A Robust Pattern Based Re-engineering Model Guided by MODA and ELM for Software Testing Effort Estimation

Yogesh Kumar, Rahul Rishi

Abstract: ***Software Testing Effort (STE) plays a big role in code development method that highly contributes in complete development effort. Reducing the testing effort while not altering the standard/quality of the final code is always imperative; thus, STE measure is incredibly essential to conduct code testing method in associate economical manner. In this paper, a MODA aided Pattern based re-engineering (PBRE) model has been proposed for the selection of desirable number of projects with their respective features from within company and cross-company projects. The five input features selected by the MODA for Software Testing Effort (STE) estimation prior to development are Project Duration, Development Personnel, Test Cases, Function Points and Project Cost. We subjected the selected projects and features to train an ELM model for estimating STE using the k-fold cross validation approach. Outcomes shows that the anticipated model for estimating STE from cross-company projects and within-company projects yielded similar results to actual effort.***

Keywords: Software Testing Effort (STE), Multi-objective Dragonfly algorithm (MODA), Pattern based reengineering (PBRE), Extreme Learning Machine (ELM), Root Means Square Estimation (RMSE).

I. INTRODUCTION

Exposing all the deformities in the framework is the trade of software testing and also keeps tracking of imperfect items to evade such items to reach clients. Optionally, programming testing is additionally completed to persuade the client that the item fits in with and satisfies the particulars and the usefulness determined and consented to [1]. Programming Testing is perceived as a vital action in programming perfection.

The growth of the project effort is directly depends on the productivity of programmer or team and size of application. By using integrated development environments (IDE's), the productivity is depends on the flexibility, technological adaptability and capability of programmers. The main aspiration of software size estimation is to estimate the effort which used for software developing [12].

Any fault happens in the estimation which leads to either underestimation or overestimation for few software projects [2]. The usage of extreme incomes may disturb the completion time or schedule. Under estimating the growing effort can lead to cost, schedule overruns, poor quality software and under-staffing. Overestimates can yields in wasted resources and inefficiency as projects to extend to fit the estimated effort [13].

Revised Manuscript Received on 8 February 2019.

Yogesh Kumar, Ph.D. Research Scholar, UIET, M. D. University, Rohtak, Haryana, India.

Dr. Rahul Rishi, Professor, UIET, M.D. University, Rohtak, Haryana, India.

Calculating approximate effort for programming test cases is one of the key and essential tasks in software code Management [3]. Software code effort estimation practice is broadly segregated in four categories - theory-based, regression, empirical, and machine learning (ML) techniques. The theory-based Procedures in view of the basic hypothetical contemplations describe a few parts of programming advancement procedures [6]. Putnam's resource allocation model, COCOMO and the SLIM models are the examples of theory-based techniques. Empirical modus operandi includes analogy, function points (FP) and rule of thumb [4]. Regression scheme uses parametric and nonparametric forecasting models [5]. Well known regression techniques are Multiple Linear Regression (MLR), the Stepwise Regression (SR), the Poisson Regression, the Standard Regression, the Ordinary Least Squares (OLS), and the Multilayer Perceptron Model (MLP).

A Neural network is the most widely recognized estimation model strategy which is utilized as a choice to mean least squares relapse. To produce better results, these estimation models can be trained using historical data.

The rest of the paper is organized as follows: Section-II presents the proposed pattern based re-engineering model for STE estimation. Formulation of objectives based on proposed model is done in section-III in addition to the explanation of MODA and ELM. Section-IV relates to the Matlab environment and the explanation of results. Finally, section-V gives the conclusion of complete research work.

II. PROPOSED PATTERN BASED RE-ENGINEERING MODEL FOR STE ESTIMATION

In this paper we purposed a robust pattern based re-engineering model guided by Multi-objective Dragonfly algorithm and Extreme Learning Machine (MODA-ELM) for software testing effort estimation.

Re-engineering focuses to accomplish rigid and delicate quality requirements and destinations, for example, ''the reengineered framework must keep running as quick as the first'', or ''the new framework ought to be more effectively viable than the first". The Pattern-Based Re-engineering which has showed that it is feasible to detect patterns on a system in order to simplify its modelling and testing as it is feasible to define the corresponding test strategy prior to the exploration[11]. Hence, this work also aims to measure software testing effort (STE) to test the object oriented software projects using pattern based re-engineering model guided by MODA.

