ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 5, July 2013

Comparative Study of Different Classification Techniques for Post Operative Patient Dataset

Satya Ranjan Dash[1], Satchidananda Dehuri[2]

School of Computer Application, KIIT University, Patia, Bhubaneswar-751024, India[1]

Department of Systems Engineering, Ajou University, San 5, Woncheon-dong, Yeongtong-gu, Suwon – 443-749, South Korea[2]

ABSTRACT: Post Operative patient dataset is a real world problem obtained from the UCI KDD archive which is used for our classification problem. In this paper different classification techniques such as Bayesian Classification, classification by Decision Tree Induction of data mining and also classification techniques related to fuzzy concepts of soft computing is used for implementation of our dataset. The parameters used to comparison of different algorithms are RMSE, ROC Area, MAE, Kappa Statistics, time taken to build the model, Relative Absolute Error, Root Relative Squared Error, and the percentage value of classifying instances.

Keywords**:** Classification and Prediction, Bayesian Classification, Decision Tree Classification, Fuzzy Classification.

I. INTRODUCTION

Simply stated, data mining refers to extracting or "mining" knowledge from large volume of data [2]. But in broadly data mining can be defined as the task of extracting implicit, non-trivial, previously unknown potential useful information or pattern from data in large databases. Data mining is "the process of using variety of data analysis tools to discover patterns and relationships in data that may be used to make valid predictions" [1]. Data mining tasks can be descriptive, (i.e., discovering interesting patterns or relationships describing the data), and predictive (i.e., predicting or classifying the behavior of the model based on available data). It is an interdisciplinary field with a general goal of predicting outcomes and uncovering relationships in data [3]-[6]. Some of the data mining techniques are Classification, Clustering and Rule Mining.

Clustering is a widely used knowledge discovery technique. It helps un-covering structures in data that were unknown in the past known. The clustering of huge data sets has attracted a lot of notice in recent years, however, clustering is a silent and difficult task since many available algorithms fail to do well in scaling with the size of the data and the amount of dimensions that illustrate the points, or in finding illogical shape of clusters, or trade efficiently with the existence of noise.

Clustering is the process of grouping objects/items into classes or clusters of similar objects. A cluster is a collection of objects that are similar to each other within the cluster and are dissimilar to objects in other clusters. Similarities and dissimilarities are assessed based on the attribute values describing the objects. Often, distance measures are used. Mathematically speaking, the norm between intra cluster objects is less than the norm between intercluster objects i.e.

$$| d_{ai}-d_{aj} |<|d_{ak}-d_{bl}|$$

Where d_{ai}, d_{aj} and d_{ak} are objects from same cluster a, d_{bl} belongs to a different cluster b and $| d_{ai}-d_{aj} |$ denote the distance between objects d_{ai} and d_{aj}. Clustering has been a folklore problem in areas like Bioinformatics [6], Data mining, pattern recognition [7], image analysis, etc. Clustering techniques used in many applications are either dominated by distance based or connectivity based. A few alike algorithms have been used in [8]. In Classification given a collection of records, containing a set of attributes, we first build a model as a function of the values of other attributes, which divides data into classes such that unseen records should be assigned to a class as accurately as possible.

II. DIFFERENT CLASSIFICATION TECHNIQUES

There are several classification techniques for classification of both multivariate and univariate dataset, but some of the basic techniques are Decision tree classifier, Bayesian classifiers, Bayesian belief networks, and Rule-

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 5, July 2013

based classifiers. Some more recent approaches to classification are Support vector machine, classification based on Association rule mining, k-Nearest-Neighbor classifiers, Case-based reasoning, and Genetic algorithms.

Bayesian classifiers [24] are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class. Bayesian classification is based on Bayes' theorem, described below. Bayesian classifiers have exhibited high accuracy and speed when applied to large databases.
Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered "naïve." Bayesian belief networks are graphical models, which unlike naïve Bayesian classifiers allow the representation of dependencies among subsets of attributes. Bayesian belief networks can also be used for classification.

