ISSN:0975-9646

Karnakar Gulla et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (4) , 2012,4585 - 4589

Robust Techniques based Hybrid Models in Classification Experiments

Karnakar Gulla[1], Ashok kumar Aithagani[2] , Satyanarayana Vollala[3],Srinivas Pilli[4]

[1,2]Dept. of CSE, *RVR INSTITUTE OF Engg .and Technology*,,Hyderabad
[3]Dept. of CSE, *Talla Padmavathi College of Engg. Warangal(AP)*
[4]Dept. of CSE, *Vidya Bharathi Institute of Technology, Pembarthy*

***Abstract* : A classification problem for objects in a particular domain is the problem of separating these objects into smaller classes, and giving criteria for determining whether a particular object in the domain is in a particular class or not. Many important problems in Machine Learning comes under classification problems. Optical Character recognition, Speech Recognition, Medical Diagnosis etc. are some of the important classification problems. The feed forward neural networks(FFNN) using back propagation learning algorithm are widely employed in classification experiments. The learning algorithm is supervised in the sense that the inputs and target outputs are available for training the network. There are two primary ways in which the performance of back propagation neural networks (BPNN) can be improved. They are feature reduction in input space and weight initialization of FFNN. Two hybrid models are implemented in this work to illustrate the effectiveness of the feature reduction and weight initialization. The hybrid models are built using rough sets and BPNN. The training time required for FFNN is directly proportional to the size of the input layer. A reduction in the size of the input layer will greatly simplify the number of weights to be trained and hence simplify the training time. Identification and removal of redundant attributes at input layer will reduce the complexity in FFNN. In the first hybrid model, rough sets are being employed for reducing the dimensionality and to study the impact on the performance of FFNN. This has been demonstrated by considering standard data sets. In the second model concepts from Rough Fuzzy Neural Computing are used to fix the number of nodes in the hidden layer and also to initialize the weights of the FFNN. The performance gain in terms of number of epochs and the percentage of classification is illustrated by experimenting with standard datasets.**

Keywords: FFNN, BPNN, DT, ANN, Fuzzy NN, Rough Fuzzy NN, Rough Fuzzy Modified NN.

1. INTRODUCTION

The main basic function executed by human brain is the classification. The human can analyze objects using some characteristics to perceive the differences and similarities. So, is possible classify animals as friendly or dangerous, healthful plant for be eat or not, etc. In all moments of your life, the human impose classification among two or more object. A classifier is a function that takes the descriptor of a pattern and decides on the patterns membership in a given set of classes. In recognition problem, the decision is considered to be correct if it is same as the one that would be made by a human being. Classification has two distinct meanings. We may be given a set of observations with the aim of establishing the existence of classes or clusters in the data. Its is known as Unsupervised Learning(or Clustering). Or we may know for certain that there are so many classes, and the aim is to establish a rule whereby we can classify a new observation into one of the existing classes. This is known as Supervised Learning. There are many issues of concern for the would-be classifier. Some of them are .

Accuracy: There is the reliability of the rule, usually represented by the proportion of correct classifications, although it may be that some errors are more serious than others, and it may be important to control the error rate for some key class.

Speed: In some circumstances, the speed of the classifier is a major issue. A classifier that is 90% accurate may be preferred over one that is ,95% accurate if it is 100 times faster in testing.

Time to Learn: Especially in a rapidly changing environment, it may be necessary to learn a classification rule quickly, or make adjustments to an existing rule in real time. “Quickly” might imply also that we need only a small number of observations to establish our rule.

2. METHODS FOR CLASSIFICATION

- **Rough Sets**
- **Neural Networks**

Rough sets and neural networks are two technologies frequently applied to classification problems. The common advantage of the two approaches is that they do not need any additional rates and robustness to noise. But neural networks have two obvious shortcomings when applied to classification problems . The first is that neural networks require long time to train the huge amount of data of large datasets. Secondly, neural networks lack explanation facilities for their knowledge. The knowledge of neural networks is buried in their structures and weights. It is often difficult to extract rules from a trained neural network.Both rough sets theory and neural networks show advantages in dealing with various imprecise and incomplete knowledge. However, they are quite different. Neural networks often have complex structures when dimensions of input data are high while rough sets have a large advantage on decreasing redundancy among the input data. Rough sets have a weak tolerance and generalization performance whereas neural networks have a better capability on performance.The combination of rough sets and neural networks is very natural for their complementary features. One typical approach is to use rough set approach as a preprocessing tool for the neural networks. By eliminating the redundant data from dataset, rough set methods can greatly accelerate the network training time and improve its classification accuracy.