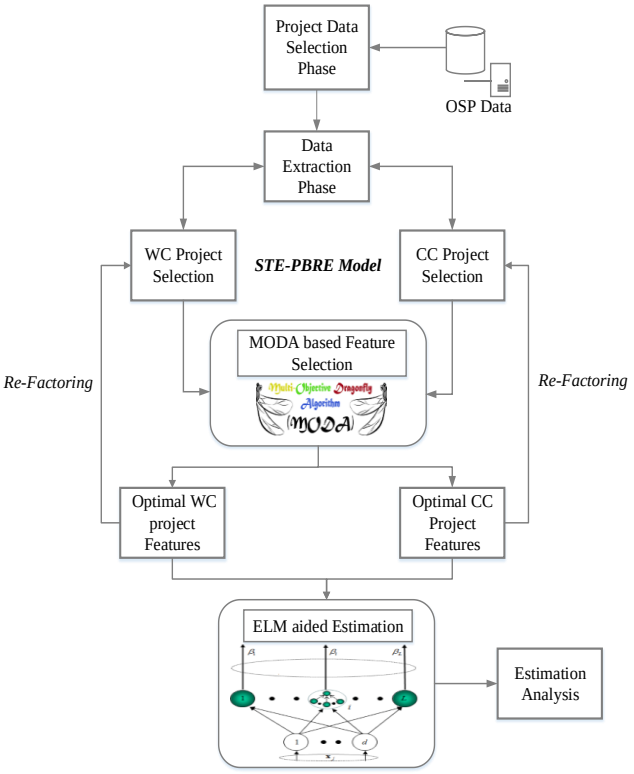

Figure 1 : Proposed PBRE Model for Software Effort Estimation

In this paper, initially we have to design pattern objectives based on software testing parameters such as core requirements, Risks and complexity, Team strength, Historical data, Resources availability. Moreover, re-engineering is permitted in developed design pattern model based on the type of object oriented software projects. Based on the formulation of design pattern objectives, a Multi-objective Dragonfly algorithm is employed to find the best trade-off between the design pattern objectives. The proposed multi-objective bio-inspired algorithm yields automatic adjustment with optimal pattern based re-engineering model for STE estimation. Finally, we conduct STE estimation on object oriented software projects based on the proposed optimal pattern based re-engineering model using the help of Hessian-Extreme Learning Machine (H-ELM). The proposed model will be implemented in MATLAB platform.

The proposed pattern based re-engineering model will be employed in an extensive scope of object oriented software project data sets. There is no earlier presumption needed to employ the proposed re-engineering based STE estimation model. This implies it is a steady, malleable, and versatile estimation display that is appropriate for bring into play in different kinds of object oriented programming ventures.

In the formulation of the main functions, we define a probability weight function, α_{ij} for each i^{th} project from the j^{th} company as shown below.

$$\alpha_{ij} = \frac{\left(\sum_{j=1}^{p} f_j\right)_i}{\sum_{i=1}^{n}\left(\sum_{j=1}^{p} f_j\right)_i} \quad (1)$$

Where p indicates the total number of f_j features for every project and n represents the total number of features from all the companies. Every probability weight is multiplied by the respective prior input feature. We incorporated two unique identifiers, λ_{ij} to uniquely label each i^{th} project from the j^{th} company and β_j to uniquely identify each of the companies.

III. OJECTIVES OF PROPOSED WORK, MODA AND ELM

3.1 Formulation Of Objectives

Two main objective functions are formulated for the STE problem. They are WC objective function and CC objective function. Every objective function is further categorized for the project and feature selection. All the objective functions are the minimization of project cost and the feature selection problems from the respective companies.

On the basis of Parkinson's Law a mathematical model is formulated for computing STE for every unsupervised project prior to the setting up of PBRE model. Here, unsupervised project refers to the dataset without STE.

$$STE = \left(\sum_{i=1}^{t} DDX \sum_{j=1}^{t} DP\right) \times TP\% \quad (2)$$

Where DD is the development duration in months; DP is the development personnel and TP is the testing proportion. To achieve consistency in the computation of the STE, for each unsupervised project dataset, we chose a threshold of 35% for TP. DD And DP are the parameters representing the duration for requirement gathering and total number of development personnel, design and testing respectively.

