ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

MACHINE LEARNING: A SURVEY

Rudranshu Sharma*, Ankur Singh Bist
* U.P.T.U.
U.P.T.U.

ABSTRACT

Machine learning [1], a branch of artificial intelligence, that gives computers the ability to learn without being explicitly programmed, means it gives system the ability to learn from data. There are two types of learning techniques: supervised learning and unsupervised learning [2]. This paper summarizes the recent trends of machine learning research.

KEYWORDS: Support vector Machine & Evolutionary Extreme Learning Machine

INTRODUCTION

Supervised learning

Supervised learning [2] is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object and a desired output value (also called the supervisory signal). This learning method is fast and accurate and is able to give the correct results when new data are given in input. The distance between the actual and the desired output vectors serves as an error measure that is used to correct adaptation.

Supervised learning involves two steps:

- Learning (training): Learn a model using the training data
- Testing: Test the model using unseen test data to assess the model accuracy

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}},$$

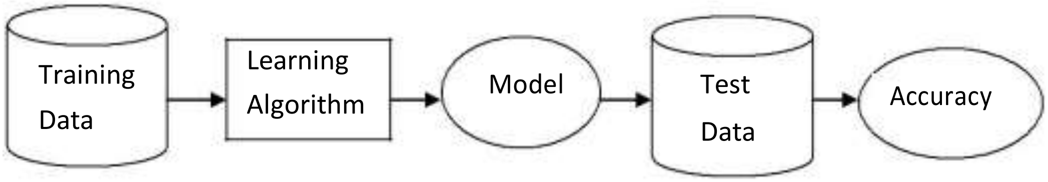

Figure 1.1: Supervised Learning

Unsupervised learning

In an unsupervised learning [2] environment, the desired response is not known. Learning is based on observations of responses to inputs. Unsupervised learning can be used to cluster the input data in classes on the basis of their statistical properties. The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes.

CLASSIFICATION

In terms of machine learning, classification is considered as illustration of supervised learning [3]. Classification [4] is a data mining technique that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. It is a frequently done brain activity in the human biological system. Practical applications of classification are ranging from medical diagnosis to business to speech recognition.

Classification is defined as, on the basis of the training set data whose category membership is acknowledged, it is a problem to recognize in which of the set of categories (sub populations) a new set of input data belongs. The algorithm which implements the . classification problem is known as classifier

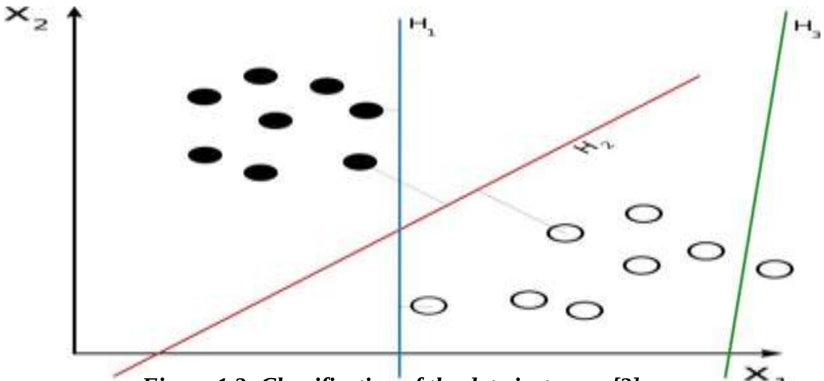

Figure 1.2: Classification of the data instances [2]

REAL VALUED DATA CLASSIFICATION

Real valued classification [5] aims at performing the classification of real world entities. The features of such type of entities have real values. An example would be assigning a given email into "spam" or "non-spam" classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.).

CLASSIFICATION PROBLEMS

There are two types of classification problems:

- Binary classification: It is the simplest type of classification problem. In binary classification, the target attribute has only two possible values.
- Multiclass classification: classify the given set of observation into more than two classes.

