International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

June 2021 | Issue 47 | Volume 13 | Number 2 | Pages 14-19

PREDICTING EFFORT OF AGILE SOFTWARE PROJECTS USING LINEAR REGRESSION, RIDGE REGRESSION AND LOGISTIC REGRESSION

M. Vyas N. Hemrajani

Department of Computer Science and Engineering, Jaipur Engineering College and Research Centre University, Jaipur, India, vyas.manju@gmail.com, hod.cse@jecrcu.edu.in

Abstract- In present times, the software industry has adapted Agile methodology for development of software. The Agile methodology gives more emphasis on incremental delivery, reduced risk as well as customer satisfaction. Also, in present times, one of the factors for successful project completion is estimation of effort. The term Estimation refers to prediction of measures like completion time in man-hours. Various models are used for Agile projects effort estimation which includes algorithmic models like mathematical models and non-algorithmic models like Planning Poker. In recent times, various Machine Learning based techniques are proposed in literature which are used to predict the effort in terms of completion time and which are expected to give better prediction accuracy as compared to non-machine learning techniques. This paper proposes an approach using different Regression techniques for prediction of effort and calculation of prediction accuracy as well as error. A comparison is made between the accuracy and various graphs are generated depicting the error between actual and predicted values. The results are further compared with the existing model proposed in the literature and it is observed that the regression-based model proposed in this work outperforms the model proposed in the existing literature.

Keywords: Effort Estimation, Agile, Story Points, Velocity, Regression.

1. INTRODUCTION

Estimation is defined as anticipating the measures like cost and effort measured in capital and individual hours in the context of software estimation [20]. Estimation is a major task in management of software project as it affects both client and developer side. If the estimation is accurate then the development can be planned and the progress can be monitored as well as negotiation of cost and completion date can be done by client side. Also, as the major reason of software failure lies in the inaccurate estimate of relevant parameters, hence estimation becomes a crucial and important task in predicting the reliability of software [1].

The effort of a software project is estimated by firstly estimating the size of the software and then the effort required is calculated [20]. The applicability of various estimation techniques in Agile methodology is dependent on the fact that in Agile the requirements are specified in every iteration of the development cycle [6]. According to various literature available and trends in current industrial scenario, story point-based technique is the most commonly used technique. In the Story point approach, the, metric used for measuring size is story points. The story points measure the user stories specifying the requirements given by the customers. The team velocity is another measure which is calculated by the number of user stories delivered in a sprint. Effort is calculated in man-hours using number of story points and velocity [6]. Previously non-algorithmic techniques like Planning Poker and expert judgment were used to predict effort. These were then replaced by more accurate algorithmic techniques which used mathematical models to calculate effort. Analogy based methods were also used which are based on case-based reasoning. Recently machine learning based techniques have been proposed by researchers for better prediction accuracy [4].

The paper is divided into sections. The first section throws a light on concepts of effort estimation in Agile software development. In the next section, a literature review showing the existing work by researchers in related areas is analyses. Further, techniques used are discussed. Next experimental evaluation and results of the proposed technique is shown followed by a comparative analysis of various results presented as well as comparison with the existing work is analyzed. In the end, conclusion along with scope for future work are explained.

The existing techniques for effort estimation use Mean Magnitude of Relative Error (MMRE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Prediction accuracy (PRED) [20].

2. LITERATURE REVIEW

Abrahamsson et al. [4] suggested a technique for effort prediction using user stories. They proposed that this technique can be applied to Agile software project

estimation as in Agile the requirements are developed iteratively. The proposed technique was applied to two industrial Agile software projects and it was concluded that effort estimation is accurate for well-structured user stories. The study identified that estimation becomes harder for Agile methods as requirements are not completely specified at the beginning or before the start of the development. To resolve these issues the paper suggested a novel method for effort prediction based only on predictors automatically extracted from user stories.

Usman et al. [14] performed a systematic review of existing literature where total 25 studies were analysed and the main observations were regarding the estimation techniques, size metrics, accuracy metrics and the cost drivers used in existing techniques.

Wen et al. [8] conducted a systematic review of existing literature comprising of empirical studies based on various soft computing models published in 1991-2001. The work analyzed machine learning models on various factors like Machine Learning model used, estimation accuracy, selection of model and context of estimation. The paper analyzed 84 primary publication and found that eight ML models are used in Software Development Effort Estimation. Further it was concluded regarding the prediction accuracy that accuracy of ML models outperformed the basic models.