A. Project Selection based on WC Objective Function

First we have to define the cost function, $Cost^{WCP}$ in relation to project selection for WC as a cost minimization including the following design variables – DP, DD, Test Cases (TC), Function Points (FP), Probability Weight Function (α), Project Cost (PC), and the Project Identification Code (λ).

$$Cost^{WCP} = \alpha_{ij}\left(\sum_t \sum_n DP_n DD_t + \sum_u TC_u + \sum_i \sum_j FP_{ij} + \sum_c PC_c\right) + \sum_i \sum_j \lambda_{ij} \quad (3)$$

B. Project Selection based on CC Objective Function

The objective cost function, $Cost^{CCP}$ for the CC project selection is termed as follows.

$$Cost^{CCP} = \alpha_{ij}\left(\left(\sum_t \sum_n DP_n DD_t + \sum_u TC_u + \sum_i \sum_j FP_{ij} + \sum_c PC_c\right) + \sum_i \sum_j \lambda_{ij}\right) + \sum_j \beta_j \quad (4)$$

C. Feature Selection based on WC Objective Function

An expression for the objective function $Cost^{WCF}$ is given for the optimal subset of features for WC. The $Cost^{WCF}$ function incorporates the Akaike Information Criteria (AIC) for the optimal selection of features for the cross modelling process.

$$Cost^{WCF} = \underset{r,\delta,k\in\Re}{\arg}\left\{\min\left\{r\log\left(\hat{\delta}^2\right)+2k\right\}_f\right\}\lambda_{ij} \quad (5)$$

Where r is the number of records in the cross projects, $\hat{\delta}^2$ denotes the mean squared error, k represents the number of estimated parameters in AIC. Here, STE is contemplated in to the fact that the needy variable and the fair-minded factors are whatever is left of the capacities to be chosen. Therefore we constructed an ordinary least square regression and utilized AIC incorporating both forward and backward selection with an objective of choosing the optimal subset of features. AIC is very good at handling much more complex models and attains a better bias-variance trade-off.

D. Feature Selection based on CC Objective Function

The objective function, $Cost^{CCF}$ for the optimal subset of features for the CC approach is defined as,

$$Cost^{CCF} = \underset{r,\delta,k\in\Re}{\arg}\left\{\min\left\{r\log\left(\hat{\delta}^2\right)+2k\right\}_f\right\}\beta_j \quad (6)$$

E. Constraints for STE estimation

Three constraints are considered namely size of projects constraints, allocation of features and allocation of projects.

F. Projects Constraint

To reduce the cost, an inequality constraint is considered for the allocation of projects expressed as $\sum_i\sum_j X_{ij} < Y_N$ whereby not all the company projects can be chosen. The variable X_{ij} indicates the i^{th} project selected from the j^{th} company. Y_N denotes the total number of N projects from the companies.

We consider the inequality constraint, $\sum_i X_i \le X^*_{.j} < Y_N$ for the selection of projects from the company. $X^*_{.j}$ is the total number of projects from the given j^{th} company. We assume that at the worst case scenario, all the projects can be chosen from most of the companies but not from all the companies.

G. Features Constraint

At last, we consider an inequality constraint in the form of $\sum_i\sum_j\sum_p f_{ijp} < f_N$ to select all the features from the total projects. So, in reducing cost, the assumption is made that, there exist a subset of features (f_{ijp}) which will equally play an important role in estimating STE as compared to all the features (f_N). f_{ijp} represents the selected p^{th} feature from the i^{th} project chosen from the j^{th} company.

H. Size of Projects Constraint

We define two inequality constraints, $\sum_i\sum_j FP_{ij} \le FP_N$ and $\sum_i\sum_j\sum_p FP_{ijp} < FP_N$ for the project sizes in relation to FP. Here, we have to consider the sum of FP in the selected projects in the provided j^{th} company to be at most equal to the total number of FP in all the projects from that j^{th} company. On the other side, the sum of FP from the chosen projects was considered to be strictly lower than the total FP in all the projects from the selected companies.

