ISSN:0975-9646

P. Vikram et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (5) , 2011, 2309-2313

Rainfall forecasting using Nonlinear SVM based on PSO

P.VIKRAM
KLUNIVERSITY ,Vaddeswaram

P.RAGHU VEER
NRI INSTITUTE OF TECHNOLOGY,Vijayawada

***ABSTRACT :*Accurate rainfall forecasting has been one of the most significant role in order to reduce the risk to life and to alleviate economic losses by natural disasters. Recently, support vector regression (SVR) provides an alternative approach for developing rainfall forecasting model due to the use of a risk function consisting of the empirical error and a regularized term which is resulting from the structural risk minimization. To put up an effectual SVR model, SVR's parameters must be positioned carefully. This study proposes a novel approach, known as particle swarm optimization algorithm (SVR–PSO), which searches for SVRs optimal parameters, and then adopts the optimal parameters to build the SVR models.The monthly rainfalls in the Andhra Pradesh during 2006–2010 were employed as the data set. The experimental results demonstrate that SVR–PSO outperforms the SVR models based on the normalized mean square error (NMSE) and mean absolute percentage error (MAPE).**

***Keywords:*Support Vector Regression; particle swarm optimization,Rainfall Forecasting;**

1. INTRODUCTION

Weather forecasting is the application of science and technology to predict the state of the atmosphere for a future time and a given location. Traditionally, the mode of "assumption–simulation–forecast" was used in rainfall forecasting, which is a confirmable data analysis with multiple regression analysis, time series analysis, and exponential smoothing analysis and so on. The global nature of this phenomenon is very complicated and requires sophisticated computer modeling and simulation to predict accurately, for example, while some regions of the world are noticing a systematic decrease in annual rainfall, other notice increases in flooding and severe storms. Artificial Neural Network (ANN), has been widely accepted as one of an alternative approaches for developing rainfall forecasting model. The success of the ANN models is attributable to its generalization capability to predict the output of new data after the neural network is trained [1], [2]. Some of these studies, however, showed that ANN had some limitations in learning the patterns because the generalization of single neural network is not unique. In the practical application, ANN often exhibits inconsistent and unpredictable performance on noisy data.

Recently, support vector machine (SVM), a novel neural network algorithm, was developed by Vapnik and his colleagues, which is a learning
machine based on statistical learning theory, and which adheres to the principle of structural risk minimization seeking to minimize an upper bound of the generalization error, rather than minimize the training error (the principle followed by ANNs). Originally, SVM has been presented to solve pattern recognition problems. However, with the introduction of Vapnik's ε-insensitive loss function, SVM has been developed to solve nonlinear regression estimation problems, such as new techniques known as support vector regression (SVR) [8], which have been shown to exhibit excellent performance.

At present, SVR has been emerging as an alternative and powerful technique to solve the nonlinear regression problem. When using SVR, the main problems is confronted: how to choose the optimal input feature subset for the SVR, and how to set the best kernel parameters. The proper parameters setting can improve the SVR regression accuracy. Inappropriate parameters in SVR lead to over-fitting or under–fitting. Different parameter settings can cause significant differences in performance. Therefore, selecting optimal hyper-parameter is an important step in SVR design. In this paper, a novel method is presented for rainfall forecasting model in terms of SVR technique based on particle swarm optimization and projection pursuit technology (SVR–PSO–PP), which use Projection Pursuit Technology based on Particle Swarm Optimization (PP–PSO) for feature selection of SVR, and then the PSO algorithm is used to evolve and design the parameters of SVR (SVR-PSO). The rainfall data of Andhra Pradesh is predicted as a case study for development of rainfall forecasting model. The rest of this study is organized as follows. Section 2 describes the PP–PSO to extract the feature of input factors for SVR. Section3 elaborates the SVR–PSO model presented in this paper. For further illustration, this work employs the method setup a prediction model for rainfall forecasting in Section 4. Finally, some concluding remarks are drawn in Section 5.

2. FEATURE EXTRACTION

If the SVR is adopted without considering feature selection, then the dimension of the input space will be large and non-clean, which degrading the performance of the SVR. If an efficient and robust feature selection method is chosen then it eliminates noisy, and irrelevant and redundant data, but while maintaining the discriminating power of the data, then it becomes critical. In such a complex rainfall system, the feature extraction from the original data is very important as inputs to the regression in the SVR.

