RESEARCH ARTICLE OPEN ACCESS

Improved Genetic Algorithm Based Feature Selection Strategy Based Five Layered Artificial Neural Network Classifier (Iga – Flann)

M. Praveena [1], Dr.V.Jaiganesh[2]

[1]Assistant Professor, Dept. of Computer Science, Dr.SNS Rajalakshmi College of Arts and Science,Coimbatore, Tamil Nadu, India.

[2]Professor, Dept. of PG & Research, Dr. NGP Arts and Science College, Coimbatore,Tamil Nadu, India.

Abstract:

Data classification is one of the investment research areas in the field of data mining. Machine learning algorithms such as naive bayes, neural network, and support vector machine are most regularly used for performing the classification task. Supervised learning is one of its kinds where the datasets consist of class labels and the machine learning classifier are trained first using that. It is to be noted that feature selection plays a vital role in developing the classification accuracy of the supervised machine learning classifiers. This research work aims in proposing an improved genetic algorithm based feature selection planning based five layered artificial neural network classifier. Around 20 datasets are collect from the UCI repository. Implementations are carried out using MATLAB tool. Performance metrics such as prediction efficiency and time taken for prediction are taken into account to conduct the performance evaluation of the expected classifier. Simulation results portrays that the proposed IGA-FLANN classifier outperforms the existing classifiers.

Keywords — **Machine learning, boosting, neural network, genetic algorithm, feature selection, data mining, MATLAB.**

I. INTRODUCTION

Data mining is one among the thrust research areas in the field of computer science. Yet there is a knowledge data discovery process helps the data mining to abstract hidden information from the dataset there is a huge scope of machine learning algorithms. Particularly supervised machine learning algorithms gain major importance in data mining research. Machine learning shortly describe as ML is a kind of artificial intelligence (AI) which makes available computers with the efficiency to be trained without being overtly programmed. ML learning spotlights on the development of computer programs which is capable enough to modify when disclosed to new-fangled data. ML algorithms are broadly classified into three categories namely supervised learning, unsupervised learning and reinforcement learning and is shown in Fig.1. The progression of machine learning is comparable to that of data mining. Both data mining and machine learning consider or explore from end to end data to seem for patterns. On the other hand, in preference to extracting data for human knowledge as is the case in data mining applications; machine learning makes use of the data to discover patterns in data and fine-tune program actions therefore.

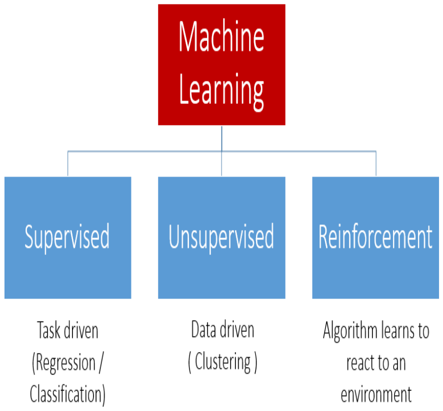
Machine Learning
Supervised
Unsupervised
Reinforcement
Task driven (Regression / Classification)
Data driven (Clustering)
Algorithm learns to react to an environment

Fig.1. Machine Learning and its Types

Supervised machine learning is the mission of assume a meaning from labelled training data which has a set of training examples. As far as supervised learning is concerned, every example is a brace containing an input object (which is usually a vector quantity) and a compulsory output value (may also be referred as supervisory signal). A supervised learning algorithm at first performs the analysis task from the training data and composes a contingent function, in order to map new examples. An optimal setting apparently facilitates the algorithm to exactly courage the class labels for concealed instances and the same needs the supervised learning algorithm to reduce from the training data to concealed situations in a "rational" manner. The supervised methods are perhaps used in various application areas that include marketing, finance, manufacturing, testing, stock market prediction, and so on. This research work aims in proposing a better genetic algorithm based feature selection strategy based five layered artificial neural network (IGA-FLANN) classifier. The aim of IGA-FLANN is to better the prediction accuracy and also to decrease the time taken for classification.

II. RELATED WORKS

2.1. Recent Works on Decision Trees

Lertworaprachaya et al., 2014 proposed a new model for compose decision trees using interval-valued fuzzy membership values. Most existing fuzzy decision trees do not examine the uncertainty associated with their membership values; however, correct values of fuzzy membership values are not always possible. Because of that, the authors represented fuzzy membership values as distance to model uncertainty and employ the look-ahead based fuzzy decision tree induction method to construct decision trees. The authors also inspected the significance of different neighbourhood values and define a new parameter unkind to specific data sets using fuzzy sets. Some examples are provided to establish the effectiveness of their approach.

Bahnsen et al. 2015 proposed an example-dependent cost-sensitive decision tree algorithm, by incorporating the different example-dependent costs into a new cost-based corruption measure and new cost-based pruning criteria. Subsequently, using three different databases, from three real-world applications namely credit card fraud detection, credit scoring and direct marketing, the authors calculate their proposed method. Their results showed that their proposed algorithm is the best performing method for all databases. Additionally, when compared against a standard decision tree, their method builds extremely smaller trees in only a fifth of the time, while having a superior performance consistent by cost savings, leading to a method that not only has more business-oriented results, but also a method that creates simpler models that are easier to analyze.