3.2 Multi Objective Dragonfly Algorithm For Optimal Feature Selection

The modus operandi of multi-objective enhancement is incredibly unique in relation to mono target advancement. Consequently to take care of multi-objective issues, Pareto optimality based non-dominated arranging/sorting is utilized to distinguish arrangements of the next generation. So any decrease in time intricacy of the non-dominated arranging calculation will likewise enhance the execution of the MODA algorithm. The principle target of any swarm is continued existence; along these lines, every one of a swarm ought to be pulled in towards sustenance sources and occupied towards outward foes. Thinking about these two practices, there are five primary issues in the position refreshing of everyone in the swarms. These five constraints incorporate alignment, control cohesion, separation, attraction (towards sustenance sources), and distraction (towards outward adversaries) of everyone in the swarm [7].

Algorithm 1: Multi-objective Dragonfly Algorithm

1: Input:

2:

$p_m, m = 1,2,.....M.Multiobjective\ problem\ with\ Mobjectives$

$S_D = search\ space\ of\ D\ decision\ variables$

$ITR = Maximum\ Number\ of\ Iterations$

3: Output:

New Population

4: **Begin Procedure**

5: for i=1 to N

6: Initialize each solution randomly with in respective search space range

7: Apply nondominated sorting according to their pareto front number.

8: Apply control cohesion, alignment, separation, attraction (towards food sources), and distraction formulas

9: Evaluate Each Solution fitness

10: Apply selection and first keep good rank solution, in case of tie solution choose solution with higher crowding distance.
11: End for
12: **End Procedure**

3.3 Extreme Learning Machine (ELM) for STE prediction

Extreme Learning Machine works on the principle of a feed forward neural network for regression, distributed approximation, feature learning, compression, and for classification with one or multi layers of hidden nodes within which the parameters are indiscriminately generated and that they are needn't be modified. The hidden nodes are allotted in discriminately and are never updated. However they will be genetic from their ancestors. In several cases the output weights of hidden nodes are usually learned during a single layer that is crucial for learning a linear model.

The structural diagram of single hidden layer feed forward NN (SLFN) is shown in figure 2.

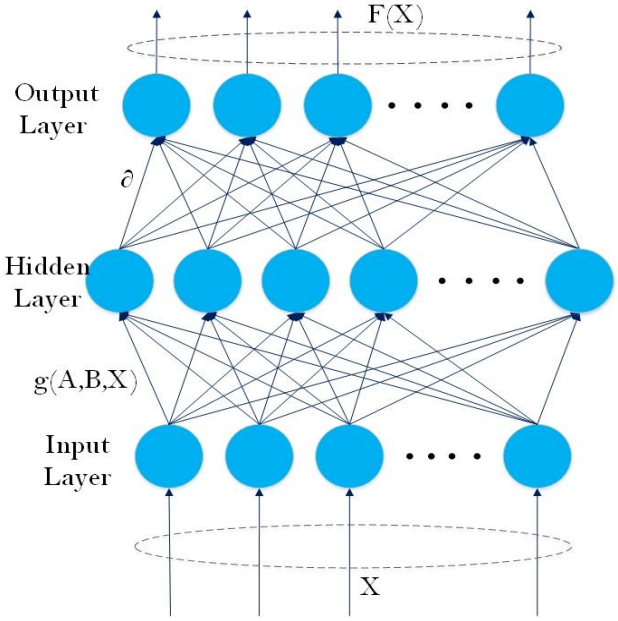

Figure 2 : Structural diagram of SLFN

The output function of SLFN is represented as $F_H(X)$.

$$F_H(X)=\sum_{i=1}^{H}\partial_i G_i(X)=\sum_{i=1}^{L}\partial_i g(A_i,B_i,X), X\in r^d, \partial_i \in r^m \quad (12)$$

Where, H represents the number of hidden nodes, G_i denotes the activation function of i^{th} hidden node, (A_i,B_i) are the parameters of the hidden layer. The number of input nodes is represented as d and the number of output nodes as m. ∂_i is the output weight which connects the hidden nodes and the output nodes.
The activation function for additive hidden nodes can be denoted as follows.