In the late 1970s, the international statistical society has developed a class of new statistical method to deal with and analysis high-dimensional data, which is called Projection Pursuit (PP) [11]. We use the PP technology based on PSO to select input feature for SVR. This method adopted Exploratory Data Analysis (EDA) in a new way as "scanning data simulation forecasting", which is suitable for dealing with non-linear, non-normal distribution data to avoid "dimension disaster". The basic idea of the method lies in: that the computer technology is used to project high-dimensional data into the low dimensional sub-space through some combination, and to and out the projection by minimizing the indicators, which can extract the original data structure or characteristics, so as to achieve the goal of the study and analysis of high dimensional data.

2.1 Particle Swarm Optimization:

Particle Swarm Optimization (PSO) is a technique used to explore the search space of a given problem to find the settings or parameters required to maximize a particular objective. This technique, first described by James Kennedy and Russell C. Eberhart in 1995, and works by maintaining a swarm of particles that move around in the search-space influenced by the improvements discovered by the other particles. In PSO algorithm it has individual best position pbest (t) and the global best position gbest (t).

- Each particle keeps track of its coordinates in the solution space which are associated with the best solution (fitness) that has achieved so far by that particle. This value is called personal best, **pbest**.
- Another best value that is tracked (specialized If a moving part of a recording machine tracks, it gets into the correct position for operating) by the PSO is the best value obtained so far by any particle in the neighborhood of that particle. This value is called **gbest**.

After calculating fitness and if the condition fails as shown in flow chart then all particles are evaluated and then particle position and velocity are updated. The velocity and position update step is responsible for the optimization ability of the PSO algorithm.

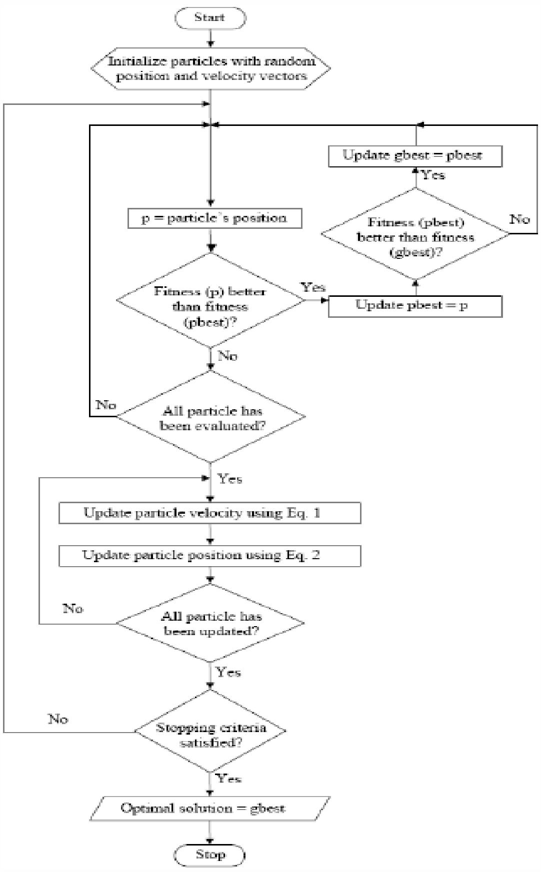

Figure1: Flowchart of particle swarm optimization algorithm (PSO)

A new velocity value for each particle or the velocity of each particle in the swarm is calculated using the following equation:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 [\hat{x}_i(t) - x_i(t)] + c_2 r_2 [g(t) - x_i(t)] \quad [1]$$

Thus, $v_i(t)$ is the velocity of particle i at time t and $x_i(t)$is the position of particle i at time t. The parameters ω, c_1, and c_2 are called user supplied coefficients. The values r_1 and r_2 are random values regenerated for each velocity update. The value $\hat{x}_i(t)$ is the individual best candidate solution for particle i at time t, and g (t) is the swarm's global best candidate solution at time t.

Once the velocity for each particle is calculated, each particle's position is updated by applying the new velocity to the particle's previous position:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad [2]$$

After all particles being updated and if stopping criteria satisfies then we can get the optimal solution gbest.