Feature selection, as a preprocessing step , has been effective in reducing dimensionality, removing irrelevant data, increasing learning accuracy, and improving com prehensibility. However, the recent increase of dimensionality of data poses a severe challenge to many existing feature selection methods with respect to efficiency and effectiveness. Reduction of pattern dimensionality and feature selection is one of the main applications of rough set theory. According to the theory, are the minimal subsets of attributes that can keep the discernibility between objects. Reduction of high dimensional samples will decrease the difficulty in building a classifying model and improve its performance. It is worth remarking that there are not only a single reduct at most time. Namely as to a high dimensional sample set we can get several minimal subsets of attributions that can preserve the discerning power between the objects as all of the attributions do.

2.1 Existing System

Feature selection has been a fertile field of research and development since 1970s and shown very effective in removing irrelevant and redundant features, increasing efficiency in learning tasks, improving learning performance like predictive accuracy, and enhancing comprehensibility of learned results [1]. In recent years, data has become increasingly larger in both rows (i.e., number of instances) and columns (i.e., number of features) in many applications such as genome projects, text categorization, image retrieval, and customer relationship management. Rough set theory provides a good basis for neural computing. This paradigm has three main threads namely production of training set description, calculus of granules, and interval analysis. This paradigm gains its inspiration from the work of Pawlak on Rough set philosophy as a basis for machine learning and from pattern recognition by Swiniararski and others in the early 1990s [14]. The first thread in rough-neural computing has a strong presence in current neural computing research. The second thread in rough-neural computing has two main components namely, information granule construction in distributed system of agents and local parameterized spaces. The third thread in neural computing systems from the introduction of rough set approach to interval analysis by Banejee, Lingras, Mitra, and Palin [13].Much research has been done in the area of dimensionality reduction. Principal Component Analysis (PCA), using the Karhunen-Loeve transformation is a common method for dimensionality reduction. While PCA gives optimal dimensionality reduction, while maintaining deity of the signal in a mean square error sense, it is not optimal with respect to any particular signal analysis task, such as target detection or classification. Other dimension reduction techniques include Isomap, Multidimensional Scaling (MDS) and clustering. Like PCA, these techniques are mostly used for signal representation and are not optimal for classification problems. Another dimensionality reduction method employs the Vora value, which is a measurement indicating the distance between two spaces. By finding the spaces with the largest distance, we are hoping that the objects can be classified with much lower dimensionality. Penalty functions are widely used for pruning artificial neural network (ANN).

2.2 Proposed System

In our work as a first hybrid model, we constructed a model for dimensionality reduction with rough set approach Fahlman [9] performed studies about random weight initialization techniques for multilayer neural networks. He proposed the use of a uniform distribution over the interval [-1.0, 1.0], but experimental results showed that the best initialization interval to the problems he dealt with varied in ranges between [-0.5, 0.5] and [-4.0,4.0]. Some researchers tried to determine the best initialization interval using other neural network parameters. Kim and Ra [6] calculated a lower bound for the initial length of the weight vector of a neuron to be $\sqrt{\frac{\alpha}{d_{in}}}$,
where α is the learning rate and d_{in} is the neuron fan-in. Boers and Kuiper [9] initialize the weights using a uniform distribution over the interval $\left[\frac{-3}{\sqrt{d_{in}}}, \frac{3}{\sqrt{d_{in}}}\right]$, without any mathematical justification.Nguyen and Widrow [9] proposed a simple modification of the random initialization process. The weights connecting the output units to the hidden units are initialized with small random values over the interval [-0.5, 0.5]. The initial weights at the first layer are designed to improve the learning capabilities of the hidden units. Using a scale factor, $\beta = 0.7(q)^{1/p}$, where q is the number of hidden units and p is the number of inputs, the weights are randomly initialized and then scaled by $v = \frac{v}{\|v\|}$, where v is the first layer weight vector.In our work as a secod hybrid model, we constructed model which extract a crude domain of knowledge from the rough set concepts. From this knowledge we construct a neural network with initialization of weights and input nodes, hidden nodes as well as output nodes.