$$G_i=g(A_i,B_i,X)=G(A_iX+B_i), A_i\in r^d, B_i\in r \quad (13)$$

For Radial Basis Function, the activation function can be represented as

$$G_i=g(A_i,B_i,X)=G(B_i\parallel X-A_i\parallel), A_i\in r^d, B_i\in r^+$$

If the training set contains n number of samples $\{(X_i,T_i)\}_{i=1}^{n}$ where X_i can be represented as $X_i=[X_{i1},X_{i2},X_{i3},.....X_{id}]^P\in r^d$ and T_i can be formulated as $T_i=[T_{i1},T_{i2},T_{i3},.....,T_{im}]^P\in r^m$, then the SLFN can approximate the n samples with zero error and that can be denoted as follows.

$$\sum_{j=1}^{n}\parallel Y_j-T_j\parallel=0 \quad (14)$$

where, Y is the actual output of the SLFN.
Matrix form of the eqn,

$$\sum_{i=1}^{H}\partial_i g(A_i,B_i,X_j)=T_j,\ j=1,2,3,....,n \text{ is } Q\partial=P. \quad (15)$$

$$Q=\begin{bmatrix}q(X_1)\\q(X_2)\\q(X_3)\\.\\.\\.\\q(X_n)\end{bmatrix}=\begin{bmatrix}g(A_1,B_1,X_1).....g(A_H,B_H,X_1)\\g(A_1,B_1,X_2).....g(A_H,B_H,X_2)\\g(A_1,B_1,X_3).....g(A_H,B_H,X_3)\\.\\.\\.\\g(A_1,B_1,X_n).....g(A_H,B_H,X_n)\end{bmatrix}_{n\times H} \quad (16)$$

$$\partial=\begin{bmatrix}\partial_1^p\\\partial_2^p\\\partial_3^p\\.\\.\\.\\\partial_H^p\end{bmatrix}_{H\times m},\quad P=\begin{bmatrix}T_1^P\\T_2^P\\T_3^P\\.\\.\\.\\T_n^P\end{bmatrix}_{n\times m} \quad (17)$$

The main feature of Extreme Learning Machine is the parameters of the hidden layer are randomly created and kept fixedly without any change by the iteration. An important step in ELM is to find the output weights ∂_i. It is solved by reducing the training error and also the rule of the output weights. If the norms of weights and training error are smaller, then the generalization performance of SLFNs tends to get better.
The main function can be termed as

$$f=\min_{\partial}\parallel Q\partial-P\parallel^2+\mu\parallel\partial\parallel^2. \quad (18)$$

Finally based on the RMSE model the estimation will be carry out in view of error prediction percentage.

IV. RESULTS AND DISCUSSION

The performance of proposed MODA-ELM algorithm is evaluated based on 45 object oriented programming JAVA projects provided by 5 different companies denoted as C1, C2, C3, C4 and C5.The objective are formulated in view of within company(WC) and Cross Company(CC) basis.

The estimation of exertion for the most part relies upon the measure of the product which thusly is commonly founded on Lines of Codes or Function Points. ELM which has numerous points of interest over traditional feed forward back spread/propogation neural system (FFN) is picking up acknowledgment in numerous regions. It is guaranteed that it has better speculation ability and plan of ELM is simple and straight forward. These persuaded in this paper we need to apply ELM for programming testing exertion estimation.

Fundamental ELM will in general experience the ill effects of the nearness of immaterial factors in the informational indexes. So as to lessen the impact of such factors on the ELM demonstrate, RELM(regularized ELM) is introduced based on wrapper methodology around the original ELM, which incorporates a course of neuron positioning advance (by means of a L1 regularization), along with a criterion (L2 regularization) used to prune out the most irrelevant neurons of the model[8].

The weighted ELM(WELM) can manage information with unbalanced class dissemination while keep up the great execution on all around adjusted information as un-weighted ELM; the weighted ELM can be summed up to cost-delicate learning by doling out various loads for every precedent as indicated by clients' needs[9].

One popular variant of ELM is the Online Regularized ELM (OR-ELM), which can deal with sequential learning tasks. However, limitations exist in the ELM such as requiring the initialization phase, pre-defined important parameters, running into singularity problem, inconsistent and potentially unreliable performance [10].

