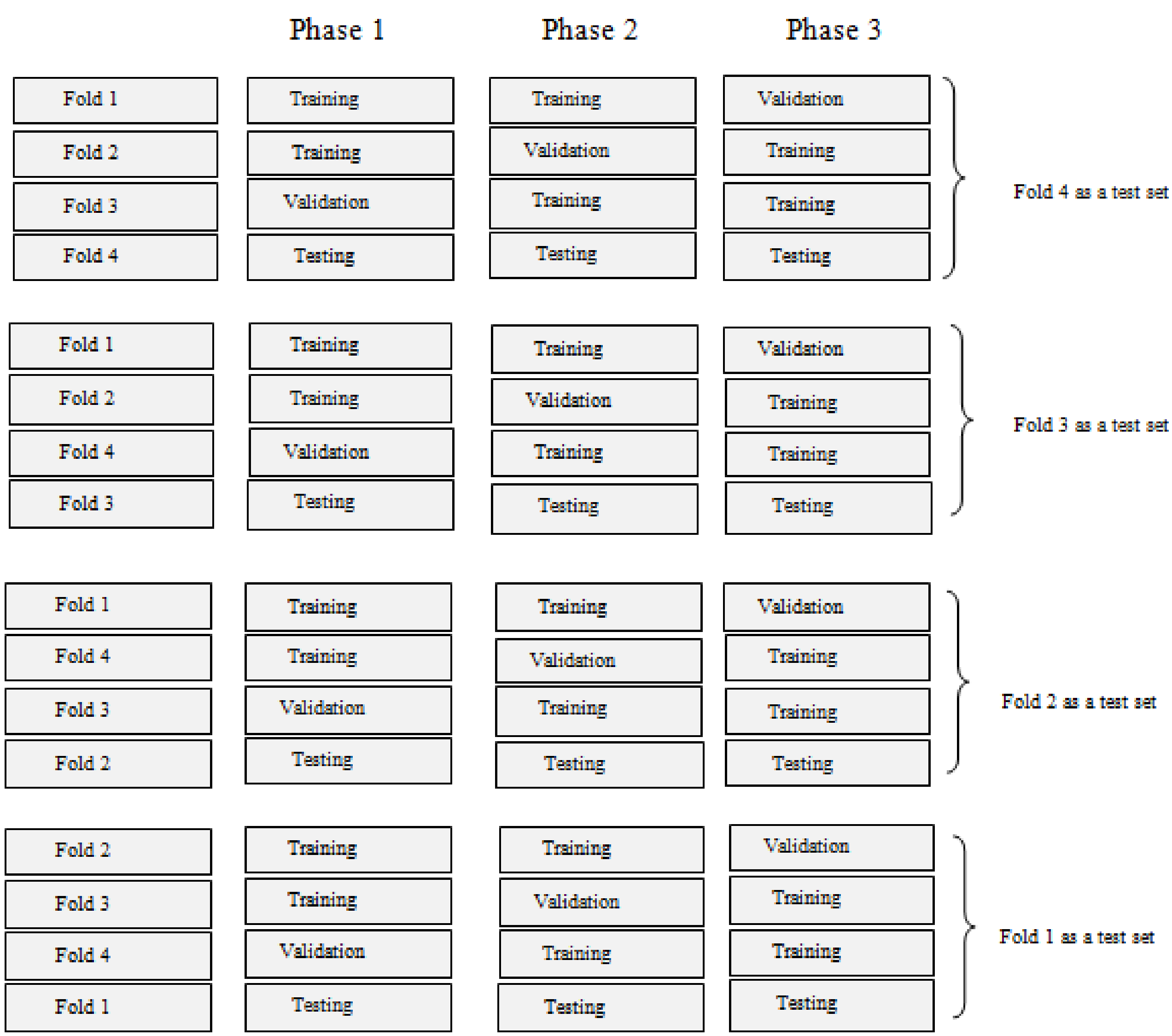


Figure 4. K-fold TVT approach with an example dataset having 4 folds describing exhaustive validation.