C. Lopez Martin [10] discussed the issues which arise when machine learning models are applied for predicting the development effort and compared the accuracy of prediction of various neural network models with statistical regression model. The dataset used was published benchmarking dataset which included function point as independent variable depicting the size of project and actual effort. The results in the paper showed that the estimation accuracy of suggested model based on neural network is better as compared to the accuracy of the mathematical model based on Regression.

S. Dragicevic, S. Celar, M. Turic [9] proposed an effort estimation model based on Bayesian network model for projects specifically developed in Agile. The paper uses data of projects developed by a software company and assesses the prediction accuracy of proposed model with the statistics like Mean Magnitude of Relative Error, Mean Absolute Error and prediction accuracy at level m. The observations indicate a very good prediction accuracy.

J. Moeyersoms [12] indicated that the prediction model must be accurate as well as comprehensible i.e., it should be easy to understand. In this paper, software faults and effort are predicted by applying various data mining techniques. Rule extraction is applied and tree structure of random forest is used and regression of SVR is used to predict the fault and effort.

P.C. Pendharkar et al. [13] used Bayesian Network Model for prediction of software development effort and compared its performance with other models like neural network and regression tree models and shows that the proposed model gives competitive accuracy with the other models.

Ziauddin et al. [3] suggested a mathematical model for estimation of measures like effort for software projects developed in Agile. The model was tested on dataset of 21 project dataset having number of story points which represents the size and team velocity which represents the number of user requirements a team completes in a particular sprint. The story point is a measure of user story (user requirement). The user story is associated with two features size and complexity. Both size and complexity have values for 1 to 5, where 1 shows a very small and least complex story that is it can be completed only in a few hours of work, is very straightforward with few unknowns, requires no research and effects are localized to that story itself while 5 denotes an extremely large story, extremely complex, requires an expert skill set and has many dependencies on other stories. Each user story is thus equal to Size*Complexity of the requirement and total user stories of the project is sum of all user stories, which gives the count of story points. The calibration of velocity is done using factors like Friction and dynamic forces which calculates Deceleration. The effort of the project is then calculated. Then the accuracy was checked using the evaluation measures MMRE, PRED. The MMRE observed was 7.19% and Prediction accuracy observed was 57.14%.

S. Kheiri [18] discussed the impact of various machine learning techniques like ANN, Bayesian, SVR with kernel RBF on the diagnosis and prediction of various factors affecting dermatological diseases. Further, they concluded that machine learning application increases the accuracy of prediction and decreases the errors in terms of various factors.

3. BACKGROUND TECHNIQUES

Following techniques are used in this paper to predict the effort of projects. The story point approach is used in the SCRUM model of Agile development framework. The regression-based models are used so as to predict learning from the historical dataset which is considered for training the model

3.1. Story Point Approach

A user story is basically a user requirement. These stories are developed in iterations, usually termed as Sprints in SCRUM Agile framework. These user stories are measured in story points. Total number of user stories in an iteration gives the number of story points. Velocity is another independent variable which gives the number of user stories the team completes in a single iteration. Using these two variables effort is calculated in man-hours [17].

3.2. Linear Regression

Linear regression is a linear approach which models the relationship between a response variable referred to as target variable and one or more dependent variables referred to as predictor variables.

3.3. Logistic Regression

Logistic regression is used when the relationship can be non-linear.

3.4. Ridge Regression

It shrinks the parameters; therefore, it is mostly used to prevent multi co linearity. It reduces the model complexity by coefficient shrinkage.

4. EXPERIMENTAL EVALUATION

This section describes the experimental evaluation of the proposed model which is based on regression. To put into effect the considered techniques, dataset used by Ziauddin et al. (2012) is used. The dataset consists of three columns and twenty-one rows. The first column and the second column represent the story points, the initial or the raw velocity and the third column is the actual effort for completing the project. This dataset is used to determine the software development effort.

