SLEAS: Supervised Learning using Entropy as Attribute Selection Measure

Kishor Kumar Reddy C [#1], Vijaya Babu B [*2], Rupa C H [#3]

[# 1]Department of CSE, Stanley College of Engg. & Tech. for Women, Hyderabad, India
[# 3]Department of CSE, V R Siddhartha Engineering College, Vijayawada, India
[*2]Department of CSE, K L University, Guntur, India
[1] kishoar23@gmail.com
[3] rupamtech@gmail.com
[2] vijaymtech28@gmail.com

Abstract—There is embryonic importance in scaling up the broadly used decision tree learning algorithms to huge datasets. Even though abundant diverse methodologies have been proposed, a fast tree growing algorithm without substantial decrease in accuracy and substantial increase in space complexity is essential to a greater extent. This paper aims at improving the performance of the SLIQ (Supervised Learning in Quest) decision tree algorithm for classification in data mining. In the present research, we adopted entropy as attribute selection measure, which overcomes the problems facing with Gini Index. Classification accuracy of the proposed supervised learning using entropy as attribute selection measure (SLEAS) algorithm is compared with the existing SLIQ algorithm using twelve datasets taken from UCI Machine Learning Repository, and the results yields that the SLEAS outperforms when compared with SLIQ decision tree. Further, error rate is also computed and the results clearly show that the SLEAS algorithm is giving less error rate when compared with SLIQ decision tree.

Keyword-Classification, Data Mining, Decision Tree, Entropy, Gini Index, SLIQ, SLEAS.

I. INTRODUCTION

The decision tree is a broadly used tool for classification in various realistic domains such as text mining, bio-informatics, speech, web intelligence, and many other fields that need to handle huge datasets [1] [2] [3] [4] [21] [22]. The leading advantage of the decision trees is its interpretability i.e., the constructed decision tree can be represented in terms of classification rules. The branching decision tree at each node is determined by the values of a certain attribute or combination of attributes and the choice of attributes are based on a certain splitting criterion that is consistent with the objective of the classification process. Each leaf node of the tree represents a class and is interpreted by the path from the root node to the leaf node in terms of a rule such as: “If A1 and A2 and A3, then class C1,” where A1, A2, and A3 are the clauses involving the attributes and C1 is the class label. Thus, each class can be described by a set of rules [2].

Various methods for classification have been proposed such as decision tree induction and non-decision tree induction among which CART is one of the popular methods of building decision trees [2] [13] [17] [19] [20]. But it generally does not do the best job of classifying a new set of records because of over fitting, as this method is based on binary splitting of the attributes and uses Gini Index as a splitting measure in selecting the best attribute. ID3 is a simple decision tree learning algorithm introduced in 1986 by Quinlan Ross [1] [2]. The basic idea of ID3 algorithm is to construct the decision tree by employing a top-down, greedy search through the given sets to test each attribute at every tree node. This algorithm uses information gain measure to choose the splitting attribute and accepts only categorical attributes in building a tree model. C4.5 [17] is an improvement of ID3 algorithm developed by Quinlan Ross in 1993 and it accepts both categorical and numerical attributes in building the decision tree. It uses gain ratio impurity method to evaluate the splitting attribute.

The SLIQ decision tree algorithm was proposed by Manish Mehtha at IBM Almaden Research Centre and many effective and efficient enhancements were indulged when compared with the previous algorithms [5] [6] [7] [8] [9] [15] [16] . The focus of the SLIQ decision tree algorithm is how to select the most appropriate attribute at each node of the decision tree. For this decision tree, Gini Index is used as an attribute selection measure and the attribute with the largest gini index is chosen as the best split attribute. The splitting attribute selection measure Gini Index used in SLIQ tends to favour attributes with a large number of distinct values. In order to overcome the marginal inaccurate measure using the Gini Index a new measure called entropy is adopted in the present research.