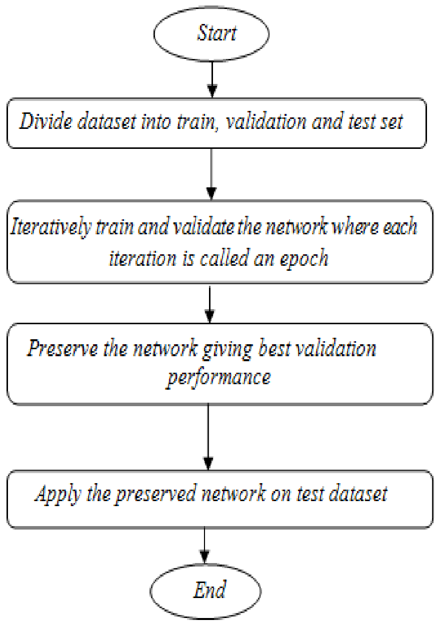


Figure 5. Normal training, validation and test approach

The reason for the same is that initialization of weights depends on the random number seed. As it is seen that initialization of weights is an important factor which contributes to the accuracy of the experiment at the end, so in the proposed methodology each time when we start the experiment, we take more than one sample of seed for random number generation and adopt the best seeded weighted configured network for starting the training, as it is all a matter of weights assigned to the network, and at the end we select network which gives the best result on the test dataset.

From the rigorous analysis, it is revealed that any 10 different random number seeds are sufficient to give optimized results among any 10 combinations of them.

The third feature which is applied in the proposed paper is the modified tan-sigmoid transfer function. The reason for this is that in almost all cases the use of the modified tan-sigmoid function with combinations of other transfer function is reflecting better results instead of applying tan-sigmoid function, which overlays all layers.

The experiments are performed with following (Table 2) types of ANNs using the proposed methodology. The details of transfer functions are shown in Table 3:

Note: Experiments are performed with 3-fold and 4-fold approach only.

VII. TRANSFER FUNCTIONS

Table 3 given below describes the transfer functions employed. The functions given in the table are all benchmark functions except the 'mtansig' function called as modified tan-sigmoid function as it is modified form of tan-sigmoid function. The reason for applying modified tan-sigmoid function is, from a rigorous experimental study, it is revealed that 'mtansig' is reflecting much better results in a lot of experiments instead of tan-sigmoid function, with other combinations of transfer functions may be 'tansig' also.

VIII. EXPERIMENTAL RESULTS AND THE STATISTICS

The results obtained on the two methodologies that are, the k-fold cross validation methodology and the k-fold methodology combined with the training-validation-test approach (k-fold TVT), on four mentioned data sets are presented below (Table 4 to Table12).

The general measures of performance are MSE (mean square error), classification error and the confusion matrix. The formulation of MSE error is mentioned above. The formula of classification error is:

$$\text{Classification Error} = \frac{\text{Number of misclassified instances}}{\text{Total instances}} \text{ X } 100$$

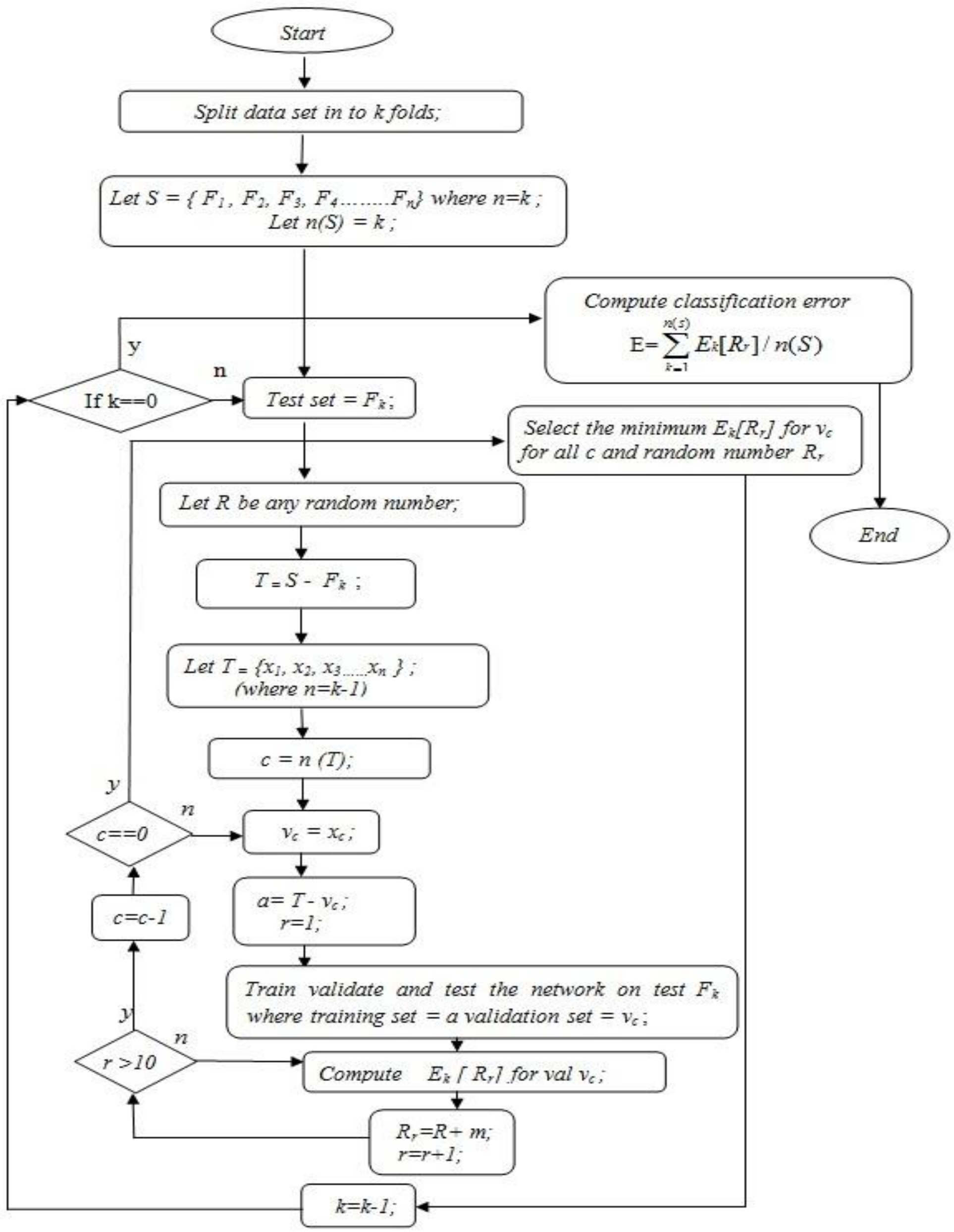
Confusion matrix for two class problems is given below:

		Actual class	
		Class 1	Not Class 1
Predicted class	Class 1	True positive(TP)	False positive(FP)
	Not class 1	False negative(FN)	True negative(TN)

Figure 6. General Confusion matrix

Table 2. Configuration parameters of ANNs

Dataset	Number of hidden layers	Transfer functions	Number of neurons in the hiddenlayer	Number of folds
Wine	1	mtansig and tansig	10	3
Iris	2	mtansig,tansig and purelin	20	3
Breast Cancer	4	mtansig,tansig,softmax and purelin	10	4
Red wine	3	mtansig,tansig,softmax and purelin	20	4



F_i=i^{th} fold, R_r=r^{th} random number, $E_k[Rr]$=Error on fold k for R_r, k= number of folds, n(S)=cardinality of S

Figure 7. Proposed k-fold TVT approach

Table 3. Transfer Functions employed in the experiments.

Function	Formula	Derivative
Tansig	*a=2 / (1+exp(-2*n))-1*	*d=1-(a*a)*
Purelin	*a=n*	*d=1*
Mtansig	*a=2/(1+exp(exp(-2*n)))-1*	*d=exp(-2*n)*(1-(a*a))*
Softmax	*a= exp(n)/sum(exp(n))*	**

**available in Matlab literature
P=Number of real positive cases in the data.
N=Number of real negative cases in the data.

Accuracy = $\frac{TP+TN}{TOTAL}$

Misclassification rate= $\frac{FP+FN}{TOTAL}$

TP: These are cases in which we predicted yes correctly predicted.
FP: Incorrectly predicted yes.
FN: Incorrectly predicted no.

Recall= $\frac{TP}{TP+TN}$

Precision= $\frac{TP}{TP+FP}$

TN: These are cases predicted no for correctly predicted no.

Table 4. Wine dataset (k-fold TVT approach)

Training function	Classification Error
Trainrp	**0.0**
Trainlm	1.67
Trainscg	2.8
Traincgb	2.8
Traincgp	3.92

Table 5. Wine dataset (k-fold approach)

Training function	Classification Error
Trainrp	3.92
Trainlm	8.39
Trainscg	**2.8**
Traincgb	59.93
Traincgp	43.32

Table 6. Iris dataset (k-fold TVT approach)

Training function	Classification Error
Trainrp	1.33
Trainlm	**0.66**
Trainscg	**0.66**
Traincgb	**0.66**
Traincgp	**0.66**

Table 7. Iris dataset (k-fold approach)

