Story Point Approach based Agile Software Effort Estimation using Various SVR Kernel Methods

Shashank Mouli Satapathy[1], Aditi Panda[2], Santanu Kumar Rath[3]
Department of Computer Science and Engineering
National Institute of Technology
Rourkela - 769008, Odisha, India.
Email: shashankamouli@gmail.com[1], aditipanda@yahoo.com[2], skrath@nitrkl.ac.in[3]

***Abstract*—Agile software development process represents a major departure from traditional, plan-based approaches to software engineering. Estimating effort of agile software accurately in early stage of software development life cycle is a major challenge in the software industry. For improving the estimation accuracy, various optimization techniques are used. The Support Vector Regression (SVR) is one of these techniques that helps in getting optimal estimated values. The main objective of the research work carried out in this paper is to estimate the effort of agile softwares using story point approach. An attempt has been made to optimize the results obtained from story point approach using various SVR kernel methods to achieve better prediction accuracy. A performance comparison of the models obtained using various SVR kernel methods is also presented in order to highlight performance achieved by each method.**

***Keywords*—Agile Software Development; Software Effort Estimation; Story Point Approach; Support Vector Regression.**

I. INTRODUCTION

Agile methods are a reaction to the traditional ways of developing software and acknowledge the "need for an alternative to documentation driven, heavyweight software development processes" [1]. Now-a-days effort estimation of softwares developed using agile methods are creating a buzz in the software development community, drawing their fair share of advocates and opponents. Agile software development emphasizes on good communication between the developers, rapid delivery of software and change on demand [2]. Due to the increasing use of agile methods in industry in the last few years, the effort estimation of software developed by agile methods has become an important issue [3].

Accurate effort estimation of an agile software project is always important for performing cost-benefit analysis and determining the feasibility of the project [4]. Agile methodologies use user stories as requirement artifacts. In the case of agile projects, story point is used to measure the effort required to implement an user story. And by adding up the estimates of user stories that were finished during an iteration (story point iteration), the project velocity is obtained. Hence in this paper, total number of story points are used along with project velocity to calculate the effort required for agile software development. In order to achieve better prediction accuracy, various kernel methods-based support vector regression techniques are introduced. In case of SVR, a linear regression function is computed in a high dimensional feature space. The input data are mapped via a nonlinear function. Further, the SVR kernel methods can be applied in transforming the input data and then based on these transformations, an optimal boundary between the possible outputs can be obtained. The results of all these SVR kernel-based techniques are compared and their performance is accessed.

II. RELATED WORK

Keaveney et al. [5] investigated the applicability of conventional estimation techniques towards agile development approaches by focusing on four case studies of agile method used across different organizations. Coelho et al. [6] described the steps followed in story point-based method for effort estimation of agile software and highlighted the areas which need to be looked into further research. Andreas Schmietendorf et al. [2] provided an investigation about estimation possibilities, especially for the extreme programming paradigm. Ziauddin et al. [7] developed an effort estimation model for agile software projects. The model was calibrated using the empirical data collected from 21 software projects. The experimental results show that model has good estimation accuracy in terms of MMRE and PRED (n). Hearty et al. [8] proposed a bayesian network model of an Extreme Programming environment and showed how it can learn from project data in order to make quantitative effort predictions and risk assessments without requiring any additional metrics.

Adriano L.I. Oliveira [9] provided a comparative study on SVR, radial basis function neural networks (RBFNs) and linear regression for estimation of software project effort. The experiment was carried out using NASA project data sets and the result showed that SVR performs better than RBFN and linear regression. Kocaguneli et al. [10] investigated nonuniform weighting through kernel density estimation and found that nonuniform weighting through kernel methods cannot outperform uniform weighting Analogy Based Estimation (ABE). Braga et al. [11] proposed and investigated the use of a genetic algorithm approach for selecting an optimal feature subset and optimizing SVR parameters simultaneously aiming to improve the precision of the software effort estimates.

III. EVALUATION CRITERIA

The performance of the various models generated using SVR kernel methods can be evaluated by using the following evaluation criteria.

- The **Mean Magnitude of Relative Error (MMRE)** can be achieved through the summation of MRE over

N observations

$$MMRE = \sum_{1}^{N} \frac{|ActualEffort_i - PredictedEffort_i|}{ActualEffort_i} \quad (1)$$

- The **Prediction Accuracy (PRED)** can be calculated as:

$$PRED = (1 - (\frac{\sum_{i=1}^{N} |actual_i - predicted_i|}{N})) * 100 \quad (2)$$

where

N = Total number of data in the test set.