3. Implementation

First Model : The objective of this model is reduce the dimensionality of the input space by using dimensionality reduction technique, to reduce the complexity of network and to increase the classification performance. The Figure 1 shows the entire process of the first model. In this model two back propagation neural networks(BPNN) are build and the comparisons are made between these two BPNNs. The first BPNN is based on the original decision table (DT). The first BPNN is constructed with number of input layer nodes equal to number of conditional attributes in DT and the number of output layer nodes equal to number of classes in decision attribute of DT. The number of hidden layer nodes is chosen arbitrarily using some heuristics. This neural network is called Normal NN.

For the second BPNN the DT is reduced by computing reduct of the DT. To find reduct if the input data is real, we apply descretisation techniques to convert real data into integer data. The reduct computation algorithm is applied to find the reduct. The DT is reduced by including only the reduct attributes and the decision column. The second BPNN is constructed as the first BPNN except the number of nodes in the input layer is now equal to the size of the reduct found.This neural network is called Reduct NN. Both neural network's are trained using the 60 percent training data and simulation is done both on training data and also the 40 percent test data. Confusion matrices are calculated for all the results and performance evaluation and analysis is done.

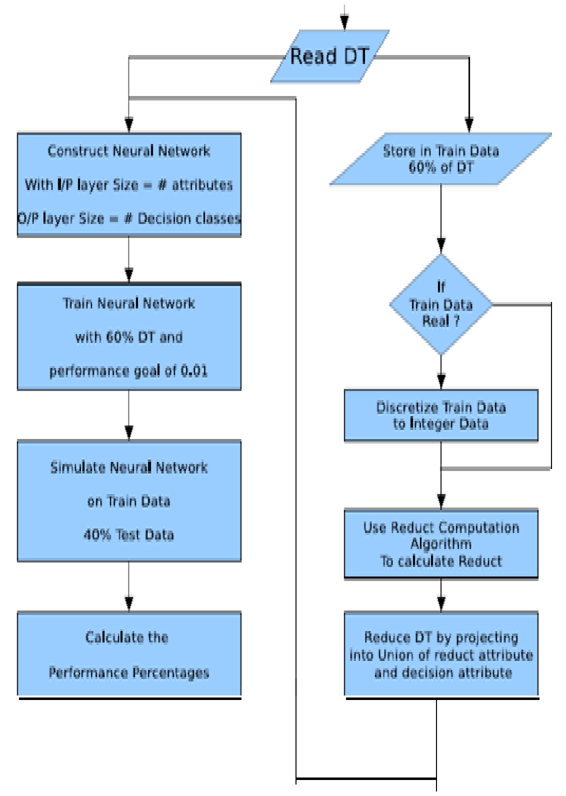

Figure 1: Flowchart representation for first model

Second Model: The objective of this model is to improve the classification performance of the neural network by initializing the weights which we obtained from the rough set induced knowledge. In this model three varieties of BPNN's are built namely Fuzzy NN, Rough Fuzzy NN, Rough Fuzzy Modified NN. The Rough Fuzzy NN is based on the work by Shankar K. Pal, Sushmita Mitra [10]. First we give the description of converting the given input data into Fuzzy Data.

Construction of Fuzzy NN: The decision table is constructed based on the given dataset. All the conditional attributes are transformed into Fuzzy Data using the procedure described above. If there are 'n' conditional attributes initially, then the transformed data consists of '3n' columns. Assume that there are m output classes having values 1 to m. For each input vector the corresponding output vector is constructed by constructing a boolean vector of size 'm' which is all zeros except having a value one in the position corresponding to the output class number the input vector belongs. The Fuzzy NN is constructed with '3n' as input layer size, 'm' as the output layer size and an arbitrary hidden layer size. As is done in the first hybrid model, 60 percent of the Fuzzy Data is used for training and 40 percent is used for testing.