Training function	Classification Error
Trainrp	5.33
Trainlm	6
Trainscg	4.66
Traincgb	**4**
Traincgp	4.66

Table 8. Breast cancer (k-fold TVT approach)

Training function	Classification Error
Trainrp	2.14
Trainlm	2.43
Trainscg	**2**
Traincgb	2.28
Traincgp	2.14

Table 9. Breast cancer (k-fold approach)

Training function	Classification Error
Trainrp	4.86
Trainlm	**4.57**
Trainscg	6.29
Traincgb	5
Traincgp	6.15

Table 10. Red wine (k-fold TVT approach)	
Training function	Classification Error
Trainrp	**37.89**
Trainlm	38.46
Trainscg	38.89
Traincgb	39.39
Traincgp	39.64

Table 11. Red wine (k-fold approach)	
Training function	Classification Error
Trainrp	41.27
Trainlm	48.08
Trainscg	**41.21**
Traincgb	46.34
Traincgp	42.19

Table 12. Confusion matrix obtained on Red wine dataset (bold facing characters are class labels)

Predicted Actual →

	3	**4**	**5**	**6**	**7**	**8**
3	0	0	8	2	0	0
4	0	0	39	13	1	0
5	0	0	528	149	4	0
6	0	0	199	407	32	0
7	0	0	13	128	58	0
8	0	0	0	12	6	0

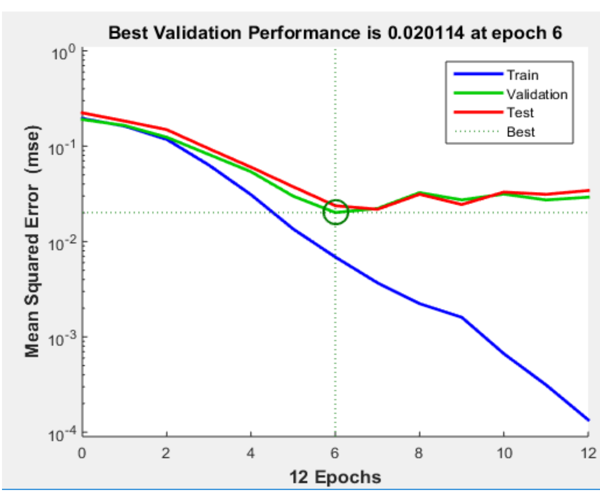


Figure 8(a)

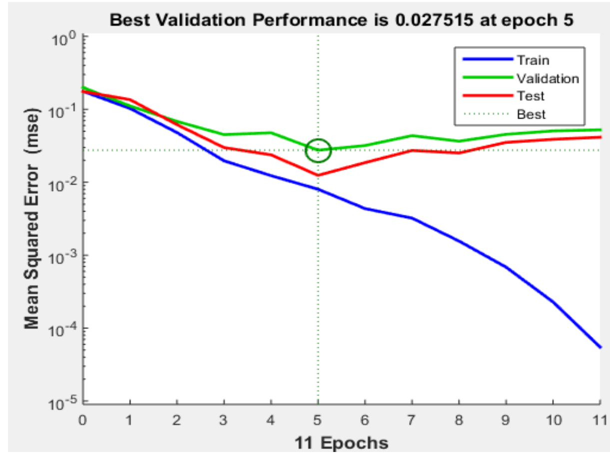


Figure 8(b)

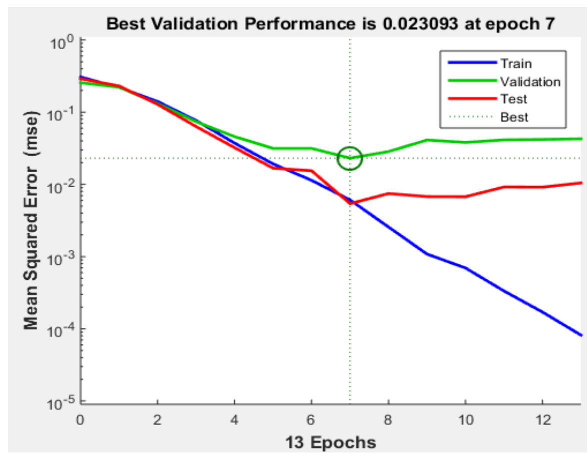


Figure 8(c)

Figure 8 (a) (b) (c). Graphs obtained by k-fold TVT approach on wine dataset by consecutively selecting test sets as fold 1, fold 2 and fold 3.