Machine Learning Models takes input of both story points and velocity and the accuracy of the predicted effort values as output [6]. The following steps as shown in Figure 1 are performed:

Step 1. Statistical Analysis of Dataset

a) Check the Data Normalization

i) Plot a Scatter plot for dependent and independent variable (No of Story points Vs. Effort)

ii) Plot a histogram of Effort values

iii) Calculation of statistical measures like Kurtosis and Skewness

iv) Outliers are checked with the help of Box Plots

b) If Data is Normalized proceed to step 3 else

Step 2. Data Transformation: Perform MinMax Scalar technique for normalizing the dataset

Step 3. Model Building Build the model using Regression.

Step 4. Partitioning of Dataset and Cross Validation

Step 5. Calculation of Prediction Accuracy and Error Rate.

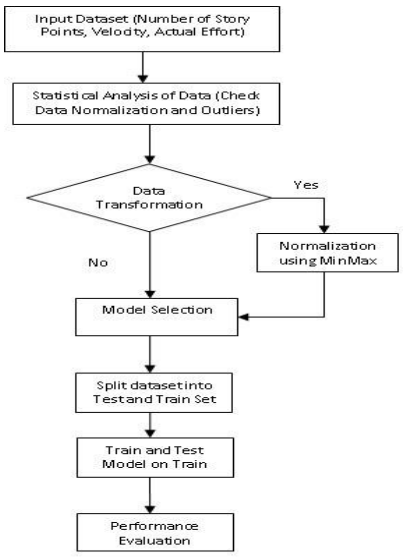

Figure 1. Flowchart of proposed approach [6]

4.1. Data Normalization

Before applying any machine learning technique, the statistical analysis of the dataset is done in which a check is made to test whether the data is normalized or not. The values of the parameters skewness and kurtosis are calculated.

Table 1 summarizes the various statistical measures with respect to actual effort. The values of skewness and kurtosis show that the data is not normalized. Further, a scatter plot is obtained to check the relationship between effort and no. of story points. The Figure 2 shows the relationship between actual effort and number of story points. Also, a histogram as shown in Figure 3 is plotted for effort for the values of number of story points. The data is converted to range 0 to 1 to show uniform results. These figures as well as the values of skewness and kurtosis indicate that the data is not normally distributed and if the data is not normally distributed then a functional transformation is applied to the data values to make it closer to the normal distribution. Here MinMax Scalar is used to transform the data. Box plots as shown in Figures 4 and 5 are plotted to check the outliers. As no significant outliers are found so we can proceed to step 3.

Table 1. Statistical Measures [6]

Mean	56.42857
Minimum	21
Maximum	112
Median	52
Standard Deviation	26.17742
Skewness	0.562172
Kurtosis	-1.0676