Construction of Rough Fuzzy NN : Here, we formulate a methodology for encoding initial knowledge in the feed forward neural network [4], following the above algorithm. Let us consider the case of feature Fj for class ck in the l–class problem domain. The inputs for the ith representative sample Fi are mapped to the corresponding 3-dimensional feature space of $\mu_{low(Fij)}(Fi)$, $\mu_{medium(Fij)}(Fi)$ and $\mu_{high(Fij)}(Fi)$. Let these be represented by Lj , Mj and Hj , respectively. We consider only those attributes which have a numerical value greater than some threshold $Th(0.5 \leq Th \leq 1)$. This implies clamping those features demonstrating high membership values with a 1, while the others are fixed at 0. In this manner an 1*3n-dimensional attribute value (decision) table can be generated from the n-dimensional data set. Next we proceed to the description of the initial weight encoding procedure. Let the dependency factor for a particular dependency rule for class c_k be α. The weight ω^h_{ki} between a hidden node ***i*** and output node ***k*** is set at $\frac{\alpha}{fac} + \epsilon$, where f_{ac} refers to the number of conjunctions in the antecedent of the rule and € is a small random number taken to destroy any symmetry among the weights. Note that $f_{ac} \geq 1$ and each hidden node is connected to only one output node.

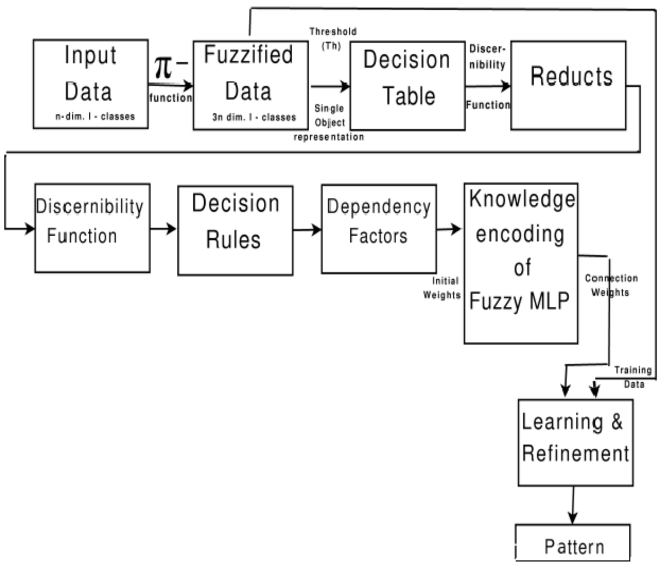

Figure 2: Block diagram for the rule generation and knowledge encoding procedure

Construction of Rough Fuzzy Modified NN : The limitation of the Rough Fuzzy NN is that each decision class members are almost linearly separable from other classes. That is one can construct convex region separation boundaries for the input members of each class. To extend the ability of the Rough Fuzzy NN with other type of datasets, a simple modification in the original algorithm is proposed. In the construction of attribute value table to calculate the D-reduct, only the best representative from each class is chosen. In the Rough Fuzzy Modified NN all the representations from each class which have a comparable frequency with the best representative also included. This attribute value table is only used to find the D-Reduct. Having found the D-reduct all the remaining computation of calculating dependency rules and dependency factors are done on the same attribute value table comprising the best representatives only. The remaining construction of NN and the initializing the weights is same as that is done for Rough Fuzzy NN.

4. RESULTS

Most of the datasets that are used in these experiments are from UCI Machine Learning Repository [16]. Vowel dataset is taken from the website of Indian Statistical Institute, Calcutta [17].

Results of the First Hybrid Model

The datasets used to experiment with first hybrid model are WDBC, Irys, Diabetes and the results for each dataset are given with respect to Neural Network with Normal data and also the Neural Network with Reduct data. In this paper we are including only one data set results i.e. WDBC results.

WDBC (Wisconsin Diagnostic Breast Cancer)

Dataset Description:

Total Number of Instances: 569
Number of Output Classes: 2
Number of Instances for Class 1: 212
Number of Instances for Class 2: 357
Number of Conditional Attributes: 30
Reducts found=[22 23 25 26 18 27 29]
Number of Attributes reduced = 23

The following table1 gives the network description , training and testing performance (%), number of epochs, and training time required for two experiments.

Experiments	NN Description			Performance		Epochs	Training Time (Sec)
	Input Nodes	Hidden Nodes	Output Nodes	Training (%)	Testing (%)		
Normal NN	30	7	2	98.6	98.23	3500	27.3
Reduct NN	7	7	2	96.4	93.4	3500	24.1