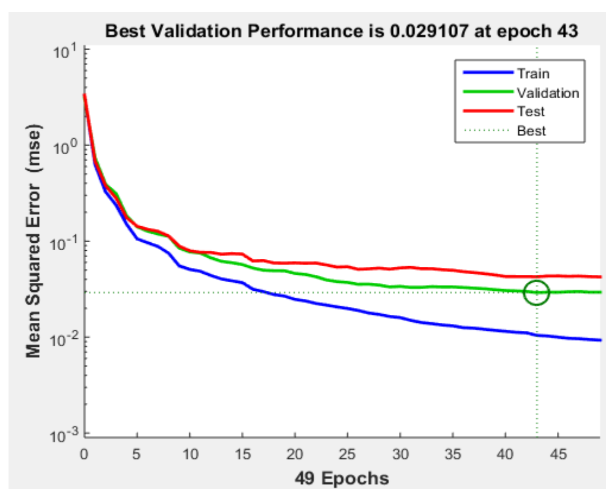


Figure 9(a)

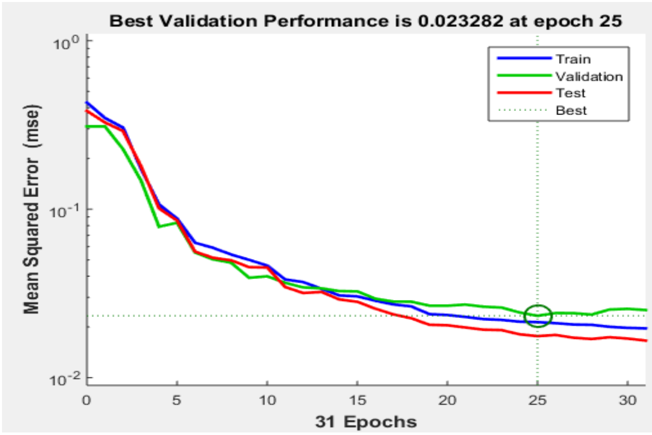


Figure 9(b)

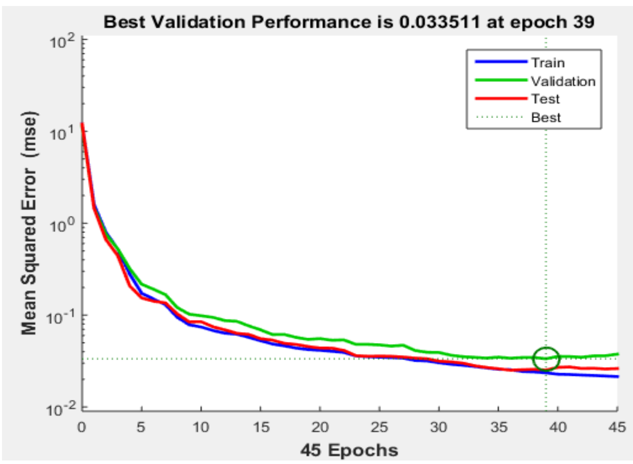


Figure 9 (c)

Figure 9 (a) (b) (c). Graphs obtained by k-fold TVT approach on Iris dataset by consecutively selecting test sets as fold 1, fold 2 and fold 3.

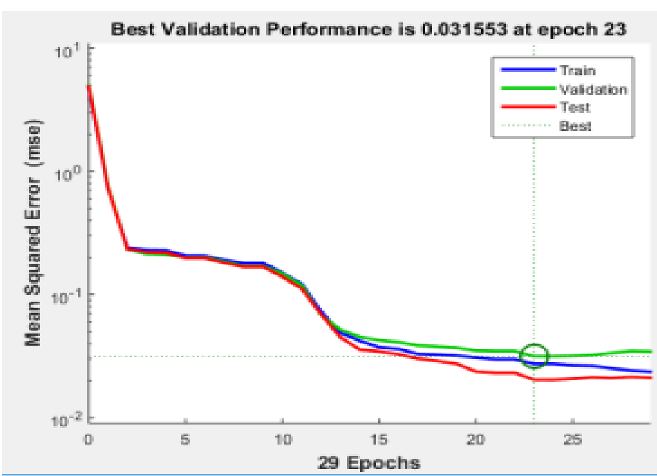


Figure 10(a)

