Volume 6, Issue 7, July 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# Experimental Recognition of Random Forest for Agile Software Effort Estimation

**Anjali Sharma***
Student of Department of CSE, UIET,
KUK, Haryana, India

**Karambir**
Assistant Professor, Department of CSE, UIET,
KUK, Haryana, India

***Abstract- Agile Software development has turn illustrious in industries for developing the software. Software effort estimation process in any software project is not essential but also a critical component. The appearance of agile methods in the software development field has presented many opportunities and challenges for researchers and practitioners. One of the most important challenges is effort estimation for agile software development. Though different types of neural networks General Regression Neural-Network (GRNN), Group Method of Data Handling (GMDH) Polynomial Neural-Network, Probabilistic Neural Network (PNN), and Cascade-Correlation Neural-Network (CCNN) are used to estimate the effort for agile software development but the results are not so much accurate. To achieve better results, effort estimation of agile projects researchers used Random Forest in the place of all types neural-network because Random Forest is simple to implement. Random Forest provides better results as compare to all types of neural-network.***

***Keywords: Agile Software Development, Neural-Network, General Regression Neural Network (GRNN),Group Method of Data Handling(GMDH) Polynomial Neural Network, Software Effort Estimation, Random Forest.***

## I. INTRODUCTION

Agile software development methodologies are applied to create the high quality software in the shorter period of time. It is a alternate of the traditional project management used in software development. Agile software development is a methodology for original process that waits for the need for flexibility and applies a level of practicality into the delivery of the complete product. Agile methods are utilized for developing software to permit organizations respond to volatility. They provide possibilities to evaluate the direction all through the software development life cycle [1]. By accenting on the replication of work cycles along with product the teams return an additive and iterative development. Instead of the assuring to market an assemble software that hasn't been developed, agile allows teams to frequently re-plan their release to optimize its value throughout development in the marketplace making them competitor [2] [3]. Predictability is the main goal of project management, we require to be able to estimate the size and complexity of the products to be built in order to decide what to do next [4]. For this, requirements need to be collected. Requirements in agile development are counted down in cards and are called user stories. These stories are estimated using story points. The team explains the relationship between story point and effort. Generally 1 story point is equal to 1 ideal working day. Total no. of story points that a team can convey in a sprint (an iteration in agile software development) is called as "team velocity" or story points per sprint. Now for obtaining better prediction accuracy, Random Forest Method is used in this study. The results found by applying this method is empirically validated and compared to measure their performance.

## II. WORK DONE

Random Forest Model provided an efficient balance between calculation times and accuracy of predictions. One of the most useful features of learning method for classification such as random forest was its ability to explore an open range of potential covariates as made available by user [17]. The classification results of two models i.e. Random Forest and J48 for classifying twenty versatile datasets. In it shown the comparison results obtained from methods i.e. random Forest a Decision Tree (J48). The classification results shown that Random Forest give better results for the similar number of attributes and large data sets i.e. with greater number of instances, while J48 is handy with small data sets [18]. Random forest (RF) was a popular tree-based ensemble machine learning tool that was extremely data adaptive, applies to "large p, small n" problems, and was abled to account for correlation as well as interactions among features. It makes RF mainly appealing for high-dimensional genomic data analysis. It methodically reviews the applications and current progresses of RF for genomic data [19]. The potential of the random forests ensemble classification and regression technique to improve rainfall rate assignment during day, night and twilight based on cloud physical properties recovered from Meteosat Second Generation (MSG) Spinning Enhanced Visible and InfraRed Imager (SEVIRI) data. Random forests (RF) models were contained a combination of characteristics that made them well suited for its application in precipitation remote sensing [20].

 

## III. DATA

The dataset of twenty one records which has used for implementing the proposed model Random Forest [22]. The inputs to the random forest models are total number of estimated time, actual Time, total number of data in the dataset and the output is the effort i.e., the completion time. The model is tested and validated for achieving better accuracy. The calculation of Velocity is pretty straightforward, i.e. Velocity = Distance / Time

For our uses, the distance is Units of Effort and Time (the denominator) is the length of our Sprint. Velocity is computed: Vi = Units of Effort complete / Sprint Time.

The examined Velocity is simply how many Units of Effort your team completes in a typical Sprint.