Online decision trees from data streams are usually unable to handle concept drift. Blanco et al., 2016 proposed the Incremental Algorithm Driven by Error Margins (IADEM-3) that mainly carry out two actions in response to a concept drift. At first, IADEM-3 resets the variables affected by the change and continue unbroken the structure of the tree, which allows for changes in which ensuing target functions are very similar. After that, IADEM-3 creates alternative models that replace parts of the main tree when they decidedly improve the accuracy of the model, thereby rebuilding the main tree if needed. An online change detector and a non-parametric statistical test based on Hoeffding's bounds are used to agreement that significance. A new pruning method is also incorporated in IADEM-3, making sure that all split tests previously installed in decision nodes are useful. Their learning model is also viewed as an in concert of classifiers, and predictions of the main and alternative models are combined to classify unlabeled examples. IADEM-3 is empirically compared with various well-known decision tree induction algorithms for concept batch detection. The authors depict that their new algorithm often reaches higher levels of accuracy with smaller decision tree models, continue the processing time bounded, irrespective of the number of instances processed.

Predicting learning styles in conversational intelligent coaching systems using fuzzy decision trees has been proposed by Crokett et al., 2017. Prediction of learning style is carried out by confining independent behaviour variables during the coaching conversation with the highest value variable. A weakness of their approach is that it does not take into consideration the communication between behaviour variables and, due to the uncertainty congenitally present in modelling learning styles, small differences in behaviour can lead to incorrect predictions. Subsequently, the learner is presented with coaching material not suited to their learning style. Because of the above mentioned challenges a new method that uses fuzzy decision trees to build a series of fuzzy predictive models connecting these variables for all dimensions of the Felder Silverman Learning Styles model. Results using live data by the authors showed that the fuzzy models have elevated the predictive certainty across four learning style dimensions and promote the discovery of some interesting relationships amongst behaviour variables.

2.2. Recent Works on Support Vector Machine (SVM)

Motivated by the KNN trick presented in the weighted twin support vector machines with local information (WLTSVM), Pan et al., 2015 expected a novel K-nearest neighbour based structural twin support vector machine (KNN-STSVM). By applying the intra-class KNN method, different weights are disposed to the samples in one class to build up the structural information. For the other class, the excessive constraints are deleted by the inter-class KNN method to speed up the coaching process. For large scale problems, a fast clip algorithm is further introduced for increase of rate. Comprehensive experimental results on twenty-two datasets determine the efficiency of their proposed KNN-STSVM.

It is noteworthy that existing structural classifiers do not balance structural information's relationships both intra-class and inter-class. Connecting the structural information with nonparallel support vector machine (NPSVM), D. Chen et al. 2016, construct a new structural nonparallel support vector machine (called SNPSVM). Each model of SNPSVM examine not only the density in both classes by the structural information but also the reparability between classes, thus it can fully accomplishment prior knowledge to directly improve the algorithms generalization capacity. Moreover, the authors applied the enhance alternating direction method of multipliers (ADMM) to SNPSVM. Both their model itself and the solving algorithm can assurance that it possibly would deal with large-scale classification problems with a huge number of occurrence as well as features. Experimental results show that SNPSVM is superior to the other current algorithms based on structural information of data in both computation time and classification accuracy.

Peng et al., 2016 formulated a linear kernel support vector machine (SVM) as a regularized least-squares (RLS) problem. By defining a set of signal variables of the errors, the solution to the RLS problem is represented as an equation that relates the error vector to the indicator variables. Through dissolution the training set, the SVM weights and bias are expressed analytically using the support vectors. The authors also determine how their approach naturally extends to sums with nonlinear kernels whilst avoiding the need to make use of Lagrange multipliers and duality theory. A fast iterative solution algorithm based on Cholesky decomposition with permutation of the support vectors is proposed as a solution method. The properties of their SVM formulation were analyzed and compared with standard SVMs using a simple example that can be illustrated graphically. The correctness and behaviour of their proposed work has been determined using a set of public benchmarking problems for both linear and nonlinear SVMs.

Utkin and Zhuk., 2017 proposed a well-known one-class classification support vector machine (OCC SVM) dealing with interval-valued or set-valued training data. Their key idea is to represent every interval of training data by a finite set of precise data with estimated weights. Their representation is based on replacement of the interval-valued expected risk produced by interval-valued data with the interval-valued expected risk produced by imprecise weights or sets of weights. It can also be considered that, the interval uncertainty is replaced with the estimated weight or probabilistic uncertainty. The authors showed how constraints for the estimated weights are incorporated into dual quadratic programming problems which can be viewed as development of the well-known OCC SVM models. With the help of numerical examples with synthetic and real interval-valued training data the authors decorate their proposed approach and investigate its properties.