Bayes' Theorem

Bayes' theorem [2] is named after Thomas Bayes (1702-1761), a nonconformist English clergyman who did early work in probability and decision theory. Let X is a data tuple. In Bayesian terms, X is considered "evidence." As usual, it is described by measurements made on a set of n attributes. Let H be some hypothesis, such as that the data tuple X belongs to a specified class C. For classification problems, we want to determine P (H|X), the probability that the hypothesis H holds given the "evidence" or observed data tuple X. In other words, we are looking for the probability that tuple X belongs to class C, given that we know the attribute description of X. P (H|X) is the posterior probability, or a posteriori probability, of H conditioned on X. Bayes' theorem is:

$$P(H|X) = \frac{P(X \mid H)P(H)}{P(X)}$$

Naïve Bayesian Classification

The naïve Bayesian classifier, or simple Bayesian classifier, works as follows:

1. Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, X = (x_1, x_2… xn), depicting n measurements made on the tuple from n attributes, respectively, A1, A2…A_n.

2. Suppose that there are m classes, C_1, C_2… C_m. Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the naïve Bayesian classifier predicts that tuple X belongs to the class Ci if and only if P (Ci|X) > P (C_j| X) for $1 \leq j \leq m$, $j \neq I$ Thus we maximize P (Ci|X). The class Ci for which P (C_i|X) is maximized is called the maximum posteriori hypothesis. By Bayes' theorem P (C_i|X) = P (X|C_i) P (C_i)/P (X).

3. As P(X) is constant for all classes, only P (X|C_i) P (C_i) need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, P (C1) = P (C2) =… = P (Cm) and we would therefore maximize P (X|C_i). Otherwise, we maximize P (X| C_i) P (C_i).

4. Given data sets with many attributes, it would be extremely computationally expensive to compute P (X| C_i). In order to reduce computation in evaluating P (X| Ci), the naive assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the tuple (i.e., that there are no dependence relationships among the attributes). Thus,

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k|Ci)$$
$$= P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i).$$

Precaution for Zero Probabilities

There is a simple trick to avoid this problem. We can assume that our training data-base, D, is so large that adding one to each count that we need would only make a negligible difference in the estimated probability value, yet would conveniently avoid the case of probability values of zero. This technique for probability estimation is known as

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 5, July 2013

the Laplacian correction or Laplace estimator, named after Pierre Laplace, a French mathematician who lived from 1749 to 1827. If we have, say, q counts to which we each add one, then we must remember to add q to the corresponding denominator used in the probability calculation.

III.FUZZY CLASSIFICATION

The ability to accommodate ambiguity in the training and test data necessitates a fuzzy approach for managing the classifier training and test data. The fuzzy k-NN classifier assigns a membership value to the unlabeled signature that provides the system with information suitable for estimating the confidence of the decision. The fuzzy membership describes what fraction of an unlabeled signature resides in each of the defined classes. If the membership is relatively high for two classes and low for three others, then there is a clear delineation between the first two classes and the other three, but there is confusion within the first two classes. This data becomes important when ultimately assigning a crisp label to the signature. For example, the classifier might assign a signature membership 0.8 to C1, 0.75 to C2, 0.2 to C3, and 0.01 to C4. For this situation the signature would likely belong to C1or C2, but the ambiguity between the two would be high making a crisp assignment difficult. One of the benefits of using a fuzzy system is that an object can be assigned to the category unknown, which in certain situations may provide a much greater advantage over crisply assigning the example to the wrong category.

Fuzzy-Rough Classification

The induction of gradual decision rules, based on fuzzy-rough hybridization, is given in [31]. For this approach, new definitions of fuzzy lower and upper approximations are constructed that avoid the use of fuzzy logical connectives altogether. Decision rules are induced from lower and upper approximations defined for positive and negative relationships between credibility of premises and conclusions. Only the ordinal properties of fuzzy membership degrees are used. More recently, a fuzzy-rough approach to fuzzy rule induction was presented in [39], where fuzzy reducts are employed to generate rules from data. This method also employs a fuzzy-rough feature selection preprocessing step.

Also of interest is the use of fuzzy-rough concepts in building fuzzy decision trees. Initial research is presented in [29] where a method for fuzzy decision tree construction is given that employees the fuzzy-rough ownership function. This is used to define both an index of fuzzy-roughness and a measure of fuzzy-rough entropy as a node splitting criterion. Traditionally, fuzzy entropy (or its extension) has been used for this purpose. In [33], a fuzzy decision tree algorithm is proposed, based on fuzzy ID3, which incorporates the fuzzy-rough dependency function as a splitting criterion. A fuzzy-rough rule induction method is proposed in [32] for generating certain and possible rule sets from hierarchical data.