Table 2 of Dataset [22]

| P.No. | Actual Time | Estimated Time |
|---|---|---|
| 1 | 63 | 58 |
| 2 | 92 | 81 |
| 3 | 56 | 52 |
| 4 | 86 | 87 |
| 5 | 32 | 29 |
| 6 | 91 | 95 |
| 7 | 35 | 29 |
| 8 | 93 | 84 |
| 9 | 36 | 35 |
| 10 | 62 | 66 |
| 11 | 45 | 41 |
| 12 | 37 | 39 |
| 13 | 32 | 35 |
| 14 | 30 | 26 |
| 15 | 21 | 22 |
| 16 | 112 | 103 |
| 17 | 39 | 40 |
| 18 | 52 | 50 |
| 19 | 80 | 76 |
| 20 | 56 | 51 |
| 21 | 35 | 34 |

## IV. ALGORITHM

1. If the number of cases in the training set is N, select from sample s from N cases at random but with replacement, from the original data .i.e. s<N.
2. This sample will be training set for growing the tree.
3. If there are M input variables, a number m< <M is identified such that at each node. The best split on this m is applied to split. The node the value of M is supposed constant during the forest growing.
4. Each tree is growth to the largest extent possible. There is no pruning.

## V. RESULTS

***Performance Metrics***

Where AT= Actual Time of i test data and $PT_i$ = Predicted Time of i test data and TD = Total Number of Data in the dataset.

I. MSE

The Mean square error (MSE) is calculated:

$$MSE = \sum_{i=1}^{TD}(AT_i - PT_i)^2/TD \quad (1)$$

II. MMRE

The Mean Magnitude of Relative Error (MMRE) is calculated:

$$\sum_{i=1}^{TD}\left(\left|AT_i - PT_i\right| / AT_i\right) \quad (2)$$

III. Squared Correlation Coefficient ($R^2$)

The squared correlation coefficient (R2) is calculated as:-

$$R^2 = 1 - \sum_{i=1}^{TD}(AT_i - PT_i)^2/\left(\sum_{i=1}^{TD} AT_i - AT\right) \quad (3)$$

IV. Prediction Accuracy (PRED)

The Prediction Accuracy (PRED) is calculated as:

$$PRED = \left(1 - \left(\sum_{i=1}^{TD}\left(\left|AT_i - PT_i\right|\right)/TD\right)\right) * 100 \quad (4)$$

1. Description of Results

The table 2 illustrates the comparison mean Square Error (MSE), squared correlation coefficient ($R^2$), Mean Magnitude of Relative Error (MMRE), Prediction Accuracy (PRED) values for different types of Neural Network (GRNN, PNN, GMDH, CCNN) and Random Forest. Probabilistic neural network solves the optimization problem in an off line manner, hence it has low accuracy of prediction. GRNN also performs the prediction in an off line manner i.e. there is no real

 

training of the network. The network parameters don't get optimal values, so the accuracy of prediction is lower than others. The Self-organizing networks perform well because proper training is done for obtaining optimal network configuration with the objective of achieving maximum accuracy. Thus, the network built at the end of training best fits the data set and yields high values of accuracy when used on testing. On comparing the results it is examined that Random Forest performs better or gives better values of MSE, $R^2$, MMRE, PRED than Neural Network.
Table 2 shows the comparison between existing (Neural Network) and proposed algorithm (Random Forest

Table 2. Comparison of Results

| ALGO | MSE | R2 | MMRE | PRED |
|---|---|---|---|---|
| **GRNN** | 0.0244 | 0.7125 | 0.3581 | 85.9182 |
| **PNN** | 0.0276 | 0.6614 | 1.5776 | 87.6561 |
| **GMDH** | 0.0317 | 0.6259 | 0.1563 | 89.6689 |
| **CCNN** | 0.0059 | 0.9303 | 0.1486 | 94.7649 |
| **RF** | 0.009 | 1.000 | 0.019 | 98.108 |

For implementing the proposed approach (Random Forest), the data set given above is used. The inputs to the Random Forest are Estimated Time, Actual Time and the total number of data in the dataset and the output is the effort i.e. completion time. Random Forest is examined and authenticated for achieving better accuracy.
The formula's that are applied for calculating the results mentioned above.

## VI. ANALYSIS

### i. Comparison of Mean Square Error (MSE)

Fig 1 indicates the comparison mean Square Error (MSE) values for different types of Neural Network (GRNN, PNN, GMDH, CCNN) and Random Forest. Among all types of Models, the presented model (Random Forest) performs better. The learning process in Random Forest is fast. The Implementation is trouble-free as compare to neural networks and also performs better than decision tree. The execution time taken for the completion task is 0.5seconds

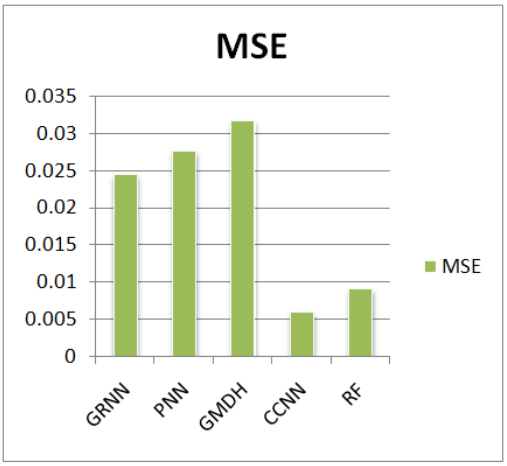


Figure 1. Comparison of MSE between Neural Networks and Random Forest

### ii. Comparison of Squared Correlation Coefficient ($R^2$)

Fig 2 shows the comparison Squared Correlation Coefficient ($R^2$) values for different types of Neural Network (GRNN, PNN, GMDH, CCNN) and Random Forest. Among all types of Models, the presented model (Random Forest) provides better value of $R^2$.