2.3. Recent Works on Adaboost

Universum data usually does not belong to any class of the training data, has been applied for training better classifiers. Xu et al., 2014 addressed a novel boosting algorithm called UAdaBoost which possibly would boost the classification performance of AdaBoost with Universum data. UAdaBoost chooses a function by minimizing the loss for labelled data and Universum data. The cost function is minimized by a greedy, stagewise, functional gradient procedure. Each training stage of UdaBoost is fast and efficient. The standard AdaBoost weights labeled samples during training emphasis while UAdaBoost gives an explicit weighting design for Universum samples as well. Also the authors described the practical conditions for the capability of Universum learning. These conditions are based on the analysis of the distribution of at once predictions over training samples. By their experimental results the authors claimed that their method can obtain superior performances over the standard AdaBoost by selecting proper Universum data.

Sun et al., 2016 quoted a representative access named noise-detection based AdaBoost (ND_AdaBoost) in order to boost the robustness of AdaBoost in the two-class classification scenario. In order to courage the dilemma a robust multi-class AdaBoost algorithm (Rob_MulAda) is proposed by the authors whose key ingredients consist in a noise-detection based multi-class loss function and a new weight updating scheme. The authors claims that their experimental study announce that their newly-proposed weight updating scheme is indeed more robust to mislabeled noises than that of ND_AdaBoost in both two-class and multi-class scenarios. As well, through the comparison experiments, the authors also verified the performance of Rob_MulAda and provide a suggestion in choosing the most convenient noise-alleviating approach according to the concrete noise level in practical applications.

Baig et al., 2017 presented a boosting-based method of learning a feed-forward artificial neural network (ANN) with a single layer of hidden neurons and a single output neuron. At first, an algorithm called Boostron is represent which learns a single-layer perceptron using AdaBoost and decision stumps. It is then continue to learn weights of a neural network with a single hidden layer of linear neurons. At last, a novel method is introduced by the authors to incorporate non-linear activation functions in artificial neural network learning. Their proposed method uses continue representation to approximate non-linearity of activation functions, learns the coefficients of nonlinear terms by AdaBoost which adapts the network parameters by a layer-wise iterative traversal of neurons and an appropriate reduction of the problem. Comparison of various neural network models learned the proposed methods and those ground using the least mean squared learning (LMS) and the resilient back-propagation (RPROP) is provided by the authors.

Miller and Soh 2015 proposed a novel cluster-based boosting (CBB) approach to address limitations in boosting on supervised learning (SL) algorithms. Their CBB approach partitions the training data into clusters consist of highly similar member data and integrates these clusters directly into the boosting process. Their CBB approach experiments to address two specific limitations for current boosting both resulting from boosting converge on incorrect training data. The first one is filtering for subsequent functions when the training data contains dangerous areas and/or label noise; and the second one is over fitting in subsequent functions that are forced to learn on all the incorrect instances. The authors demonstrated the capability of CBB through extensive empirical results on 20 UCI benchmark datasets and proclaimed that CBB achieves superior predictive accuracy that use selective boosting without clusters.

III. PROPOSED WORK

The proposed work has contributions three fold. At first an improved genetic algorithm based feature selection action is portrayed. Once when the features are selected then in the next stage, verdict association is done. At the third stage, five layered artificial neural network is proposed.

3.1. Improved Genetic Algorithm based Feature Selection Strategy

Let, $P=\{P_1,P_2,\ldots,P_n\}$ be the feature set where *n* is the number of features and $U=\{O_1,O_2,\ldots,O_m\}\in R^{m\times n}$ be a given dataset with *m* objects where $m \geq n$. The feature selection method can be viewed as a mapping $g(P,U,D)\rightarrow\{P'\}$, where, $g(\cdot)$ is the feature selection method, *D* is the decision attribute representing class labels and $P'\subseteq P$, where, $|P'|=k$, number of selected features, $k \ll n$. The goal of proposed method is to find P' which will be highly relevant to D and less related to each other.

To select the most feasible and compact feature subset, the improved GA with mutation pool is being proposed in this research work. The search for feature subset is done based on δ and conditional mutual information measure. Every dataset is first fed into the proposed theoretical search based feature selection algorithm to obtain a subset of features, which are evaluated and finally after convergence, the non-dominated feature subset is selected.

As GA is a population based stochastic search algorithm, the initial population for proposed improved GA is created at random. The dimension of population set is $p\times n$ where p the defined population size is and n is the number of features in the dataset. Binary string representation of chromosome is chosen in this work. The length of the string is same with total number of features present in the dataset. Each gene consists of only two values, ‘1’ and ‘0’, which implies that the index feature is present and absent respectively in current subset. So, a chromosome C of length K is represented as follows:

In the proposed improved GA, every population member has equal probability of being a parent. Single point crossover approach is used to get offspring from parent chromosomes. The main reason behind application of alteration is to diversity in the population. The method uses a alteration pool of different mutation strategies. At any moment for any current offspring, alteration strategy to be applied is chosen dynamically from the pool. The reason behind selecting such alteration strategy is to maximize quality of population member by utilizing effectiveness of those individual mutual strategies. To acquire the robust and noise resilient feature subset, the main dataset is first sampled using sample with replacement strategy (SWR) into *E* number of dataset of equal dimension with main dataset. The SWR strategy is used to make *E* number of dataset from the main dataset with intentional perturbation into it and reason behind choosing SWR strategy is that in this strategy two sample values are independent. Goal of propose method is to apply multiple improved GAs on multiple sub datasets to make robust and generalized feature subset. Among the final reduct set, pair wise dominance check is opted and at last the final selected feature subset is that one which has its domination count 0 implies it is a non-dominated solution. If multiple feature subsets are present with dominance count zero then objective wise weak dominance concept is opted to get most feasible one. The improved GA algorithm is given below.