An attribute selection measure is a heuristic for selecting the splitting criterion that best separates a given data partition, T of class-labelled training tuples into individual classes. If we were to split T into smaller partitions according to the outcomes of the splitting criterion, ideally each partition would be pure. The attribute selection measure provides a ranking for each attribute describing the given training tuples. The attribute having the best score for the measure is chosen as the splitting attribute for the given tuples. If the splitting attribute is continuous-valued or if we are restricted to binary trees then, respectively, either a split point or a splitting subset must also be determined as part of the splitting criterion. The tree node created for partition *T* is labelled with the splitting criterion, branches are grown for each outcome of the criterion, and the tuples are partitioned accordingly. This paper proposes new methodology SLEAS, in which instead of Gini Index, entropy is used while selecting the best attributes to form a decision tree [2] [11] [12].

For the experimentation, we collected 12 datasets from UCI Machine Learning Repository varying with number of instances and class labels. Further we splitted each dataset to training set and testing set. For training purpose we have chosen 75% of the dataset and the rest for testing purpose. Next, we computed the accuracy levels for both SLIQ and SLEAS decision trees. We also computed error rate for SLIQ and SLEAS decision trees and the results show that the proposed methodology is outperforming for most of the datasets.

The rest of the paper is organised as follows: Section II gives the related work about decision trees. Section III introduces the proposed SLEAS algorithm. Section IV provides the results for SLEAS and SLIQ for UCI machine Learning Repository datasets [10]. Finally, conclusions are provided in section V, followed by Acknowledgements, References and Author Profile.

II. RELEVANT WORK

Currently, in order to generate decision trees three kinds of algorithms have been developed: Algorithms for quantitative data, qualitative data and mixed data.

A. Algorithms for qualitative data

ID3 and k-dimensional algorithms are the most known algorithms in this category; both the algorithms use the dataset while constructing the decision tree [1] [18]. The major divergence between these two algorithms is the split rule used to generate the nodes. ID3 use the information gain, and k-dimensional [14] use confusion induce by a feature. Other method that only works with qualitative data is a fuzzy decision tree algorithm [7] that is only an extension of ID3. This algorithm uses membership's functions to represent the values of a feature. All the above mentioned three algorithms follow top-down approach.

B. Algorithms for quantitative data

C4.5 and CART are the most known algorithms in this category [8] [13]. Both of them apply a pruning process at the end of the decision tree construction, and their split rule is generated by using information gain as an attribute selection measure. The processes of these techniques are known as gain proportion, and Gini diversity index, respectively [11]. Other significative difference between C4.5 and CART is that the second only generates binary decision trees. Other algorithms found in this family are FACT and QUEST [14]. FACT is the predecessor of QUEST, and their main difference is the number of branches created for each node, FACT form as many branches as number of classes have, and QUEST generate binary trees. These algorithms realize its process of split in two steps, at each node, an analysis of variance F-statistic is calculated for each feature. The feature with largest F-statistic is selected, and a linear discriminant analysis is applied to find the split point selection.

C. Algorithms for mixed data

Support vector Machine is the most known algorithm of this family [2]. This tool is used to transform the quantitative data in synthetic Boolean features, changing the initial space representation of the features. Besides, it uses the ID3 algorithm to process the qualitative features, in this way the total set of features never is manipulated for only one method. LMDT is another algorithm that works with mixed data; it builds a decision tree in the well known top-down manner [14]. The LMDT algorithm trains a linear machine, which then serves as a multivariate test for the decision trees, this indicated the split rule that it uses to generate the internal nodes. In order to construct that linear machine test, each instance must be represented as a vector consisting of a constant threshold value of ones, and numerically encoded features that describe the objects of data set.

