Statistical Analysis of the Most Popular Software Service Effort Estimation Datasets

Amid Khatibi Bardsiri[1], Seyyed Mohsen Hashemi[1], Mohammadreza Razzazi[2]
[1]*Computer Engineering Department, Science and Research Branch, Islamic Azad University, Tehran, IRAN*
[2]*Computer Engineering and IT Department, Amirkabir University of Technology, Tehran, IRAN*
a.khatibi@srbiau.ac.ir

***Abstract*—Considering the complex nature of software projects, we have to use historical data and past experiences to execute them better. In previous years, a large number of software engineering datasets were introduced for different purpose. One of the important groups among these datasets is the use of software effort estimation repositories as a framework for analyzing diverse methods and models of estimation. In recent decades, researchers have worked on the different types of these datasets for various purposes and have tried to find the features of each one. DPS, ISBSG, Desharnais, Maxwell, and CF are among the most popular of these datasets. Insufficient or unstructured documentation causes problems for researchers in recognizing and working with datasets that are suitable for their purposes. This article intends to perform a thorough statistical analysis of the five the most popular datasets for software effort estimation to provide researchers with useful information and to help them select the appropriate repositories. In this paper, a thorough statistical analysis of software effort datasets is performed, and sufficient explanations are offered so that researchers have better options for their particular purposes. It is suggested that software engineering community should be aware of and account for the software effort dataset related issues when evaluating the validity of research outcomes.**

***Index Terms*—Software effort estimation, Repository, Software dataset, Statistical analysis**

I. Introduction

Software development is a very complex process with many risks involved. The software engineering community has widely adopted the use of repositories for estimating development effort, number of defects prediction, project maintenance costs, and other similar items [5-8]. Historical data is very vital and valuable for the growth of the software development industry. The quality of the repository greatly influences the outcome and efficiency of effort estimation models [9]. A software engineering repository is a well-defined, useful, and real set related to software projects. These sets usually contain qualitative and quantitative information regarding resources, artifacts, techniques, and the data they include are required for software frameworks, models, estimation process, development methods, and for upgrading production process quality [10]. These datasets allow specialists to perform their analyses in a repeatable and comparable manner on a single field of endeavor [11]. The mentioned data is useful in empirical and experimental studies and in frameworks, and has many applications. Although it is difficult to access and collect this data, recognition of its features and applications is important. Unfortunately, there are many datasets related to software effort estimation, and they are not sufficiently documented [12]. Satisfying documentation requirements requires different types of document from informal working documents to the professionally produced user manuals. Software engineers are usually responsible for producing most of this documentation although professional technical writers may assist with the final result [13-15]. Moreover, no comprehensive study has been conducted so far on dataset features, and on the information included in requirements [16, 17]. This makes the selection of a suitable dataset difficult for researchers. There is a growing number of software repositories, with varying content types (e.g. articles, data sets, images, etc.) and disciplinary foci. If your funder has not specified a repository/data center or a disciplinary repository, other repositories are available [15]. Topics, such as recognition of the most important dataset features, their data distribution and its deviation, correlations between dataset variables, and the relationships between the variables have not been comprehensively studied. To assess whether a repository is suitable for your research, you should consider the following questions:

- Will others be able to find your data
- Under what license terms are datasets made available for reuse?
- Can you apply an access embargo period if you need to?
- Is there a community to support the repository?
- What is the growth rate for data deposit?
- What file formats does the repository support?

Figure 1 shows a diagram of the most important datasets related to software effort estimation, and the example references used in each one of the dataset. Repositories in PROMISE and ISBSG, which are among the most popular datasets, have so far been used in numerous studies [5, 10, 17, 18]. In this article, a thorough statistical analysis of these datasets is performed, and sufficient explanations are offered so that researchers have better options for their particular purposes. Statistical discussions include a large number of techniques that clearly shows the meaning of quantities, their related concepts, and their relationships with each other. Without such an understanding, results obtained from estimation models will be biased and devoid of real meaning.

The rest of this article is organized in the following sequence. Section 2 introduces five popular datasets and their descriptive statistics and Section 3 studies the correlations between variables. Additionally, Section 4

visualizes the data, followed by Section 5 and 6 which are devoted to data distribution methods and regression analysis. Finally, Section 7 deals with the conclusions and future work.