3. Evolution and Design the Parameters of SVR

SVR has newly emerged as an alternative and well effective means of solving the nonlinear regression problem. SVR has been rather successful in both academic and industrial platforms due to its many attractive features and shows potential generality performance. There are three significant features; the first thing is it can model nonlinear relationships. Secondly, the SVR training process is equivalent towards solving linearly constrained quadratic programming problems and the SVR surrounded resolution meaning is unique, optimal and suspect to generate local minima. Finally, it chooses only the essential data points to solve the regression function, which results in the sparseness of solution [13], [14], [15].

3.1 *Support Vector Regression*

Assume we are given training data $(x_i, d_i)_{i=1}^{N}$, where $x_i \in R^n$ is the input vector; d_i is the output value and *n* is the total number of data dimension. The modeling aim is to identify a regression function, $y = f(x)$, that accurately predicts the outputs d_i corresponding to a new set of input–output examples, (x_i, d_i). The linear regression function (in the feature space) is described as follows:

$$F(x) = \omega \, \Phi(x) + b,$$
$$\emptyset : R^n \rightarrow F, w \in F \quad [3]$$

where ω and b are coefficients; $\Phi(x)$ denotes the high dimensional feature space, which is nonlinearly mapped from the input space x. Therefore, the linear regression in the high-dimensional feature space responds to nonlinear regression in the low-dimension input space, disregarding the inner product calculation between ω and $\Phi(x)$ in the high dimensional feature space. The coefficients ω and b can thus be estimated by minimizing the primitive problem of SVR as follows:

$$\text{MinR}(\alpha,\alpha^*) = \sum_{i=1}^{N} d_i(\alpha_i-\alpha_i^*) - \epsilon \sum_{i=1}^{N} d_i (\alpha_i+\alpha_i^* - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(x_i,x_j)$$

$$\text{s.t.} \sum_{i=1}^{N}(\alpha_i-\alpha_i^*) = 0$$

$$0 \leq \alpha, \alpha^* \geq C, i = 1,2,\ldots N \quad [4]$$

Where α_i and α_i^* are the Lagrangian multipliers associated with the constraints, the term $K(x_i,x_j)$ is defined as kernel function, where the value of kernel function equals the inner product of two vectors x_i and x_j in the feature space φ (x_i) and φ(x_j), meaning that $K(x_i,x_j)$ = φ(x_i) × φ(x_j). The typical examples of kernel function are the polynominal kernel and the Gaussian kernel. In this paper, the Gaussian kernel function is selected for SVR model as follows:

$$K(x_i,x_j) = exp\{-\frac{\|x-x_i\|^2}{\sigma^2}\} \quad [5]$$

3.2 *SVR Parameters*

Different parameter settings can cause significant differences in performance [15, 16]. An extremely large value for parameters in SVR show the way to over-fitting, otherwise a suspiciously small value leads to under-fitting. Therefore, selecting the most favorable hyper-parameter is an important step in the SVR design.

The parameters consist of:

(1) Regularization parameter *C*: *C* determines the tradeoff cost between minimizing the training error and minimizing the models complexity.

(2) Bandwidth of the kernel function (σ^2): σ^2 represents the variance of the Gaussian kernel function.

(3) The tube size of e-insensitive loss function (ε): It is equivalent to the approximation accuracy placed on the training data points.

SVR generalization performance (estimation accuracy) and efficiency depends on the hyper-parameters (*C*, *ε* and kernel parameters σ^2), being set correctly.

3.*3 Apply PSO to SVR Parameters*

In this paper, rainfall data is randomly divided into three subsets of approximately equal size, such as training data set, validation data and test data. Generally, most of the researchers still follow the trial and error procedure in order to select the parameters of SVR, first building a few SVR models based on different parameter sets, and then test them on a validation set to obtain optimal parameters. However, this procedure is time consuming. If it is not carefully used, then there is a possibility for irrelevant information (noises) in the system (over-fitting). This paper presents a new method SVR–PSO, which optimizes all SVR's parameters simultaneously.