III.SLEAS: PROPOSED METHOD

The SLIQ algorithm uses the gini index as a split measure to generate a binary decision tree. With the help of gini index, it is decided that which attribute is to be split and the splitting point of the attribute. The gini index is minimized at each split, so that the tree becomes less diverse as we progress. One of the main draw backs of SLIQ is regarding the attribute selection measure used. The splitting attribute selection measure gini index used in SLIQ tends to favour attributes with a large number of distinct values. This drawback was overcome to a greater extent by introducing a new measure called entropy and the algorithm is as follows:

Algorithm:

1. Read the training dataset T
2. Sort T in ascending order and choose the initial attribute along with the associated class label.
3. Evaluate the split points and the procedure will be as follows:
    a. Initially check for change in the class label.

b. If there is a change in the class label, evaluate the split points and the midpoint of changed class labels is the split point. For instance, Let V be the initial record and V_i be the second record: such that take Mid Point (V, V_i) only when there is change in the class label, shown in formula (1).

$$Split\,Po\mathrm{int} = Midpo\mathrm{int}\,(V, V_i) \tag{1}$$

4. Choose the split point 1 and apply entropy attribute selection measure and evaluate the entropy value and continue this for all the split points obtained for initial attribute and the procedure is as follows:
   a. Initially, consider attribute and also along with its associated class label and evaluate attribute entropy and it is shown in formula (2) [2].

$$Attribute\,Entropy = \sum_{j=1}^{N} P_j \left[-\sum_{i=1}^{M} P_i \log_2 P_i \right] \tag{2}$$

Where P_i is the probability of class entropy belonging to class i. Logarithm is base 2 because entropy is a measure of the expected encoding length measured in bits.

b. Further, consider class label and evaluate class entropy and is as follows:

Class entropy is a measure in the information theory, which characterizes the impurity of an arbitrary collection of examples. If the target attribute takes on M different values, then the class entropy relative to this M-wise classification is defined in formula (3) [2].

$$Class\,Entropy = -\sum_{i=1}^{M} P_i \log_2 P_i \tag{3}$$

Where P_i is the probability of class entropy belonging to class i. Logarithm is base 2 because entropy is a measure of the expected encoding length measured in bits.

5. Now, compute the entropy: it is the difference of class entropy and attribute entropy and is shown in formula (4) [2].

$$Entropy = Class\,Entropy - Attribute\,Entropy \tag{4}$$

6. Once the entropy values are evaluated for all split points, choose the maximum entropy value as the best split point and continue this for the remaining attributes also.
7. Finally, if the number of attributes are N, we will get N best split points for individual attributes. As decision tree is a binary tree, there will be only one root node and for this reason, among the N entropy values choose one best entropy value to form the root node and it will be as follows:

Consider, all the attribute best split points along with entropy values. Choose, the maximum entropy value is the best entropy value. Now, consider the maximum entropy value attribute as the root node and take its split point and divide the tree in binary format i.e. keep the values which are lesser to split point at the left side of the tree and keep the values which are greater and equals to the right side of the tree, and continue the process till it ends with a unique class label.

IV. Results and Discussion

This section presents a detailed performance evaluation of SLEAS decision tree. We conducted experiments by implementing our proposed algorithm in Java Net Beans IDE 7.2. All experiments were performed on intel i3 core processor and 4 GB RAM with windows 7 operating system. We also divided our data set in to two parts: training set (75%), which is used to create the model, and a test set (25%), which is used to verify that the model is accurate and not over fitted. In order to reveal the performance of our proposed SLEAS algorithm, we presented comparison between SLIQ and SLEAS decision trees in terms of classification accuracy, using twelve datasets, taken from the UCI machine learning repository [10]. The detailed description of the datasets is shown in Table 1.