Algorithm 1 Improved genetic algorithm based feature selection (FSGA).

```
1: **Input:** Sampled Dataset
2: **Begin**
$N$ 3: Generate Population $(P)$ of size $N$
4: Evaluate both fitness functions for all population members
5: Calculate global best of each fitness function
6: **Repeat**
$i=1..N$ 7:    *for* $i=1..N$
$F_i$ 8:    First_parent = $F_i$
9:    Select another parent $F_i$ from the population
10:    Apply single point crossover between $F_i$ and $F_i$ to produce offspring
11:    Dynamically select mutation strategy from mutation pool
12:    Apply mutation to offspring
13:    Compute two fitness values for offspring
14:    **if** both fitness values of offspring is optimal than the global best **then**
           Offspring replaces one of its parents and update the global best
       **Else if** Offspring dominates any parent **then**
           Offspring replaces that parent
       **End if**
15: **End** *for*
16: **Until** predefined number of generations are exhausted
17: **Return** Non-dominated set of population members as solution
18: **End**
19: **Output:** Non-dominated reduct set
```

3.2. Verdict Consolidation

In proposed work a nondominated solution based verdict consolidation is considered for final feature subset selection. The ensemble feature selector runs E number of individual feature selectors in parallel which provides $reduct$ set $Reduct_i$ for $i=1,2,...,E$. As a result, final $reduct$ set

R is obtained by combining all such $reducts$, where, $R = Reduct_1 \cup Reduct_2 \cup ... \cup Reduct_E$.

Now each pair of $reduct$ x and y in $reduct$ set R is compared based on their objective functions and if all objective function values for *x* are optimum compare to that of y, we say that x dominates y and denote it as $x \succ y$. For each reduct in R, a dominating factor is set initially as zero and it increases by one when it dominates another $reduct$ in R . Finally, a $reduct$ is selected in set F if it dominates all the reducts, i.e., if it's dominating factor is $|R|-1$. If multiple such $reduct$ is there in F then one is chosen at random as the final reduct.

Algorithm 2 Find optimal features by verdict consolidation.

```
$R = Reduct_1 \cup Reduct_2 \cup ... \cup Reduct_E$1: **Input** : All generated reducts $R = Reduct_1 \cup Reduct_2 \cup ... \cup Reduct_E$
$F = \phi$ 2: **Begin** $F = \phi$
$d_x = 0$ 3: **for** each $reduct$ $x$ in $R$ set dominating factor $d_x = 0$
$R$4: **for** each $reduct$ $x$ in $R$
$R$5:     **for** each $reduct\ y(\neq x)$ in $R$
6:         **if** $(x \succ y)$ then increase $d_x$ by one
$F = F \cup \{x\}$7: **if** $(d_x == |R|-1)$ then $F = F \cup \{x\}$
8: **End-for**
$F$ 8: **Return** $F$
9:     **End**
10: **Output:** Set $F$ of optimal feature subset
```

3.3. Five Layered - ANN (FL – ANN) Classifier

Once after the feature selection is carried out, FL – ANN is used to classify. FL – ANN is a five layered RBF based classifier neural network that makes use of gradient descent approach and regression based classification. It optimizes flattening parameter of RBF kernel through grade descent approach.

Applied input vector x is transmitted to classification layer through input layer. Pattern layer includes one neuron for each training datum with RBF kernel. Squared Euclidean distance between input vector x and training data vector t is calculated as in (1) where p denotes total number of training data at pattern layer.

$$dist(j) = \|x - t_j\|^2, 1 \le j \le p \quad ... (1)$$

Calculated squared Euclidean distances are used in RBF kernel function as in (2) where r (j) denotes output of jth training data and σ represents flattening parameter. Outputs of RBF kernel function are the output values of pattern layer neurons. Moreover, this layer includes N target values of each training datum determined by corresponding class.

$$r(j) = e^{\left(-1 \times \frac{dist(j)}{2\sigma^2}\right)}, 1 \le j \le L \quad ... (2)$$

When a training datum belongs to ith class then its ith value will be 0.9 and others will be 0.1, as given in (3).

$$y(j,i)=\begin{cases}0.9, & x_j \text{ belongs to } i^{th} \text{ class} \quad 1\le i\le N\\ 0.1 & \text{else} \quad 1\le j\le p\end{cases}$$ ... (3)