For robust optimal feature extraction we employed MODA algorithm for both within company (WC) and Cross Company (CC) projects. Final, STE estimation is carried out through Extreme Learning Machine (ELM) classifier models such as basic ELM, Regularized ELM, WRELM and ORELM.The control parameters of MODA algorithm is illustrated in Table 1.

Table 1: Control Parameters of MODA

S.No	Parameter	Description	Value
1	Pop	Population Size	100
2	M	No. of objectives	Test Problem
3	N	No. of variables	Test Problem
4	ITR	No. of max iteration	100
5	AS	Archieve Size	100

In this paper, we mean to gather the exertion information from OSPs consequently. Thusly, we embrace the Automated Function Point (AFP) process into our investigation/study for estimating a product venture, which is reliable with the capacity point Counting Practices Manual (CPM) kept up by the International Function Point Users' Group (IFPUG). We can utilize it to consequently dissect the undertaking data and compute the capacity point size of a task with couple of setups portrayed in Figure 3.

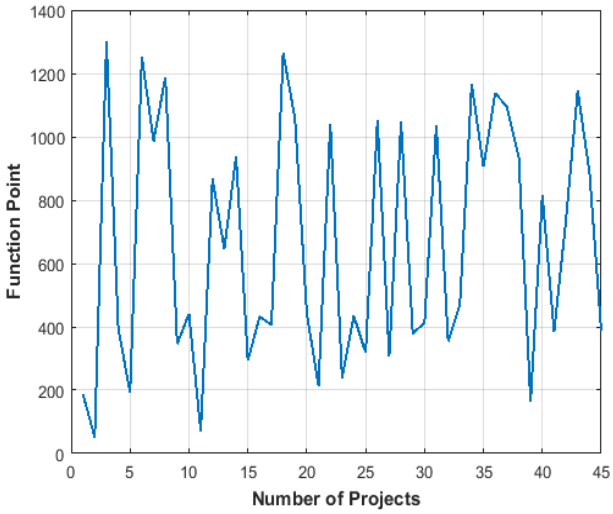

Figure 3 Function point analysis

For WC project selection based on the pattern model design variables such as DD, DP FP and PC. In this paper we consider five types of different company's data, based on the first objective our proposed MODA based algorithm performance is evaluated in view of size of the projects and algorithm convergence capability. Figure 4 depicted the convergence of five different companies with different project sizes.

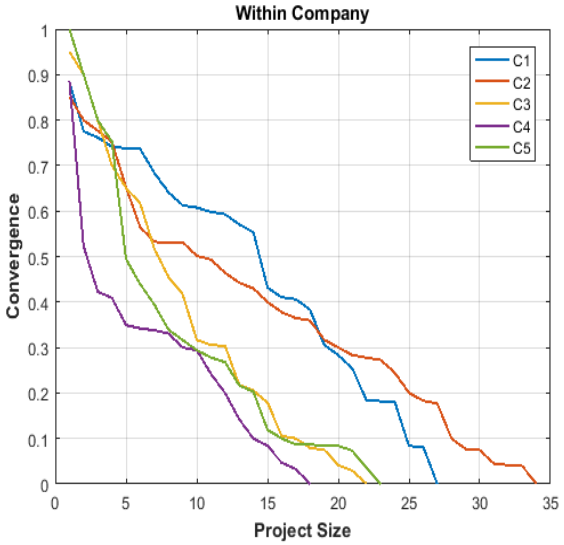

Figure 4 WC project convergence of MODA

The Convergence function incorporates the Akaike Information Criteria (AIC) for the optimal selection of features for the cross modelling process.

$$Covergence = \underset{r,\delta,k\in\Re}{\arg}\left\{\min\left\{r\log\left(\hat{\delta}^2\right)+2k\right\}_f\right\}\lambda_{ij}$$

Where r is the number of records in the cross projects, $\hat{\delta}^2$ denotes the mean squared error, k represents the number of estimated parameters in AIC.

For CC project selection based on the pattern model design variables such as DD, DP FP and PC. In view of CC based objective five types of different company's data are combined which provides ten types of cross company combinations, based on the second CC objective our proposed MODA based algorithm performance is evaluated in view of size of the projects and algorithm convergence capability. Figure 5 depicted the convergence of ten cross company combinations with different project sizes. Here C12 means the cross company projects between the company C1 and C2, and same is for other denotations.