Well Known Classifier Available For Prediction

- Support vector Machine

Support Vector Machine (SVM) [2][6] is primarily a classification method that performs classification tasks by constructing hyper planes in a multi dimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables. To separate two classes there will number of hyper plane but the hyper plane with largest margin will give good generalization performance. Margin is minimum distance between hyper plane and instance. If the training data is linearly separable, then a pair of weight w and bias b exists such that $w^T x + b \geq 1$ for positive class and $w^T x + b \leq -1$ for negative class. Decision function can be given as

$$f(w, b) = sign(w^T x + b)$$

An optimum hyper plane dividing instances into two regions is obtained by reducing the squared norm of the weight vector of feature. The minimization can be set up as a convex quadratic programming problem:

$$\begin{array}{ll} Minimize & f = \frac{1}{2}\|w\|^2 \\ Subjectto & y_i(wx_i + b) \geq 1 \end{array}$$

Once optimal hyper plane is obtained the instances on margin are identified. These instances are called support vector. Only support vector are used while classifying new instance and other instances are ignored.

EXTREME LEARNING MACHINE

Extreme learning Machine (ELM) [7] is introduced by G. B. Huang in 2006. It is a single hidden layer feed forward network (SLFN). In ELM, the weights between input and hidden neurons and the bias for each hidden neuron are assigned randomly. The weight between output neurons and hidden neurons are analytically determined using the Moore Penrose Generalized Inverse [8][9], which makes ELM a fast learning classifier. Architecture of ELM is shown in Figure 1.3. It surmounts various traditional gradient based learning algorithms such as Back Propagation (BP) [10] and well known classifier Support Vector Machine (SVM).

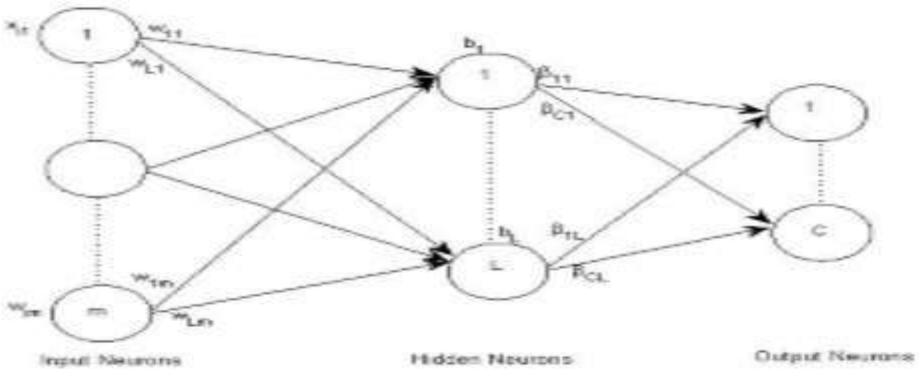

Figure 1.3: Architecture of ELM

In current scenario ELM takes much attention of researchers in both theoretic study and applications due to its performance and speed in these fields. Many variants of ELM are also present based on their methodology.

VARIANTS OF ELM

- Evolutionary Extreme Learning Machine

Evolutionary extreme learning machine (E-ELM) [11] is a learning algorithm, which takes the advantages of both Extreme Learning Machine and Differential Evolution (DE)[12]. DE is known for its ability and efficiency to locate global optimum over other EAs. Instead of tuning, E-ELM uses the fast minimum norm least-square scheme to analytically determine the output weights and for optimizing the input weights and hidden biases, a modified form of DE is used. In E-ELM, there is no need of the activation functions to be differentiable, which implied that E-ELM can be used to train SLFNs with many nonlinear hidden units such as threshold units, which are easier for hardware implementation. In comparison of ELM, E-ELM has faster learning speed and higher testing. E-ELM can obtain much more compact network architecture which would increase the response speed and be helpful in fast response (to unknown testing data) applications.

Incremental Extreme Learning Machine

Incremental based Extreme Learning Machine (I-ELM)[13] randomly chooses hidden nodes and then only need to adjust the output weights linking the hidden layer and the output layer. So here advantage presents in the terms that good performance achieved by only tuning of single parameter.