N+1 neurons are placed at summation layer where N is the total number of classes and additional one term to N is for one neuron to obtain denominator. FL-ANN uses diverge effect term at summation layer to increase the distances among classes. Diverge effect term value is calculated as in (4) where d (j, i) denotes diverge effect term of j^{th} training data and i^{th} class. y_{max} is initialized to 0.9 which denotes the maximum value of y(j, i). y_{max} value is updated with the maximum value of output layer after each iteration of optimization. Diverge effect term is calculated by N neurons of summation layer. This calculation includes aggressive form of y (j, i) − y_{max} to increase the effect of y (j, i).

$$d(j,i)=e^{(y(j,i)-y_{max})}*y(j,i)$$ ... (4)

Diverge effect term is used in calculating nominator values at summation layer as in (5). Moreover, denominator value is also calculated at this layer as in (6).

$$u_i=\sum_{j=1}^{p} d(j,i)*r(j), 1\le i\le N$$ ... (5)

When N neurons, represented with u_i, calculate nominator values by summing dot product of diverge effect terms and classification layer outputs, other neuron calculates denominator value the same as FL-ANN represented by D.

$$D=\sum_{j=1}^{p} r(j)$$ ... (6)

Each class is represented with a neuron at normalization layer. These neurons divide corresponding nominator value by denominator value calculated at summation layer, according to (7) where c_i denotes normalized output of i^{th} class.

$$c_i=\frac{u_i}{D}, 1\le i\le N$$ ... (7)

Class of input vector is determined at output layer through the winner decision mechanism as given in (8) where c is the output vector of normalization layer, c_{id} and id denote winner neuron value and indices of the class, respectively.

$$[c_{id}, id]=\max(c)$$ ... (8)

Gradient descent based collective learning is utilized in FL-ANN for obtaining optimized flattening parameter value. Each training datum at pattern layer is sequentially applied to neural network and three steps are executed until maximum iteration limit exceeds. Firstly, squared error e is calculated for each input, as in (9) where y(z, id) represents the value of z^{th} training input data for id^{th} class and c_{id} is value of winner class.

$$e=(y(z,id)-c_{id})^2$$ ... (9)

IV. RESULTS AND DISCUSSIONS

Table 1. Dataset Name, No. of Instances and No. of Features

Dataset Name	No. of Instances	No. of Features
Blood	748	5
Bupa	345	7
Car	1728	6
Contraceptive	1473	9
Credit	30000	24
Diagnostic	569	32
Ecoli	336	8
Ionosphere	351	34
Mammography	961	6
monks – 1	432	7
monks – 2	432	7
monks – 3	432	7
Parkinsons	197	23
Pima	768	8
Prognostic	198	34
Sonar	208	60
Spect	267	22
Tic-Tac-Toe Endgame	958	9
Vert	310	6
Yeast	1484	8

Table 2. Predictive Accuracy of the Algorithms

Dataset	Algorithms				
	MLP	SVM	TREE	RBF	IGA – FLANN
blood	0.8	0.78	0.78	0.78	0.83
bupa	0.7	0.72	0.67	0.69	0.76
car	1	0.99	0.96	0.97	1
cont.	0.72	0.69	0.7	0.68	0.77
credit	0.87	0.87	0.86	0.86	0.89
diag.	0.97	0.98	0.95	0.98	0.99
ecoli	0.99	0.99	0.99	0.98	0.99
iono.	0.91	0.91	0.89	0.94	0.96
mamm.	0.82	0.82	0.82	0.81	0.85
mks-1	1	1	0.99	0.96	1
mks-2	1	1	0.68	0.71	1
mks-3	1	1	1	1	1
park	0.92	0.94	0.9	0.9	0.97
pima	0.77	0.77	0.75	0.77	0.81
prog.	0.8	0.77	0.79	0.8	0.84
sonar	0.81	0.82	0.79	0.83	0.89
spect	0.81	0.82	0.84	0.85	0.88
Tic	0.97	1	0.94	0.97	1
vert	0.82	0.82	0.83	0.83	0.87
yeast	0.65	0.66	0.66	0.64	0.68