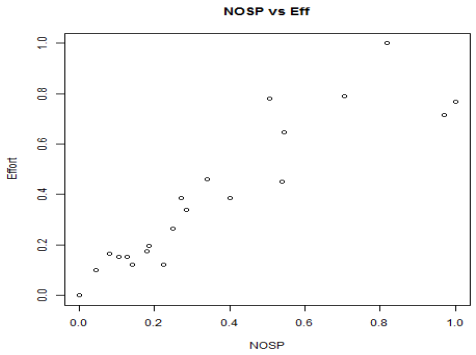

Figure 2. Scatter plot between NOSP vs. effort

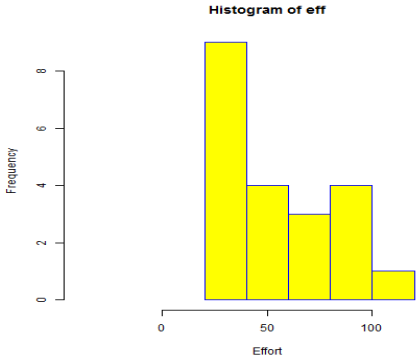

Figure 3. Histogram of effort values

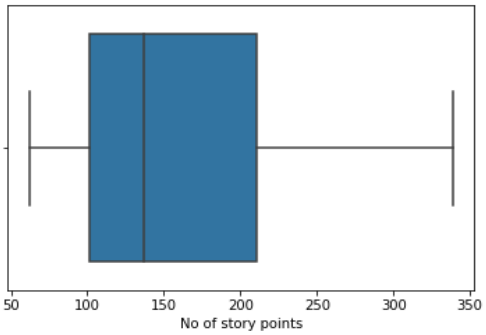

Figure 4. Box plot for NOSP

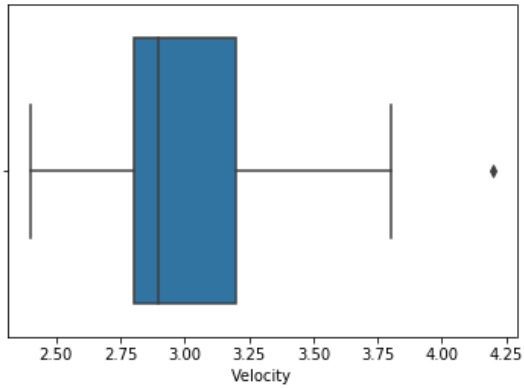

Figure 5. Box plot of effort

4.2. Model Building

The selected model for this work is Regression. It is a model which establishes relationship between dependent and independent variables. Here a curve is fit to the data points such that the distance of data points from the curve is minimized.

4.2.1. Linear Regression

It is applied when dependent variable is continuous and independent variables can be continuous or discrete. The prediction value is given by [8]

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + ..w_p x_p \quad (1)$$

where, y is the target variable and x_i are the predictor variables with w_i weights.

4.2.2. Logistic Regression

This model applies non-linear log transformation to the predicted values as it doesn't require a linear relationship necessarily.

4.2.3. Ridge Regression

This is used when independent variables are highly correlated. It solves this problem of multi co linearity through shrinking parameter.

4.3. Calculate Prediction accuracy and Error Rate

In general, the metrics which are used to evaluate the performance of estimation techniques are mean magnitude of relative error (MMRE), Root Mean Square Error (RMSE) and percentage of prediction accuracy at a certain level (PRED) [8].

- Mean Magnitude of Relative Error is given by

$$MMRE = \frac{1}{m} \sum_{k=1}^{m} MRE_k \quad (2)$$

- Root Mean Square Error is given by

$$RMSE = \sqrt{\sum_{i=1}^{TP} (AE_n - PE_n)^2 / TP} \quad (3)$$

where, TP is total number of projects, AE_n is Actual Effort of the nth test data and PE_n is Predicted Effort of the nth test data

Prediction Accuracy is given by $PRED(X)$ which is the percentage of estimates that are within X% of the original value. Normally the value of X is set to 25.

4.4. Cross-Validation on Random Sample

If prediction by the model is satisfactory on 20% division, then the model must be tested to check its performance on this size split. The performance is ensured by building it on separate subset taken as training data and prediction is done on the remaining data. For this data is divided into random sample partitions say K which are mutually exclusive and one of the portions is kept on test data and remaining (k-1) portion is used to build the model and mean squared error is calculated.

5. RESULTS AND COMPARATIVE ANALYSIS

The regression technique takes a training dataset $\{(x_1,y_1), ... (x_i,y_i)\} \subset \chi \times R$, where, χ indicates the input pattern space and finds the function $f(x)$ which optimizes the training data. In this paper, our target is to create a regression model which estimates the effort of the software projects in man-hours on the training data. In linear regression this is performed by finding a line which maximizes the sum of square error on the training set. Further it is evident from the literature [5, 6, 8] that the models which are based on machine learning techniques mostly proves to be advantageous over other statistical and non-algorithmic models which are based on intuition of experts for prediction of measures for software projects like effort, cost etc.

Considering the metrics RMSE, MMRE and PRED(25) as evaluation measures and applying the Linear Regression, logistic regression, and Ridge Regression the results are obtained.

The Table 2 depicts the mean magnitude of relative error and root mean square error along with the prediction accuracy at 25% level of significance of all the applied models The proposed regression-based model is then compared with the results of the model proposed by Zia [3] and the results show that regression-based models outperform the model proposed by Zia which is a statistical model. Figures 4-8 depicts the scatter plots depicting comparison of actual effort values vs. predicted effort values using all three models of Regression.