Online Sequential Extreme Learning Machine [14]

Sometimes, it is not necessary that whole data set is available at a time, so learning process can be continuous in nature. In batch learning we have to train the network with old as well as new dataset. So there is a need of developing a network which automatically adjusts with availability of newly arrived data. Online sequential extreme learning machine (OS-ELM) proposed by Liang is a fast and accurate online sequential learning algorithm for single hidden layer feed forward networks (SLFNs) with additive and radial basis function (RBF) hidden nodes. In this paper a new learning algorithm EOS-ELM has been proposed based on OS-ELM as the base learning algorithm in which final prediction is done by the simple averaging of components model. With fixed or varying chunk size, EOS-ELM can learn data one by one or chunk by-chunk. EOS-ELM uses sigmoid function and RBF function as an activation function.

Activation function in ELM

In ELM, each neuron consists of an activation function [15] which describes the output produced by a neuron to a given input. In the learning process of ELM updating of various parameter of network depends on the steepness (slope) of activation function. Performance of ELM also depends upon the selection of appropriate activation function because mapping between input node and hidden node is provided by activation function. Several functions have been proposed in the literature like Sigmoid, Multi quadratics, Radial basis function etc.

Imbalanced Data

Limitation of ELM is that it considers that data has balanced class distribution. But now a days, data with imbalanced class distribution or we can say, imbalanced data [16] is present everywhere. This work focuses on imbalance data for classification purpose.

Data set have large number of instances. These instances belong to any one class from set of classes. If some class instances are in large number and other class instances are in very less in number then data set is called imbalanced data set. In other words, imbalanced data can be thought of, as any data set that exhibits an unequal distribution between its classes. Figure 1.4 presents a data set with imbalanced data. Class with less number of instances is called minority class and class with large number of instances is called majority class. For example consider Mammography data consisting of mammography images of various patients. Patients can be cancerous (positive instance) or non cancerous (negative instance). It is clear that number of positive instance will be very less compared to number of negative instance. So this data set is imbalance by nature.

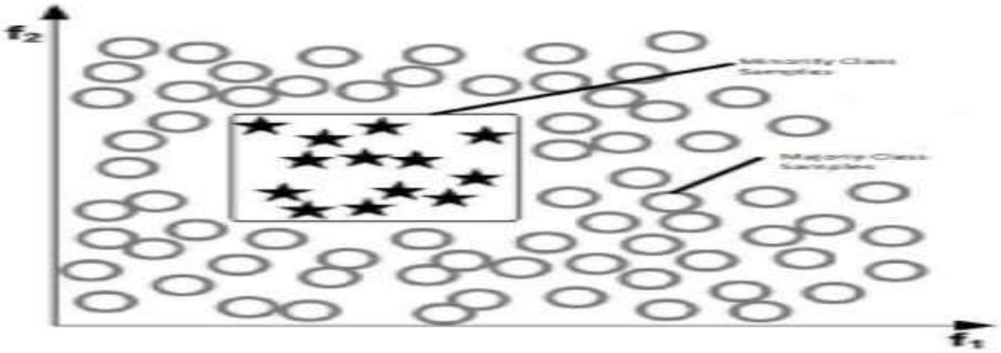

Figure1.4: A data set with imbalance class distribution

Due to recent advantages in the field of science and technology, imbalanced data is widely spread ranging from daily civilian life to national security, from enterprise information processing to governmental decision-making support systems, from micro scale data analysis to macro scale knowledge discovery. Traditionally, most standard algorithms assume balanced class distributions or equal misclassification costs between its classes. But, due to the unequal distribution between classes, these algorithms unable to properly classify the data to correct class and misclassification problem arise.

Nature of imbalance

Based on nature, imbalance can be present in any form. It may be between Class, with-In Class, intrinsic, extrinsic or relative imbalance [16]. In between class imbalance one class severely outrepresents another which means that data distribution between classes is highly unbalanced. It seems that between class imbalance is binary, but it may be multiclass also.

With-In class Imbalance: It involves only one class in which distribution of data at one place is more in comparison with other place.

Intrinsic imbalance: Intrinsic imbalance is the imbalance of nature. It analyzes classes in the binary sense, means we can take the example of

"Positive" or "Negative" for an image representative of a "cancerous" or "healthy" patient, respectively. From experience, one would expect the number of noncancerous patients to exceed greatly the number of cancerous patients. This data set contains 9876 "Negative" (majority class) samples and 260 "Positive" (minority class) samples. So, popular classifier supports majority class. Imbalance of this form is commonly referred as intrinsic imbalance.