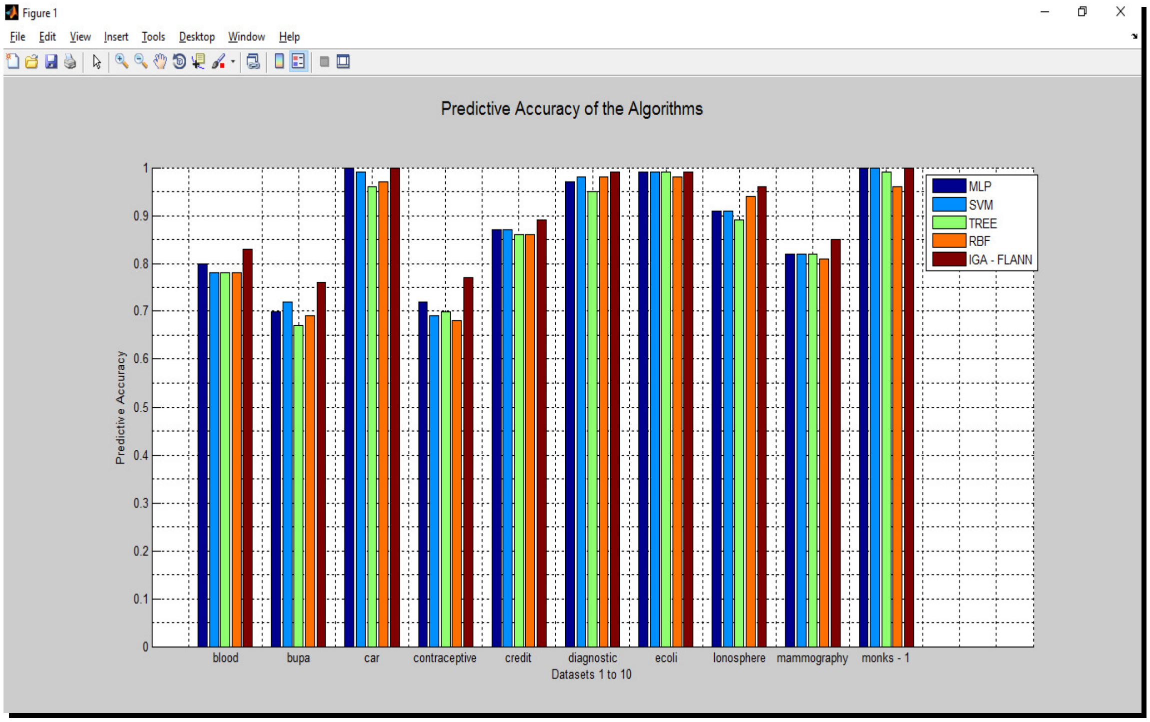

Fig.2. Predictive Accuracy Comparison for the datasets 1 to 10

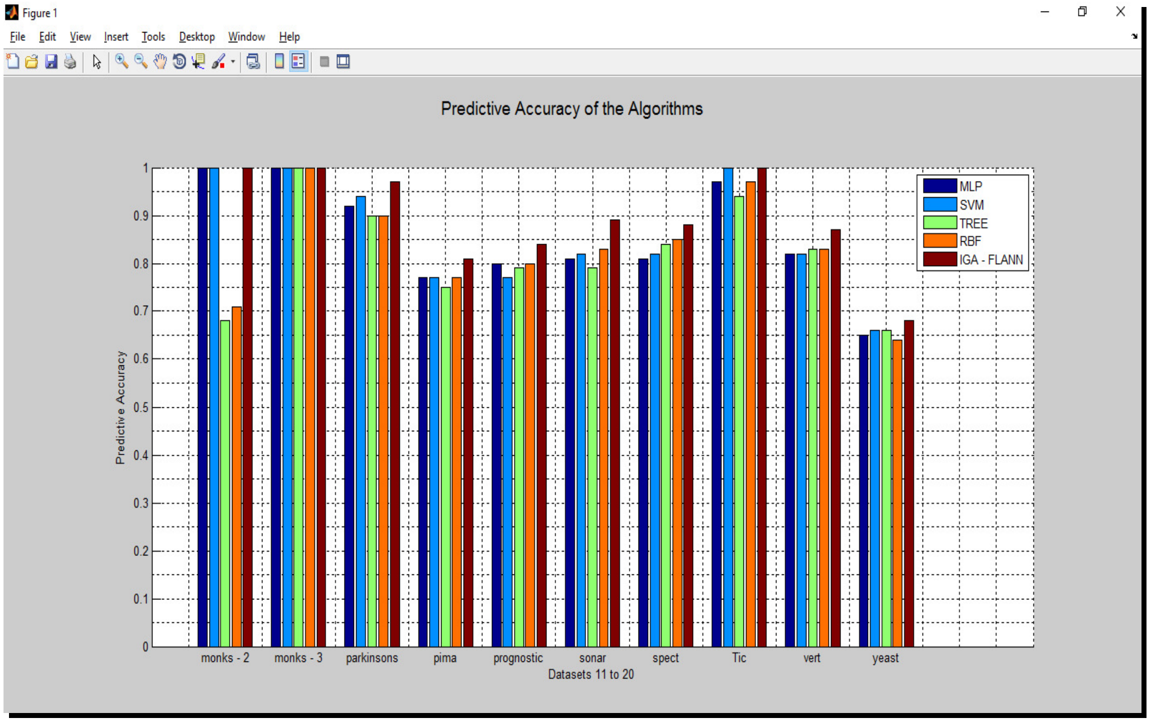

Fig.3. Predictive Accuracy Comparison for the datasets 11 to 20

Table 3. Time Taken by the Algorithms for Classification (in milliseconds)