IV. Proposed Approach

The proposed approach is based on twenty one project data set [7]. The data set is used to evaluate software development effort and to validate the improvement. The results obtained in the validation process have provided initial experimental evidence of the effectiveness of story point approach. The block diagram, shown in Figure 1, displays the proposed steps used to determine the predicted effort using various kernel-based SVR techniques. To calculate the effort of a given

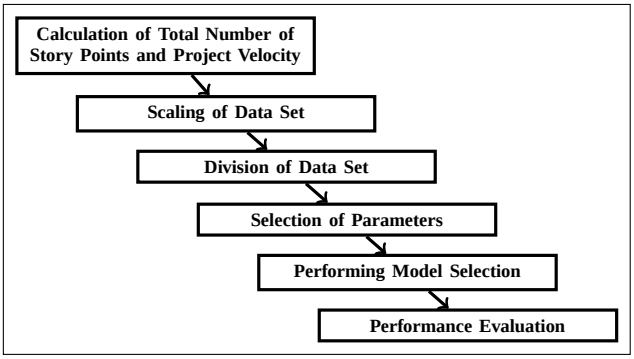

Fig. 1: Proposed Steps Used for Effort Estimation using Various SVR Kernel Methods

software project, basically the following steps have been used.

Steps in Effort Estimation

1) **Calculation of Total Number of Story Points and Project Velocity**: After collecting the data from other developed projects, the number of story point and their corresponding project velocity is calculated from the user stories using the steps provided in section 3.
2) **Scaling of Data Set**: In this step, the generated number of story points and project velocity values are used as input arguments and are scaled within the range [0,1]. Let X be the data set and x is an element of the data set, then the scaled value of x can be calculated as :

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (3)$$

where
x' = Scaled value of x within the range [0,1].
$\min(X)$ = the minimum value for the data set X.
$\max(X)$ = the maximum value for the data set X.
If $\max(X)$ is equal to $\min(X)$, then Normalized(x) is set to 0.5.

3) **Division of Data Set**: In this step, the data set is divided into three parts using five-fold cross validation approach. These are learning set, validation set and test set.
4) **Selection of Parameters**: The tunable parameters have been selected to find the best parameter C and γ using a five-fold cross validation procedure .
5) **Performing Model Selection**: In this step, a five-fold cross validation approach is implemented for model selection. The model that provides lowest MeanMagnitude of Relative Error (MMRE) value and highest Prediction Accuracy (PRED) value is selected as the best model for each fold.
6) **Performance Evaluation**: The performance of the model is evaluated using final MMRE and PRED values obtained from test samples. The performance of the model is evaluated using final MMRE and PRED values obtained from test samples. The average of MMRE and PRED accuracy(from each fold) are found out for this.

The SVR kernel-based methods are implemented using the above steps. Finally, a comparison of results obtained using various kernels-based SVR effort estimation model is presented to assess their performances.

V. Experimental Details

In this paper to implement the proposed approaches, data set given in [7] is used. The detailed description about the data set has already been provided in the proposed approach section. The inputs of the model are total number of story points and project final velocity and the output is the effort i.e., time required to complete the project. Initially,three sets of data are extracted from the available data set i.e., training data, testing data and validation data.First of all, every fifth data out of those two data sets, is extracted for testing purpose and rest data will be used for training purpose. Then the training data is partitioned into the learning and validation sets. After partitioning data into learning set and validation set, the model selection for ϵ and γ is performed using five fold cross validation process. In this paper, to perform model selection, the ϵ and γ values are varied over a particular range. The γ value ranges from 2^{-7} to 2^{7} and ϵ value ranges from 0 to 5. Hence, ninety number of models are generated to perform the model selection operation.

Table I and II show the validation error of the ninety models generated using SVR linear kernel and SVR polynomial kernel respectively based on the value of ϵ and γ. For SVR Linear kernel, *0.0132* value has been chosen as the minimum validation error. Hence based on the minimum validation error, the best model is C = 0.78121, $\gamma = 0.0078125$ and $\epsilon = 0$. Similarly for SVR Polynomial kernel, *0.0536* value has been chosen as the minimum validation error. Hence based on the minimum validation error, the best model is C = 0.78121, $\gamma = 8$ and $\epsilon = 0$.