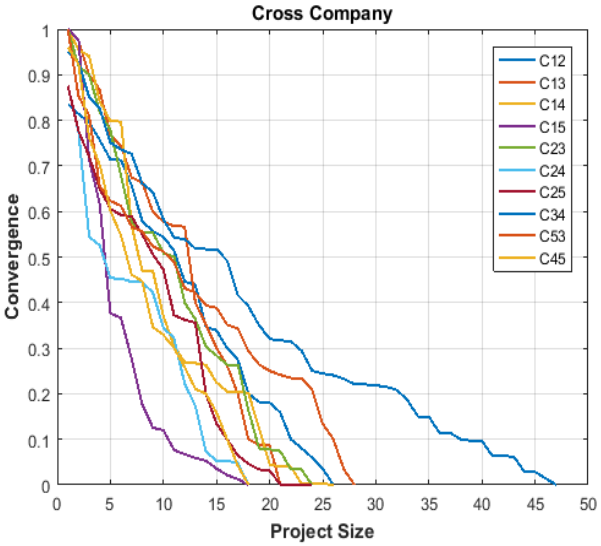

Figure 5 CC project convergence of MODA

Based on the second objectives our proposed MODA algorithm selects the best optimal set of feature using the help of AIC aided fitness formula. Afterwards, the set of optimal features our ELM classifier construct the training set for both WC and CC projects. For the purpose of desired output the STE for unknown project is estimated in predefined manner. The training set construction is over the ELM performing testing for selected WC and CC projects.

After applying the Mixed-Integer Linear Programming (MILP) to the project datasets, a subset of 30 projects and five features were selected. These features were project duration in months (PD), number of development personnel (DP), number of test cases (TC), number of function points (FP) and the cost of each project (PC) in USD.

In order to compare the evaluation performance of STE estimated from both WC and CC, we find the RMSE accuracy measure through the guidance of ELM with various parameters.

RMSE (Root Mean Square Error):

$$\sqrt{\frac{\sum_{i=1}^{n}(O_i - P_i)^2}{N}}$$

Where,

O_i , P_i - Observed & Predicted Value

Initially for WC RMSE estimation we utilized training set features for STE estimation. In this paper we consider fifth company project as a testing dataset in order to ensure the efficiency of the proposed PBRE model. Figure 6 shows the RMSE estimation of WC projects based on test features.

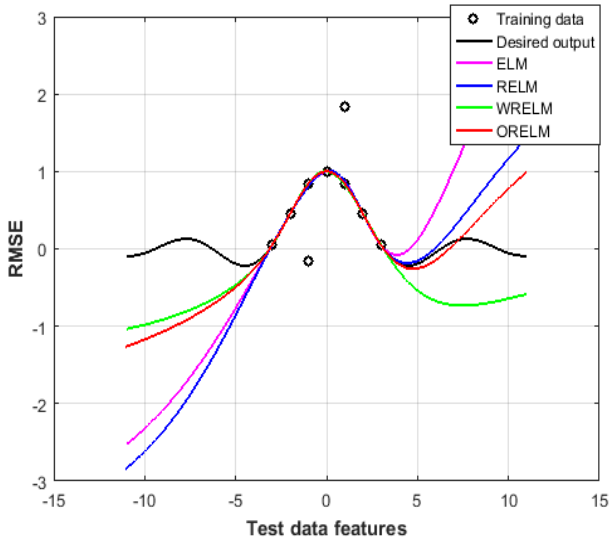

Figure 6 RMSE estimation of WC features using ELM

Figure 7 shows the RMSE estimation of CC projects based on test features.