Dataset	Algorithms				
	MLP	SVM	TREE	RBF	IGA - FLANN
blood	2396	2093	2377	2839	1032
bupa	2436	2699	2056	2496	1948
car	2251	2117	2157	2385	1682
cont.	2157	2369	2978	2635	1732
credit	19377	19507	19563	19674	15782
diag.	7361	7962	7233	7676	3982
ecoli	2268	2894	2367	2665	1846
iono.	7588	8000	7558	7569	3901
mamm.	2875	2586	2210	2612	1738
mks-1	2039	2612	2185	2076	1936
mks-2	2329	2956	2940	2120	1726
mks-3	2024	2003	2062	2497	1639
park	4047	4646	4526	4879	2393
pima	2052	2533	2813	2992	1392
prog.	2686	2283	2208	2644	1888
sonar	8686	8681	8297	8788	5189
spect	5348	5821	5655	5455	2291
Tic	2250	2290	2574	2745	1749
vert	2869	2497	2054	2645	1638
yeast	2088	2604	2971	2222	1843

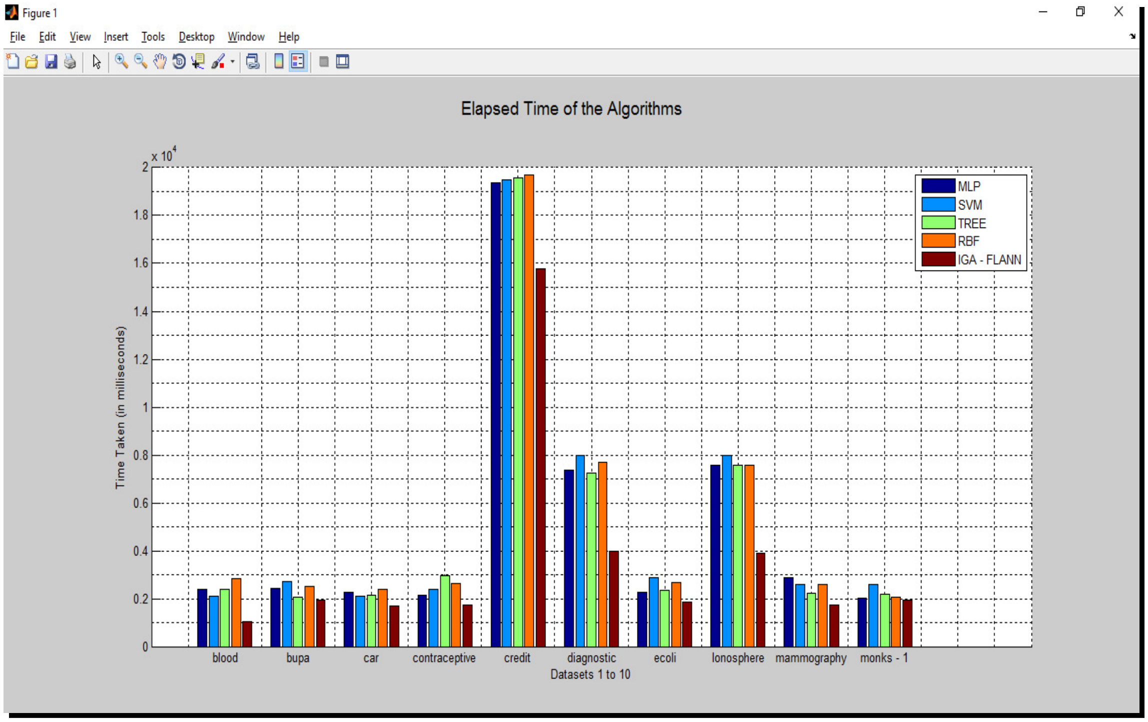

Fig.4. Time Taken for Classification by the Algorithms Comparison for the datasets 1 to 10

