Perspectives on Intelligent System Techniques used in Data Mining

Poonam Verma
United Institute of Technology

Abstract— With the uprising trend of the social media, for marketing purposes and for personal communication, the data mining techniques used in the early decade can no longer handle the data for pattern recognition .It becomes necessity that the data mining techniques become intelligent enough to deal with the volume and variety of the data being searched for recognizing patterns or trends. Thus in this article, I have explored the various neural networks and their techniques being implemented on data to find a particular trend. Various applications of such hybrid intelligent systems have been discussed. Perceptron and it's varieties have been long used for machine learning without the human intervention. Moreover their various features are inspired by the biological strategies endowed by nature on humans to recognize, learn and innovative ideas to solve their problems. Humans have various sensory organs to help them receive their inputs from the surroundings, however it is their brain that helps them to process the large data and get the required data as an output. Neural Networks are based on the Human brain and the nervous system. So we shall explore various intelligent systems of Neural Networks to help in data mining.

Key words: Intelligent Systems, Neural Networks, Data Mining

I. INTRODUCTION

The field of the Intelligent Systems has phenomenally increased in the range of techniques and the applications. Intelligent Systems includes a range of techniques that are capable of the data processing capabilities in the real life situations. These intelligent systems techniques are inspired by the biologically strategies that are used in this universe to handle the real life situations. Major categories of the intelligent systems are the Neural Networks, Fuzzy Logics, Genetic algorithms. In most of the data rich environment, these techniques help in optimizing search criteria of different factors. Such techniques are useful in today's scenario as the data has grown exponentially. To recognize pattern from this data using the earlier techniques might not result into optimal results.

II. INTRODUCTION TO NEURAL NETWORKS

An artificial Neural Network is an information processing model that is basically inspired by the nervous systems of the most important organ of human body that is brain. It consists of a large number of highly interconnected processing system that work together to solve pattern recognition or data classification problems. Human Brain consists of neurons that collect signals from dendrites and sends it to the required neuron with the help of axon, only when the neuron is excited enough to surpass the restraint activity on it.

A. Human Vs Artificial Neuron

Generally in the human brain, Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes. Artificial Neurons can be developed by first imitating the essential features of the human Neurons and their interconnections. A computer is then programmed to simulate these features.

B. An Engineering Method of Neurons

Imitating the Human Brain, an artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be taught to fire (or not), for particular input patterns with various learning examples. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the trained, the firing rule is used to determine whether to fire or not. If the input pattern is not in the list, then combinations of the firing rules are made to decide the firing policy of the neurons, there by infusing the artificial intelligence into the artificial neurons that can take decisions without the human intervention and on the past experiences.

C. Firing Rules

The firing rule is an important concept in neural networks and has an important feature of high flexibility. A firing rule determines how one calculates whether a neuron should fire for any input pattern or not. It relates to all the input patterns, apart from patterns that the neuron nodes have been trained, which instigate the artificial intelligence in the neurons. A simple firing rule can be implemented by using Hamming distance technique. The rule goes as follows:

Take a collection of Input patterns that can be used to train a node. Out of these input patterns, some of the patterns cause the node to fire (1) and others which prevent it from doing so (0). Then the patterns not in the collection cause the node to fire if, they have more input elements in common with the 'nearest' pattern in the 1 set than with the pattern in the 0 set. If there is a tie, then the pattern remains in the undefined state. Thus in this manner the neural network is trained to fire at particular consequences than other.

D. An Example of Pattern Recognition

One of the most appropriate application of neural networks in current scenario is pattern recognition. Pattern recognition can be implemented by using a feed-forward neural network that has been trained accordingly. Feed Forward networks are one of the simplest neural networks that do not consist of loops or cycles, and the information flows only in one direction. When the network is used, it identifies the input pattern from the trained set and tries to output the associated output pattern. The power of neural networks comes to activity when an input pattern that has no output associated with it, is given as an input. In such cases, the neural network will give the answer corresponding to the trained input which has the most resemblance to the input.