Table 2. Comparison of proposed model with existing model [6]

Approach	RMSE	MMRE	PRED(25)
Linear Regression	16.86	0.15	71.42
Logistic Regression	14.06	0.19	71.42
Ridge Regression	7.75	0.13	85.71
Zia [3]	-	0.0714	57.14

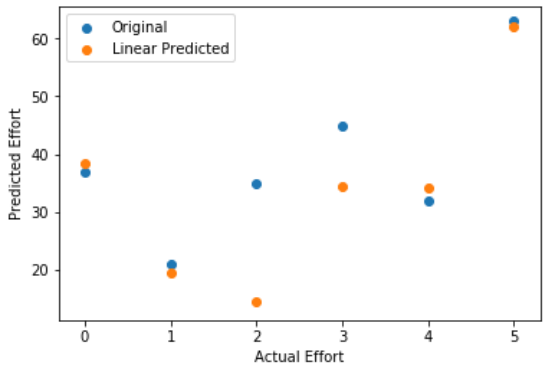

Figure 6. Actual effort vs. linear predicted

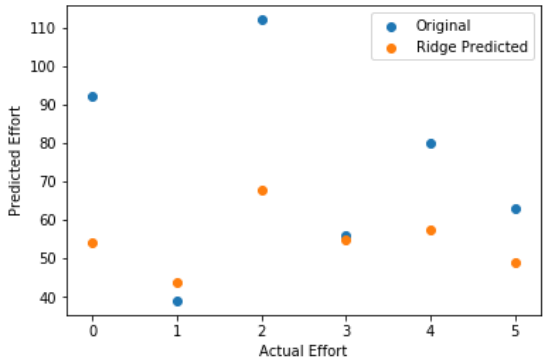

Figure 7. Actual effort vs. ridge predicted

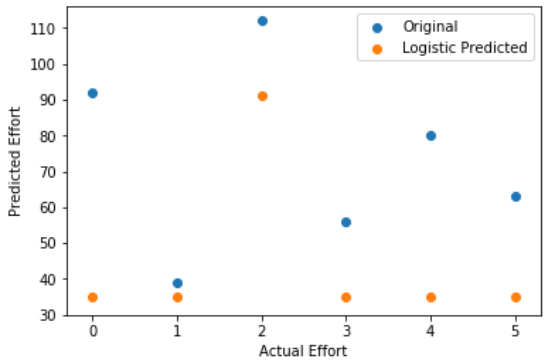

Figure 8. Actual effort vs. logistic regression

6. CONCLUSION AND FUTURE SCOPE

From the existing literature, it is observed that Agile is the most popular framework for development of project now-a days in software industry, Also, it is evident that effort estimation is an important task in project management. This paper considers the story point approach for effort calculation and for optimization, Linear, Ridge and Logistic regression models are used. The results obtained are compared based on certain evaluation measures like MMRE, RMSE and PRED(25). It has been observed that the Ridge Regression outperformed other models. Further the suggested regression-based model is compared with a statistical model given in [3]. The computations in this paper are performed using Python and the outputs are generated. This work can be further extended using other machine learning techniques as well as the use of ensemble algorithms may give better prediction accuracy.

ACKNOWLEDGEMENTS

The great work of Mr. Satapathy that was a doctoral thesis was a great help for developing this paper. The authors sincere acknowledgment to the Ph.D. thesis's supervisor for the cooperation, effort and constant support and motivation to spent a valuable part of the time for the paper.

REFERENCES

[1] A. Schmietendorf, M. Kunz, R. Dumke, "Effort Estimation for Agile Software Development Projects", 5th Software Measurement European Forum, pp. 113-123, May 2008.

[2] E. Coelho, A. Basu "Effort Estimation in Agile Software Development Using Story Points", International Journal of Applied Information Systems (IJAIS), Issue 3 Vol. 7, pp. 7-10, August 2012.

[3] S.K.T. Ziauddin, S. Zia, "An Effort Estimation Model for Agile Software Development", Advances in Computer Science and its Applications (ACSA), Vol. 2, No. 1, pp. 314-324, 2012.

[4] P. Abrahamsson, I. Fronza, R. Moser, J. Vlasenko, W. Pedrycz, "Predicting Development Effort from User Stories", Empirical Software Engineering and Measurements (ESEM) International Symposium, pp. 400-403, Baniff, AB, Canada, 22-23 Sept. 2011.

[5] B. Baskeles, B. Turhan, A. Bener, "Software Effort Estimation Using Machine Learning Methods", Computer and Information Sciences, 2nd IEEE International Symposium on Computer and Information Sciences, pp. 1-6, 2007.

[6] S.M. Satapathy, S.K. Rath, "Empirical Assessment of Machine Learning Models for Agile Software Development Effort Estimation Using Story Points", Innovations in Systems and Software Engineering, Springer, pp. 191-200, 2017.