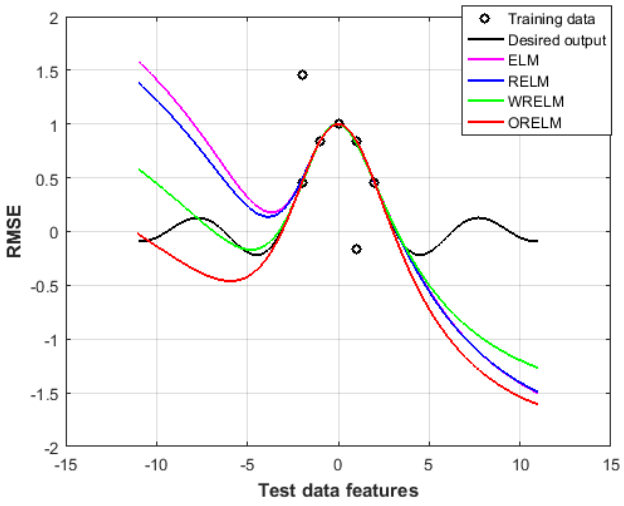

Figure 7 RMSE estimation of CC features using ELM

As per our evaluation ORELM model predict the RMSE efficiently for both WC and CC projects.

This means that, our proposed MODA based optimization selection framework can select optimal prior features which yield similar STE results based RMSE estimation. Without the optimal feature selection using MODA, we realized that RMSE estimation of both WC and CC is not sufficient for STE estimation. Thus, results from our optimization selection framework and ELM estimation modelling approach reveal that, models constructed from WC and CC yield approximately similar STE results when datasets were subjected WC and CC manner.

V. CONCLUSION

A MODA aided PBRE model has been proposed for the selection of desirable number of projects with their respective features from within company and cross-company projects. The five input features selected by the MODA for Software Testing Effort (STE) estimation prior to development are Project Duration, Development Personnel, Test Cases, Function Points and Project Cost. We subjected the selected projects and features to train an ELM model for estimating STE using the k-fold cross validation approach. Outcomes of this research show that the proposed model for estimating STE from cross-company projects and within-company projects yielded similar results to actual effort.

REFERENCES

1. Chemuturi M, "Mastering software quality assurance: best practices, tools and techniques for software developers", 2010.
2. Bardsiri VK, Jawawi DN, Hashim SZ, Khatibi E, "Increasing the accuracy of software development effort estimation using projects clustering", IET software,Vol.6,No.6,pp.461-473,2012.
3. Benestad HC, Anda B, Arisholm E, "Understanding cost drivers of software evolution: a quantitative and qualitative investigation of change effort in two evolving software systems", Empirical Software Engineering, Vol.15, No.2, pp.166-203, 2010.
4. Pai DR, McFall KS, Subramanian GH, "Software effort estimation using a neural network ensemble", Journal of Computer Information Systems, Vol.53, No.4, pp.49-58, 2013.
5. Jorgensen M, Shepperd M, "A systematic review of software development cost estimation studies", IEEE Transactions on software engineering, Vol.33, No.1, pp.33-53, 2007.
6. Jorgensen M, Shepperd M, "A systematic review of software development cost estimation studies", IEEE Transactions on software engineering, Vol.33, No.1, pp.33-52, 2007.
7. Seyedali Mirjalili, "Dragonfly algorithm : a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems", Neural Computing and Application, Vol. 27, Issue 4, pp 1053-1073, May-2016.
8. Hieu, Trung, Huynh, Yonggwan and Won, "Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks", Pattern Recognition Letters, Volume 32, Issue 14, 15 October 2011, Pages 1930-1935.
9. WeiweiZong, Guang-BinHuang and Yiqiang Chen, "Weighted extreme learning machine for imbalance learning", Neurocomputing, Volume 101, 4 February 2013, Pages 229-242.
10. Zhifei, Shao and Meng Joo, "An online sequential learning algorithm for regularized Extreme Learning Machine", Neurocomputing, Volume 173, Part 3, 15 January 2016, Pages 778-788.
11. Yogesh Kumar, Rahl Rishi, "Dragonfly algorithm guided extreme learning machine based prediction model for software testing effort estimation", in Journal of advanced research in dynamical and control system, Special Issue-07, 2018. Pp. 1948-1958.
12. Yogesh Kumar, "Comparative analysis of software size estimation techniques in project management", in International journal for research in applied science & engineering technology, Vol. 5, Issue VIII, Aug-2017. Pg 1470-1477.
13. Tannu, Yogesh Kumar, "Comparative Analysis of Different Software Cost Estimation Methods", International Journal of Computer Science and Mobile Computing, Volume 3, Issue 6, 04 July 2014, pg.547-557.