Fuzzy Nearest Neighbour Classification

The fuzzy K-nearest neighbour(FNN) algorithm[34] was introduced to classify test objects based on their similarity to a given number K of neighbours(among the training objects), and these neighbours 'membership degrees to(crisp or fuzzy class labels. For the purpose of FNN, the extent C(y) to which an unclassified object y belongs to a class C is computed as:

$$C(y) = \sum_{x \in N} R(x, y)C(y)$$

Where N is the set of object y's K nearest neighbours, and R(x, y) is the [0, 1]-valued similarity of x and y.

The fuzzy KNN algorithm FNN (U, C, y, K)

U, the training data; C, the set of decision classes;
y, the object to be classified; K, the number of nearest neighbour;

(1) N $\leftarrow$ getNearestNeighbours(y, K);
(2) $\forall$ C $\in$ C
(3) $C(y) = \sum_{x \in N} R(x, y)C(x)$
(4) output $\underset{C \in C}{\arg\max}$ (C(y))

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 5, July 2013

(5)

Initial attempts to combine the FNN algorithm with concepts from fuzzy rough set theory were presented in [37, 38].

Fuzzy-Rough nearest Neighbour (FRNN) Algorithm

The algorithm given below, combining fuzzy-rough approximations with the ideas of the classical FNN approach, in what follows, FRNN-FRS and FRNN-VQRS denote instances of the algorithm where traditional, and VQRS, approximations are used, respectively. The rationale behind the algorithm is that the lower and the upper approximation of a decision class, calculated by means of the nearest neighbours of a test object y, provide good clues to predict the membership of the test to that class.

In particular, if (R↓C)(y) is high, it reflects that all (most) of y's neighbours belong to C, while a high value of (R↑C)(y) means that at least one (some) neighbour(s) belong(s) to that class, depending on whether the FRE or VQRS approximations are used. A classification will always be determined for y due to the initialization of τ to zero in line (2) of Fig. 2.To perform crisp classification, the algorithm outputs the decision class with the resulting best combined fuzzy lower and upper approximation membership, in the fuzzy lower and upper approximations to determine class membership. The complexity of the algorithm is O (|C|. (2|U|)).

The FRNN algorithm for classification FRNN (U, C, y)

U, the training data; C, the set of decision classes;
y, the object to be classified.

(1) N $\leftarrow$ getNearestNeighbours(y, K)
(2) $\tau \leftarrow$ 0, Class $\leftarrow \phi$
(3) $\forall$ C $\in$ C
(4) if (((R↓C)(y) + (R↑C)(y)) / 2 $\geq \tau$)
(5) Class $\leftarrow$ C
(6) $\tau \leftarrow$ (R↓C)(y) + (R↑C)(y)) / 2
(7) output Class

When using FRNN-FRS, the use of K is not required in principle: as R(x, y) gets smaller, x tends to have only have a minor influence on (R↓C)(y) and (R↑C)(y). For FRNN-VQRS, this may generally not be true, because R(x, y) appears in the numerators as well as the denominator. In which Ra(x, y) is the degree to which objects x and y are similar for attribute a.

IV. RESULTS AND DISCUSSION

DATA SET INFORMATION

The classification task of this database is to determine where patients in a postoperative recovery area should be sent to next. Because hypothermia is a significant concern after surgery, the attributes correspond roughly to body temperature measurements.

Attribute Information:

1. L-CORE (patient's internal temperature in C), high (> 37), mid (>= 36 and <= 37),low (< 36)
2. L-SURF (patient's surface temperature in C), high (> 36.5), mid (>= 36.5 and <= 35), low (< 35)
3. L-O2 (oxygen saturation in %), excellent (>= 98), good (>= 90 and < 98), fair (>= 80 and < 90), poor (< 80)
4. L-BP (last measurement of blood pressure), high (> 130/90), mid (<= 130/90 and >= 90/70), low (< 90/70)
5. SURF-STBL (stability of patient's surface temperature): stable, mod-stable, unstable
6. CORE-STBL (stability of patient's core temperature): stable, mod-stable, unstable
7. BP-STBL (stability of patient's blood pressure): stable, mod-stable, unstable
8. COMFORT (patient's perceived comfort at discharge, measured as an integer between 0 and 20)
9. Decision ADM-DECS (discharge decision):

