Application of Evolutionary Algorithms for Software Maintainability Prediction using Object-Oriented Metrics

Ruchika Malhotra
DSE, DTU, Delhi-110042, India and
DCIS, Indiana University-Purdue University,
ruchmalh@cs.iupui.edu,
ruchikamalhotra2004@yahoo.com

Anuradha Chug
University School of Information and Communication Technology, GGS IP University,
Dwarka, New Delhi-110077, India
a_chug@yahoo.co.in

ABSTRACT
The cost incurred during maintenance phase of any software consists of nearly 60-70% of the total project cost. In order to control, it needs to be measured in the earlier phases of software development life cycle (SDLC). Software Maintainability Prediction (SMP) is desirable because firstly the resource planning can be optimized in advance and secondly it helps in producing cost effective software systems. Significance of the Evolutionary Algorithms (EA) has substantially increased in recent times due to their capability of maximizing the quality function. Inspired by the evolutionary algorithms, we have conducted an empirical study for exploring the application of the EA for SMP. Although several traditional methods such as statistical and machine learning were applied in past, we experimented to apply EA for the first time for SMP. Two open source software projects Apache Poi 3.9 and Apache Rave 0.21.1 written in Java languages were used to carry out this empirical investigation and the results were analyzed using prevalent prediction accuracy measures. We observed that the optimization values were achieved more accurately and precisely with EA than the traditional methods, thus can be successfully applied for SMP.

Categories and Subject Descriptors
D.2 SOFTWARE ENGINEERING D.2.7 Distribution, Maintenance, and Enhancement

General Terms
Experimentation

Keywords: Empirical Validation, Evolutionary Algorithms, Object-Oriented Metrics, Prediction Modeling and Analysis, Software Maintainability Prediction.

1. INTRODUCTION
Software maintainability is defined as the ease with which the software could be modified during operational phase. It cannot be measured until the software system is put to use for a certain period of time. In this regard, it becomes important to develop models which can assess maintainability during early phases of project development with the help of some measurable software characteristics. In literature, it is empirically proved that there exist indeed a strong link between design metrics and subsequent maintainability [1, 4, 6, 10, 12, 13, 14, 15, 16, 17, 19 and 21]. Although so many statistical or machine learning techniques have been developed for SMP in last decade such as Fioravanti & Nesi [10] used Multi Linear Regression Analysis, Aggarwal et al. [1] and Thwin & Quah [19] used neural networks, Koten & Gray [12] used Bayesian network, Misra [17] used Linear Regression, and Zhou et al. [21] used Multivariate Adaptive Regression Splines (MARS) model; however they have been reported to be low in prediction accuracies as per the criteria laid down by Conte et al. [9] and Kitchenham et al. [17]. Therefore, it is necessary to explore the new techniques which are not only easy but also cost effective. We are interested in SMP using relatively new biological inspired EA which are successfully used for the predictions in many fields other than SMP. Overall, study deals with the following three Research Questions (RQs):
RQ1: Empirically validate the relationship between design metrics and subsequent maintainability, if exists?
RQ2: Can Evolutionary Algorithms (EA) be applied for SMP?
RQ3: If EA can be used for SMP, do they perform better or worse than traditional statistical and machine learning methods?
A set of 14 algorithms including 2 statistical, 6 machine learning, and 6 EA based techniques were selected for making the prediction model using the KEEL tool [13]. Preliminary results show that it was able to achieve 78% precision accuracy which is quite competitive. Rest of the paper is organized as follows: Section 2 describes evolutionary algorithms, Section 3 presents research methodology, Section 4 discusses the results and finally Section 5 concludes the paper along with future directions.

BICT 2014, December 01-03, Boston, United States

DOI 10.4108/icst.bict.2014.258044

2. EVOLUTIONARY ALGORITHMS
The evolutionary algorithms are the set of algorithms inspired by the metaphor of natural biological evolution. Each time, at each generation, various operators such as selection, recombination, mutation, migration, locality and neighborhood are applied on potential solutions to produce better and better results [2, 3, 5, 9, 18 and 20]. As shown in Figure 1, when this process is repeated again and again, it leads to the evolution of populations consisting of potential solutions which are optimized. EA techniques are used effectively in software engineering like prediction of development effort [3], prediction of maintenance effort [5], and prediction of preventive maintenance [18], however their use in SMP is found to be extremely limited. In order to compare the performances of EA with traditional methods, we identified the following set of 14 algorithms divided into three major categories as under:

(a) **Statistical Regression**: Linear-LMS-R (LLMSR) and ProQuadratic-MS-R (PQMSR)
(b) **Traditional Machine Learning Algorithms**: Decision Tree (CART and M5-Rules), Neural Networks (Decr-RBFN-R and Ensemble), Support Vector Machine (EPSILON-SVR-R and NU-SVR-R)
(c) **Hybrid Algorithms**: Evolutionary Fuzzy Rule Learning (GFS-GPG-R and THRIFT), Evolutionary Fuzzy Symbolic

Regression (GGSR and GSSR), Evolutionary Neural Network (GANN-R and NNEP-R)

The details of each algorithm are given by Fernandez et al [13] however these are briefly explained as under:

- **Linear-LMS-R (LLMSR):** It is an adaptive algorithm, uses an iterative procedure to minimize the mean square error.
- **Pro Quadratic -MS-R (PQMSR):** In this algorithm, terms are placed in respective groups for classifications.
- **CART**: Classification and Regression Trees are machine learning methods for constructing prediction models obtained by recursively partitioning the data space.
- **M5-Rules**: Same as CART but here the tree can have multivariate linear model to tackle the high dimensionality.
- **Decr-RBFN-R**: Generalization in terms of interpolation between known points is created for such networks.
- **Ensemble**: Instead of random space, constructive non linear projections are created using neural networks.
- **EPSILON-SVR-R**: First order approximation of objective function is used to achieve faster convergence during the working set selection using Support Vector Machine.
- **NU-SVR-R**: Instead of first order information, it uses second order information to achieve faster convergence.
- **GFS-GPG-R**: It combines genetic programming and genetic algorithms to solve symbolic regression problems.
- **THRIFT**: Given by Philip Thrift, the discrete nature of fuzzy strategies is used during the discovery process.
- **GFS-GAP-Sym-R (GGSR)**: Designed especially for electrical engineering, fuzzy arithmetic-based GA procedure is applied for the search of an analytic expression.
- **GFS-SAP-Sym-R (GSSR):** It is a Symbolic Fuzzy-Valued Data Learning based on Genetic Programming Operators and Simulated Annealing.
- **GANN-R**: As it stands for Genetic Algorithm with Neural Network, both techniques are combined to generate more regular connective patterns.
- **NNEP-R**: As it stands for Neural Network Evolutionary Programming for Classification which leads to overall performance gain for real world high order functions.

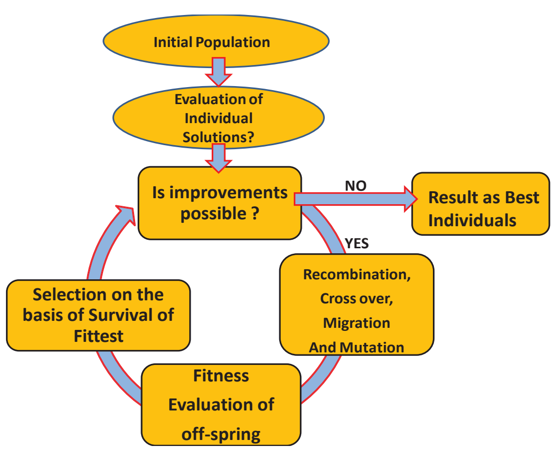

Figure 1: Functioning of Evolutionary Algorithms

3. RESEARCH METHODOLOGY

3.1 Independent and Dependent Variables

Various characteristics of open source software were measured using Chidamber and Kemerer [7 and 13] metric suite given as follows:

- WMC (Weighted Methods per Class): The sum of McCabe's Cyclomatic complexities of all local methods in a class.
- DIT (Depth of Inheritance Tree): The depth of a class in the inheritance tree where the root class is zero.
- NOC (Number of Children) : It counts number of immediate sub classes of a class in a hierarchy
- CBO (Coupling between Objects): It represents the number of classes to which the given class is coupled.
- RFC (Response For a Class): The number of local methods and non-local methods called by current methods.
- LCOM (Lack of Cohesion of Methods): The number of disjoint sets of local methods is calculated by counting the instance variables common to the two or more members.
- LOC (Lines of code): The number of lines of code excluding comments.
- Change: The total number of lines Added, Deleted and Modified in new version w.r.t. the previous version.

3.2 Empirical Data Collection

Two open source software Apache Poi 3.9 and Apache Rave 0.21.1 were analyzed which are briefly described here:

3.2.1 Appache Poi: Written in java language, POI Stands for Poor Obfuscation Implementation. As the name suggests whenever the communication is confusing or harder to interpret, this free and open source java library is used for converting different formats[1].

3.2.2 Appache Rave: It is the web based data integration application software[2]. It's a light weight java platform to combines the data, presentations or functionality from two or more sources to create new services, for example Flicker.

