Multi-Objective Optimization for Software Testing Effort Estimation

Solomon Mensah[1], Jacky Keung[1], Kwabena Ebo Bennin[1] and Michael Franklin Bosu[2]
[1]Department of Computer Science, City University of Hong Kong, Hong Kong, China
{smensah2-c, kebennin2-c}@my.cityu.edu.hk, Jacky.Keung@cityu.edu.hk
[2]Centre for Business, Information Technology and Enterprise
Wintec, Hamilton, New Zealand
michael.bosu@wintec.ac.nz

Abstract— **Software Testing Effort (STE), which contributes about 25-40% of the total development effort, plays a significant role in software development. In addressing the issues faced by companies in finding relevant datasets for STE estimation modeling prior to development, cross-company modeling could be leveraged. The study aims at assessing the effectiveness of cross-company (CC) and within-company (WC) projects in STE estimation. A robust multi-objective Mixed-Integer Linear Programming (MILP) optimization framework for the selection of CC and WC projects was constructed and estimation of STE was done using Deep Neural Networks. Results from our study indicate that the application of the MILP framework yielded similar results for both WC and CC modeling. The modeling framework will serve as a foundation to assist in STE estimation prior to the development of new a software project.**

Keywords- Software Testing Effort; Cross-Company; Within-Company; Optimization; Deep Neural Networks

I. INTRODUCTION

Software Testing Effort (STE) is one of the critical phases in software development as it is estimated to contribute about 25-40% [1] of the total development cost. STE is the effort in relation to duration, resources and source code for testing the software before deployment to the client or market [1]. The acquisition of relevant data for effort estimation has been a focus for prior work [2] concerning the effective use of within-company (WC) and cross-company (CC) datasets for predictive modeling. Researchers and practitioners who lack the relevant local data rely on data imported from other repositories or companies. The challenge is that this approach might not always be useful for WC estimation. In order for WC estimation to benefit from CC datasets, the selection of an optimal "mix" of project datasets and features is required for efficient estimation.

The WC and CC modeling have been considered in the domains of software development effort estimation and defect prediction [2][4]. To the best of our knowledge, this is the first study to employ the use of CC projects in the estimation of STE. A study by Burak et al. [4] indicates that models that used CC datasets for defect prediction yielded similar performance to WC models when normalization and Nearest Neighbor filtering techniques were applied to the datasets.

In this study, we incorporated a *z*-score normalization technique into our proposed project selection approach after it proved superior to log transformation and box cox transformation.

The two research questions addressed by this study are as follows: ***RQ1.*** Do models constructed from CC projects yield similar STE results to models from WC projects with/without normalization? ***RQ2.*** Does STE estimated from "prior" features yield similar results to STE estimated from "posterior" features?

In order to minimize the cost (effort) associated with project and feature selection in STE estimation, this study proposes a multi-objective Mixed-Integer Linear Programming (MILP) optimization framework. The selection of an optimal "mix" of N^* projects with their respective f^* features from m companies to estimate STE is subjected to the following constraints: 1) allocation of projects constraint; 2) allocation of features constraint; 3) size of project constraint.

The rest of the paper is organized as follows: Section II describes the related work. Section III presents the methodology. Section IV describes the results from the study. Finally, Section V presents the conclusion and future work.

II. RELATED WORK

Bareja and Singhal [1] studied various effort estimation techniques in order to minimize STE. They realized that machine learning and data mining testing techniques adopted by developers assist in the reduction of effort expended in the software development process. A study by Turhan et al. [4] investigated how cross-company (CC) projects can be used for defect prediction modeling. They found that within-company (WC) modeling performed better than CC modeling. In the field of effort estimation, Kocaguneli and Menzies [2] in the quest for finding relevant dataset for effort estimation concluded that there is little significant difference in the use of CC or WC datasets for modeling. Ansari et al. [7] proposed a regression test case optimization approach to assist in the reduction of error cost and testing time to achieve a more quality software product. Their test selection approach makes use of prioritized test cases for testing riskier components of the product in order to enhance system reliability and stability.

This study differs from previous works since to the best of our knowledge this is the first study to apply a multi-objective MILP for project and feature selection for STE estimation.

