ISSN: 2277-9655
Impact Factor: 4.116

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A PROPOSED MODEL OF AGILE METHODOLOGY IN SOFTWARE DEVELOPMENT

Anjali Sharma*, Karambir
* Student of Department of CSE, UIET, KUK, India
Assistant Professor, Department of CSE, UIET, KUK, India

DOI: 10.5281/zenodo.57031

ABSTRACT

Agile Software development has been increasing popularity and replacing the traditional methods of software develop-ment. This paper presents the all neural network techniques including General Regression Neural Networks (GRNN), Prob-abilistic Neural Network (PNN), GMDH Polynomial Neural Network, Cascade correlation neural network and a Machine Learning Technique Random Forest. To achieve better prediction, effort estimation of agile projects we will use Random Forest with Story Points Approach (SPA) in place of neural network because Random Forest is easy to implement and better than decision tree. In this paper Neural Network is the existing model and the proposed model is Random Forest. Random Forest performs better as compare to General Regression Neural Network (GRNN).The researchers will perform comparison between Random Forest and all types (GRNN, PNN, GMDH, and CCNN) of Neural Network.

KEYWORDS: Agile Software Development; General Regression Neural Network; Probabilistic Neural Network; GMDH Polynomial Neu-ral Network; Cascade Correlation Neural Network, Random Forest.

INTRODUCTION

Agile means able to move rapidly and easily and this is what an agile software development methodology refers to. An agile software development methodology completely accepted these days. It is an iterative approach to maintain action with dynamic development environments. During the past years, a new software development approaches were initiated to fits new cultures of the software development companies. Most software companies today aim to generate high quality software in short time period with minimal costs, and within unstable, changing environments. Agile Methodologies were launched to assemble the new requirements of the software development companies [1].

Agile methodologies are used to develop and implement software rapidly according to customer requirements. Agile SDMs (Software Development Methodologies) share numerous features including prototyping, iterative development, and minimal documentation. Agile software development methodologies are used to produce the high quality software in the shorter period of time. It is an alternative to the traditional project management used in software development. Agile software development is a methodology for creative process that expects the need for flexibility and applies a level of practicality into the delivery of the complete product. Agile software development centers on keeping code simple, testing and delivering functional bits of the application. The goal of the ASD is to build upon small client approved parts as the project progresses as opposite to delivering one large application at the end of the project.

Agile software development encourages promotes adaptive planning, evolutionary development, continuous improve-ment, early delivery and promotes quick and flexible response to change. Agile software development is a methodology for the creative process that expects the need for flexibility and applies a level of practicality into the delivery of the com-plete product. It focuses on keeping code simple, testing frequently, delivering functional bits of the application as soon as they are prepared. The goal of ASD is to build upon small client approved parts as the project progresses, as opposite to delivering one large application at the end of the project. It is a lightweight software engineering framework that supports iterative development during the life-cycle of the project.

Agile software development is a group of software development methods based on incremental and iterative development, where requirements and solutions develop through collaboration between cross functional and self-organizing teams. It is a most popular model getting used in the software industry. It helps in rapid software development and gives great results in the form of a better quality and reusability. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages fast and flexible response to change.

AGILE SOFTWARE DEVELOPMENT MODEL

Agile Software Development Methodology is currently widely in use due to its characteristics of rapid software development and accommodation to changing requirement at any level of development [2]. Agile software development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, fast delivery and continuous improvement. It is a time-boxed itera-tive approach, and encourages rapid and flexible response to change. It is a conceptual framework that promotes foreseen interactions throughout the development cycle.