The performance of the parameter set is measured by the RMSE (root mean square error) on the training subset and validation data. Averaging the RMSE over the training data (or validation data) can be computed as:

$$R1 = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i - d_i)^2} \quad [6]$$

where n is the number of training data samples; d_i is the actual value and f (x_i) is the predicted value. The fitness is defined as follows:

$$fitness = \frac{R2}{R1 + R2} \quad [7]$$

where *R*1 is the RMSE over the training data and *R*2 is the RMSE over the validation data. First, the population of particles is initialized and each particle is represented as (*τ*1, *τ*2, *τ*3), where *τ*1, *τ*2 and *τ*3 denote the regularization parameters *C*, σ^2 and *ε*, respectively. The initial group is randomly generated, where each group having a random position and a random velocity for each dimension. Second, each particles fitness for the SVM is evaluated. The each particles fitness in this study is the regression accuracy. If the fitness is better than the particle's best fitness, then the position vector is saved for the particle. If the particle's fitness is better than the global best fitness, then the position vector is saved for the global best. Finally the particle's velocity and position are updated until the termination condition is figure2.

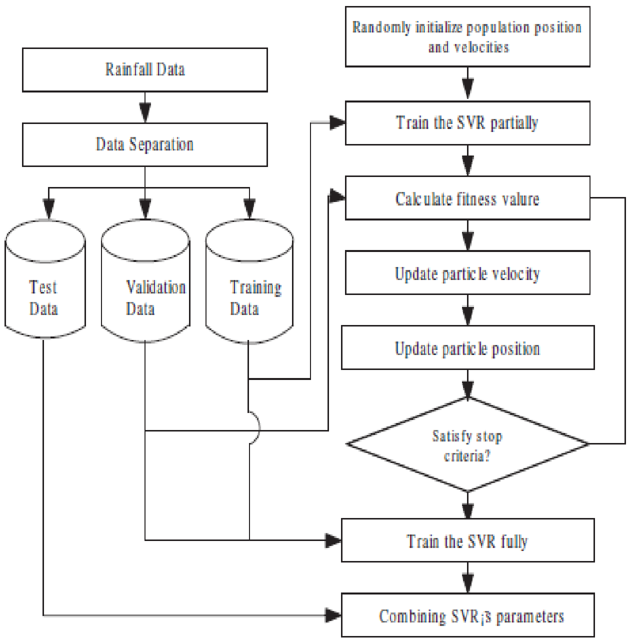

Figure2: A Flow Diagram of the Proposed SVR-PSO Model

4. MODELING FOR RAINFALL FORECASTING

The platform adopted to develop the PP-PSO approach and the SVR–PSO approach is a PC with the following features: Intel Celeron M 1.86 GHz CPU, 1.5 GB RAM, a Windows XP operating system and the Visual C++ 6.0 development environment. Table 1 gives overview of PSO parameter settings.

Table1. PSO parameter settings

Iteration times	100
Population size	60
The minimum inertia weight	0.1
The maximum inertia weight	0.9
The minimum velocity	0.1
The maximum velocity	0.9
Learning rate	2.0

4.1 The Presentation Assessment

The experimental results demonstrate that SVR–PSO outperforms the SVR models based on the errors. In order to compute the efficiency of the proposed method, we compare the results of SVR-PSO model. Four types of errors are presented such as,

Normalized Mean Squared Error (NMSE):

$$NMSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y_i})^2} \quad [8]$$

Pearson Relative Coefficient (PRC):

$$PRC = \frac{\sum_{i=1}^{n}(y_t - \bar{y_t})(\hat{y_t} - \bar{\hat{y_t}})}{\sqrt{\sum_{i=1}^{n}(y_t - \bar{y_t})^2 * (\hat{y_t} - \bar{\hat{y_t}})^2}} \quad [9]$$

Root Mean Squares Error (RMSE):

$$R1 = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i - d_i)^2} \quad [10]$$

and Mean Absolute Error (MAE) which have been found in many papers, are also used here, Interested readers can referred to [16] for more details.

The Error Value more than 25mm:

$$F1 = \sum_{i=1}^{n} I_i \quad [11]$$

Where

I_i =1, $|y_i - \hat{y_i}| > 15$

0, $|y_i - \hat{y_i}| \leq 15$

The Error Value less than 5mm

$$F2 = \sum_{i=1}^{n} L_i \quad [12]$$

Where

L_i =1, $|y_i - \hat{y_i}| < 5$

0, $|y_i - \hat{y_i}| \leq 5.$

y_i is original value, $\hat{y_i}$ is forecast value.