3.3 Prediction Accuracy Measures (PAM)

We analyzed results using accuracy measures given by Conte [8] and Kitchenham [11] as follows:

3.3.1 Magnitude of Relative Error (MRE): It is a normalized measure of the difference between predicted values obtained through model and the respective actual value.

$$MRE = \frac{|\, Actual\ Value - Predicted\ Value\,|}{Actual\ Value}$$

3.3.2 Mean of Magnitude of Relative Error (MMRE): Mean of MRE is calculated as follows

$$MMRE = \frac{\sum_{i=1}^{N} MRE_i}{N}$$ N is the number of observations

3.3.3 Prediction Accuracy at 25% and 30% (Pred 0.25 and Pred 0.30): It measures the portion of the predicted values who have MRE less than or equal to specified value in percentage.

$Pred(q) = \frac{K}{N}$ Where q is the specified value and K is the number of observations.

[1] Apache Poi available online at http://poi.apache.org/

[2] Apache Rave is available at https://rave.apache.org/.

Table 2: Results of Models applied on Apache Poi dataset and Apache RAVE dataset

Category of the algorithm	Name of the Algorithm	POI Dataset				RAVE Dataset			
		Max MRE	MMRE	Pred(0.25)	Pred(.3)	MaxMRE	MMRE	Pred(.25)	Pred(.3)
Statistical Regression	LLMSR	12.94	1.511	0.34	0.38	14.63	1.047	0.45	0.49
	PQMSR	16.83	1.003	0.23	0.29	15.89	0.809	0.39	0.48
Decision Tree	CART	14.27	1.007	0.42	0.48	13.73	0.814	0.37	0.46
	M5-Rules	20.69	0.856	0.52	0.55	10.98	0.758	0.24	0.32
Neural Networks	Decr-RBFN	10.96	0.617	0.49	0.57	11.96	0.791	0.43	0.51
	Ensemble	15.65	0.557	0.37	0.44	13.92	0.845	0.48	0.56
Support Vector Machine	EPSILON-SVR-R	8.28	0.739	0.29	0.36	16.82	0.692	0.46	0.51
	NU-SVR-R	6.76	0.619	0.41	0.48	15.49	0.540	0.39	0.48
Evolutionary Fuzzy (EF)	GFS-GPG-R	3.67	0.246	0.59	0.67	7.02	0.251	0.61	0.64
	THRIFT	6.89	0.224	0.64	0.69	8.06	0.238	0.58	0.62
EF Symbolic Regression	GGSR	5.73	0.396	0.52	0.58	8.78	0.434	0.55	0.59
	GSSR	6.88	0.422	0.57	0.61	9.68	0.354	0.49	0.54
Evolutionary Neural	GANN-R	4.56	0.364	0.46	0.49	7.81	0.362	0.46	0.58
	NNEP-R	6.37	0.257	0.51	0.58	8.22	0.395	0.42	0.48

4. RESULTS AND DISCUSSIONS

The results of various accuracy measures achieved by applying SMP models are compiled in Tables 2 for Poi and Rave datasets. First column represent the category of the algorithm, second column represents the name of the algorithm, third and seventh column represents the maximum value of MRE, fourth and eighth column represents MMRE, and fifth, sixth, ninth and tenth column represents prediction accuracy at 25% and sixth represents prediction accuracy at 30% for POI and RAVE respectively.

RQ1: Is relationship exists between design metrics and maintainability? We divided the data into 3:1 ratio between training and testing respectively. The value of MMRE represents the goodness of fit of the proposed models. When we calculated the average MMRE values of all 14 algorithms selected in the current empirical study, its value is 0.364 and 0.362 for Poi and Rave respectively. It is quite evident that the values of MMRE are significantly better for both the datasets and quite competitive as per the standards of accuracy measurements [8 and 11], we claim that there exists indeed a strong relationship between design metrics and maintainability.

RQ2: Can EA be applied for SMP?

On both the datasets, THRIFT is found to be most accurate if MMRE is taken as the accuracy indicator and GFS-GPG-R algorithm of was found to be more accurate if MaxMRE is considered as accuracy indicator. Both algorithms under the category of EFA were found to be the most accurate even when prediction accuracy at 25% and 30% is checked. Thus, from the results it is safe to claim that EFA can be used for more precise maintainability predictions.

RQ3: Do EA perform significantly better than traditional methods?