[7] M. Cohn, "Agile Estimating and Planning", Pearson Education, 2005.

[8] J. Wen, S. Li, Z. Lin, C. Huang, "Systematic Literature Review of Machine Learning Based Software Development Effort Estimation Models", Information and Software Technology, Elsevier, Vol. 54 Issue 1, pp. 41-59, January 2012.

[9] S. Dragicevic, S. Delar, M. Turic, "Bayesian Network Model for Task Effort Estimation in Agile Software Development", Journal of Systems and Software, Vol. 127, pp. 109-119, May 2017.

[10] C. Lopez Martin, "Predictive Accuracy Comparison between Neural Networks and Statistical Regression for Development Effort of Software Projects", Applied Soft Computing, Elsevier, Vol. 27, pp. 434-449, 2015.

[11] O. Malgonde, K. Chari, "An Ensemble-Based Model for Predicting Agile Software Development Effort", Empirical Software Engineering, Springer, Vol. 2, Issue 24, pp. 1017-1055, 2018,

[12] J. Moeyersoms, E.J. De Fortuny, K. Dejaeger, B. Baesens, "Comprehensible Software Fault and Effort Prediction: A Data Mining Approach", Journal of Systems and Software, Elsevier, Vol. 100, pp. 80-90, 2015.

[13] P.C. Pendharkar, G.H. Subramanian, J.A. Rodger, "A Probabilistic Model for Predicting Software Development Effort", IEEE Transactions on Software Engineering, Vol. 11, Issue 7, pp. 615-624, August 2005.
[14] M. Usman, E. Mendes, F. Weidt, R. Britto, "Effort Estimation in Agile Software Development: A Systematic Literature Review", Proceedings of the 10th International Conference on Predictive Models in Software Engineering, pp. 82-91, New York, US, September 2014.
[15] S. Bilgaiyan, S. Mishra, M. Das, "A Review of Software Cost Estimation in Agile Software Development Using Soft Computing Techniques", Computational Intelligence and Networks (CINE), IEEE, pp. 112-117, January 2016.
[16] R. Britto, M. Usman, E. Mendes, "Effort Estimation in Agile Global Software Development Context", International Conference on Agile Software Development, Springer, Vol. 199, pp. 182-192, Cham, May 2014.
[17] E. Coelho, A. Basu, "Effort Estimation in Agile Software Development using Story Points", International Journal of Applied Information Systems, Foundation of Computer Science, Vol. 3, Issue 7, pp. 182-192, New York, USA, 2012.
[18] S. Kheiri, V. Yousefi, S. Rajebi, "Evaluation of K-Nearest Neighbor, Bayesian, Perceptron, RBF and SVM Neural Networks in Diagnosis of Dermatology Disease", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 42, Vol. 12, No. 1, pp. 114-120, March 2020.
[19] M. Zile, "Improved Control of Transformer Centers Using Artificial Neural Networks", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 40, Vol 11, No. 3, pp. 28-33, September 2019.

BIOGRAPHIES

Manju Vyas was born in Rajasthan, India, 1980. She received the B.E. degree from University of Rajasthan, India and the M.Tech. degree from Jodhpur National University, India. Currently, she is pursuing her Ph.D. degree from in Computer Science and Engineering, Jaipue Engineering College and Research Centre (JECRC) University, Jaipur, India. Currently, she is working as Assistant Professor in the same university.

Naveen Hemrajani was born in Rajasthan, India, 1970. He has received his B.E. degree in Computer Science and Engineering and M. Tech. (CSE) in 1992 and 2004, respectively. His Research Topic for Ph.D. was Admission Control for Video Transmission over IP Networks. He is HOD, Computer Science and Engineering, JECRC University, Jaipur, India for more than 26 years of teaching and research experience. Simultaneously, he is also working as Professor and Director Internship Program at JECRC University. He is also working on government funded incubation center at JECRC University. He was Principal (Eng.), SGVU and is former Chairman of Computer Society of India (Jaipur Chapter). He has expertise in Network Security, MANET and Software Engineering. He has published three books and many research papers in international and national journals of repute and presented several papers in international and national conferences. He is also Editorial Board member of many international journals of repute. He has also organized various international conferences, workshops and seminars.