Do Not Take It for Granted: Comparing Open-Source Libraries for Software Development Effort Estimation

This document contains additional information for the article "Do Not Take It for Granted: Comparing Open-Source Libraries for Software Development Effort Estimation" currently under review.

I. Dataset Details

To empirically investigate our RQs we used the largest SEE publicly available datasets (namely, China, Desharnais, Kitchenam, Maxwell, Miyazaki) containing a diverse sample of industrial software projects developed by a single company or several software companies [1]. These datasets exhibit a high degree of diversity both in terms of number of observations (from 48 to 499), number and type of features (from 3 to 17), technical characteristics (e.g., software projects developed in different programming languages and for different application domains), number of companies involved and their geographical locations. Furthermore, all these datasets have been widely used in several SEE studies (see e.g., [2, 3, 4, 5, 6, 7, 8, 9]). A description of each of these datasets can be found below, while in Table I we report for each of the datasets its type (Within-Company –WC– or Cross-Company –CC– [10]), number of projects, and the descriptive statistics of the dependent variable (i.e., Effort) and the independent variables used to build the prediction models.

The **China** dataset [11] consists of a total of 499 projects developed by various Chinese companies. The independent variables comprise of the basic elements used to calculate Function Points (i.e., Input, Output, Inquiry, File, Interface) whereas the dependent one is the variable Effort.

The **Desharnais** dataset [12] contains up to 81 software projects acquired from a Canadian software company. We considered the dependent variable as the total effort not the length of the code. We also excluded the categorical variables (i.e., Language and YearEnd) and four projects that have missing values from our analysis, as suggested and applied in previous work (e.g., [2, 13, 8]). To this end, we used the following variables as independent ones: TeamExp (i.e., the team experience measured in years), ManagerExp (i.e., the manager experience measured in years), Entities (i.e., the number of the entities in the system data model), Transactions (i.e., the number of basic logical transactions in the system), AdjustedFPs (i.e., the Adjusted Function Points).

The **Kitchenham** dataset [14] comprises of a total of 145 maintenance and development industrial projects managed by a single outsourcing company, including effort estimates and actuals (dependent variable), and function points count (independent variable). The estimates were made as part of the company's standard project estimating process that involved producing two or more estimates for each project and selecting one estimate to be the basis of the client-agreed budgets.

The **Maxwell** dataset [15] consists of 62 industrial software projects developed for one of the biggest commercial banks in Finland. We use 17 of features available, namely Function Points (SizeFP) and 16 ordinal variables, i.e., number of different development languages used (Nlan), customer participation (T01), development environment adequacy (T02), staff availability (T03), standards used (T04), methods used (T05), tools used (T06), software's logical complexity (T07), requirements volatility (T08), quality requirements (T09), efficiency requirements (T10), installation requirements (T11), staff analysis skills (T12), staff application knowledge (T13), staff tool skills (T14), and staff team skills (T15). Whereas, for the Desharnais dataset, none of the categorical variables were used.

The **Miyazaki** dataset [16] consists of 48 industrial software projects developed by a total of 20 different software companies of the Fujitsu Large Systems Users Group. For this dataset, we considered the following independent variables: SCRN (i.e., the number of different input or output screens), FORM (i.e., the number of different report forms), and FILE (i.e., the number of different record format). Similarly to the aforementioned datasets, the dependent variable used is Effort, defined as the number of person-hours needed from system design to system test, including indirect effort such as project management.

II. Machine Learners and Their Settings

Table II and Table IV report the settings used in our empirical study for the *out-of-the-box-ml* scenario and *tuned-ml* scenario, respectively.

TABLE I: Datasets used in our study.

Dataset	Type	Variable	Min	Max	Mean	Std. Dev.
China	CC	Input	0.00	9404.00	167.10	486.34
(499 projects)		Output	0.00	2455.00	113.60	221.27
		Enquiry	0.00	952.00	61.60	105.42
		File	0.00	2955.00	91.23	210.27
		Interface	0.00	1572.00	24.23	85.04
		Effort	26.00	54620 .00	3921.00	6481.00
Desharnais	WC	TeamExp	0.00	4.00	2.30	1.33
(77 projects)		ManagerExp	0.00	4.00	2.65	1.52
		Entities	7.00	386	121.54	86.11
		Transactions	9.00	661.00	162.94	146.09
		AdjustedFPs	73.00	1127.00	284.48	182.26
		Effort	546.00	23490.00	4903.95	4188.19
Kitchenham	CC	AFP	15.36	18140	527.70	1521.99
(145 projects)		Effort	219.00	113900.00	3113.00	9598.00
Maxwell	CC	SizeFP	48.00	3643.00	673.31	784.04
(62 projects)		Nlan	1.00	4.00	2.55	1.02
		T01	1.00	5.00	3.05	1.00
		T02	1.00	5.00	3.05	0.71
		T03	2.00	5.00	3.02	0.89
		T04	2.00	5.00	3.19	0.70
		T05	1.00	5.00	3.05	0.71
		T06	1.00	4.00	2.90	0.69
		T07	1.00	5.00	3.24	0.90
		T08	2.00	5.00	3.81	0.96
		T09	2.00	5.00	4.06	0.74
		T10	2.00	5.00	3.61	0.89
		T11	2.00	5.00	3.42	0.98
		T12	2.00	5.00	3.82	0.69
		T13	1.00	5.00	3.06	0.96
		T14	1.00	5.00	3.26	1.01
		T15	1.00	5.00	3.34	0.75
		Effort	583.00	63694.00	8223.20	10500.00
Miyazaki	CC	SCRN	0.00	281.00	33.69	47.24
(48 projects)		FORM	0.00	91.00	22.38	20.55
		FILE	2.00	370.00	34.81	53.56
		Effort	896.00	253760.00	13996.00	36601.56

TABLE II: Machine learners investigated in this study and corresponding class/method name in SCIKIT-LEARN, CARET and WEKA.