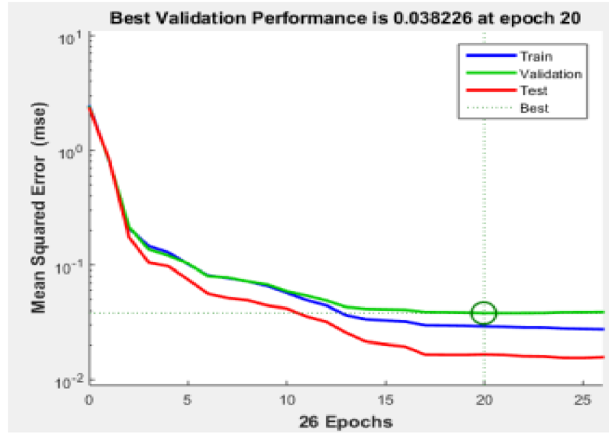


Figure 10(b)

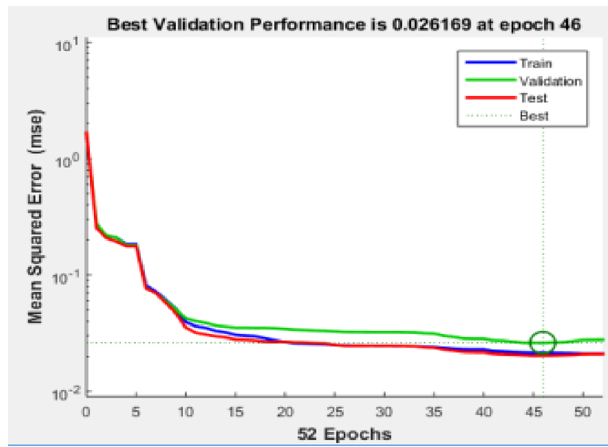


Figure 10(c)

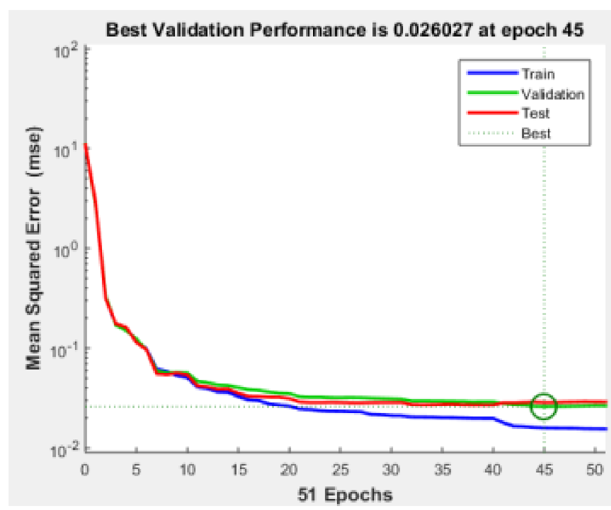


Figure 10(d)

Figure 10 (a) (b) (c) (d). Graphs obtained by k-fold TVT approach on Cancer dataset by consecutively selecting test set as fold 1, fold 2, fold 3 and fold 4.

From all the graphs obtained during the experiments performed using k-fold TVT approach, it is reflected that curves belonging to validation set (green) and test set (red) are imitating each other with very little margin. This is the reason that over-fitting is almost reduced.

VII.I. Statistical analysis

Performance of k-fold TVT classifier

Table 13

Dataset	Classification accuracy	Kappa statistic
Iris	99.33	0.99
Wine	100	1
Cancer	97.99	0.96

Table 14. Confusion matrix statistics on all datasets.

Dataset	Class	Recall(%)	Precision(%)
Iris	Setosa	100	100
	Versicolour	98.03	100
	Virginica	100	98
Cancer	Benign	99.15	97.8
	Malignant	95.95	98.34
Wine	Class-I	100	100
	Class-II	100	100
	Class-III	100	100

IX. Discussion

The new facts which are explored while performing the experiments on the above two methodologies are as follows:

(a) Initialization of weights as well as contents of each fold (that is indices of instances in each fold) effects the accuracy of the experiment.

(b)The combination of transfer functions gives better result in comparison to the single transfer function which overlays all layers/neurons.

(c) If we employ validation set in the experiment than chances of over-fitting are almost null.

(d) The use of a validation set in the experiment gives robust network.

The experiments are employing five well-known training algorithms (training functions) that is 'tarinlm', 'trainrp', 'trainscg', 'traincgb' and 'traincgp' present in Matlab. All these algorithms are variants of the back-propagation algorithm.

X. Advantages and disadvantages of the proposed approach

Advantages

(1) The main advantage of this approach is that it is very robust and most optimum results are obtained in either case, that is ANN domain or comparing with a non-ANN domain.