E. A Complex Neuron

McCulloch and Pitts model (MCP) is one of the sophisticated neuron and is also known as linear threshold gate. It has a simple mathematical precise definition that has fixed and threshold value and gives a binary output. The difference between a simple artificial neuron and MCP is that these inputs consists of weights and these weights at each input help in the decision making, as it helps to prioritize the inputs and the effect of the input with higher weight will be given more consideration in case of the output processing. The weight of an input is a number which when multiplied with the input gives the weighted input. These weighted inputs are then added together and if they exceed a pre-set threshold value, the neuron fires. In any other case the neuron does not fire.

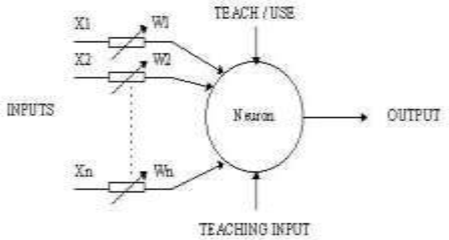

Fig. 1: An MCP neuron

In mathematical terms, the neuron fires if and only if;

$$X_1W_1 + X_2W_2 + X_3W_3 + ... > T$$

The addition of input weights and of the threshold makes this neuron a very flexible and powerful one. Although in a particular scenario the MCP has fixed weights and thresholds, but MCP neuron has the ability to adapt to a particular situation by changing its weights and/or threshold. Various algorithms exist that cause the neuron to 'adapt'; the most used ones are the Delta rule and the back error propagation. The Delta rule is used in feed-forward networks and the Back Error Propagation is used in feedback networks.

F. Architecture of Neural Networks

1) Feed – Forward Networks

Feed-forward ANNs (figure 1) allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organisation is also referred to as bottom-up or top-down.

2) Feed Back Networks

Feedback networks (figure 1) can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organisations.

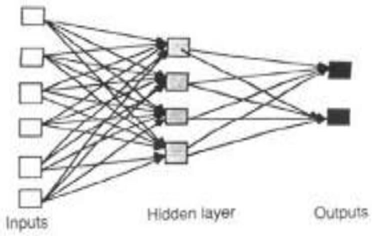

Fig 8 An example of a simple feedforward network

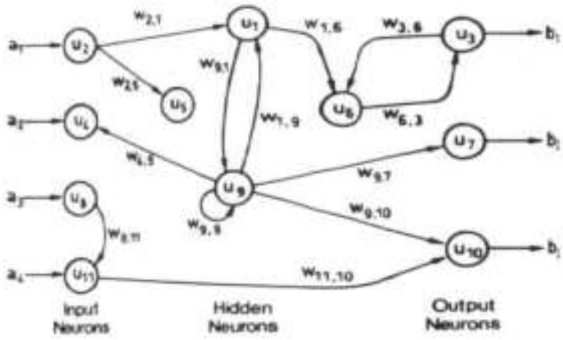

Fig. 9: An example of a complicated network

3) Neural Networks

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units. (see Figure 4.1)

- The activity of the input units represents the raw information that is fed into the network.
- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.
- The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

4) Perceptrons

The most influential work on neural nets in the 60's was proposed in the form of 'perceptrons' a term coined by Frank Rosenblatt. The perceptron (figure 4.4) turns out to be an MCP model where the input neurons consists of weights with some additional, fixed, pre-processing. Units labelled A1, A2, Aj , Ap are called association units and their task is to extract specific, localised featured from the input images. Perceptrons imitate the human perception system where the humans can recognise the patterns in given sequences with much ease. They were mainly used in pattern recognition even though their capabilities extended a lot more.

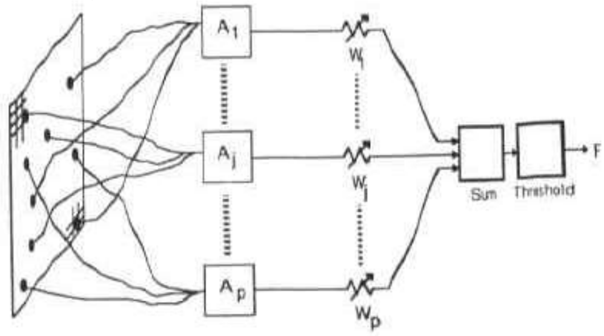

Fig. 2: Perceptrons and Hidden nodes

Given the appropriate training, multilevel perceptrons can carry out many operations making perceptrons copying the mammalian visual system.