TABLE 1
Datasets Description

Dataset	Instances	Training	Testing	Class Labels
Letter	11250	8438	2812	26
Australia	540	405	135	02
Breast Cancer	500	375	125	02
Diabetes	400	300	100	02
Segment	2310	1733	577	07
Shuttle	43500	32625	10875	07
Vehicle	696	522	174	04
Waveform	5000	3750	1250	03
Satimage	6435	4826	1609	06
Glass	214	160	54	06
Iris	110	82	18	03
Abalone	4177	3133	1044	03

A model is said to be more efficient, if it yields maximum performance. In the present research, we computed the performance measure accuracy for the SLIQ and our proposed SLEAS decision tree algorithms. Accuracy is the ratio of correct predictions and total prediction. Apparently, almost all accuracy results for SLEAS are better than those of SLIQ, and the increase of accuracy on Breast Cancer, Diabetes, Shuttle, Vehicle, Waveform, Satimage, Glass and Abalone are very obvious, shown in bold font in Table 2. The graphical comparison of classification accuracy between SLIQ and SLEAS is shown in figure 1. The proposed method is out performing in most of the cases and decreasing their performance levels for a fewer datasets like Letter, Australia and Segment.

TABLE 2
Accuracy comparison of SLEAS over SLIQ using UCI datasets

Dataset	SLIQ	SLEAS
Letter	84.60	84.42
Australia	77.77	77.03
Breast Cancer	94.40	**96.00**
Diabetes	67.00	**70.00**
Segment	81.62	78.33
Shuttle	99.97	**99.98**
Vehicle	63.21	**65.51**
Waveform	74.88	**75.76**
Satimage	83.53	**83.65**
Glass	18.51	**22.22**
Iris	96.42	**96.42**
Abalone	48.18	**50.19**

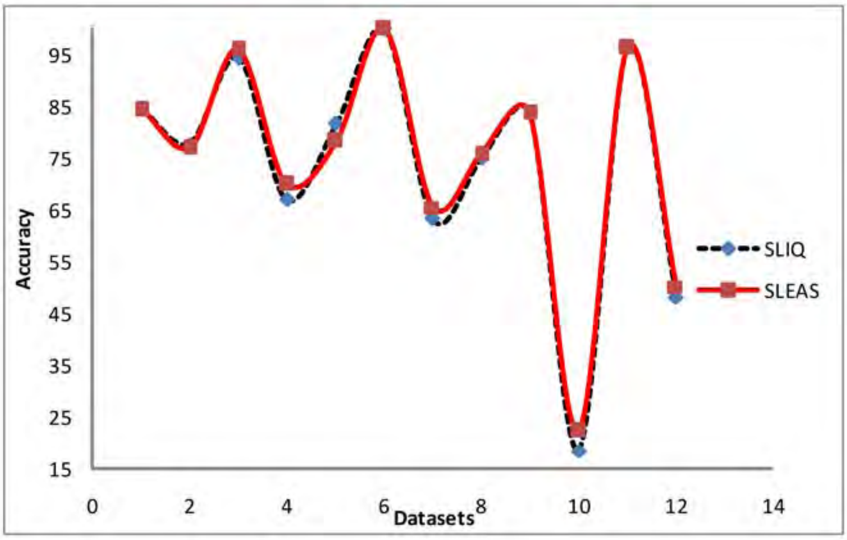


Fig: 1. Comparison of classification accuracy between SLIQ and SLEAS

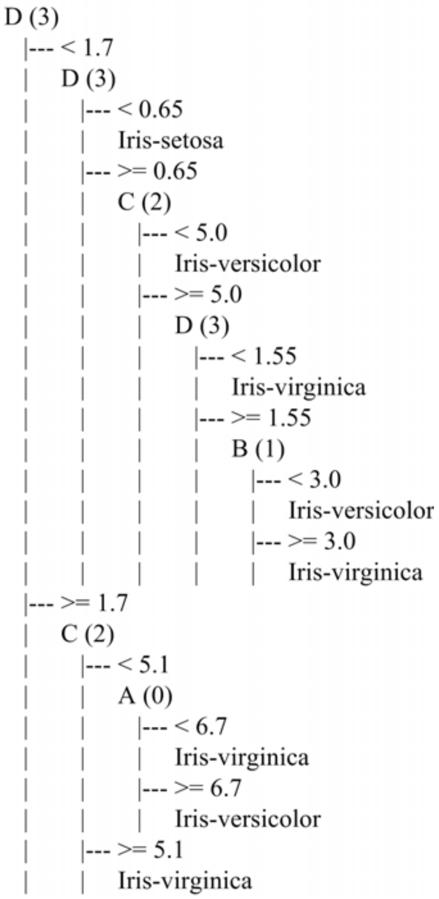
Further, we also evaluated the error rate for the proposed model. Error rate is the ratio of incorrect predictions and total predictions. A model is said to be good if it yields a very low error rate, and the proposed model yields low error rate for most of the datasets except for Letter, Australia, and Segment, shown in Table 3.