The original rainfall data is used as the predicted variables. Thus, we can build a SVR-PSO prediction method. When all the training results satisfy the request of error, the new factor matrix and the parameters of SVR have been obtained well by the learning data.

Related work:

@relationrainfall3-DM.PREPROCESS.InterquartileRange-Rfirst-last-O3.0-E6.0

@attribute'date {6/28/2003,6/29/2003,6/30/2003,7/1/2003,7/2/2003,7/3/2003,7/4/2003,7/5/2003,7/6/2003,7/7/2003,7/8/2003,7/9/2003,7/10/2003,7/11/2003,7/12/2003,7/13/2003,7/14/2003,7/15/2003,7/16/03,1/9/2004,1/10/2004,1/11/2004,1/12/2004,1/13/2004,1/14/2004,1/15/2004,1/16/2004,1/17/2004,1/18/2004,1/19/2004,1/20/2004,1/21/2004,1/22/2004,1/23/2004,1/24/2004,1/25/2004,1/26/2004,1/27/2004,1/28/2004,1/29/2004,1/30/2004,1/31/2004,2/1/2004,2/2/2004,2/3/2004,2/4/2004,2/5/2004,2/6/2004,2/7/2004,2/8/2004}

@attribute 'Rain fall' numeric

@attribute sunshine numeric

@attribute Dewpoint numeric

@attribute Humidity numeric

@attribute pressure numeric

@attribute'wind speed' {5.45102,4.189796,3.585714,4.942857,0.863265,2.57551,2.881632653,2.12244898,4.443969,6.379591837,0.371428571,2.144897959,8.816326531,6.363265306,5.589795918,4.381632653,5.073469388,3.555102041,3.879591837,6.132653061,4.044897959,5.126530612,5.642857143,1.783673469,4.818367347,5.185714286,6.010204082,12.75510204,11.53265306,9.869387755,4.065306122,2.589795918,5.22244898,8.483673469,2.320408163,2.732653061,3.132653061,1.734693878,0.66122449}

@attribute 'Max temp' numeric

@attribute 'Min temp' numeric

@attribute Outlier {no,yes}

@attribute ExtremeValue {no,yes}

@data

1/1/2003,19.3,0,6.6571,95.6122,992.1632,5.45102,10.9,2.4,no,yes

1/2/2003,14.8,0,6.5326,93.3061,982.6734,4.189796,10,5.2,no,yes

1/3/2003,0.9,0,2.553,95.8367,995.3469,3.585714,6,-0.4,no,no

1/4/2003,0,4.3,-0.9959,89.3061,1011.2857,4.942857,2.8,-0.8,no,no

1/5/2003,0,2.8,0.2,89.653,1013.7551,0.863265,3.2,-1.6,no,no

1/6/2003,0,4.3,-2.0469,84.4285,1017.1428,2.57551,2.4,-1.2,no,no

1/7/2003,0,1.3,-4.6653,82.6938,1018.6938,2.881632653,0,-4,no,no

1/8/2003,0,0,-2.1061,92.9387,1016.1428,2.12244898,0.4,-4,no,no

INFORMATION GAIN ATTRIBUTE SELECTION RESULT

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 8 Min temp):
        Information Gain Ranking Filter

Ranked attributes:
5.655   1 date
5.6     4 Dewpoint
5.494   6 pressure
5.442   5 Humidity
3.565   3 sunshine
3.22    7 Max temp
1.58    2 Rain fall