5) Multilayer Perceptron (MLP)

Unlike MCP neurons, the multilayer perceptron (MLP) is able to learn complex non-linear relationships between the different inputs. A simple MLP is a network of perceptrons. This model consists of basically three layers where neurons exist in the input layer, hidden layer and an output layer. This form of MLP is the most simple network model that can be used in the everyday life.The processing elements are organized in a regular architecture (or topology) of three distinct groups of neurons (input, hidden and output layer) interconnected using unidirectional weights. The number of input nodes is depended to correspond with the number of dimensions in the problem. The number of neurons in the hidden layer is determined experimentally and the number of output classes in the analyzed dataset determines the number of outputs.

MLP networks are generally used in the supervised learning problem .The MLP network can be solved using back propagation algorithm, which consists of forward bias and backward bias. Each neuron in MLP performs a weighted sum of inputs and transforms it using a nonlinear activation function (e.g. sigmoid transfer function) (Haykin, 1999). During the forward pass, the MLP processes all of its input in a feedforward manner; where the input from the neurons flow only in one direction towards the output of each neuron is calculated and feeds the next layer through to the output. Let us consider a neuron h, with j^{th} input vector which consists of n elements x_{ij} ($i = 1, \dots n$), the summation function a_{jh} accumulates the sum of the products of the input signals x_{ij} with associated weights w_{ih} . It assumes a fixed weight,$_h$, which is then transformed by the activation function *f(.)* (e.g. sigmoid) to produce the single output z_{jh} , the overall computations follows that given in equation (1):

$$z_{jh} = f(a_{jh}) = \frac{1}{1+\exp(-a_{jh})} = f\left(\sum_{i=1}^{n}(w_{ih}x_{ij} - \theta_h)\right)$$

The error is then calculated by determining the difference between the actual generated output zj and the target output tj using the expression $\delta_{jh} = z_{jh} - t_{jh}$. The error term is often called delta and hence when the delta learning rule is used, the component difference expression becomes $\delta_{jh} = (z_{jh} - t_{jh})(1 - t_{jh})$. In the backward pass, the stochastic approximation procedure back-propagates this error to adjust the weight values during each presentation of the j^{th} training sample on each iteration (or epoch) τ. Various training algorithms can be used to improve the operation of BP (Hagan, et al., 1996):

6) Self-Organizing Map (SOM)

Self-Organizing Map (SOM) was chosen as the structure of neural networks for analyzing the material data set in the thesis, because it is an unsupervised learning technique suitable for datasets with no pre-defined classes. The SOM algorithm was developed by Kohonen to transform a data set of arbitrary dimensions into a one or more dimensional discrete map(Kohonen T, 1990). It consists of a two dimensional array of *m x m* discrete units and the Kohonen network associates each of the vector inputs to a representative output.

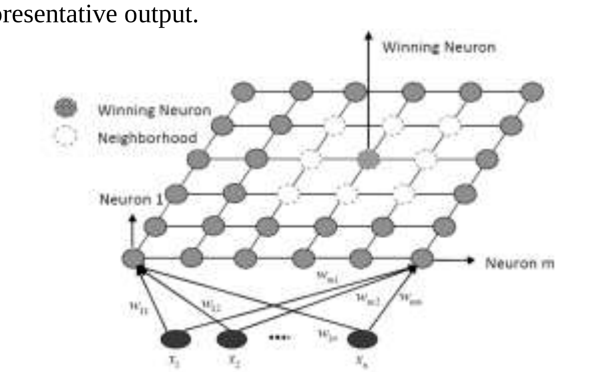

Fig 3: Schematic of a Self Organizing Map topology

Suppose wij are the components of the weight vector W , connecting the inputs i to output node j; the xi are components of the input vector X, the output of the neuron is the quadratic (Euclidean) distance dj between the weight vector and the input vector (see Figure 2-3).