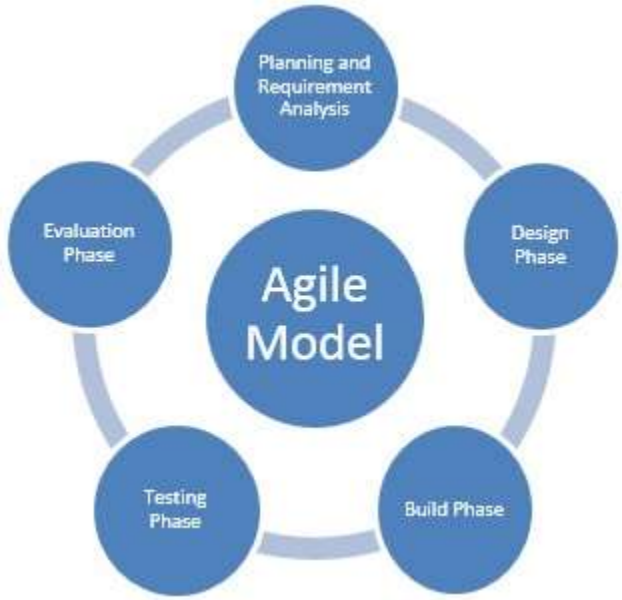

Fig. Agile Software Development Model

In this agile model Planning and requirement analysis, Design phase, Build phase, testing phase and after that Evaluation phase is defined.

RELATED WORK

Andreas Schmietendorf [3] presented an analysis of the effort estimation possibilities within agile software development methodologies. Agile methods of the software development were increasingly used for industrial projects. The application of effort estimation methods in such kind of projects was very difficult, but an important task. Classical estimation methods were needed well defined requirements. It provided an investigation about estimation possibilities, especially for the extreme programming paradigm. A classification of possible estimation methods was defined like top-down estima-tion and bottom-up estimation. Sakshi Garg and Daya Gupta [4] proposed a new cost estimation model for agile soft-ware development projects. The methodology was found to think appropriate for agile projects as it uses constraint pro-gramming to openly check for satisfaction of agile manifestos. The methodology also used in case of unavailability of historical data or expert opinion. The proposed cost estimation approach increases the correctness and accuracy of esti-mates and it was applied for the Agile Software Development Projects. Aditi Panda et al. [5] described agile software development process had become famous in industries and substituting the traditional methods of software development. The industry was capable to estimate the effort necessary

for software development using agile methodology efficiently. An effort had been made to enhance the prediction accuracy of agile software effort estimation process using SPA. For it, different types of neural networks General Regression Neural Network (GRNN), Probabilistic Neural Network (PNN), Group Method of Data Handling (GMDH) and Cascade-Correlation Neural Network) was used.

NEURAL NETWORK DETAILS

Generalized Regression Neural Networks

A generalized regression neural network (GRNN) is used for function approximation. It has a special linear layer and ra-dial basis layer. General Regression Neural Networks (GRNN) were proposed by Donald F. Specht in 1990. It consists of four layers. The Input layer includes of one neuron for every input variable. The proceeds of this layer are maintained to all the hidden layer neurons. The hidden layer has the similar number of neurons as there are lines in the training set. Hid-den neurons discover the Euclidean distance of the input from the center of the neuron and later apply the RBF kernel function. Then, the result found from this is given to the pattern layer, which has two neurons, namely the denominator summation unit and numerator summation unit. Results found from all the neurons of hidden layer are summed up and given to the denominator summation unit. For the numerator summation unit, the progress from hidden layer neurons are multiplied by the actual effort and after that summed up. The last layer divides the estimation of the numerator summation unit by the estimation of the denominator summation unit. This value is the estimated effort value for input. For improving the accuracy, unnecessary neurons are eliminated from the network and then the network is retrained. Three different criteria can be used for taking out this i.e., Minimize number of neurons, Minimize error, and fixed num-ber of neurons.

Probabilistic neural networks

Probabilistic networks are mainly used for classification (Specht1990). A probabilistic neural network (PNN) is a feed forward neural network, which was obtained from the Bayesian network and a statistical algorithm called Kernel Fisher discriminant analysis. It was recognized by D.F. Specht in near the beginning 1990s. In a PNN, the operations are ar-ranged into a multilayered feed forward network with four layers:-

i. Input layer
ii. Hidden layer
iii. Pattern layer
iv. Output Layer

PNN is used for classification problems. When an input is present, the first layer calculates the distance from the input vector to the training input vectors. This generates a vector where its elements show how close the input is to the training input. The second layer sums the role for each class of inputs and generates its net output as a vector of probabilities. Fi-nally, contend transfer function on the output of the second layer picks the maximum of these probabilities, and produc-es a 1 (positive identification) for that class and a 0 (negative identification) for non-targeted classes.

For regression analysis i.e. hiring PNN for prediction, first of all the number of classes in the dataset need to be find out. This can be done using any clustering mechanism. After finding out the number of classes and the inputs included under each class, some input vectors from each class are taken as example vectors and the dot product of example vectors and input vectors is find out.