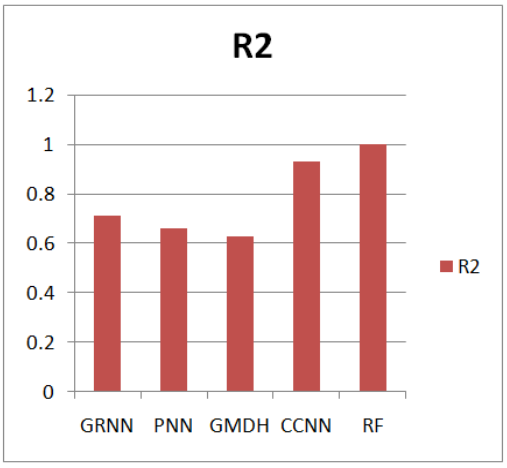


Figure 2. Comparison of $R^2$ between Neural Networks and Random Forest

 

**iii. Comparison of Mean Magnitude of Relative Error (MMRE)**

Fig 3 demonstrates the comparison Mean Magnitude of Relative Error (MMRE) values for different types of Neural Network (GRNN, PNN, GMDH, CCNN) and Random Forest. Among all types of neural networks, Random Forest presents better value of MMRE.

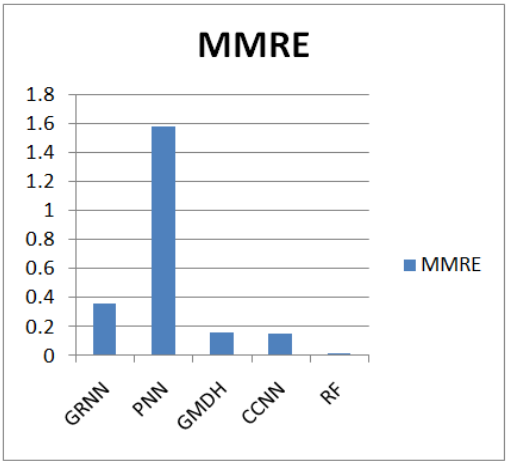


Figure 3 Comparison of MMRE between Neural Networks and Random Forest

**iv. Comparison of Prediction Accuracy**

Figure 4 indicates the comparison Prediction Accuracy (PRED) values for different types of Neural Network (GRNN, PNN, GMDH, CCNN) and Random Forest. Among all types of neural networks, Random Forest predicts better accuracy.

Figure 4 clearly shows that the presented model provides better accuracy as compare to existing model.

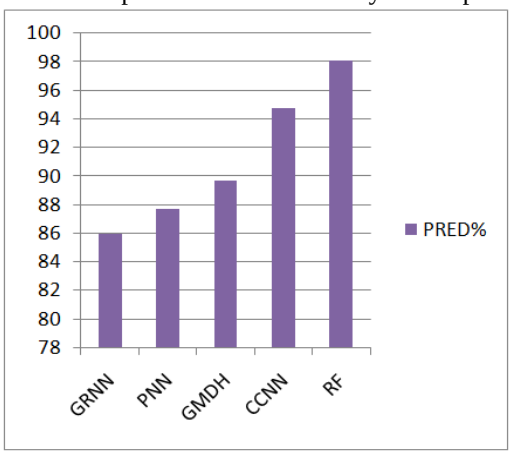


Figure 4. Comparison of Prediction Accuracy between Neural Networks and Random Forest

## VII. CONCLUSION

In this paper Random Forest and Story Point approach was used for estimation the effort of a real life case study. Random forest is a notion of general technique of random decision forests that are an ensemble learning method for classification, regression and other tasks. Story point approach is one of the method that can be used for developing mathematical models for agile software effort estimation. At the end of this paper results obtained from Random Forest and all types of Neural Network (GRNN, PNN, GMDH and CNN). The comparison of both was shown in the table and graph. As shown in this paper random forest gives better results as compare to all types of Neural Network (GRNN, PNN, GMDH and CNN). The computations for above methodologies were executed, and results were obtained using MATLAB. The existing work accuracy is 94.76% and the proposed work accuracy is 98.108 %. So, the proposed model (Random Forest) performs 3.3431% better results as compare to existing work.