- Initialize all the weights to random values between 0 and 1.
- Randomly select an input vector X; present it to all the neurons of the network and evaluate the corresponding quadratic distance output dj according to the following equation:

$$d_j = \|X - W_j\|^2 = \sum_{i=1}^{n}(w_{ij} - x_i)^2 \qquad (4)$$

where n is the number of input vector components.

- Select the neuron with the minimum output dj as the wining neuron, i.e. the nearest vector to the input vector. Let $j*$denote the index of the winner, the minimum output will be:

$$d_{j^*} = \min_{j\in[1,2,\dots,m\text{^}2]} \|X - W_j\|^2 \qquad (5)$$

where *m x m* is the number of neurons.

- Update the weight of the wining neuron according to Equation (6).

$$wij * \tau + 1 = wij * \tau + \eta\ \tau\ [xi\ \tau - wij *] \qquad (6)$$

where τ is the learning iteration count and η is the gain term.

The neighbors of the wining neuron, defined by the neighborhood function ($j*$) (the neighborhood function ($j*$) defines how many neurons in the neighborhood of the winning one will be updated for each learning input) are also updated following Equation (7) and (8):

$$wij\ \tau + 1 = wij\ \tau + \eta\ \tau\ [xi\ \tau - \ ] \qquad (7)$$

If $\in (j*)$, neighborhood of $j*$

$$wij\ \tau + 1 = wij\ \tau \qquad (8)$$

If $\notin (j*)$, neighborhood of $j*$

- Repeat the learning process until all the input vectors X have been used at least once.

7) Applications of Neural Networks

Neural networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries. Since neural networks are best at identifying patterns or trends in data, they are well suited for prediction or forecasting needs including:

- Sales Forecasting
- Industrial Process Control
- Customer Research
- Data Validation
- Risk Management
- Target Marketing
- Neural Networks In Medicine

1) Modelling and Diagnosing the Cardiovascular System
2) Electronic Noses
3) Instant Physician

G. Hybrid Neural fuzzy systems:

Hybrid Neural fuzzy systems are based on an architecture which integrates in an appropriate parallel structure a neural network and a fuzzy logic based system (Tsoukalas & Uhrig, 1996). These two parts work as one synchronized entity. Hybrid neuro-fuzzy systems have a parallel architecture, and exploit similar learning paradigms, as is the case for neural networks. The parallelism of the system can be viewed as a mapping of the fuzzy system into a neural network structure (N. K. Kasabov, 2002).

Hybrid neuro-fuzzy systems have the same architecture as that of traditional fuzzy systems except that a layer of hidden neurons performs each of the tasks of the fuzzy inference system. This type of architecture is the most commonly used among neuro-fuzzy inference systems, and it uses the Sugeno-type fuzzy inferencing (J.-S. R. Jang & Sun, 1997).

1) ANFIS

ANFIS is a multilayer feed forward network where each node performs a particular function on incoming signals. It is normally represented by a six-layer FNN as shown in Figure 2-6. To perform a desired input-output mapping, adaptive learning parameters are updated based on gradient learning rules (J. S. R. Jang, 1993; Soyguder & Alli, 2009). Both square and circle node symbols in Figure 2-6 are used to represents different properties of adaptive learning, among which the rule layer represents a set of fuzzy rules.

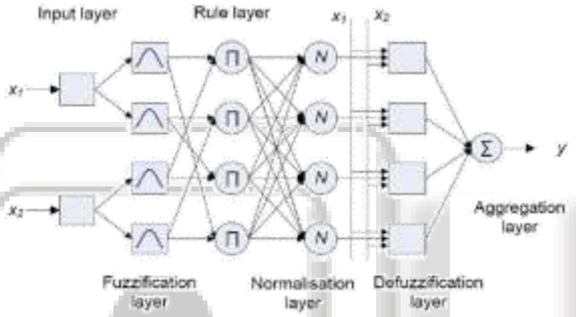

Fig 4: A six layer ANFIS structure

Specifically, ANFIS only supports Sugeno-type systems, and these must have the following properties:

Be first or zeroth order Sugeno-type systems. Have a single output, obtained using weighted average defuzzification. All output membership functions must be the same type and either be linear or constant.