GMDH polynomial neural networks

GMDH networks date back to 1968 (first invented by Prof Alexey G. Ivakhnenko in Ivakhnenko11). Group method of data handling (GMDH) is a relation of inductive algorithms for computer-based mathematical modeling of multi-parametric datasets that features fully automatic structural and parametric optimization of models. GMDH is used in such fields as data mining, complex systems modeling, prediction, optimization and pattern recognition. GMDH algo-rithms are qualified by inductive procedure that performs sorting-out of slowly complicated polynomial models and se-lecting the best solution by means of the so-called external criterion. A GMDH model with multiple inputs and one output is a subset of components of the base function. GMDH networks are self-organizing networks i.e., the network connec-tions are not defined initially, they are determined when the network is trained, with the objective of optimizing the net-work. Maximum accuracy is obtained by restricting the number of layers to be added to the network. It also

helps prevent over-fitting. Originally a GMDH network starts with input layer only. This layer contains one unit for each input variable. The successive layer neurons derive their inputs from any two units of the previous layer. The final layer i.e., the output layer derives two of its inputs from the previous layer and gives increase to one value (output of the network). Neurons in any layer can take input from any of the previous layers. Two variable-quadratic polynomials are used as transfer func-tions in the neurons.

Cascade Correlation neural networks

Cascade correlation neural networks are "self-organizing". Cascade-correlation (CC) is an architecture and generative, feed-forward, supervised learning algorithm. Cascade Correlation begins with a minimal network, then automatically trains and adds new hidden units one by one making a multi-layer structure. The cascade-correlation architecture, issued by Fahlman and Lebiere (1990), has two key ideas. First it develops the network on demand, so it only adds new neurons when they can help for solving the problem. Second the new neurons are added and trained one by one which can elimi-nate many of the problems presented in the previous section. [4] Cascade Correlation neural networks initially have only an input layer and an output layer. While training the network, neurons are chosen from a list of neurons and comprised in the hidden layer. The new neurons get their inputs from all the living neurons of the network, hence it is called cascade. The training algorithm attempts to increase the amount of correlation between the newly added and the network's residual error. In this study, Gaussian kernel function is applied.

PROPOSED MODEL

Random forest is a notion of general technique of random decision forests that are an ensemble learning method for clas-sification, regression and other tasks, that operate by making a multitude of decision trees at training time and outputting the class that is mode of the classes or mean prediction (regression) of the individual trees. Random decision forests exact for the decision trees habit of over fitting to their training set. The selection of a random subset of features is an example of the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" ap-proach to classification proposed by Eugene Kleinberg. The general method of random decision forests was first proposed by Ho in 1995. Random forest is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees. The method merges Breiman's "Bagging" idea and the random selec-tion of features.