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 5, July 2013

a. I (patient sent to Intensive Care Unit)
b. S (patient prepared to go home)
c. A (patient sent to general hospital floor)

From all the attributes of the dataset mentioned above, all attributes are well-defined but the attribute Comfort isn't clearly defined. The attribute Comfort is not depended on all other attributes and also the calculation of its value is not given. So, we can remove that attribute from the implementation. The implementation in WEKA [42] will be done by taking 5 different algorithms. To analyze all the algorithms following parameters are used:

1. Kappa statistic

The Kappa Statistic can be defined as measuring degree of agreement between two sets of categorized data (reliability of the data collected and validity) [50]. Kappa result varies between 0 to 1 intervals. Higher the value of Kappa means stronger the agreement/ bonding. If Kappa = 1, then there is perfect agreement. If Kappa = 0, then there is no agreement. If values of Kappa statics are varying in the range of 0.40 to 0.59 considered as moderate, 0.60 to 0.79 considered as substantial, and above 0.80 considered as outstanding [41].

2. MAE (Mean Absolute Error)

Mean absolute error can be defined as sum of absolute errors divided by number of predictions. It is measure set of predicted value to actual value i.e. how close a predicted model to actual model.

3. ROC Area

ROC area provides comparison between predicted and actual target values in a classification. It describes the performance of a model with complete range of classification thresholds or in other words, it has been used for model comparisons. ROC area varies between 0 to 1 intervals [40]. By default classification threshold for binary classification is .5. When the probability of a prediction is 50% or more, the model predicts that class. Changes in the classification threshold affects the predictions made by the model; if the threshold for predicting the positive class is changed from 0.4 to 0.7 then fewer positive predictions will be made. This will affect the distribution of values in the confusion matrix.

4. RMSE (Root Mean Square Error)

Root mean square error is defined as square root of sum of squares error divided number of predictions. It is measure the differences between values predicted by a model and the values actually observed. Small value of RMSE means better accuracy of model. So, minimum of RMSE & MAE better is prediction and accuracy.

First we classify the dataset using two test options i.e.: using training method and 10 cross fold method. The test values of those parameters are given in the Table 1.

Classifier	Use Training Set Method				Use 10 Cross Fold Method			
	Kappa Statistics	MAE	ROC Area	RMSE	Kappa Statistics	MAE	ROC Area	RMSE
Naïve Bayes	0.1074	0.2677	0.643	0.3641	-0.0413	0.2982	0.395	0.4018
ID3	0.7024	0.0859	0.965	0.2073	-0.1481	0.3209	0.432	0.5433
J48	0	0.2818	0.5	0.3754	-0.021	0.2842	0.441	0.3789
FuzzyRoughNN	0.7024	0.3029	0.892	0.3656	-0.06	0.3565	0.433	0.4292
FuzzyNN	-0.0413	0.2074	0.485	0.4554	-0.06	0.3565	0.433	0.4292

Table 1: Use Training Set Method

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 5, July 2013

From the Table 1 statistics, it is clear that the "Use Training Set Method" has better performance than "10 Cross Fold Method".

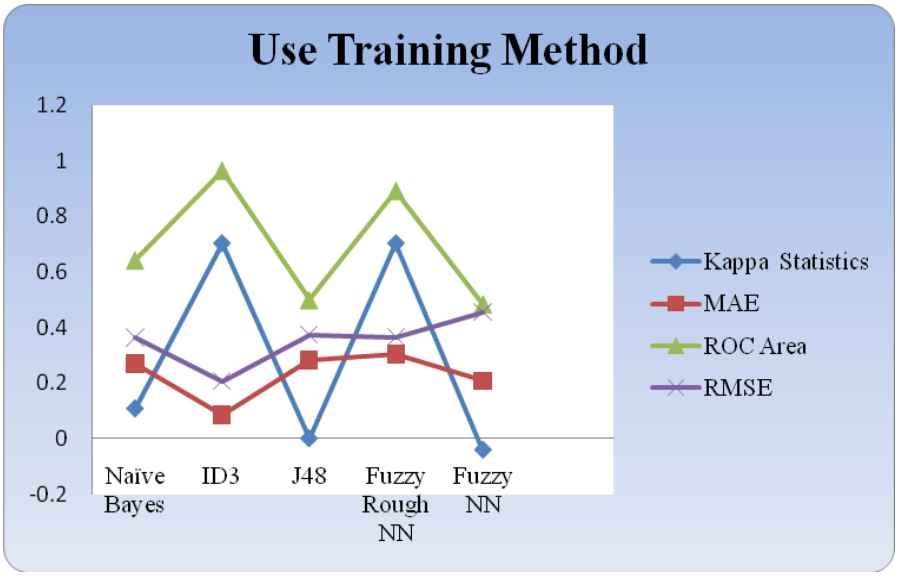

Fig 1: Graph showing comparison of algorithms using "Use Training Method"