TABLE 3
Error rate comparison of SLEAS over SLIQ using UCI datasets

Dataset	SLIQ	SLEAS
Letter	15.40	15.58
Australia	22.23	22.97
Breast Cancer	5.60	**4.00**
Diabetes	33.00	**30.00**
Segment	18.38	21.67
Shuttle	0.03	**0.02**
Vehicle	36.79	**34.49**
Waveform	25.12	**24.24**
Satimage	16.47	**16.35**
Glass	81.49	**77.78**
Iris	3.58	**3.58**
Abalone	51.82	**49.81**

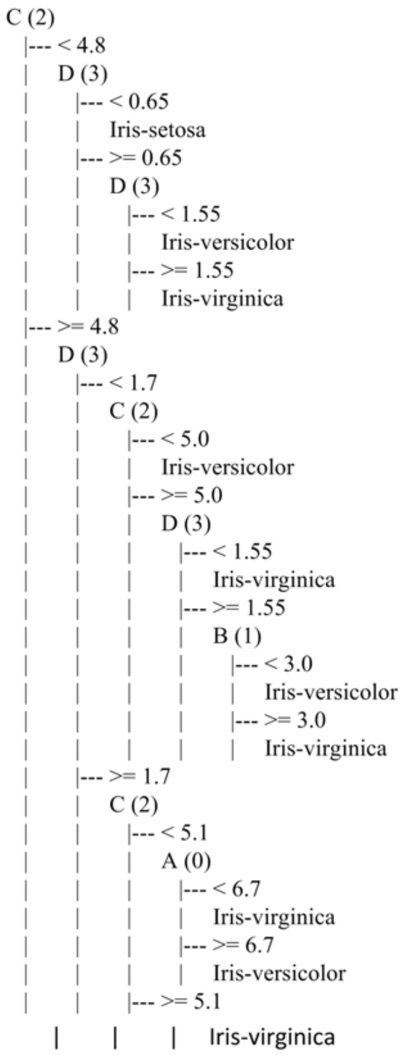
Further, we also constructed decision trees for the datasets presented in Table 1 and for a sample we presented decision tree for iris dataset using SLIQ decision tree in Figure 2 and our proposed methodology SLEAS in figure 3. The dataset comprises of 110 instances, out of which it is splitted 82 instances as training dataset and 18 instances as testing instances.

The main advantage with decision trees is: it generates classification rules. Further, we also constructed classification rules for the datasets presented in Table 1 and for a sample we presented rules for iris dataset using SLIQ decision tree and our proposed methodology SLEAS. The iris dataset comprises of 110 instances, out of which it is splitted 82 instances as training dataset and 18 instances as testing instances. With the help of rules the constructed decision tree can be analysed easily.



```
D (3)
  |--- < 1.7
  |    D (3)
  |       |--- < 0.65
  |       |    Iris-setosa
  |       |--- >= 0.65
  |       |    C (2)
  |       |       |--- < 5.0
  |       |       |    Iris-versicolor
  |       |       |--- >= 5.0
  |       |       |    D (3)
  |       |       |       |--- < 1.55
  |       |       |       |    Iris-virginica
  |       |       |       |--- >= 1.55
  |       |       |       |    B (1)
  |       |       |       |       |--- < 3.0
  |       |       |       |       |    Iris-versicolor
  |       |       |       |       |--- >= 3.0
  |       |       |       |       |    Iris-virginica
  |--- >= 1.7
  |    C (2)
  |       |--- < 5.1
  |       |    A (0)
  |       |       |--- < 6.7
  |       |       |    Iris-virginica
  |       |       |--- >= 6.7
  |       |       |    Iris-versicolor
  |       |--- >= 5.1
  |       |    Iris-virginica
```