Extrinsic Imbalance: Extrinsic Imbalance involves variable factors such as time, storage etc. We can understand it with an example like if we think of rainy data which are collected by two people. If one person collected data in rainy season and another person in non-rainy season, then automatically difference in amount is present. This is known as extrinsic imbalance.

Relative Imbalance: In relative Imbalance the ratio between classes always remains same. Consider a data set with 1,000 examples and a 10:1 between-class imbalance. We would expect this data set to contain 100 minority class examples; clearly, the majority class dominates the minority class. Now, we then double the sample space by testing more patients, and suppose further that the distribution does not change, i.e., the minority class now contains 200 examples. Clearly, the minority class is still outnumbered.

Techniques to handle imbalanced data

A number of techniques are introduced to handle imbalanced data which are as follows:

Undersampling and Oversampling

Undersampling and oversampling [16] are data distribution techniques which tries to provide re-balance between classes.

In undersampling, a fraction of the majority samples are removed, and in oversampling approach, minority samples are duplicated. Here, we can see that oversampling and undersampling, both methods appear to be functionally equivalent since both methods change the size of the original data set and provides the same proportion of balance.

Both of these methods have a problem in identifying which are informative samples and which are the redundant samples and, the assumption that the neighborhood of a minority sample share the same label is not always satisfied with different types of data.

There are some drawbacks in both of these methods. In undersampling method, a fraction of majority class examples are removed. It may be possible that these are important examples of particular interest of classifier and due to this, classifier performance degraded. In oversampling, the condition is little worse: oversampling simply adds duplicated data to the original data set which leads to the problem of overfitting [17].

Cost Sensitive Learning

Cost sensitive method [16] considers misclassification weight or cost associated with each instance of datasets for finding optimal classifier [17][18], whereas same cost for misclassification is used by previous classifiers. Cost sensitive method is a learning method related to the field of data mining. Initially, it was developed for some other purpose; but as the time spends research in imbalanced learning increases and, it was observed that cost sensitive learning and imbalanced learning has close relationship [19][20].

As cost sensitive learning gives cost to misclassification of each class instance, therefore, the more is the cost of misclassification to any particular class instances more is the importance of that class instance or we can say that, giving more importance to some classes and less to others. This concept can be directly applied to imbalanced learning. In imbalance data, high imbalance is present between numbers of instances of some classes compare to other classes. Class having less number of instances is called minority class and class having large number of instances is called majority class.

When new instances come then there are more chances to classify them into majority class. In other words classifier generated from imbalanced data is biased towards the majority class instances. By giving more weight to minority class instance this problem can be rectified. So it is clear that cost sensitive learning concept can be applied directly to imbalanced data learning, thereby making it a better method to use for imbalanced data learning.

Cost-Sensitive Learning Framework

Cost matrix shows misclassification cost for each class instance and it is the fundamental of the cost-sensitive learning methodology. The entries in the cost matrix represents penalty of classifying examples from one class to another. For example, in a binary classification scenario, we define C (Min, Maj) as the cost of misclassifying a majority class example as a minority class example and let C (Maj, Min) represents the cost of the contrary case. Typically, there is no cost for correct classification of either class and the cost of misclassifying minority examples is higher than the contrary case, i.e., C (Maj, Min) > C (Min, Maj). The main objective of cost-sensitive learning then is to develop a hypothesis that minimizes the overall cost on the training data set.

Target Class

Predicted Class

i *j*	1	2	. . .	m-1	m
1	C(1,1)	C(1,2)	. . .	C(1,m-1)	C(1,m)
2	C(2,1)	C(2,2)	. . .	C(2,m-1)	C(2,m)
.	.	.		.	.
.	.	.	C(i,j)	.	.
.	.	.		.	.
m	C(m,1)	C(m,2)	. . .	C(m,m-1)	C(m,m)

Figure 1.5: Cost Matrix

Weighted ELM

As we have seen that each technique has its own problems and consequences, So G.-B.Huang in 2013 proposed two new variants of ELM, which are Weighted ELM W1 and Weighted ELM W2.

Weighted ELM uses weighting scheme W1 and W2 for pushing the boundary towards the majority class. We can assign different misclassification cost for each instance, but for simplicity these algorithms choose a weighing scheme which is automatically generated from the class information.