DOI reference number: 10.18293/SEKE2016-017

III. Methodology

A. Datasets

For the purpose of this study, 45 *PHP* projects from 5 software development companies were used. We present the projects with the means of their respective sampled features - Function Points (FP), Development Duration (DD) in months, Lines of Codes (LOC) and STE in person-months in Table I. Other features from the projects are Number of Development Personnel (DP), Test Cases (TC), Project Cost (PC), number of defects, user stories and marketers. For confidentiality reasons, we denote the names of the companies with $\boldsymbol{X_j}$ and their respective project names as $\boldsymbol{X_{ij}}$.

TABLE I. Descriptive Summary of Company Projects

Company	No. of Projects	Mean			
		FP	DD	LOC	STE
X_1	15	47.2	13.7	14589	151.7
X_2	7	44.4	8.4	3942	25.0
X_3	4	301.0	16.0	719786	123.3
X_4	4	58.8	4.8	14654	18.6
X_5	15	238.0	7.3	65543	25.5

B. Within-Company and Cross-Company Modeling

Given $\{\boldsymbol{X_1, X_2 \ldots X_m}\}$ denoting a set of $\boldsymbol{m}$ companies and each $\boldsymbol{X_j}$ consisting of a set of $\{\boldsymbol{X_{1j}, X_{2j} \ldots X_{mj}}\}$ projects, then we formulate a within-company (WC) project modeling as a combination of selected $\boldsymbol{X_{ij}}$ projects from a single $\boldsymbol{X_j}$ company. Similarly, cross-company (CC) project modeling as a combination of projects from a number of companies. In each case of the modeling, the selected projects by the MILP were used for constructing the predictive model for estimating the STE of a target project.

Based on findings by Turhan et al. [4], we also found the need of preprocessing datasets prior to model construction. We applied a 3-step preprocessing approach. 1) Data Cleaning: In order to assess if the datasets were normally distributed, we made use of a graphical analytical technique namely Quantile-Quantile (QQ) plot. It should be noted that, no missing values were observed in the datasets used. 2) Data Transformation: Because of variations within the datasets, we used the *z*-score normalization to transform the data points into specific range in order to leverage the outweighing discrepancies. 3) Data Reduction: The MILP framework did not select three projects and this was due to the weak correlation between the independent features and the dependent feature (STE).

C. Proposed MILP Algorithm

The MILP optimization framework is formulated for the project and feature selection. Deep Neural Network (DNN) is incorporated into the framework for estimating the STE. A 7-step procedure of the proposed framework is presented below.

Step 1: Preprocess project datasets.

Step 2: Given *m* companies each with an archival set of projects, construct the MILP for the project selection.

Step 3: For every selected optimal set of X_i^* projects from each j^{th} company, compute the multivariate Spearman rank correlation between the respective independent features and the dependent feature (STE).

Step 4: Select the projects based on the project selection optimization method and based on the higher correlation values in a non-increasing order until the optimal number of projects for every j^{th} company is achieved.

Step 5: Combine the selected projects from each j^{th} company. For the given project datasets applied, it was assumed that each project has similar set of features for estimating STE.

Step 6: Select the features based on the feature selection optimization method.

Step 7: Construct both within-company and cross-company STE models in the domain of DNN and evaluate the performance with MdMRE, a robust metric free from outliers [9].

D. Optimization Model Formulation

A generalized optimization model formulation specifically MILP is composed of three main components namely; the objective function, the constraints and decision variable bounds. The optimization model minimizes or maximizes an objective function subject to certain constraints, which can be in the form of equality or inequality constraints. The decision variable (normally independent) is said to be discrete if it can assume a finite set values and continuous if it can assume values within a specified interval. Optimization problems with both discrete and continuous decision variables are said to be MILP. Based on Mridul and Rana [6] categorization of software size into functional and technical size, we constructed the generalized objective function for the MILP incorporating the functional size in the form of FP. Thus, since our ultimate aim is to estimate STE prior to development, we eliminated LOC which is a technical size from developers' perspective and made use of FP from users' perspective together with other prior input features - estimated project duration, estimated project cost and number of development personnel. FP is defined as the sum of five components – external inputs, external outputs, external query, external interface files and internal logical files with their respective weights [6]. Posterior features from the projects include LOC, coding time, test cases and number of defects.