Fig: 2. Decision Tree using SLIQ for Iris Dataset



```
C (2)
 |--- < 4.8
 |    D (3)
 |      |--- < 0.65
 |      |    Iris-setosa
 |      |--- >= 0.65
 |      |    D (3)
 |      |      |--- < 1.55
 |      |      |    Iris-versicolor
 |      |      |--- >= 1.55
 |      |      |    Iris-virginica
 |--- >= 4.8
 |    D (3)
 |      |--- < 1.7
 |      |    C (2)
 |      |      |--- < 5.0
 |      |      |    Iris-versicolor
 |      |      |--- >= 5.0
 |      |      |    D (3)
 |      |      |      |--- < 1.55
 |      |      |      |    Iris-virginica
 |      |      |      |--- >= 1.55
 |      |      |      |    B (1)
 |      |      |      |      |--- < 3.0
 |      |      |      |      |    Iris-versicolor
 |      |      |      |      |--- >= 3.0
 |      |      |      |      |    Iris-virginica
 |      |--- >= 1.7
 |      |    C (2)
 |      |      |--- < 5.1
 |      |      |    A (0)
 |      |      |      |--- < 6.7
 |      |      |      |    Iris-virginica
 |      |      |      |--- >= 6.7
 |      |      |      |    Iris-versicolor
 |      |      |--- >= 5.1
 |      |      |    Iris-virginica
```

Fig: 3. Decision Tree using SLEAS for Iris Dataset

Classification Rues for Iris Dataset using SLIQ Decision Tree

[1] If [(D < 1.7) and (D < 0.65)] Then (prediction = Iris-setosa)
[2] If [(D < 1.7) and (D >= 0.65) and (C < 5.0)] Then (prediction = Iris-versicolor)
[3] If [(D < 1.7) and (D >= 0.65) and (C >= 5.0) and (D < 1.55)] Then (prediction = Iris-virginica)
[4] If [(D < 1.7) and (D >= 0.65) and (C >= 5.0) and (D >= 1.55) and (B < 3.0)] Then (prediction = Iris-versicolor)
[5] If [(D < 1.7) and (D >= 0.65) and (C >= 5.0) and (D >= 1.55) and (B >= 3.0)] Then (prediction = Iris-virginica)
[6] If [(D >= 1.7) and (C < 5.1) and (A < 6.7)] Then (prediction = Iris-virginica)
[7] If [(D >= 1.7) and (C < 5.1) and (A >= 6.7)] Then (prediction = Iris-versicolor)
[8] If [(D >= 1.7) and (C >= 5.1)] Then (prediction = Iris-virginica)