(2) It avoids over-fitting.

Disadvantages

(1) For large datasets, it is too time consuming as we have seen from the proposed flow chart that there are three loops to be implemented and it is too complex to implement.

(2) Another disadvantage of this approach is time consuming as we have to evaluate the results on different random number seeds.

XI. Comparison of results

It has been observed that the results obtained by performing the proposed approach outperform with the results obtained by other researchers. (P. Cortez [16] got classification accuracy (success rate) as 62.3% (tolerance level=0.5) and 43.2%(tolerance level=0.25)) for red wine based on Support Vector Machine (SVM) [4, 12, 24] and our experiment (Table 12) gives accuracy (success rate) as 62.1%. Further, our experiment gives the precision for class 5 as 67.09% and for class 6 it is 57.24% in comparison to the precision given by others [16] for class 5(tolerance=0.5) is 67.5% and for class 6(tolerance=0.5) it is 57.7%. SVM is giving optimum results when tolerance value is minimum as .001, approximately. So, we expect that SVM with so minimal [16] or no tolerance level cannot be able to give results obtained from ANN methodology. Further our experiment gives 0.66% as classification error on iris data set and 2.00% on cancer dataset whereas other techniques (Paolo P ([17]) gave 3.07% as classification error on Iris dataset and 6.15% on Cancer dataset. Some other research findings (Zhun-ga Liu [26](Table 15)) on Breast cancer, Wine and Iris dataset

(Misclassification rate or average error rate) are also compared and our finding give classification errors as 2.00%, 0.00% and 0.66% on Breast cancer, Wine and Iris dataset respectively, which is better than the results as shown in Table 15.

Table 15 Comparison of results displaying classification error

Technique	Breast cancer	Wine	Iris
K-NN	3.16	30.45	2.79
CART	5.59	11.67	5.33
ANN	3.97	63.33	4.67
SVM	3.95	**5.0**	2.67
BCKN	**2.54**	23.84	**2.55**
N-fold TVT	**2.0**	**0.00**	**0.66**
N-fold cross-validation approach	4.57	2.8	4.0

XII. Conclusion

From the results of the experiments performed, it is observed that, k-fold TVT approach gives better results in comparison to the pure k-fold approach or pure validation approach, either it is in ANN domain itself or if it is compared with other technologies like K-NN [1, 8, 17],CART (Classification and Regression Tree) [26] or SVM etc. Also, it is revealed that all results obtained from k-fold TVT approach on different datasets with different algorithms are very consistent, implying that the proposed technique is also robust. Also, we have compared the results of proposed ANN technique with other ANN techniques as well as various non-ANN techniques in the classification domain like support vector machines (SVM), probabilistic methods, decision trees, rule-based methods and it is observed that the proposed ANN approach is giving better results.

References

[1] B.W. Silverman, M. C. Jones, E. Fix, J.L. Hodges, "*An important contribution to nonparametric discriminant analysis and density estimation*", Int. Stat. Rev, Vol.57, Issue. 3, pp. 233–247, 1989.

[2] B. Widrow, M. E. Hoff, "*Adaptive switching circuits*", IRE1 WESCON Convention Record, pp. 96-104, 1960.

[3] C. B. Bishop, "*Neural Networks for pattern Recognition*", Oxford University Press, Oxford, 1995.

[4] C. Cortes, V. Vapnik, "*Support-vector networks*", Mach. Learn., Vol. 20, Issue. 3, pp. 273–297, 1995.

[5] D. E. Rumelhart, G.E. Hinton, R.J. Williams, "*Learning internal representation by error propagation*", Parallel distributed processing: Explorations in the microstructure of cognition, Vol.1, Bradford books, Cambridge, MA, 1986.

[6] D. S. Broomhead, Lowe, "*Multivariate functional interpolation and adaptive networks*", Complex Syst., Vol. 2, pp. 302–355, 1988.

[7] F. Rosenblatt, "*The perceptron: A probabilistic model for information storage and optimization in the brain*", Psychological Review, Vol. 65, Issue. 6, pp. 386-408, 1958.

[8] G. Shakhnarovich, T. Darell, P. Indyk, "*Nearest-Neighbors Methods in Learning and Vision*", MIT Press, 2006.

[9] I. Guyon, A. Saffari, G. Dror, G. Cawley, "*Model selection: beyond the Bayesian–frequentist divide"*, The Journal of Machine Learning Research, Vol. 11, pp. 61–87, 2010.