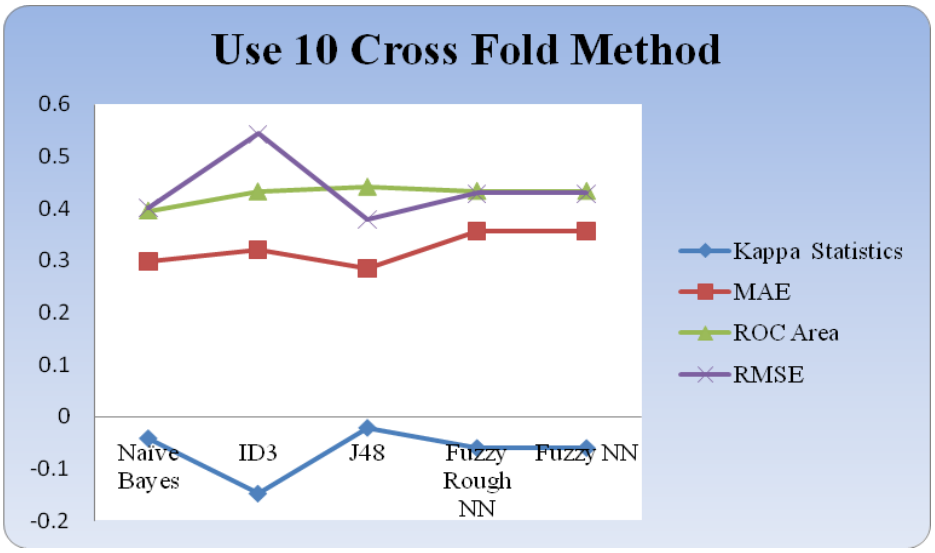

Fig 2: Graph showing comparison of algorithms by using "10 Cross Fold Method"

Figure 1 and 2 show that the comparison of classifiers with the help of two test options. These figures also state that "Use Training Set Method" has better performance than "10 Cross Fold Method" under these 3 observations:

- Firstly, the value of RMSE and MAE .Lower these values, better the prediction. So, "Use Training Method "has minimum value of RMSE and MAE as compare to "10 Cross Fold Method" (except the value of RMSE of FuzzyNN).
- Secondly, the value of Kappa statistics. Comparing above discussed method; "Use Training Set Method" has better value than "10 Cross Fold Method" The values of Kappa statistics are all negative values in "10 Cross Fold Method", so obviously it is not better.
- Third, The value of ROC .Higher the value of ROC means higher positive predictions that can affect confuse matrix that provides the value of sensitivity, specification etc .From above table we can clearly see that the

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 5, July 2013

value of ROC area of all algorithm by using "Use Training Set Method" higher than using "Use 10 Cross Fold Method".

So, from above analysis we can now only consider "Use Training Set Method" to compare all algorithms. Kappa statistics of ID3 and FuzzyRoughNN are same i.e. 0.7024. ROC area's values are higher than other 3 algorithms. Also MAE and RMSE values are comparatively smaller, so we can take these two algorithms for further analysis.

Next we take only "Use Training Method" for comparing classifiers but with other new parameters like time taken to build the model, Relative Absolute Error, Root Relative Squared Error, and the percentage value of classifying instances. We take those parameters for next test and found the values as given in the Table 2.