Have no rule sharing. Different rules cannot share the same output membership function, namely every rule has a specific output function so the number of output membership functions must be equal to the number of rules. Have unity weight for each rule.

2) Evolving Fuzzy Neural Network (EFuNN)

The Evolving Fuzzy Neural Network (EFuNN) proposed by Kasabov (N. Kasabov, 1998, 2007; N. Kasabov, 2008) implements a strategy of dynamically growing and pruning the connectionist (i.e. NN) architecture and parameter values.

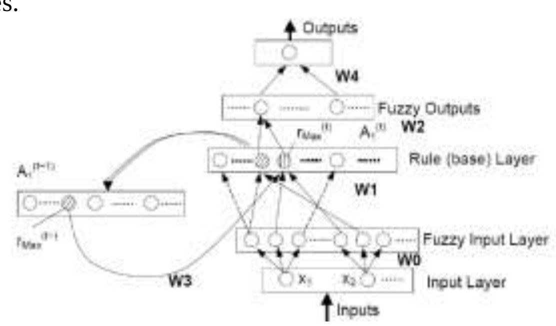

Fig. 5: Architecture of Evolving Fuzzy Neural Network (N. Kasabov, 2007)

The input layer represents input variables. The second layer of nodes (fuzzy input neurons or fuzzy inputs) represents fuzzy quantization of each input variable space. The third layer contains rule (case) nodes that evolve through supervised and/or unsupervised learning. The fourth layer of neurons represents fuzzy quantization of the output variables, similar to the input fuzzy neuron representation. Here, a weighted sum input function and a saturated linear activation function is used for the neurons to calculate the membership degrees to which the output vector associated with the presented input vector belongs to each of the output membership functions. The fifth layer represents the values of the output variables. Here a linear activation function is used to calculate the defuzzified values for the output variables (N. Kasabov, 2001).

3) Fuzzy ARTMAP

Fuzzy ARTMAP is a self-organizing architecture that is capable of rapidly learning to recognize, test hypotheses and predict the consequences of virtually any input. It involves a combination of neural and fuzzy operations that together give these useful capabilities. Fuzzy ARTMAP adapts a competitive learning model based on ART.

Fuzzy ARTMAP has proven itself as a supervised incremental learning system in pattern recognition and *M-to-N* dimensional mapping (Downs, Harrison, Kennedy, & Cross, 1996). The two fuzzy ARTs are connected using a series of weight connections between the $F2$ layers both in $ARTa$ and$A$. Those connections are weighted with a value between 0 and 1. Learning in Fuzzy ARTMAP encompasses the recruitment of new prototype vectors and expansion of the boundary of existing prototype vectors in the feature space. Like other incremental NNs, the Fuzzy ARTMAP growth criterion is subject to a similarity measure between the input pattern and the prototypes stored in the network (S.C. Tan, Rao, & Lim, 2008).

4) GNMM

GNMM was proposed by Jianhua Yang in 2009 (Yang, 2009). Utilizing GAs and MLPs, this technique is capable of pattern classification and analysis. By incorporating GA, GNMM can optimize the number of input features to the MLP automatically. It consists of three steps: (1) GA-based input feature selection, (2) Multi-Layer Perceptron (MLP) modeling and (3) rule extraction based on mathematical programming (Yang, 2009). In the first step, GAs are introduced in order to get an optimal set of MLP inputs.

Before the training process, an Independent Component Analysis (ICA) based weight initialization algorithm is used to determine optimal weights. The LM algorithm is used to achieve a second-order speedup compared to conventional BP training (Chow, 2007; 2 CNS Tech Lab, Boston University, Ham & Kostanic, 2000). In the third step, mathematical programming can be used to identify the parameters of extracted multivariate polynomial rules. In addition to that, mathematical programming can also explore features from the extracted rules based on data samples associated with each rule. Therefore, the methodology can provide regression rules and features not only in the separate multi-dimensional spaces with data instances, but also in the spaces without data instances.