These new weighing schemes are better than the previous ones because, they assign weights to binary class as well as carry all the advantages of ELM which are: 1) Simplicity in theory and ease of implementation. 2) Can be applied directly into binary classification tasks. 3) Faster Learning speed. These algorithms generate weight automatically according to the class distribution, which are inversely proportional to the number of instances in the training data. Hence, these algorithms belong to the family of cost-sensitive learning [16].

In Weighted ELM W1 and W2, weights assigned to binary class, generates according to the number of instances, which creates dependency of WELM on number of instances. While for some data sets Weighted ELM W1 gives better performance, for others Weighted ELM W2 outperforms it. So, out of these two weighing schemes none of them is a standalone solution. The need for a technique, which eliminates dependency by deciding optimal weight from a large search space so that better generalization performance can be achieved, exists. The next section discusses the available optimization techniques for parameter tuning.

Optimization Techniques Overview

Optimal weights are necessary for getting good generalization performance. So there is a need of optimization technique that can provide optimal weights from overall optimal weight search space. Optimization techniques are widely deployed in machine learning and various such techniques are available. We may express the general constrained optimization problem as follows.

$$\begin{aligned} &Minim\ ize: \quad f(x) \\ &Subject\ to: \quad \phi(x) = 0 \\ &\qquad\qquad\quad \psi(x) \le 0\,,\ x \in R^{n} \end{aligned}$$

Classical Optimization Techniques
Classical optimization techniques are useful in finding the optimum solution or unconstrained maxima or minima of continuous and differentiable functions. These are analytical methods and make use of differential calculus in locating the optimum solution. Also, these methods have limited scope in practical applications as some of them involve objective functions which are not continuous and/or differentiable.

Numerical Methods of Optimization
Linear programming: It studies the case in which the objective function is linear and equalities and inequalities are also linear.

Integer programming: It studies linear programs in which some or all variables are constrained to take on integer values.

Quadratic programming: It allows the objective function to have quadratic terms, while equalities and inequalities are linear.

Nonlinear programming: It studies the general case in which the objective function or the constraints or both contain nonlinear parts.

Stochastic programming: It studies the case in which some of the constraints depend on random variables. It includes dynamic programming.

Dynamic programming: Dynamic programming is a stochastic method which studies the case in which the optimization strategy is based on splitting the problem into smaller sub-problems.

Evolutionary Algorithm: An Evolutionary Algorithm (EA) is an iterative and stochastic process that operates on a set of individuals (population). Each individual represents a potential solution to the problem being solved. This solution is obtained by means of an encoding/decoding mechanism. Initially, the population is randomly generated (perhaps with the help of a construction heuristic). Every individual in the population is assigned, by means of a fitness function, a measure of its goodness with respect to the problem under consideration. This value is the quantitative information the algorithm uses to guide the search. Various evolutionary algorithms are as follows:

Particle Swarm Optimization: Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. The system is initialized with a population of random solutions and searches for optima by updating generations. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

Simulated Annealing: Simulated annealing is a probabilistic method proposed in Kirkpatrick, Gelett and Vecchi(1983) and Cerny(1985) for finding the global minimum of a cost function that may possess several local minima. It works by emulating the physical process whereby a solid is quickly cooled so that when eventually its structure is “frozen”, this happens at a minimum energy configuration.

Genetic Algorithm: Genetic algorithm was developed by Prof. John Holland, his colleagues and students at the University of Michigan around 1975. It is a well known global search and optimization technique, based on the survival of the fittest concept, means the individual which is best in population will survive till the end.

Advantages of Genetic Algorithm are: 1) Traditional methods start search from a single point, while genetic algorithm considers the whole population. So, the search space of genetic algorithm is wider than traditional methods, which improves the robustness of algorithm and reduce the chances of trapping in a local stationary point. 2) It operates with coded versions of the problem parameters rather than parameters themselves.3) It uses probabilistic approach in comparison with traditional methods, which uses deterministic approach.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Performance of Genetic Algorithm depends on the fitness function and genetic parameters which are selection, crossover and mutation. Genetic Algorithm starts with a large population of individuals or weights. With each generation weights are selected by Selection, and corresponding fitness value is calculated by fitness function. After determining fitness value, rank individuals by their fitness value. Individuals with high fitness can be termed as promising candidates. These promising candidates are kept and allowed to reproduce and also random changes are introduced by mutation. This process is repeated in each generation. The expectation is that the fitness of the population will increase with each generation. Therefore, with each generation more classification accuracy is achieved.