Classifier	Correctly Classified Instances (%)	Incorrectly Classified Instances (%)	Time Taken (seconds)	Relative Absolute Error (%)	Root Relative Squared Error (%)
Naïve Bayes	73.3333	26.6667	0	93.2633	96.9547
ID3	88.8889	11.1111	0.02	29.9334	55.2018
J48	71.1111	28.8889	0	98.1724	99.97
FuzzyRoughNN	88.8889	11.1111	0	105.5255	97.3575
FuzzyNN	68.8889	31.1111	0	72.2531	121.2881

Table 2: Comparison of algorithms using only "Use Training Method"

From the above Table 2, we can see that the followings:

- The correctly classified instances of ID3 and FuzzyRoughNN is 88.8889 i.e. same and also the value of incorrectly classified instances.
- Time taken to build the model of ID3 is 0.02 but of FuzzyRoughNN is 0.
- But the Root Absolute Error and Root Relative Squared Error value of both algorithms are different but lower the value of ID3.

From the above experiments it is clearly seen that ID3 and FuzzyRoughNN are the best algorithm for classifying post operative patient dataset. But comparing these two algorithms the value of the Kappa statistics is correctly classified instances are same for both. The value of FuzzyRoughNN for time taken is only less than ID3.But all other error values of ID3 are less than FuzzyRoughNN. So, it is clearly told that the ID3 algorithm is better than all other algorithms for classifying post operative patient data set.

V. CONCLUSION AND FUTURE WORK

In the first phase we took two test option like "Use training set" and "Use 10 cross fold" and take some parameters like RMSE, ROC Area, MAE, Kappa Statistics to test that which test option is better. In this phase we implement those 5 algorithm and the results are recorded in a table. Then it is clearly observable that "Use training set" test options is more better than "Use 10 cross fold ".After analysis of all parameters it is clearly seen that ID3 and FuzzyRoughNN are better than other algorithm.

In the second phase we took some other parameters like time taken to build the model, Relative Absolute Error, Root Relative Squared Error, and the percentage value of classifying instances .Again we noted the value of all parameters of all algorithm. After all observation it is it is finally said that FuzzyRoughNN and ID3 are best algorithm for classification and prediction of Post operative patient dataset and ID3 algorithm is the best of twos.

The default settings of the various techniques in WEKA were used to perform the experiments. This may have affected the performance of the techniques. An exploration of different parameters could be made to assess the impact on the results obtained from this project. This project has identified the best technique that will offer the most efficient performance when applied to classify Post Operative dataset in the medical organization. In theory, this has proven to yield better results or returns. It is also recommended that this is applied in the field and monitored to validate these results.