In the formulation of the objective functions, we define a probability weight function, α_{ij} for each i^{th} project from the j^{th} company as shown in (1).

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

where $\boldsymbol{p}$ denotes the total number of $\boldsymbol{f_j}$ features for every project and $\boldsymbol{n}$ denotes the total number of features from all companies. Each of the probability weights is multiplied by the respective prior input features. We incorporated two unique identifiers, $\boldsymbol{\lambda_{ij}}$ to uniquely label every i^{th} project from the j^{th} company and $\boldsymbol{\beta_j}$ to uniquely identify each of the companies as shown in (3) and (4).

Objective Function

In this study, we formulated two main objective functions for the MILP problem – WC and CC objective functions. Each objective function is further categorized for project and feature selection. All objective functions are minimization of project cost and feature selection problems from the respective companies.

A mathematical model based on Parkinson's Law [5] is formulated in (2) for computing the STE for each unsupervised project prior to the setting up of the MILP model. Here, unsupervised project refers to dataset without STE.

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

where DD = Development Duration in months; DP = Development Personnel; TP = Testing Proportion. For consistency in the computation of the STE for each unsupervised project dataset, we chose a threshold of 40% for TP [1]. DP and DD are parameters denoting the total number of development personnel and duration for requirement gathering, design and testing respectively.

WC Objective Function for Project Selection

We first define the cost function, *Cost*^{WCP} in relation to project selection for WC in (3) as a cost minimization involving the following design variables – DD, DP, Function Points (FP), Test Cases (TC), Project Cost (PC), Probability Weight Function (α), Project Identification Code (λ)

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

CC Objective Function for Project Selection

The objective cost function, *Cost*^{CCP} for the CC project selection is defined in (4).

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

WC Objective Function for Feature Selection

We define an expression for the objective function, *Cost*^{WCF} for the optimal subset of features for WC in (5). The *Cost*^{WCF} function incorporates the Akaike Information Criteria (AIC) for the optimal selection of features for the cross modeling process.

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

where r = number of records in the cross projects, $\hat{\delta}^2$= mean squared error, k = number of estimated parameters in AIC. Here, we consider STE as the dependent variable and independent variables are the rest of the features to be selected. We therefore constructed an ordinary least squares regression and used AIC incorporating both forward and backward selection with the aim of selecting the optimal subset of features. AIC is very good at handling much more complex models and achieves a better bias-variance tradeoff [10].

CC Objective Function for Feature Selection

The objective function, *Cost*^{CCF} for the optimal subset of features for the CC approach is defined in (6).

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

Constraints

We considered three constraints namely; allocation of projects, allocation of features and size of projects constraints.

Allocation of Projects Constraint

In order to minimize cost, we considered an inequality constraint for the allocation of projects defined as $\sum_i \sum_j X_{ij} < Y_N$ whereby not all the company projects can be selected. The variable X_{ij} denotes the i^{th} project selected from the j^{th} company. Y_N denotes the total number of N projects from the companies.

Secondly, we considered the inequality constraint, $\sum_i X_i \leq X_{\cdot j}^* < Y_N$ for the selection of projects from within-company. $X_{\cdot j}^*$ denotes the total number of projects from a given j^{th} company. We assumed that at the worst case scenario, all projects can be selected from most of the companies but not all companies.

Allocation of Features Constraint

We define an inequality constraint for the allocation of features in terms of Variance Inflation Factor (VIF) to deal with the multicollinearity issues among the predictor (independent) features. After experimental fine tuning of the parameters, we considered multicollinearity as an issue if VIF is more than a specified threshold, k = 10 in each AIC model. Hence, for optimal predictor features to be obtained we needed correlation between predictor features to be very minimal.

An adjusted R^2 of 0.7 was chosen after parameter fine tuning. Hence, for optimal feature subset, we needed at least 70% of the total variation in STE to be explained by the predictor features.