CONCLUSION

This paper gives a brief overview about machine learning techniques. There are lots of advancements going on in this specific domain. Continuous evolution in this area has added various dimensions in base atoms of concerned area. This study will be helpful for those working in the area machine learning

REFERENCES

1. Rumelhart D., Hinton G., "Williams R. Learning representations by back-propagation errors," Nature, pp. 533–536. 1986.
2. Huang G., Zhu Q., Siew C., "Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks. International Joint Conference on Neural Networks", pp. 985–990, 2004.
3. Huang G., Zhu Q., Siew C., "Extreme learning machine: Theory and applications. Neurocomputing 70," vol. 1 issue 3, pp. 489-501. 2006.
4. Huang G., Ding X., Zhou H., "Optimization method based extreme learning machine for classification. Neurocomputing" pp. 155-163, 2004.
5. Huang G., Chen L., Siew C.,"Universal approximation using incremental constructive feedforward networks with random hidden nodes," IEEE Transactions on Neural Networks, pp.879-892.2006.
6. Huang G., Zhou H., Ding X., Zhang R., "Extreme Learning Machine for Regression and Multiclass Classification", IEEE Transactions on Systems, Man, and Cybernetics, Part B, pp. 513-529.2012.
7. He H., Garcia E, "Learning from Imbalanced Data. IEEE Transactions on Knowledge And Data Engineering", pp. 1263-1284. 2009.
8. Deng W., Zheng Q., Chen L., "Regularized Extreme Learning Machine. IEEE Symposium on Computational Intelligence and Data Mining". pp. 389 - 395.2009.
9. Zong W., Huang G., Chen Y. , "Weighted extreme learning machine for imbalance learning. Neurocomputing", pp. 229-242. 2013.
10. Holland J., "Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, Michigan; re-issued by MIT Press". 1992.
11. Srinivas M., Patnaik L., "Genetic algorithms: A survey IEEE" pp. 17-26. 2009.
12. Rao C., Mitra S., "Generalized inverse of a matrix and its applications. In: Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics, Berkeley, Calif. Sixth Berkeley Symposium on Mathematical Statistics and Probability. University of California Press", pp 601-620.1972.
13. Serre D., "Matrices:Theory and Applications. Springer-Verlag, New York, Inc". 2002.
14. Fletcher R., "Practical Methods of Optimization: Constrained Optimization. Wiley, New York 2". 2002.
15. Reeves C., "Genetic Algorithms. In: Glover F, Kochenberger G (eds) Handbook of Metaheuristics, vol 57. International Series in Operations Research & Management Science. Springer US," pp 55-82. doi:10.1007/0-306-48056-5_3. 2003.
16. Mitchell M., "An introduction to genetic algorithms. MIT Press".2003.
17. Sharapov R., "Genetic Algorithms: Basic Ideas, Variants and Analysis. In: Vision systems: segmentation and pattern recognition" pp 407-422.2007.
18. Sivaraj R., Ravichandran T., "A Review of selection methods in genetic algorithm.

International Journal of Engineering Science & Technology" vol. 3 issue 5, pp. 3792-3797.2011.

19. Herrera F., Lozano M., "Heuristic crossovers for real-coded genetic algorithms based on fuzzy connectives. In: Voigt H-M, Ebeling W, Rechenberg I, Schwefel H-P (eds) Parallel Problem Solving from Nature — PPSN IV, vol 1141. Lecture Notes in Computer Science. Springer Berlin Heidelberg", pp 336-345. doi:10.1007/3-540-61723-X_998.1996.
20. Alcala-Fdez J., Fernandez A., Luengo J., Derrac J., Garcia S. , "KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithm and Experimental Analysis Framework. Multiple-Valued Logic and Soft Computing 17" vol. 2 issue3, pp.255-287.2011.