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 5, July 2013

REFERENCES

1. Two Crows Corporation, “Introduction to Data mining and Knowledge discovery” 3rd ed., 2006..
2. J. Han and M. Kamber, “Data Mining: Concepts and Techniques”, 2nd ed., pp.1-10, 2006.
3. S. Mitra, S. K. Pal, and P. Mitra,"Data mining in soft computing framework: A survey", IEEE Transactions on Neural Networks, vol.13, pp.3-14, 2002.
4. D. Hand, H. Mannila, and P. Smyth, “Principles of Data Mining”, London: MIT Press, 2001.
5. M. Kantardzic, “Data Mining: Concepts, Models, Methods, and Algorithms”, Hoboken, N J: Wiley Inter science, IEEE Press, 2002.
6. R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases", International Conference on Management of Data (Washington, D.C.), pp. 207-216, May 1993.
7. R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases”, 20th International Conference on Very Large Databases, pp.478-499, September 1994.
8. L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing”, Communications of the ACM, vol.37, pp.77- 84, 1994.
9. S. K. Pal and S. Mitra,” Neuro-fuzzy Pattern Recognition: Methods in Soft Computing”, New York: John Wiley & Sons, 1999.
10. S. Mitra, S. K. Pal, and P. Mitra, "Data mining in soft computing framework: A survey”, IEEE Transactions on Neural Networks, vol.13, pp.3-14, 2002.
11. L. A. Zadeh, "Fuzzy sets", Information and Control, vol.8, pp.338-353, 1965.
12. L .A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning”, Part1, 2, and3, "Information Sciences, vol.8, 9, pp.199-249, 301-357, 43-80, 1975.
13. L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility", Fuzzy Sets and Systems, vol.1, pp.3-28, 1978.
14. S. Haykin, “Neural Networks: A Comprehensive Foundation”, New York: Macmillan College Publishing Co., 1994.
15. J. Hertz, A. Krogh, and R. G. Palmer,” Introduction to the Theory of Neural Computation”, MA: Addison-Wesley, 1994.
16. D .E. Rumelhart and J. L. McClelland, eds.,”Parallel Distributed Processing: Explorations in the Microstructures of Cognition”, vol.1, Cambridge, MA: MIT Press, 1986.
17. T. Kohonen, “Self-Organization and Associative Memory”, Berlin: Springer-Verlag, 1989.
18. J. M. Zurada, “Introduction to Artificial Neural Systems”, New York: West Publishing Company, 1992.
19. D. R. Hush and B. G. Home, "Progress in supervised neural networks”, IEEE Signal Processing Magazine, pp.8-39, January 1993.
20. D. O. Hebb, “The Organization of Behaviour”, New York: John Wiley & Sons, 1949.
21. D. E. Goldberg,” Genetic Algorithms in Search, Optimization and Machine Learning”, MA: Addison-Wesley, 1989.
22. Z. Michalewicz, “Genetic Algorithms + Data Structures = Evolutionary Programs”, Berlin: Springer-Veriag, 1994.
23. P. Langley, W. Iba, K. Thompson, “An analysis of Bayesian classifiers, in: Proceedings of the tenth National Conference on artificial intelligence”, pp.223-228, 1992.
24. C. Apt´ e and S. Weiss. “Data mining with decision trees and decision rules”, Future Gener. Comput. Syst., 13(2-3):197–210, 1997.
25. I. H. Witten, “Text mining”, Computer Science, University of Waikato, Hamilton, New Zealand, 2004.
26. E.Frank, I.H. Witten, “Data Mining: Practical Machine Learning Tools and Techniques”, Morgan Kaufman Publishers, 2005.
27. Y.K. Jain, Upendra,”An efficient Intrusion Detection based on decision tree classifier using feature reduction”, International Journal of Scientific and Research Publications, vol.2, Issue 1, January 2012.
28. R.B. Bhatt and M. Gopal, “FRID: Fuzzy-Rough Interactive Dichotomizers”, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'04), pp. 1337-1342, 2004.
29. C. Cornelis, M. De Cock, A.M. Radzikowska, “Fuzzy Rough Sets: from Theory into Practice, Handbook of Granular Computing “(W. Pedrycz, A. Skowron, V. Kreinovich, eds.), Wiley, 2008.
30. S. Greco, M. Inuiguchi, and R. Slowinski, “Fuzzy rough sets and multiple-premise gradual decision rules”, International Journal of Approximate Reasoning, vol. 41, pp. 179-211, 2005.
31. T.P. Hong, Y.L. Liou, and S.L. Wang, “Learning with Hierarchical Quantitative Attributes by Fuzzy Rough Sets”, Proceedings of the Joint Conference on Information Sciences, Advances in Intelligent Systems Research, 2006.
32. R. Jensen and Q. Shen, “Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches”, Wiley-IEEE Press, 2008.
33. J.M. Keller, M.R. Gray and J.A. Givens, “A fuzzy K-nearest neighbor algorithm”, IEEE Trans. Systems Man Cybernet., vol. 15, no. 4, pp. 580-585, 1985.
34. Z. Pawlak, “Rough Sets: Theoretical Aspects of Reasoning about Data”, Kluwer Academic Publishing, 1991.
35. A.M. Radzikowska and E.E. Kerre, “A comparative study of fuzzy rough sets, Fuzzy Sets and Systems”, vol. 126, no. 2, pp. 137-155, 2002.
36. M. Sarkar, “Fuzzy-Rough nearest neighbor’s algorithm, Fuzzy Sets and Systems”, vol. 158, pp. 2123-2152, 2007.
37. X. Wang, J. Yang, X. Teng and N. Peng, ”Fuzzy-Rough Set Based Nearest Neighbor Clustering Classification Algorithm," Lecture Notes in Computer Science, vol. 3613/2005, pp. 370-373, 2005.
38. X. Wang, E.C.C. Tsang, S. Zhao, D. Chen and D.S. Yeung, “Learning fuzzy rules from fuzzy samples based on rough set technique", Information Sciences, vol. 177, no. 20, pp. 4493-4514, 2007.
39. T. Fawcett, “An introduction to ROC analysis”, Pattern Recognit Lett, vol.No.27:861–874, 2006.
40. J.R. Landis and G.G. Koch, “The measurement of observer agreement for categorical data”, Biometrics 33: pp.159–174, 1977.
41. WEKA at http://www.cs.waikato.ac.nz/~ml/weka.