Classification Rues for Iris Dataset using SLEAS Decision Tree

[1] If [(C < 4.8) and (D < 0.65)] Then (prediction = Iris-setosa)
[2] If [(C < 4.8) and (D >= 0.65) and (D < 1.55)] Then (prediction = Iris-versicolor)
[3] If [(C < 4.8) and (D >= 0.65) and (D >= 1.55)] Then (prediction = Iris-virginica)
[4] If [(C >= 4.8) and (D < 1.7) and (C < 5.0)] Then (prediction = Iris-versicolor)
[5] If [(C >= 4.8) and (D < 1.7) and (C >= 5.0) and (D < 1.55)] Then (prediction = Iris-virginica)
[6] If [(C >= 4.8) and (D < 1.7) and (C >= 5.0) and (D >= 1.55) and (B < 3.0)] Then (prediction = Iris-versicolor)
[7] If [(C >= 4.8) and (D < 1.7) and (C >= 5.0) and (D >= 1.55) and (B >= 3.0)] Then (prediction = Iris-virginica)
[8] If [(C >= 4.8) and (D >= 1.7) and (C < 5.1) and (A < 6.7)] Then (prediction = Iris-virginica)
[9] If [(C >= 4.8) and (D >= 1.7) and (C < 5.1) and (A >= 6.7)] Then (prediction = Iris-versicolor)
[10] If [(C >= 4.8) and (D >= 1.7) and (C >= 5.1)] Then (prediction = Iris-virginica)

V. CONCLUSION

Classification is an important problem in data mining. Although classification has been studied extensively in the past, the various techniques proposed for classification do not scale well for large datasets. In this paper, we presented a fast and scalable algorithm named: SLEAS, designed specifically for scalability and also to increase the performance levels. An experimental performance evaluation shows that, compared to SLIQ, SLEAS achieves comparable and better classification accuracy. Further, we also evaluated the error rate for both SLIQ and SLEAS decision trees. In the results, we clearly demonstrated that SLEAS achieves good scalability and performs better for datasets comprising of large number of instances, attributes, and classes.

ACKNOWLEDGEMENT

The authors would like to thank the referees for their valuable comments and suggestions, which greatly enhanced the clarity of this paper. We are indebted to L V Narasimha Prasad for his guidance during the experimentation and paper set up. We will also like to thank the www.wunderground.com for providing us the precipitation historical datasets. Finally, we are thankful to the Management and Department of Computer Science and Engineering of Stanley College of Engineering & Technology for Women, Hyderabad, V R Siddhartha Engineering College, Vijayawada and K L University, Guntur for providing their maximum support during the experimentation.

REFERENCES

[1] J.R. Quinlan: Induction of Decision Trees. Journal of Machine Learning (1986) 81-106.

[2] J. Han: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers. (2001)

[3] Anuja Priyama, Abhijeeta, Rahul Guptaa, Anju Ratheeb and Saurabh Srivastavab: Comparative Analysis of Decision Tree Classification Algorithms, International Journal of Current Engineering and Technology (2013) 334-337.

[4] Masud Karim and Rashedur M. Rahman: Decision Tree and Naive Bayes Algorithm for Classification and Generation of Actionable Knowledge for Direct Marketing. Journal of Software Engineering and Applications (2013) 196-206.

[5] Manish Mehta, Rakesh Agarwal and Jorma Rissanen: SLIQ: A Fast Scalable Classifier for Data Mining. International Conference on Extending Database Technology (1996) 18-32.

[6] Rodrigo Coelho Barros, Marcio Porto Basgalupp, Andre C.P.L.F. De Carvalho and Alex A. Freitas: A Survey of Evolutionary Algorithms for Decision Tree Induction, IEEE Transactions on Systems, Man and Cybernetics (2012) 291-312.

[7] B. Chandra and P. Paul Varghese: Fuzzy Sliq Decision Tree Algorithm: IEEE Transactions on Systems, Man and Cybernetics (2008) 1294-1301.

[8] J.R. Quinlan: C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, (1993).

[9] S. Safavian and D. Landgrebe: A Survey of Decision Tree Classifier Methodology, IEEE Transactions on Systems, Man And Cybernetics (1991) 660 -674.

[10] UCI Repository, ftp://ftp.ics.uci.edu/pub/machine-learning-databases.

[11] W.Y. Loh and Y.S. Shih: Split selection methods for classification trees. Statistica Sinica (1997) 815-840.

[12] W.Y.Loh and N.Vanichsetakul: Tree-structured classification via generalized discriminant analysis. Journal of the American Statistical Association (1988).