Table 1: Result Comparison Table for WDBC data set

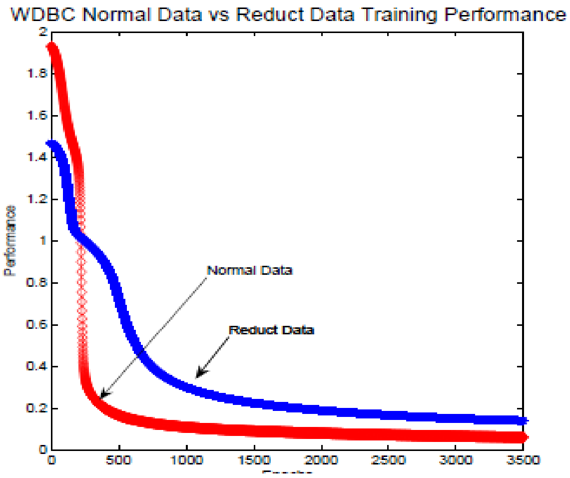

Figure 3: Plot of Epochs vs Performance for first Model

Results of the Second Hybrid Model

The datasets used to experiment with second hybrid model are Vowel, Irys, Diabetes and Wine. Here we are showing only one data set result (i.e. Vowel data set). The results of Vowel dataset is given below with respect to Fuzzy Neural Network, Rough Fuzzy Neural Network, Rough Fuzzy Modified Neural Network. For the first experiment we use fuzzy data and construct network with random initialization of weights(i.e.,Fuzzy NN). For the second experiment we construct network with weights which obtain from rough set theoretical knowledge(i.e., Rough Fuzzy NN). For the third experiments we modify the decision table and construct network with weights which obtain from rough set theoretical knowledge(i.e., Rough Fuzzy Modified NN).

Vowel Dataset Results

Dataset Description:

Total Number of Instances: 871
Number of Output Classes: 6
Number of Instances for Class 1: 71
Number of Instances for Class 2: 90
Number of Instances for Class 3: 72
Number of Instances for Class 4: 151
Number of Instances for Class 5: 207
Number of Instances for Class 6: 180
Number of Conditional Attributes: 3

The first experiment is done with the fuzzy data. There is no reduction in the conditional attributes. In second experiment we get the reduct [1 5 9], the reduction in the conditional attributes is 66.6%. In third experiment we get the reduct [3 5 7 6 8], the reduction in the conditional attributes is 44%. The following table 2 gives network description, training and testing performance (%), the number of epochs, and training time required for three experiments.

Experiments	NN Description			Performance		Epochs	Training Time (Sec)
	Input Node	Hidden Node	output Node	Training %	Testing %		
Fuzzy NN	9	12	6	82	81	50000	240.5
Rough Fuzzy NN	3	14	6	86	83.9	200	71.6
Rough Fuzzy Mod-ified NN	5	16	6	93.4	83	120	69.8

Table 2: Results Comparison Table for Vowel Data

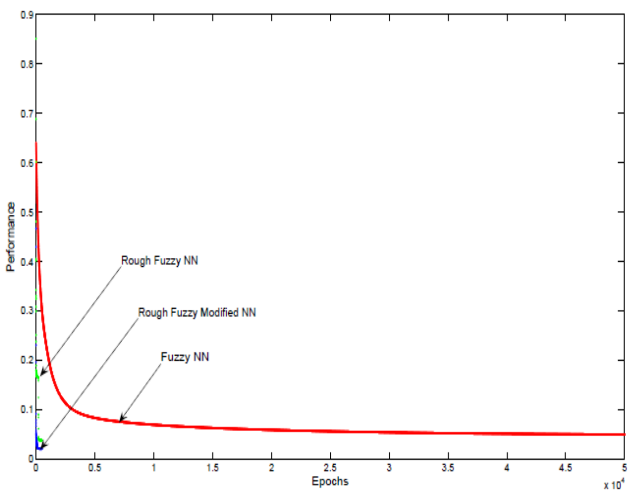

Figure 4: Plot of Epochs vs Performance in Second Model for Vowel Dataset for 50000 epochs

CONCLUSION

The important result of the first model is

- NN with reduct data is achieving comparable performance with NN with normal data while resulting in reduction of the size of the network.

The important result of the second model is

- Rough Fuzzy Modified NN has given better performance compared to Rough Fuzzy NN and Fuzzy NN. Rough Sets knowledge encoded models have reached the convergence to a minima of energy function in very few epochs and have less size compared to Fuzzy NN.
- As in the Rough models size is less they are suitable for testing of the network in time critical applications like real time embedded controllers.