The present work contains dataset of twenty one records but it can be increased for the further study purposes. For the future work on this field the large size of dataset should be available for better performance.

## REFERENCES

[1] Martin Fowler and Jim Highsmith. The agile manifesto, "Software Development. San Francisco, CA: Miller Freeman, Inc", pp. 28-35, 2001.

[2] David Cohen and Mikael Lindvall and Patricia Costa, "An introduction to agile methods. Advances in Computers", Elsevier, pp. 1-66, 2003.

[3] Rashmi Popli and Naresh Chauhan. Estimation in agile environment using resistance factors. Information Systems and Computer Networks (ISCON), International Conference, IEEE, pp. 60-65, 2014.

 

[4] Shashank Mouli Satapathy, Mukesh Kumar and Santanu Kumar Rath. Fuzzy-class point approach for software effort estimation using various adaptive regression methods. CSI Transactions on ICT, Springer, pp. 367-380, 2013.

[5] Zia, Z.; Rashid, A.; Uzzaman, K., "Software cost estimation for component based fourth-generation-language software applications, IET Software", vol. 5, pp. 103-110, 2011.

[6] Matson, J. E., Barrett, B. E. & Mellichamp, J. M., " Software Development Cost Estimation Using Function Points", IEEE Transactions on Software Engineering, vol. 20**,** pp. 275-287, 1994.

[7] Keaveney S. and Conboy K., "Cost Estimation in Agile Development Projects", Proceedings of the 14th European Conf. Information Systems (ECIS), 200

[8] Boehm, B. W., ABTS, C. and Chulani S., "Software Development Cost Estimation Approaches: A Survey. USC-CSE", 2000.

[9] Burgess, C. J. and Lefley M., "Can Genetic Programming Improve Software Effort Estimation? A Comparative Evaluation. Information and Software Technology", Vol. 43, pp. 863-873, 2001.

[10] Briand, L. C., El emam, K. and Bomarius, F.," COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment", Proceedings of the 20th International Conference on Software Engineering. Kyoto, Japan, 1998.

[11] Mukhopadhyay, T. and Kekre, S.," Software Effort Models for Early Estimation of Process Control Applications", IEEE Transactions on Software Engineering, Vol. 18**,** pp. 915-924,1992.

[12] Mendes, E., Watson, I., Triggs, C., Mosley, N. and Counsell, S.," A Comparison of Development Effort Estimation Techniques for Web Hypermedia Applications", Proceedings of the 8th IEEE Symposium on Software Metrics,2002.

[13] Jones, C.," By Popular Demand: Software Estimating Rules of Thumb. Computer, Vol. 29**,** pp. 116-118, 1996.

[14] Ferens, D. V.," Software Size Estimation Techniques. Proceedings of the IEEE National Aerospace and Electronics Conference", 1991.

[15] Boehm, B. W. and Sullivan, K. J., "Software Economics: Status and Prospects. Information and Software Technology", Vol. 41, 937-946, 1991.

[16] Ruhe, M., Jeffery, R. and Wieczorek, I.," Cost Estimating for Web Applications", Proceedings of the 25th International Conference on Software Engineering. Portland, Oregon, 2003.

[17] D. Vitorino, S. T. Coelho, P. Santos, S. Sheets, B. Jurkovac and C. Amado, "A Random Forest Algorithm Applied to Condition-Based Wastewater Deterioration Modeling and Forecasting", 16th Conference of Water Distribution System Analysis, WDSA, Elsevier Ltd., pp. 401-410, 2014.

[18] Jehad Ali, Rehanullah Khan, Nasir Ahmad and Imran Maqsood, "Random Forests and Decision Trees", IJCSI International Journal of Computer Science Issues, ISSN (Online): 1694-0814, Vol. 9, Issue 5, No 3, pp. 272-278, September 2012.

[19] Xi Chen and Hemant Ishwaran, "Random forests for genomic data analysis ", www. Elsevier.com, Department of Biostatistics, Vanderbilt University, Nashville, TN 37232, USA, pp. 323-329, 2012.

[20] Meike Kühnlein, Tim Appelhans, Boris Thies and Thomas Nauss, "Improving the accuracy of rainfall rates fromoptical satellite sensors with machine learning - A random forests-based approach applied to MSG SEVIRI ", www.elsevier.com, pp. 129-143, 2014.

[21] Zia, Z.; Rashid, A. and Uz zaman, K. (2011) Software cost estimation for component based fourth-generation-language software applications, IET Software , 5, Page(s): 103-110.

[22] Ziauddin, Shahid Kamal Tipu and Shahrukh Zia, " An Effort Estimation Model for Agile Software Development", Advances in Computer Science and its Applications (ACSA) 314 Vol. 2, No. 1, ISSN 2166-2924, 2012.

 