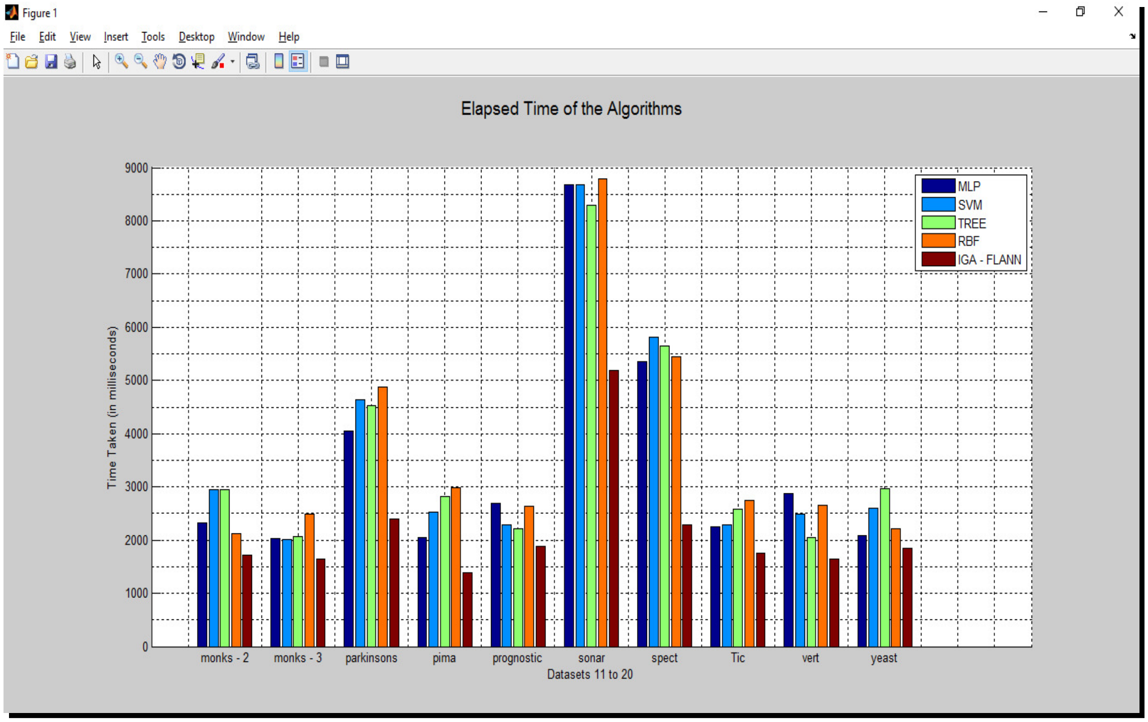

Fig.5. Time Taken for Classification by the Algorithms Comparison for the datasets 11 to 20

20 datasets are taken from the UCI machine learning repository namely blood, bupa, car, contraceptive, credit, diagnostic, ecoli, Ionosphere, mammography, monks – 1, monks – 2, monks – 3, parkinsons, pima, prognostic, sonar, spect, Tic-Tac-Toe Endgame, vert and yeast. The dataset details such as name of the dataset,

number of instances and number of features are portrayed in the Table 1. The performance are done using MATLAB tool. The system configuration is Core I3 processor with 8 GB RAM and 1 TB hard disk that runs on Microsoft Windows 8 operating system. Implementation metrics such as predictive accuracy and time taken for classification are obtained. For better visual depiction at first 10 datasets are involved in the simulation followed up with next 10 datasets. It is evident from the results that the proposed IGA-FLANN outperforms all the other algorithms in terms of prediction accuracy. Next, we compared the implementation of the proposed IGA-FLANN in terms of time taken for classification. From that results too, it is obvious that the proposed IGA-FLANN consumes less time than that of all the algorithms.

V. CONCLUSIONS

This research manuscript aims in design and development of an improved genetic algorithm based feature selection approach based five layered artificial neural network classifier. Around 20 datasets are obtained from the UCI repository. Implementations are carried out using MATLAB tool. Implementation metrics such as prediction accuracy and time taken for prediction are taken into account to conduct the Implementation evaluation of the proposed classifier. Simulation results presents that the proposed IGA-FLANN classifier outperforms the existing classifiers.

REFERENCES

[1] Y. Lertworaprachaya, Y. Yang, R. John, “Interval-valued fuzzy decision trees with optimal neighbourhood perimeter,” Applied Soft Computing, vol. 24, pp. 851-866, 2014.

[2] A. C. Bahnsen, D. Aouada, B. Ottersten, “Example-dependent cost-sensitive decision trees,” Expert Systems with Applications, vol. 42, pp. 6609-6619, 2015.

[3] F. Blanco, J. C. Ávila, G. R. Jiménez, A. Carvalho, A. O. Díaz, R. M. Bueno, “Online adaptive decision trees based on concentration inequalities,” Knowledge-Based Systems, vol. 104, pp. 179-194, 2016.

[4] Crockett, A. Latham, N. Whitton, “On predicting learning styles in conversational intelligent tutoring systems using fuzzy decision trees,” International Journal of Human-Computer Studies, vol. 97, pp. 98-115, 2017.

[5] X. Pan, Y. Luo, Y. Xu, “K-nearest neighbor based structural twin support vector machine,” Knowledge-Based Systems, vol. 88, pp. 34-44, 2015.

[6] D. Chen, Y. Tian, X. Liu, “Structural nonparallel support vector machine for pattern recognition,” Pattern Recognition, vol. 60, pp. 296-305, 2016.

[7] X. Peng, K. Rafferty, S. Ferguson, “Building support vector machines in the context of regularized least squares,” Neurocomputing, volume. 211, pp. 129-142, 2016.

[8] V. Utkin, Y. A. Zhuk, “An one-class classification support vector machine model by interval-valued training data,” Knowledge-Based Systems, vol. 120, pp. 43-56, 2017.

[9] J. Xu, Q. Wu, J. Zhang, Z. Tang, “Exploiting Universum data in AdaBoost using gradient descent,” Image and Vision Computing, vol. 32, pp. 550-557, 2014.

[10] B. Sun, S. Chen, J. Wang, H. Chen, “A robust multi-class AdaBoost algorithm for mislabeled noisy data,” Knowledge-Based Systems, vol. 102, pp. 87-102, 2016.

[11] M. Baig, M .M. Awais, E. M. El-Alfy, “AdaBoost-based artificial neural network learning,” Neurocomputing, vol. 16, pp. 22 – 41, 2017.

[12] L. D. Miller and L. K. Soh, "Cluster-Based Boosting," IEEE Transactions on Knowledge and Data Engineering, vol. 27, pp. 1491-1504, 2015.