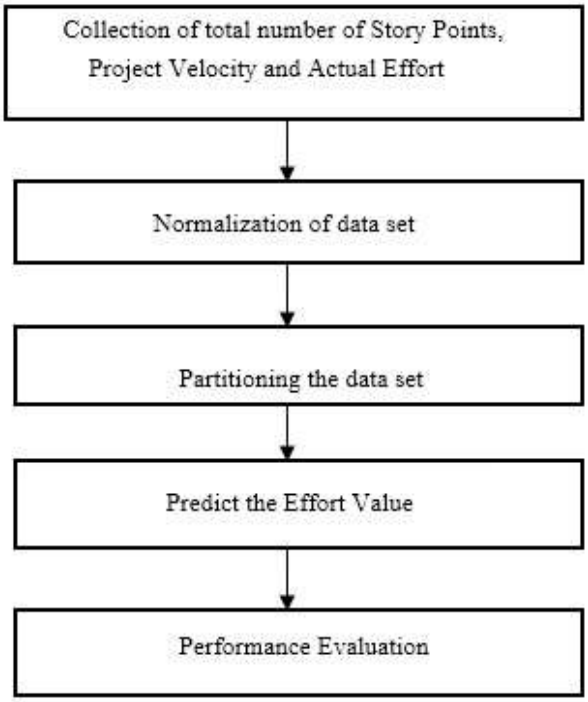

Fig. 2. Proposed Steps of Random Forest for Agile Software Effort Estimation

Features and Advantages of Random Forest:-

- It is one of the most correct learning algorithms available. For many data sets, it produces a highly correct classifier.
- It runs professionally on large databases.
- It can hold thousands of input variables without variable deletion.
- It gives estimates of what variables are significant in the classification.
- It produces an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an efficient method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
- It has methods for balancing error in class population unbalanced data sets.
- Produced forests can be saved for future use on other data.
- Prototypes are calculated that give information about the relation between the variables and the classification.
- It calculates proximities between pairs of cases that can be used in clustering, locating outliers, or give interesting views of the data.
- The capabilities of the above can be extended to unlabeled data, guiding to unsupervised clustering, data views and outlier detection.
- It presents an experimental method for detecting variable interactions.
- Random Forest is well-organized, interpretable and non-parametric for various types of datasets.
- Random Forest is very unique among popular machine learning methods.

PROPOSED WORK

D. Vitorino *et al*. presented the Random Forest Model that 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. It reported on the detailed application on the random forest classification algorithm in the context of the utilities thorough ongoing CCTV predicting sewer condition and in directing investment in inspections. Jehad Ali compared 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 an 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. Xi Chen and Hemant Ishwaran introduced 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 review the applications and current progresses of RF for genomic data, including classification, pathway analysis, prediction variable selection, genetic association, epistasis detection, and unsupervised learning. Meike Kuhnlein investigated 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. One of the key advantages was the ability to capture non-linear association of patterns between predictors and response which become important when dealing with complex non-linear events like precipitation.

RESULTS

For implementing the proposed approach (Random Forest), the data set given in the (Zia *et al.*) is applied. The inputs to the Random Forest are the total number of the story points and the output is the effort i.e. Predict time. Random Forest is tested and validated for achieving better accuracy.

Fig 2 demonstrates the comparison of MSE, R^2, MMRE, PRED values of General Regression Neural Network and Random Forest. The Comparison between GRNN and Random Forest shows that Random Forest performs better. The learning process in Random Forest is quick. The Implementation is easy as compare to neural networks and also performs better than decision tree. Random Forest provides better accuracy as compare to GRNN.

ISSN: 2277-9655
Impact Factor: 4.116

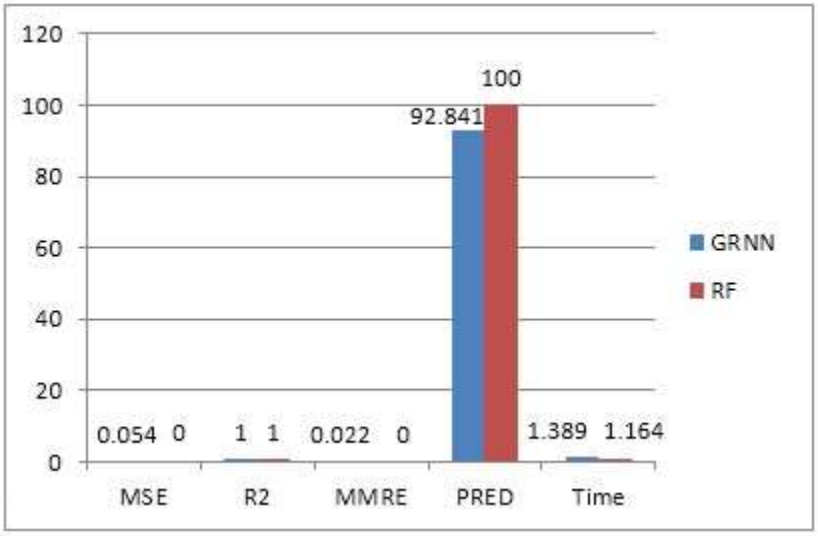

Fig. Comparison between GRNN and Random Forest

Performance Metrics

A **performance metric** is that which decides an organization's behavior and performance. Performance metrics determine an organization's activities and performance.

- **MSE**

The **mean squared error** (**MSE**) or **mean squared deviation** (**MSD**) of an estimator calculates the average of the squares of the errors, i.e., the difference between the estimator and what is estimated.
The Mean Square Error (MSE) is considered:

$$\mathbf{MSE} = \sum_{\mathbf{i=1}}^{\mathbf{TD}} (\mathbf{AE_i} - \mathbf{PE_i})^{\,2} / \mathbf{TD}$$

Where AE_i = Actual Effort of i test data and PE_i = Predicted Effort of i test data and TD = Total Number of Data.