Similarly, Table III and IV show the validation error of ninety models generated using SVR RBF kernel and SVR

TABLE I: Validation Errors Obtained Using SVR Linear Kernel

	ϵ = 0	1	2	3	4	5
$\gamma = 2^{-7}$	**0.0132**	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-6}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-5}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-4}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-3}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-2}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-1}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712
2^{0}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712
2^{1}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712
2^{2}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712
2^{3}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712
2^{4}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712
2^{5}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712
2^{6}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712
2^{7}	0.0132	0.0712	0.0712	0.0712	0.0712	0.0712

TABLE II: Validation Errors Obtained Using SVR Polynomial Kernel

	ϵ = 0	1	2	3	4	5
$\gamma = 2^{-7}$	0.0916	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-6}	0.0916	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-5}	0.0916	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-4}	0.0916	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-3}	0.0914	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-2}	0.0896	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-1}	0.0746	0.0712	0.0712	0.0712	0.0712	0.0712
2^{0}	0.0764	0.0712	0.0712	0.0712	0.0712	0.0712
2^{1}	0.2870	0.0712	0.0712	0.0712	0.0712	0.0712
2^{2}	0.1987	0.0712	0.0712	0.0712	0.0712	0.0712
2^{3}	**0.0536**	0.0712	0.0712	0.0712	0.0712	0.0712
2^{4}	0.0646	0.0712	0.0712	0.0712	0.0712	0.0712
2^{5}	0.0646	0.0712	0.0712	0.0712	0.0712	0.0712
2^{6}	0.0643	0.0712	0.0712	0.0712	0.0712	0.0712
2^{7}	0.0619	0.0712	0.0712	0.0712	0.0712	0.0712

TABLE III: Validation Errors Obtained Using SVR RBF Kernel

	ϵ = 0	1	2	3	4	5
$\gamma = 2^{-7}$	0.0874	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-6}	0.0833	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-5}	0.0756	0.0897	0.0897	0.0897	0.0897	0.0712
2^{-4}	0.0601	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-3}	0.0364	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-2}	0.0139	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-1}	0.0113	0.0712	0.0712	0.0712	0.0712	0.0712
2^{0}	**0.0088**	0.0712	0.0712	0.0712	0.0712	0.0712
2^{1}	0.0115	0.0712	0.0712	0.0712	0.0712	0.0712
2^{2}	0.0158	0.0712	0.0712	0.0712	0.0712	0.0712
2^{3}	0.0222	0.0712	0.0712	0.0712	0.0712	0.0712
2^{4}	0.0315	0.0712	0.0712	0.0712	0.0712	0.0712
2^{5}	0.0370	0.0712	0.0712	0.0712	0.0712	0.0712
2^{6}	0.0453	0.0712	0.0712	0.0712	0.0712	0.0712
2^{7}	0.0525	0.0712	0.0712	0.0712	0.0712	0.0712

Sigmoid kernel respectively based on the value of ϵ and γ. For SVR RBF kernel, *0.0088* value has been chosen as the minimum validation error. Hence based on the minimum validation error, the best model is C = 0.78121, $\gamma = 1$ and $\epsilon = 0$. Similarly for SVR Sigmoid kernel, *0.0169* value has been chosen as the minimum validation error. Hence based on the minimum validation error, the best model is C = 0.78121, $\gamma = 1$ and $\epsilon = 0$.

Finally, based on the model's parameter values, the model is again trained and tested using training and testing data

TABLE IV: Validation Errors Obtained Using SVR Sigmoid Kernel

	ϵ = 0	1	2	3	4	5
$\gamma = 2^{-7}$	0.0895	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-6}	0.0874	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-5}	0.0833	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-4}	0.0754	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-3}	0.0593	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-2}	0.0348	0.0712	0.0712	0.0712	0.0712	0.0712
2^{-1}	0.0115	0.0712	0.0712	0.0712	0.0712	0.0712
2^{0}	**0.0080**	0.0712	0.0712	0.0712	0.0712	0.0712
2^{1}	0.0169	0.0712	0.0712	0.0712	0.0712	0.0712
2^{2}	0.1693	0.0712	0.0712	0.0712	0.0712	0.0712
2^{3}	0.4131	0.0712	0.0712	0.0712	0.0712	0.0712
2^{4}	0.4285	0.0712	0.0712	0.0712	0.0712	0.0712
2^{5}	0.4503	0.0712	0.0712	0.0712	0.0712	0.0712
2^{6}	0.2520	0.0712	0.0712	0.0712	0.0712	0.0712
2^{7}	0.1211	0.0712	0.0712	0.0712	0.0712	0.0712

set respectively to estimate the effort. After implementing the SVR-based model using four different kernel methods for software effort estimation, the following results are generated.