In order to find the answer of research question RQ3, we performed paired t-test which is used to identify the significant existence of statistical differences between various treatments applied to the data. While finding the answer of previous question RQ2, both the algorithms under the category of EFA were found to be as out-performer. With the help of paired T-test, we investigated further the performance THRIFT and GFS-GPG-R, both belongs to the category of EFA. In total 24 pairs were made for each datasets; in the first set, THRIFT of EFA category was paired with rest 12 algorithms and in second set, GFS-GPG-R of EFA category was paired with remaining 12 algorithms. Paired T-Test was conducted for Rave as well as Poi datasets at 95% significance level with 9 degree of freedom and results are summarized in Table 3 and 4 respectively. The performance of the paired-T-test is measured using p-value and t-value. If p-value is less than 0.05, we claim that the results of the given pair are significantly different. Whether this significance is better or worse depends upon the t-value. If t-value is negative that means the first algorithm is better, otherwise second algorithm is better. In Table 3 and 4, 1st column represent the pair with THRIFT and 4th

Table 3: Results of Paired T Test with Poi dataset

Pair of Algorithm with THRIFT	t-value	p-value	Pair of Algorithm with GFS-GPG-R	t-value	p-value
CART	-3.255	0.010	CART	-2.614	0.028
M5RULES	-3.691	0.005	M5RULES	-2.715	0.024
LLMSR	-2.827	0.020	LLMSR	-2.644	0.027
PQMSR	-3.359	0.008	PQMSR	-.887	0.398
Ensemble	-3.601	0.006	Ensemble	-2.493	0.034
DecrRBFN	-2.749	0.023	DecrRBFN	-1.996	0.077
Epsilon	-3.541	0.006	Epsilon	-2.821	0.020
NU_SVR	-3.869	0.004	NU_SVR	-2.662	0.026
GGSR	-1.623	0.139	GGSR	-.266	0.796
GSSR	-2.099	0.065	GSSR	-1.457	0.179
GANN	-1.401	0.195	GANN	-.633	0.542
NNEP	-2.380	0.041	NNEP	-2.203	0.055

Table 4 : Results of Paired T Test with Rave dataset

Pair of Algorithm with THRIFT	t-value	p-value	Pair of Algorithm with GFS-GPG-R	t-value	p-value
CART	-1.408	0.019	CART	-1.408	0.003
M5RULES	-2.046	0.017	M5RULES	-2.053	0.040
LLMSR	-2.148	0.006	LLMSR	-1.750	0.014
PQMSR	-3.107	0.013	PQMSR	-2.840	0.019
Ensemble	-2.090	0.009	Ensemble	-2.185	0.057
DecrRBFN	-1.220	0.023	DecrRBFN	-1.546	0.056
Epsilon	-2.360	0.043	Epsilon	-2.012	0.017
NU_SVR	-1.747	0.015	NU_SVR	-2.094	0.006
GGSR	-4.351	0.002	GGSR	-4.075	0.003
GSSR	-1.649	0.134	GSSR	-1.649	0.134
GANN	-.590	0.370	GANN	-.632	0.543
NNEP	-2.109	0.264	NNEP	-1.734	0.117

column with GFS-GPG-R, 2nd and 5th column represents the corresponding t-values and 3rd and 6th column represents corresponding p-value in both the tables for POI and RAVE dataset respectively. It is quite evident that THRIFT is significantly different with 9 out of 12 algorithms for Rave dataset and 8 out of 12 algorithms for poi datasets. GFS_GPG_R is significantly different with 8 out of 12 algorithms for Rave dataset and 7 out of 12 algorithms with poi datasets. Thus, we claim that the evolutionary fuzzy algorithms are significantly different than other algorithms. Next job was to determine if they are better or worst. As all the value of t-value is negative when THRIFT and GFS-GPG-R is paired with other algorithms, we also claim that the evolutionary fuzzy algorithms are significantly better than their counterpart.

5. CONCLUSION & FUTURE DIRECTION

In this empirical study, we evaluated the performance of evolutionary algorithms for software maintainability predictions. We compared the prediction performance of evolutionary fuzzy, evolutionary neural and evolutionary neural symbolic regression methods with traditional statistical and machine learning models. Datasets were collected using two open source software systems Apache Poi and Apache Rave. The results indicate EA are much better than traditional algorithms as they could achieve accuracy within the range of 22% to 25%. We conducted paired T-test to compare their performance with traditional methods. We found that the EA performs significantly better than the traditional algorithms. An important contribution of this work is that since we have compared results using the OO metrics suite using two open source software systems, we can generalize our results. This study confirms that construction of EA for SMP is feasible, adaptable and useful in predicting software maintainability. One of the biggest drawbacks of EA is that the suggested solution produced in each iteration is better only in comparison with the previous one and the candidate solution is not guaranteed optimal. As the final choice of when to stop always lies with the user, it is very important to attain the maxima of cost and efforts trade off. In future we are planning to determine the parameter which makes the evolutionary algorithms not only effective but equally efficient as well.