Selected attributes: 1,4,6,5,3,7,2 : 7
```

5. CONCLUSIONS

The rainfall system is one of the most active dynamic weather systems, and its interaction is also the most complex. And because it is influenced by many changeable factors, it is very difficult to predict the results. By using PSO algorithms evolve and design the parameters (C, ε and kernel parameter σ2) of SVR, and the actual rainfall is predicted. This paper present that PSO is applied to evolve and design the parameters (C, ε and kernel parameter σ2) of SVR, the experimental results indicate that the SVR–PSO-PP model is superior to the BP–NN model for the training samples and testing samples of rainfall forecasting in terms of different measurement. From the experiments presented in this paper we can draw the following conclusions that the SVR–PSO model can be used as an alternative tool for rainfall forecasting to obtain greater forecasting accuracy and improve the prediction quality further in view of empirical results.

REFERENCES

1. K. C. Luk, J. E. Ball and A. Sharma. An application of artificial neural networks for rainfall forecasting. Mathematical and Computer Modeling, 33:683–693, 2001.
2. M. Nasseri, K. Asghari and M. J. Abedini. Optimized scenario for rainfall forecasting using genetic algorithm coupled with artificial neural network. Expert Systems with Application, 35: 1414–1421, 2008.
3. G. J. Bowden, G. C. Dandy and H. R. Maier. Input determination for neural network models in water resources applications, Part 1-background and methodology. Journal of Hydrology, 301: 75–92, 2005.
4. J. S. Wu. A novel nonparametric regression ensemble for rainfall forecasting using particle swarm optimization technique coupled with artificial neural network. Lecture Note Computer Science, 5553(3): 49–58, Springer-Verlag Berlin Heidelberg, 2009.
5. G. F. Lin and L. H. Chen. Application of an artificial neural network to typhoon rainfall forecasting. Hydrological Processes, 19:1825–1837, 2005.
6. V. Vapnik. The nature of statistical learning theory. NewYork: Springer Press, 1995.
7. F. E. Tay and L. Cao. Modified support vector machines in financial time series forecasting. Neuro computing, 48(1–4): 847–861, 2002.
8. V. Vapnik, S. Golowich and A. Smola. Support vector method for function approximation, regression estimation and signal processing. In Edited by M.Mozer, M. Jordan and T. Petsche, Advance in neural information processing system, 9: 281–287, Cambridge, MA: MIT Press, 1997.
9. C. L. Huang and C. J. Wang. A GA–based feature selection and parameters optimization for support vector machines. Expert Systems with Applications, 31: 231–240, 2006.
10. S. W. Lin, K. C. Ying, S. C. Chen and Z. J. Lee. Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Systems with Applications, 35: 1817–1824, 2008.
11. J. H. Friedman and J. W. Turkey. A projection pursuit algorithm for exploratory data analysis. IEEE Transaction on Computers, 23(9): 881–889, 1974.
12. O. Demirci, P. C. Vincent and V. D. Calhoun. A projection pursuit algorithm to classify individuals using f-MRI data: Application to schizophrenia. NeuroImage, 39: 1774–1782, 2008.
13. D. Yocheved and I. Nathan. Multimodality exploration by an unsupervised projection pursuit neural network. IEEE Transaction on Neural Network, 9: 464–472, 1998.
14. K. Y. Chena and C. H.Wang. Support vector regression with genetic algorithms in forecasting tourism demand. Tourism Management, 28: 215–226, 2007.
15. Particle Swarm Optimization: A Tutorial James Blondin September 4, 2009
16. S. S. Keerthi. Efficient tuning of SVM hyper-parameters using radius/margin bound and iterative algorithms. IEEE Transaction on Neural Networks, 13(5): 1225–1229, 2002.
17. K. Duan, S. Keerthi and A. Poo. Evaluation of simple performance measures for tuning SVM hyper parameters. Technical report, Singapore: National University of Singapore, Department of Mechanical Engineering, 2001.
18. A Tutorial on Support Vector Machines, CHRISTOPHER J.C. BURGES
19. James Kennedy and Russell Eberhart. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, volume IV, pages 1942–1948, Piscataway, NJ, 1995. IEEE Press.

AUTHORS:

Penumacha Vikram received the B.Tech degree from the JRNRV university, and the M.Tech degree in computer science and Engineering from the KL University,vaddeswaram, in 2011. He is currently working as Assistant Professor in the Department of Electronics and Computer EngineeringKLUniversity. His research interests include computer networks,wireless communication networks,adhoc networks.

Penumacha Raghuveer received the B.Tech degree from the sir crr college of Engineering, and the MS degree in computer science and Engineering from the silicon valley University, in 2009. He is currently working as Assistant Professor in the Department of Computerand science and Engineering, in NRI Institute of Technology.
His research interests include computernetworks,wirelessnetworks, Adhocnetworks