SVR Linear Kernel Result:
Param: -s 3 -t 0 -c 0.78121 -g 0.0078125 -p 0
* Mean Squared Error (MSE_TEST) = 0.2068
* Squared correlation coefficient = 0.8936

SVR Polynomial Kernel Result:
Param: -s 3 -t 1 -c 0.78121 -g 8 -p 0
* Mean Squared Error (MSE_TEST) = 0.2057
* Squared correlation coefficient = 0.9006

SVR RBF Kernel Result:
Param: -s 3 -t 2 -c 0.78121 -g 1 -p 0
* Mean Squared Error (MSE_TEST) = 0.0030
* Squared correlation coefficient = 0.9843

SVR Sigmoid Kernel Result:
Param: -s 3 -t 3 -c 0.78121 -g 1 -p 0
* Mean Squared Error (MSE_TEST) = 0.0194
* Squared correlation coefficient = 0.8734

The *squared correlation coefficient* (r^2) is also known as the *coefficient of determination*. It is the proportion of variance in actual effort that can be accounted for by knowing story point value for test set. In the output generated, it is quite clearly observed that the *squared correlation coefficient* value for RBF kernel is very high (greater than 0.95). Thus it can be observed that a strong positive correlation exists between the story point, velocity and the predicted effort required to develop the software, and minor changes in one lead to significant changes in another. The proposed model generated using the SVR

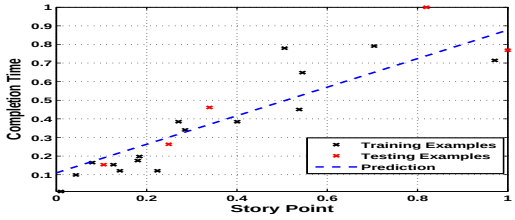

Fig. 2: SVR Linear Kernel based Effort Estimation Model

linear, polynomial, RBF and sigmoid kernel have been plotted based on the training and testing sample data set as shown in Figure 2, 3, 4 and 5. The graphs show the variation of

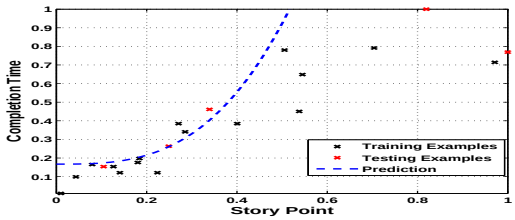

Fig. 3: SVR Polynomial Kernel based Effort Estimation Model

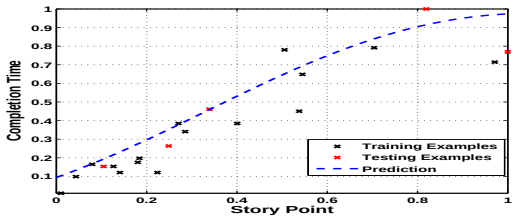

Fig. 4: SVR RBF Kernel based Effort Estimation Model

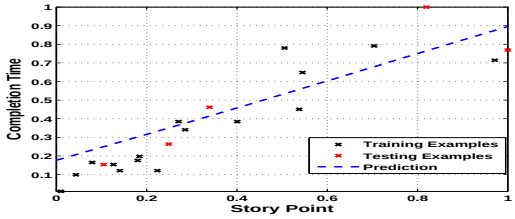

Fig. 5: SVR Sigmoid Kernel based Effort Estimation Model

predicted effort value with respect to its corresponding class point value. In these graphs, it is clearly shown that the data points are very little dispersed than the regression line. Hence the correlation is higher. While comparing the dispersion of data points from the predicted model in the above graphs, it is clearly visible that the data points are less dispersed for SVR RBF kernel-based model than other models. Hence, SVR RBF kernel-based effort estimation model exhibits lower error value and higher prediction accuracy value.