Machine Learner	Library	Class/Method Name	Default Parameters
CART	Caret	rpart	cp =0, maxdepth = 30, minbucket = 7, minsplit = 20, maxcomplete = 4, maxsurrogate = 5, usersurrogate = 2, surrogatestyle= 0, xval = 0
	SkLearn	DecisionTreeRegressor	criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, ccp_alpha=0.0
	Weka	REPTree	-L=-1, -M=2, -S=1, -N=3, -V=0.001, -I=0.0
KNN	Caret	knn	k={5, 7, 9}
	SkLearn	KNeighborsRegressor	n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None
	Weka	IBk	-A=LinearNNSearch, -K=1, -W=0 , -R =first-last
LR	Caret	lm	intercept=True
	SkLearn	LinearRegression	fit_intercept=True, normalize=False, copy_X=True, n_jobs=None, positive=False
	Weka	SimpleLinear	
SVR	Caret	svmRadial	C={0.25, 0.5, 1}, epsilon = 0.1, sigma is execution dependent
	SkLearn	SVR	kernel='rbf', degree=3, gamma='scale', coef0=0.0, tol=0.001, C=1.0, epsilon=0.1, shrinking=True, cache_size=200, verbose=False, max_iter=- 1
	Weka	SMOReg	-C=1, -C=1, -E= 1, -L=0.0001, -K=PolyKernel, -T=0.001, W=1, -N=0, -I=RegSMOImproved, -P=1e-12

Techniques	Parameter	Grid
CART	Maximum Depth	[1 - 21] with a step of 2
KNN	Neighbours	[1 - 17] with a step of 4
SVM	C	[0.25 - 4] with a step of 0.25
	Gamma	[0.1 - 0.9] with a step of 0.2

TABLE III: RQ2: The Parameters and Grid used for the tuning of CART, KNN and SVM.

REFERENCES

[1] "The seacraft repository of empirical software engineering data," 2017.

[2] F. Sarro, A. Petrozziello, and M. Harman, "Multi-objective software effort estimation," in *Proc. of ICSE'16*, 2016, pp. 619–630.

[3] F. Sarro, F. Ferrucci, and C. Gravino, "Single and multi objective genetic programming for software development effort estimation," in *Proc. of SAC'12*. ACM, 2012, pp. 1221–1226. [Online]. Available: http://doi.acm.org/10.1145/2245276.2231968

[4] F. Ferrucci, M. Harman, and F. Sarro, "Search-based software project management," in *Software Project Management in a Changing World*, G. Ruhe and C. Wohlin, Eds. Springer Berlin Heidelberg, 2014, pp. 373–399.

[5] E. Kocaguneli, T. Menzies, and J. W. Keung, "On the value of ensemble effort estimation," *IEEE TSE*, vol. 38, no. 6, pp. 1403–1416, 2012.

[6] B. Sigweni, M. Shepperd, and T. Turchi, "Realistic assessment of software effort estimation models," in *Proc. of EASE'16*. ACM, 2016, pp. 41:1–41:6.

[7] F. Sarro and A. Petrozziello, "Linear programming as a baseline for software effort estimation," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 27, no. 3, pp. 12:1–12:28, 2018.

[8] M. Shepperd and C. Schofield, "Estimating software project effort using analogies," *IEEE TSE*, vol. 23, no. 11, pp. 736–743, 2000.

[9] F. Sarro, R. Moussa, A. Petrozziello, and M. Harman, "Learning from mistakes: Machine learning enhanced human expert effort estimates," *IEEE Transactions on Software Engineering*, 2020.

[10] E. Mendes, M. Kalinowski, D. Martins, F. Ferrucci, and F. Sarro, "Cross- vs. within-company cost estimation studies revisited: An extended systematic review," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '14. New York, NY, USA: Association for Computing Machinery, 2014.

[11] F. H. Yun, "China: Effort estimation dataset," 2010. [Online]. Available: https://doi.org/10.5281/zenodo.268446

[12] J. M. Desharnais, "Analyse statistique de la productivitie des projets informatique a partie de la technique des point des fonction," Ph.D. dissertation, Unpublished Masters Thesis, University of Montreal, 1989.

[13] G. Kadoda and M. Shepperd, "Using simulation to evaluate predictions techniques," in *Proc. of Int. Software Metrics Symposium*. IEEE press, 2001, pp. 349–358.

[14] B. Kitchenham, S. Lawrence Pfleeger, B. McColl, and S. Eagan, "An empirical study of maintenance and development estimation accuracy," *Journal of Systems and Software*, vol. 64, no. 1, pp. 57–77, 2002.

[15] K. Maxwell, *Applied Statistics for Software Managers*. Software Quality Institute Series, Prentice Hall, 2002.

[16] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki, "Robust regression for developing software estimation models," *JSS*, vol. 27, no. 1, pp. 3–16, 1994. [Online]. Available: http://dx.doi.org/10.1016/0164-1212(94)90110-4