6. REFERENCES

[1] Aggarwal, K.K., Singh, Y., Kaur, A., Malhotra, R. 2006. Application of Artificial Neural Network for Predicting Maintainability Using Object Oriented Metrics. *Proceedings of World Academy of Science, Engineering and Technology*, 2(10), 2008, 285- 289

[2] Alba, E. Parallelism and evolutionary algorithms, IEEE Trans. on Evolutionary Computation, 2002, 6(5), 443-462.

[3] Balogh, G., Zoltan, A., Baszedes, A. Prediction of Software Development Effort Enhanced by a Genetic Algorithm, *Symposium on search based software engineering, SSBSE*, 28-30 September, 2012, Trento, Itly.

[4] Bandi R. Predicting maintenance performance using object-oriented design complexity metrics, *IEEE Transactions on Software Engineering*, 2003,29(1), 77-87.

[5] Baqais, B., Alshayeb, M., Baig, Z.A. Hybrid Intelligent Model for Software Maintenance Prediction *in Proceedings of the World Congress on Engineering*, 3rd -5th July 2013, London, UK, 358-362.

[6] Briand, L.C., Bunse, C., and Daly, J.W. A controlled experiment for evaluating quality guidelines on the maintainability of object oriented design, *IEEE Transaction on software Engineering*, 27(6), 513-530

[7] Chidamber, S. and Kemerer, C. A Metrics Suite for Object Oriented Design, *IEEE Transactions on Software Engineering*, 20(6), 476-493, 1994

[8] Conte,S.H., Dunsmore, Shen,V. Software Engineering Metrics and Models, *Menlo Park, Benjamin Cummings*,1986

[9] Fernandez A., Luengo J., Derrac J., AlcalaFdez J and Herrera F. Implementation and Integration of Algorithms into the KEEL Data-Mining Software Tool, Intelligent Data Engineering and Automated Learning–IDEAL 2009, Lecture Notes in Computer Science, Volume 5788, 562-569, 2009.

[10] Fioravanti, F. and Nesi, P. Estimation and prediction metrics for adaptive maintenance effort of object oriented systems, *IEEE Transactions on Software Engineering*, 27(12), 1062-1084, 2001.

[11] Kitchenham, B.A., Pickard, L.M. MacDonell, S.G. and. Shepperd M.J What accuracy statistics really measure, *IEE Proceedings-Software*,2001, 148 (3), 81–85.

[12] Koten, C.V., Gray, A.R. An application of Bayesian network for predicting object-oriented software maintainability, *Information and Software Technology*, 2006, 48(1), 59-67.

[13] Li, W. and Henry, S. Object-Oriented Metrics that Predict Maintainability, *Journal of Systems and Software*, 1993, 23(2), 111-122,

[14] Malhotra, R., and Chug, A., Software Maintainability Prediction using Machine Learning Algorithms, Software Engineering: An International Journal (SEIJ), Sept, 2012, 2(2), 19-36.

[15] Malhotra, R., and Chug, A., Application of Group Method of Data Handling model for software maintainability prediction using object oriented systems. Int. J. Systems Assurance Engineering and Management , International Journal, Springer Vol 5, Issue 2, page : 165-173 (2014),

[16] Malhotra, R., and Chug, A., A Metric Suite for Predicting Software Maintainability in Data Intensive Applications, Transactions on Engineering Technologies, Springer, page 161-175

[17] Misra, S. Modeling design/coding factors that drive maintainability of software systems, *Software Quality Journal*, 2005, 13(3), 297-320, 2005.

[18] Sun, P. and Wang, A. Application of Ant Colony Optimization in Preventive Software Maintenance Policy, *IEEE international Conference on Information Science and Technology*, 23-25 March 2012, Yangzhou, China, 141-144.

[19] Thwin, M. and Quah, T. Application of neural networks for software quality prediction using object oriented metrics, *Journal of Systems and Software*, 76(2),2005, 147-156.

[20] Vivanco, N. Pizzi, Finding Effective Software Metrics to Classify Maintainability Using a Parallel Genetic Algorithm, Genetic and Evolutionary Computing, GECCO 2004, Lecture Notes in Computer Science, 30(13), 2004, 1388-1399.

[21] Zhou Y, Leung H. Predicting object-oriented software maintainability using multivariate adaptive regression splines. The Journal of Systems and Software 2007, 80(8):1349-1361