Lastly, we considered an inequality constraint in the form of $\sum_i \sum_j \sum_p f_{ijp} < f_N$ in order not to select all features from the total projects. Thus, in minimizing cost, we made the assumption that, there exist a subset of features (f_{ijp}) that will equally play a significant role in estimating STE as compared to all features (f_N). f_{ijp} denotes the selected p^{th} feature from the i^{th} project selected from the j^{th} company.

Size of Projects Constraint

We define two inequality constraints, $\sum_i \sum_j FP_{ij} \leq FP_N$ and $\sum_i \sum_j \sum_p FP_{ijp} < FP_N$ for the project sizes in relation to FP [6]. Here, we considered the sum of FP in the selected projects in a given j^{th} company to be at most equal to the total number of FP in all projects from that j^{th} company. On the other hand, the sum of FP from the selected projects was considered to be strictly less than the total FP in all the projects from the selected companies.

E. Deep Neural Networks (DNN) Model

In the estimation of STE for the selected projects using their respective prior and posterior features, we made use of DNN. DNN was considered for the within-company (WC) and cross-company (CC) modeling approach since it makes use of multiple layers to automatically learn from a set of features and gives better predictive results [8]. After series of fine tuning of network architecture parameters, we considered the DNN for the STE estimation to be best at 3 hidden layers with 5, 2 and 1 neuron(s) respectively and an output layer with a single neuron. The Levenberg-Marquardt backpropagation optimization training function was used to update the weights and the hyperbolic tangent activation function was used in each of the neurons for giving the respective outputs. We employed the *k*-fold cross validation approach for setting up the DNN model. The formation of the training set and test set differ slightly for the CC and WC projects. For CC modeling, we used all projects from four of the five companies to form the training set whilst the fifth company project formed the test set. This is repeated in a leave-one-out (LOO) cross validation manner till all projects from the companies were part of the training set and test set respectively. For WC modeling, we used all but one project from a single company to form the training set whilst the remaining project formed the test set using the LOO method similar to the CC modeling. We then used the Median Mean Relative Error (MdMRE) [9] in evaluating the DNN model. Experiment was conducted in MATLAB toolkit (version R2014b).

IV. Results and Discussion

A. Project and Feature Selection by MILP

After applying the MILP to the project datasets, a subset of 42 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.

RQ1: Do models constructed from CC projects yield similar STE results to models from WC projects with/without normalization?

In order to compare the evaluation performance of STE estimated from both approaches, we used the MdMRE accuracy measure [9] and presented results in Tables II-III. The results in relation to normalized and un-normalized datasets are presented using the win/tie/loss metric. The win/tie/loss metric enabled us to make an evaluation and comparative performance using MdMRE for the within-company (WC) and cross-company (CC) modeling [2]. For example, in Table II, in relation to MdMRE evaluation, we realized that WC and CC modeling with *z*-score normalization yielded similar predictive results in 3 cases (that is 3 ties in the 2nd, 3rd and 5th cases where one project from X_4, X_3 and X_1 were used for testing respectively). Without the normalization technique, we realized that WC dominated in estimating STE (Table III). Thus, results from our optimization selection framework and DNN estimation modeling approach reveal that, models constructed from WC and CC yield approximately similar STE results when datasets were subjected to the *z*-score normalization technique.

TABLE II. MDMRE EVALUATION (NORMALIZED)

CC Train set	WC Train set	Test set	Win	Tie	Loss
X_{i1}, X_{i2}, X_{i3}, X_{i4}	$X_{i-1,5}$	LOO	WC		CC
X_{i1}, X_{i2}, X_{i3}, X_{i5}	$X_{i-1,4}$	LOO		✓	
X_{i1}, X_{i2}, X_{i4}, X_{i5}	$X_{i-1,3}$	LOO		✓	
X_{i1}, X_{i3}, X_{i4}, X_{i5}	$X_{i-1,2}$	LOO	WC		CC
X_{i2}, X_{i3}, X_{i4}, X_{i5}	$X_{i-1,1}$	LOO		✓	

TABLE III. MDMRE EVALUATION (UN-NORMALIZED)