VI. Comparison

On the basis of results obtained, the estimated effort using various SVR kernel methods are compared. While using the MMRE and PRED in evaluation, good results are implied by lower value of the MMRE and higher value of the PRED.

TABLE V: Comparison of Efforts obatined using various SVR kernel methods

	Actual Effort	SVR Linear Effort	SVR Polynomial Effort	SVR RBF Effort	SVR Sigmoid Effort
1	0.4615	0.3436	0.2332	0.4194	0.3567
2	0.7692	0.7698	1.1027	0.7865	0.8067
3	0.2637	0.2295	0.2059	0.2325	0.2090
4	1.0000	0.7030	1.9282	0.8908	0.7156
5	0.1538	0.1635	0.1691	0.1489	0.1841

Table. V shows the comparison of actual effort with an estimated effort by various SVR kernel methods for agile software on the five data taken for testing out of a data set of twenty one data.

TABLE VI: Comparison of errors and prediction accuracy values obtained using various SVR kernel methods

	Various SVR Kernel Methods	MMRE	PRED
1	SVR Linear Kernel	0.1492	90.8112%
2	SVR Polynomial Kernel	0.4350	68.7382%
3	SVR RBF Kernel	0.0747	95.9052%
4	SVR Sigmoid Kernel	0.1929	89.7646%

The Table VI displays the final comparison of MMRE and PRED values for different SVR kernel methods. The result shows that effort estimation using SVR RBF kernel-based model gives lower MMRE value and higher prediction accuracy value than those obtained using other SVR kernel methods.

VII. Conclusion

The story point approach is one of the effort estimation models that can be used for agile software's effort estimation. In this paper, first the total number of story points and project velocity are used to estimate the effort involved in developing an agile software product. The results obtained are optimized using four different support vector regression kernel methods. At the end of the study, the results generated are compared in order to access their accuracy. While comparing the results obtained using various SVR kernel methods, it can be concluded that RBF kernel-based support vector regression technique outperformed other three kernel methods. The computations for above procedure have been implemented, and outputs were generated using MATLAB. This approach can also be extended by applying other machine learning techniques such as SGB, Random Forest etc. on the story point approach.

References

[1] M. Fowler and J. Highsmith, "The agile manifesto," *Software Development*, vol. 9, no. 8, pp. 28–35, 2001.

[2] A. Schmietendorf, M. Kunz, and R. Dumke, "Effort estimation for agile software development projects," in *5th Software Measurement European Forum*, 2008.

[3] D. Cohen, M. Lindvall, and P. Costa, "An introduction to agile methods," *Advances in Computers*, vol. 62, pp. 1–66, 2004.

[4] S. M. Satapathy, M. Kumar, and S. K. Rath, "Fuzzy-class point approach for software effort estimation using various adaptive regression methods," *CSI Transactions on ICT*, vol. 1, no. 4, pp. 367–380, 2013.

[5] S. Keaveney and K. Conboy, "Cost estimation in agile development projects." in *ECIS*, 2006, pp. 183–197.

[6] E. Coelho and A. Basu, "Effort estimation in agile software development using story points," *development*, vol. 3, no. 7, 2012.

[7] Z. K. Zia, S. K. Tipu, and S. K. Zia, "An effort estimation model for agile software development," *Advances in Computer Science and its Applications*, vol. 2, no. 1, pp. 314–324, 2012.

[8] P. Hearty, N. Fenton, D. Marquez, and M. Neil, "Predicting project velocity in xp using a learning dynamic bayesian network model," *Software Engineering, IEEE Transactions on*, vol. 35, no. 1, pp. 124–137, 2009.

[9] A. L. Oliveira, "Estimation of software project effort with support vector regression," *Neurocomputing*, vol. 69, no. 13, pp. 1749–1753, 2006.

[10] E. Kocaguneli, T. Menzies, and J. W. Keung, "Kernel methods for software effort estimation," *Empirical Software Engineering*, vol. 18, no. 1, pp. 1–24, 2013.

[11] P. L. Braga, A. L. Oliveira, and S. R. Meira, "A ga-based feature selection and parameters optimization for support vector regression applied to software effort estimation," in *Proceedings of the 2008 ACM symposium on Applied computing*. ACM, 2008, pp. 1788–1792.