- **MMRE**

The Mean Magnitude of Relative Error, MMRE, is probably the most widely used evaluation criterion for assessing the performance of competing software prediction models.
The Mean Magnitude of Relative Error (MMRE) is considered:

$$\sum_{\mathbf{i=1}}^{\mathbf{TD}} \left(\left| \mathbf{AE_i} - \mathbf{PE_i} \right| \; / \; \mathbf{AE_i} \right)$$

- **Squared Correlation Coefficient (R2)**

It is a statistic used in the context of statistical models whose main purpose is either the prediction of future outcomes.
The squared correlation coefficient (R2) is calculated as:-

$$\mathbf{R^2} = \mathbf{1} - \sum_{\mathbf{i=1}}^{\mathbf{TD}} (\mathbf{AE_i} - \mathbf{PE_i})^2 / \left(\sum_{\mathbf{i=1}}^{\mathbf{TD}} \mathbf{AE_i} - \mathbf{AE} \right)$$

- **Prediction Accuracy (PRED)**

It is a description of systematic errors and random errors.
The Prediction Accuracy (PRED) is calculated as:

$$\mathbf{PRED} = \left(\mathbf{1} - \left(\sum_{\mathbf{i=1}}^{\mathbf{TD}} \left(\left| \mathbf{AE_i} - \mathbf{PE_i} \right| \right) / \mathbf{TD} \right) \right) * \mathbf{100}$$

CONCLUSION

Agile methodologies are widely used in a variety of software industry projects, their flexibility provides the means to deal with many common problems faced in the development of software systems. In this paper Random Forest is the pro-posed model. Random Forest is a concept of general technique of random decision forests that are an ensemble learning method for classification, regression and other tasks.

This paper parents the comparison between General Regression Neural Network (GRNN) and Random Forest. The result shows that Random Forest performs better as compare to GRNN.

As a future work, the researchers will perform comparison between Random Forest and all types (GRNN, PNN, GMDH, and CCNN) of Neural Network. Estimate the effort required during the agile software development using machine learn-ing techniques i.e. Random Forest with story point approach.

REFERENCES

[1] Agile Alliance. Manifesto for Agile Software Development. [Online] Retrieved 16th March 2009. Available at: http://www.agilemanifesto.org.
[2] Dyba T., and Dingsoyr T., "Empirical Studies and Agile Software Development: A Systematic Review", Information and Software Technology, 2008, vol. 50, pp. 833-859.
[3] Andreas Schmietendorf, Martin Kunz and Reiner Dumke, "Effort Estimation for Agile Software Development Pro-jects", Proceedings 5thSoftware Measurement European Forum, Milan 2008, pp. 113-126, 2008.
[4] Sakshi Garg and Daya Gupta, "PCA Based Cost Estimation Model for Agile Software Development Projects", Pro-ceedings of the 2015 International Conference on Industrial Engineering and Operations Management Dubai, United Ar-ab Emirates (UAE), 2015.
[5] Aditi Panda, Shashank Mouli Satapathy and Santanu Kumar Rath, "Empirical Validation of Neural Network Models for Agile Software Effort Estimation based on Story Points", Elsevier B.V., 3rd International Conference on Recent Trends in Computing, pp. 772-781,2015.
[6] Gabor Balazs, "Cascade-Correlation Neural Networks: A Survey", Department of Computing Science, University of Alberta, Edmonton, Canada, pp. 1-6, 2009.
[7] 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.
[8] Jehad Ali, Rehanullah Khan, Nasir Ahmad and Imran Maqsood, "Random Forests and Decision Trees", IJCSI Inter-national Journal of Computer Science Issues, ISSN (Online): 1694-0814, Vol. 9, Issue 5, No 3, pp. 272-278, September 2012.
[9] 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.
[10] Meike Kühnlein, Tim Appelhans, Boris Thies and Thomas Nauss, "Improving the accuracy of rainfall rates fro-moptical satellite sensors with machine learning - A random forests-based approach applied to MSG SEVIRI ", www.elsevier.com, pp. 129-143, 2014.
[11] ZIA, Z.; RASHID, A.; UZ ZAMAN, K. (2011) Software cost estimation for component based fourth-generation-language software applications, IET Software , 5, Page(s): 103-110.