CC Train set	WC Train set	Test set	Win	Tie	Loss
X_{i1}, X_{i2}, X_{i3}, X_{i4}	$X_{i-1,5}$	LOO	WC		CC
X_{i1}, X_{i2}, X_{i3}, X_{i5}	$X_{i-1,4}$	LOO		✓	
X_{i1}, X_{i2}, X_{i4}, X_{i5}	$X_{i-1,3}$	LOO	WC		CC
X_{i1}, X_{i3}, X_{i4}, X_{i5}	$X_{i-1,2}$	LOO	WC		CC
X_{i2}, X_{i3}, X_{i4}, X_{i5}	$X_{i-1,1}$	LOO		✓	

LOO – Leave one out; WC – within company; CC – cross company

RQ2: Does STE estimated from "prior" features yield similar results to STE estimated from "posterior" features?

Here, we compared performance of STE estimation from both "prior" and "posterior" features with respect to WC and CC modeling. Due to space limitation, the MdMRE evaluation of results are presented in Tables IV-V available in the link[1]. In relation to cross-company (CC) modeling approach, the average MdMRE value was 79.7%. In relation to within-company (WC), the average MdMRE value was 82.0%. Result from Table IV shows that, in relation to the application of the normalization technique, STE estimated from CC modeling yielded similar results for both prior and posterior features. We further confirmed this result using Friedman's test statistic [3] which yielded *p*-values of 0.4240 and 0.0597 at 5% significance level for the prior and posterior features respectively. This indicates that there is no statistical difference in the STE estimation results from the prior and posterior features. In the WC modeling, prior features dominated best in the STE estimation as illustrated in Table V.

This means that, our proposed multi-objective MILP optimization selection framework can select optimal prior features which yield similar STE results as compared to posterior features provided a *z*-score normalization is applied.

V. CONCLUSION AND FUTURE WORK

A Mixed-Integer Linear Programming (MILP) optimization framework 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 MILP 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 a Deep Neural Network (DNN) model for estimating STE using the *k*-fold cross validation approach. Results show that the DNN model for estimating STE from cross-company projects yielded similar results to within-company projects provided that the *z*-score normalization method was applied.

Going forward, we intend to incorporate a filter in our MILP framework to further improve the CC project and feature selection approach for STE estimation.

ACKNOWLEDGEMENT

This research is supported by the City University of Hong Kong research funds (Project No. 7200354, 7004222, 7004474).

REFERENCES

[1] K. Bareja and A. Singhal. "A Review of Estimation Techniques to Reduce Testing Efforts in Software Development." *Advanced Computing & Communication Technologies (ACCT), 2015 Fifth International Conference on*. IEEE, 2015.

[2] E. Kocaguneli and T. Menzies. "How to find relevant data for effort estimation?." *Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on*. IEEE, 2011.

[3] M. Friedman. "The use of ranks to avoid the assumption of normality implicit in the analysis of variance." *Journal of the american statistical association* 32.200 (1937): 675-701.

[4] B. Turhan, T. Menzies, A. B. Bener and J. D. Stefano. "On the relative value of cross-company and within-company data for defect prediction." *Empirical Software Engineering* 14.5 (2009): 540-578.

[5] C. N. Parkinson. *Parkinson's law, and other studies in administration*. Vol. 24. Boston: Houghton Mifflin, 1957.

[6] B. Mridul and A. Rana. "Impact of Size and Productivity on Testing and Rework Efforts for Web-based Development Projects." ACM SIGSOFT Software Engineering Notes 40.2 (2015): 1-4.

[7] A. S. A. Ansari, K. K. Devadkar and P. Gharpure. "Optimization of test suite-test case in regression test." *Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on*. IEEE, 2013.

[8] K. Sangwook, M. Lee and J. Shen. "A novel deep learning by combining discriminative model with generative model." *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015.

[9] T. Foss, E. Stensrud, B. Kitchenham and I. Myrtveit. "A simulation study of the model evaluation criterion MMRE." Software Engineering, IEEE Transactions on 29.11 (2003): 985-995.

[10] D. Ruppert. *Statistics and finance: An introduction*. Springer Science & Business Media, 2004.

[1] MdMRE evaluation results - http://tinyurl.com/WinTieLoss