III. SUMMARY

The current article has briefly reviewed some intelligent DM techniques including NNs, FL, DT and some hybrid techniques including ANFIS, EFuNN, Fuzzy ARTMAP, GNMM etc. These intelligent DM techniques can be combined to further improve and optimize the efficiency of hybrid techniques introduced in this chapter such as ANFIS, EFuNN in terms of its accuracy and the number of rules.

REFERENCES

[1] Babuska, R. (1998). Fuzzy Modelling for Control. Kluwer Academic Publishers, The Netherlands.
[2] Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm.Computers & Geosciences, 10(2-3), 191-203.
[3] Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. Neural Networks, IEEE Transactions on, 3(5), 698-713.
[4] Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks, 4(6), 759-771.
[5] Chow, T. W. S. (2007). Neural Networks and Computing: Learning Algorithms and Applications: World Scientific Publishing Co., Inc.
[6] Downs, J., Harrison, R. F., Kennedy, R. L., & Cross, S. S. (1996). Application of the fuzzy ARTMAP neural network model to medical pattern classification tasks. Artificial Intelligence in Medicine, 8(4), 403-428.
[7] Georgiopoulos, M., Huang, J., & Heileman, G. L. (1994). Properties of learning in ARTMAP. Neural Networks, 7(3), 495-506.
[8] Godin, N., Huguet, S., & Gaertner, R. (2005). Integration of the Kohonen's self-organizing map and k-means algorithm for the segmentation of the AE data collected during tensile tests on cross-ply composites. NDT&E International, 38, 299-309.
[9] Haykin, S. (1999). Neural networks: a comprehensive foundation. New York: Macmillan.
[10] Höppner, F., Klawonn, F., Kruse, R., & Runkler, T. (1999). Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition. Sussex: John Wiley & Sons Ltd.
[11] Ian, C., & Jacek, M. Z. (2000). Knowledge-based neurocomputing: MIT Press.
[12] Jang, J.-S. R., & Sun, C.-T. (1997). Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence: Prentice-Hall, Inc.
[13] Kartalopoulos, S. V. (1997). Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications: Wiley-IEEE Press.
[14] Kasabov, N. (1998). Evolving Fuzzy Neural Networks - Algorithms, Applications and Biological Motivation. Paper presented at the Methodologies for conception, design and application of soft computing.
[15] Leeson, M. M. Ramón, M. Pardo, E. Llobet, D. D. Iliescu & J. Yang (Eds.), Intelligent Systems: Techniques and Applications: Publisher: Shaker publishing.
[16] Li, X., Ramirez, C., Hines, E. L., Leeson, M. S., Purnell, P., & Pharaoh, M. (2008b). Pattern recognition of fiber-reinforced plastic failure mechanism using computational intelligence techniques. Paper presented at the Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on.
[17] Singh, H., Li, X., Hines, E. L., & Stocks, N. (2007). Classification and feature extraction strategies for multi channel multi trial BCI data. International Journal of Bioelectromagnetism, 9(4), 233–236.
[18] Soyguder, S., & Alli, H. (2009). An expert system for the humidity and temperature control in HVAC systems using ANFIS and optimization with Fuzzy Modeling Approach. Energy and Buildings, 41(8), 814-822.
[19] Tan, P.-N., Steinbach, M., & Kumar, V. (2005). Introduction to Data Mining, (First Edition): Addison-Wesley Longman Publishing Co., Inc.
[20] Tan, S. C., Rao, M. V. C., & Lim, C. P. (2008). Fuzzy ARTMAP dynamic decay adjustment: An improved fuzzy ARTMAP model with a conflict resolving facility. [doi: DOI: 10.1016/j.asoc.2007.03.006]. Applied Soft Computing, 8(1), 543-554.
[21] Tsoukalas, L. H., & Uhrig, R. E. (1996). Fuzzy and Neural Approaches in Engineering: John Wiley & Sons, Inc.
[22] Umanol, M., Okamoto, H., Hatono, I., Tamura, H., Kawachi, F., Umedzu, S., et al. (1994). Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems. Paper presented at the Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on.
[23] Yang, J. (2009). Intelligent Data Mining using Artificial Neural Networks and Genetic Algorithms: Techniques and Applications. University of Warwick.