[10] I. V. Tetko, D. J. Livingstone, A.I. Luik, "*Neural network studies. 1. Comparison of over-fitting and overtraining*", Journal of chemical information and computer sciences, Vol. 35, Issue.5, pp. 826-833, 1995.

[11] J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational abilities". Proc. Natl. Acad. Sci. USA, Vol. 79, pp. 2554-2558, 1982.

[12] N. Barakat, A. P. Bradley, "Rule extraction from support vector machines: a review", Neurocomputing, Vol. 74, Issue. 1–3, pp.178–190, 2010.

[13] N. Rezazadeh, "Initialization of weights in deep belief neural network based on standard deviation of feature values in training data vectors", International journal of scientific research in computer science and engineering, Vol.5, Issue.4, pp.1-8,2017.

[14] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, "Choosing multiple parameters for support vector machines", Machines Learning, Vol. 46, Issue.1-3, pp. 131–159, 2002.

[15] O. D. Hebb, "*Organisation of Behaviour: A Neuropsychological Theory*", John Wiley and Sons, New York, 1949.

[16] P. Cortez, A. Cerderia, F. Almeida, T. Matos, J. Reis, "*Modelling wine preferences by data mining from physicochemical properties*", Decision Support Systems, Elsevier, Vol.47, Issue.4, pp. 547-553, 2009.

[17] P. Piro, R. Nock, F. Nielsen, M. Barlaud, "*Leveraging k-NN for generic classification boosting*", Neurocomputing, Vol. 80, pp. 3–9, 2012.

[18] P. Jian, M.Kamber, "*Data Mining: Concepts and Techniques*", 3rd Ed, 2012.

[19] S. N. Sivanandam, M. Paulraj, "Introduction to Artificial Neural Networks", Vikas Pub. Pvt. Ltd., India, 2009.

[20] Satish Kumar, "Neural Networks A Classroom Approach", Tata McGraw Hill, 2013.

[21] T. Jackie, C. Hung, "*Neural Network Model: Loss Minimization control of a PMSM with core Resistance Assesment*", International journal of scientific research in Network security and Communication,Vol.4, Issue.6, 2016.

[22] T. Kohonen, "*Self-organized formation of topologically correct feature maps*", Biological Cybernetics, Vol. 43, Issue. 1, pp. 59–69, 1982.

[23] V. N. Vapnik, "*Statistical learning theory*", Wiley-Interscience, 1998.
[24] W. An, M. Liang (2013), "*Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises*", Neurocomputing, Vol. 110, June, pp. 101–110, 2013.
[25] W. Pitts, & W. S. McCulloch, "*A logical calculus of the ideas immanent in nervous activity*", Bull. Math. Biophys., Vol. 5, pp.115–133, 1943.
[26] Zhun-ga Liu, Quan Pan, Jean Dezertb, "*Classification of uncertain and imprecise data based on evidence theory*", Neurocomputing,Vol. 133, pp. 459–470, 2014.

Authors Profile:

Mr. Kshitij Tripathi did his graduation in Mathematics and Masters in Computer Science from Barkatullah University Bhopal. Thereafter he completed M.C.A. He has obtained diploma in advance computing from C-DAC (A research & development organization belongs to Pune University). He has cleared Ugc accredited examination for the post of Assistant Professor. Currently, he is pursuing Ph.D. in Neurocomputing and working as a faculty in Department of Computer Applications, The Maharaja Sayajirao University of Baroda, Vadodara, India.

Mr. Rajendra G. Vyas did his Ph.D. in Mathematics, in the Field of Fourier Analysis from Department of Mathematics, The Maharaja Sayajirao University of Baroda, Vadodara, India,in 1996. He is currently working as a Professor in Department of Mathematics, Faculty of Science, The Maharaja Sayajirao University of Baroda, Vadodara, India. He has published more than 40 research papers in national and international journals of repute. He is a member of the editorial board in many Mathematics journals, a reviewer for Mathematical Reviews, published by American Mathematical Society as well as a reviewer for Zentralblatt, published by European Mathematical Society.

Mr. Anil Kumar Gupta is actively involved in Data Mining and Pattern Recognition research. His research work brings many new ideas in Classification and Clustering techniques. He has Ph.D. in Computer Science. He is currently serving as H.O.D of the Department of Computer Science and Applications, Barkatullah University, Bhopal. Currently he is guiding 7 Ph.D. research scholars. He has ten years of research and an over twenty years of teaching experience.