[13] R. Olshen, L. Breiman, J. Friedman: Classification and Regression Trees. Wadsworth International Group (1984).

[14] Anilu Franco Arcega, Guillermo Sanchez Diaz and Jose Ruiz Shulcloper: ADT: A Decision Tree Algorithm Based on Concepts. International Symposium on Robotics and Automation (2006) 1 -6.

[15] Jiang Su and Harry Zhang: A Fast Decision Tree Learning Algorithm. American Association for Artificial Intelligence (2006).

[16] B. Chandra, Sati Mazumdar, Vincent Arena and N. Parimi: Elegant Decision Tree Algorithm for Classification in Data Mining. IEEE International Conference on Web Information Systems Engineering (2002) 160-169.

[17] Manpreet Singh, Sonam Sharma and Avinash Kaur: Performance Analysis of Decision Trees. International Journal of Computer Applications (2013) 10-14.

[18] Shikha Chourasia: Survey paper on improved methods of ID3 decision tree classification. International Journal of Scientific and Research Publications (2013) 1-4.

[19] A.L.C. Bazzan: Cooperative induction of decision trees. IEEE Symposium on Intelligent Agent (2013) 62-69.

[20] U. Johansson, H. Bostrom and T. Lofstrom: Conformal Prediction Using Decision Trees. IEEE International Conference on Data Mining (2013) 330-339.

[21] Fahim Irfan Alam, Fateha Khanam Bappee, Md. Reza Rabbani and Md. Mohaiminul Islam: An Optimized Formulation of Decision Tree Classifier. Springer Advances in Computing, Communication, and Control Communications in Computer and Information Science (2013) 105-118.

[22] S. B. Kotsiantis: Decision trees: a recent overview. Springer Artificial Intelligence Review (2013) 261-283.

AUTHORS PROFILE

C Kishor Kumar Reddy obtained his B.Tech in Information Technology from JNTU Anantapur in 2011, M.Tech in Computer Science and Engineering from JNTU Hyderabad in 2013 and currently pursuing Ph. D in Computer Science Engineering from K L University, Guntur. Presently, he is working as Assistant Professor in CSE department of Stanley College of Engineering & Technology for Women, Hyderabad. He has been awarded with Interscience Scholastic Award in the International Conference on Computer Science and Information Technology in 2012. His research area are Data Mining, Image Processing and Remote Sensing. He is the member of ISTE, IAENG, IEEE, UACEE, IACSIT.

Dr Ch. Rupa, obtained her B.Tech in Computer Science & Information Technology from JNTU, Hyderabad in 2002, M.Tech in Information Technology from Andhra University in 2005 and Ph.D in Computer Science Engineering from Andhra University. She has published 33 papers in National and International Conferences and Journals. She received the Best Paper Award in the International Conference on Systemics, Cybernetics and Informatics, Pentagram Research Center, Gov. of A. P in 2009. She was awarded Young Engineers Award of 2010-2011 by JNTU, Kakinada and National Young Engineers Award for 2011-2012 by IEI, Kolkata. She was awarded Young Engineers Award of 2012- 2013 by Govt. of A. P and IEI. She is the life member of IEI, CSI, ISTE, IAENG, ICCSIT.

Dr B.Vijaya Babu is presently working as Professor in CSE department of K L University, Vaddeswaram, Guntur, Andhra Pradesh. He obtained his B.Tech (ECE) from College of Engineering, JNTU Kakinada in the year 1993, M.Tech (CSE) from College of Engineering, JNTU Anantapur in the year 2004. He obtained his PhD (CSSE) from Andhra University, Visakhapatnam in 2012. He has teaching experience of about 20 years, in various private engineering colleges of Andhra Pradesh, in various positions. His research areas are Knowledge Engineering/Data Mining and published about 25 research papers in various International/Scopus indexed journals. He is the life member of ISTE.