REFERENCES

[1] Blum, A.,Langley, P. (1997). Selection of relevant features and examples in machine learning. Artificial Intelligence, 97, 245271

[2] Ilona, J., Chris, M. and Tim, W.: An investigation into the application of neural networks, fuzzy logic, genetic algorithms, and rough sets to automated knowledge acquisition for classification problems, Neurocomputing 24(13) (1999), 3754.

[3] S. K. Pal and S. Mitra, Fuzzy versions of Kohonens net and MLPbased classification: Performance evaluation for certain nonconvex decision regions, Inform. Sci., vol. 76, pp. 297337, 1994

[4] S. K. Pal and S. Mitra, Multilayer perceptron, fuzzy sets and classification, IEEE Trans. Neural Networks, vol. 3, pp. 683697, 1992.

[5] Fahlman, S.E. An Empirical Study of Learning Speed in Back-Propagation Networks, Tech. Rep., CMU-CS- 88-162, School of Computer Science, Carnegie Mellon University, Pittsburg, PA, September 1988.

[6] Kim, Y.K. and Ra, J.B. Weight Value Initialization for Improving Training Speed in the Backpropagation Network, Proc. of the IEEE International Joint Conf. on Neural Networks, vol. 3, pp. 2396-2401, 1991

[7] Mohua Banerjee, Susmita Mitra , and Sankar K pal,"Rough Fuzzy Mlp: Knowledge Encoding and Classification," ,IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 9, NO. 6, NOVEMBER 1998

[8] S. Mitra and S. K. Pal, Fuzzy multilayer perceptron, inferencing and rule generation, IEEE Trans. Neural Networks, vol. 6, pp. 5163, 1995.

[9] R. P. Lippmann, An introduction to computing with neural nets, IEEE Acoust., Speech, Signal Processing Mag., vol. 61, pp. 4-22, 1987.

[10] L. A. Zadeh, Fuzzy logic, neural networks, and soft computing, Commun. ACM, vol. 37, pp. 7784, 1994.

[11] J. C. Bezdek and S. K. Pal, Eds., Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data. New York: IEEE Press, 1992.

[12] RamadeviYellasiri1,C.R.Rao,Vivekchan Reddy DECISION TREE INDUCTION USING ROUGH SET THEORY COMPARATIVE STUDY Dept. of CSE, Chaitanya Bharathi Institute of Technology, Hyderabad, DCIS, School of MCIS, University of Hyderabad, Hyderabad, INDIA.

[18] Neural Networks in a Softcomputing Framework by K.-L. Du and M.N.S. Swamy:Springer

[13] Rough Sets and Current Trends in Computing by James J.Alpigini, James F.Peters, Andrzej Skrowron, Ning Zhong(Eds.), Third International Conference, RSCTC2002 Malvern, PA, USA, October2002 procedings

[14] Z. Pawlak, Rough Sets, Theoretical Aspects of Reasoning about Data, Kluwer Academic: Dordrecht, 1991.

[15] The First International Workshop on Rough Sets: State of the Art and Perspectives by Wojciech Ziarko, U. of Regina, Saskatchewan

[16] http://archive.ics.uci.edu/ml/

[17] http://iscal.ac.in/susmita/data/vowsy.html

AUTHOR'S BIOGRAPHY

KARNAKAR GULLA is currently working as Associate Professor in the department of CSE & IT at **RVR INSTITUTE OF Enggineering .and Technology**, shriguda, Ibrahimpatnam,Hyderabad(A.P). He completed his B.Tech(CSE) and M.Tech(CSE) from JNTUH,Hyderabad. His main research includes Artificial Intelligence, Neaural Networks and Network Security.

SATYANARAYANA VOLLALA is currently working as Assistant Professor in the department of CSE& IT in **Talla padmavathi college of Enggineering**, somidi, Kazipet, Warangal(A.P) His interested research areas are Artificial Intelligence, Neaural Networks and Network Security.

ASHOK KUMAR AITHAGANI is currently working as Assistant Professor in the department of CSE& IT at **RVR INSTITUTE OF Enggineering and Technology**,shriguda,Ibrahimpatnam, Hyderabad(A.P). He completed his B.Tech(CSE) and M.Tech(CSE) from JNTUH,Hyderabad. His interested research areas are Artificial Intelligence, Neaural Networks,Network Security and Data warehousing and Data Mining

SRINIVAS PILLI, He is currently pursuing M.Tech and his specialization is Software Engineering at Vidya Bharathi Institute of Technology Pembarthy, Jangaon, Warangal(AP).