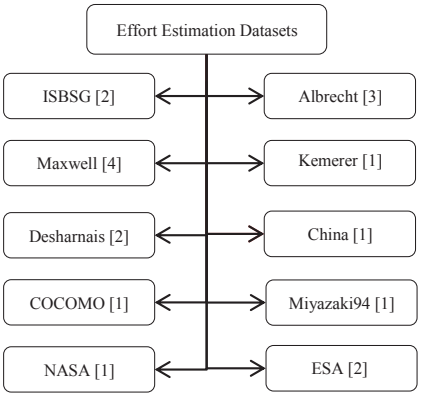

Figure 1: Different Effort Estimation Datasets

II. Research Datasets

In order to explore the real performance of estimation models, an evaluation must be carried out by applying real datasets. During the past decades, a large number of datasets have been constructed, each with its own particular purposes [19]. In this article, the following datasets developed by various companies engaged in different studies have been selected. The information included in each of these repositories will be explained below. Filters considered for ISBSG dataset were generally derived from the documents of the company itself (Guidelines for use of the ISBSG data) and previous works [20]. Authors have carefully studied this report and reviews of previous research, and selected an appropriate subset. In fact, in order to use and select an appropriate subset of the data, we must fully understand their concept and meaning, as an apple with apple comparison. Moreover, the statistical information for each dataset was collected to fully inform the readers. Generally, the issue of pre-processing and data preparation is an important and essential task in the data mining domain and there are many articles in this context. For instance, [5] presents a credible example of research in the effort estimation domain, which explains the method and reason of performing pre-processing on datasets. A filtration process must be conducted to select an appropriate and reliable subset of dataset projects. It is important that its users have a sound knowledge of the data, are aware of its strengths, limitations and its positioning, prior to analyzing or using it. The first important step in each data mining exercise is data preprocessing.

A. DPS dataset

The first dataset is related to the IBM data processing services (DPS) organization [21]. It consists of 24 projects developed by third-generation languages. Five numerical attributes that may affect the project effort are input count (IC), output count (OC), query count (QC), file count (FC), and adjustment function point (AFP). In this dataset, there is a project whose effort is quite far from the second smallest project. In practice, this project is unsuitable as an analogue for other projects. Therefore, it is excluded and regarded as a missing project in order to compare the results with the previous findings. The DPS dataset is also called Albrecht in some references. Table 1 shows the statistical information about this dataset.

B. CF dataset

The second dataset is related to a major Canadian financial (CF) organization [22] which is comprised of 21 projects. The collected projects are within the same application domain and are developed using a standard development process model. Most of the collected projects are developed on the IBM mainframe. Input count (IC), output count (OC), inquiry count (IQC), internal logical files count (ILF), external interface files (EIF) and adjustment function point (AFP) are the main attributes considered in the model construction. Statistical information related to this dataset is presented in Table 2.

C. Desharnais dataset

Desharnais is one of the most common datasets in the field of software effort estimation [23]. Although this dataset is relatively old, it has been widely employed in many of recent research studies [2-4, 19]. This dataset includes 77 software projects, and has 8 independent features. Each project is described by nine attributes. One of the attributes (language) is categorical and the remaining ones are numerical. Table 3 provides the statistical information about this dataset.

Table 1
Description of DPS dataset

Attribute	Description	Min	Max	Mean	Median	Std Dev
IC	Input Count	7	193	41.3	34	37.3
OC	Output Count	12	150	48.7	40	35.3
QC	Query Count	0	75	17.3	14	19.6
FC	File Count	5	60	18	12	15.6
FP	Function Point	199	1902	647	506	488
EF	Effort (1000h)	2.9	105.2	22.8	11.8	28.7

Table 2
Description of CF dataset

Variable	Description	Min	Max	Mean	Median	Std Dev
EIF	External Interface Files	2	67	29.9	30	16.2
ILF	Internal Logical Files	0	45	16.6	16	11.3
IC	Input Count	0	46	17.2	16	11.5
OC	Output Count	0	69	27.4	25	15.0
IQC	Inquiry Count	0	33	9	8	9.1
FP	Function Point	31	232	123.8	132	56.1
EF	Effort (day)	52	544	331.8	369	151.0

Table 3
Description of Desharnais dataset

Attribute	Description	Min	Max	Mean	Median	Std Dev
TeamExp	Team experience	0	4	2.30	2	1.33
ManagerExp	Manager's Experience	0	7	2.65	3	1.52
Length	Length of project	1	36	11.30	10	6.79
Transactions	Number of transactions	9	886	177.47	134	146.08
Entities	Number of entities	7	387	120.55	96	86.11
AdjustFactor	Sum of complexity factors	5	52	27.45	28	10.53
PointsAdjust	Number of adjusted function points	73	1127	298.01	247	182.26
Language	Programming language	1	3	1.56	1	0.72
DE	Development Effort (h)	546	23940	4833	3542	4188

D. Maxwell dataset

Maxwell was selected as the fourth dataset because it is a relatively new dataset comprised of 62 software projects [24]. Each project is described by 26 attributes in this dataset in which four attributes are numerical, six attributes are categorical, and the rest sixteen attributes are ordinal. Twenty five attributes are treated as the independent attributes while the effort is treated as the dependent attribute. Table 4 provides the statistical information about the Maxwell dataset.

E. ISBSG dataset

International software benchmarking standard group (ISBSG) is a company located in Australia. It collects the information related to software projects from all over the world [20]. In this study, ISBSG release 11 was used as the basic dataset. It contains the detailed information about 5052 software projects. 70% of the projects are less than nine years old. Each project in the ISBSG dataset is described by numerous attributes. These data have been gathered from 24 countries, and the distribution of contribution comprises of the United States (31% of all projects), Japan (17%), Australia (16%), Finland (10%), the Netherlands (8%), India (6%), Canada (5%), Denmark (3%), Brazil (2%), the United Kingdom (2%), and China (1%). The statistical information related to attributes of ISBSG is presented in Table 5. From the table, there are seven numerical and three categorical attributes in the selected subset of ISBSG. The selected attributes are the Input count (INPCont), output count (OutCont), enquiry count (EnqCont), file count (FileCont), Interface count (IntCont) adjusted function point (AFP), development type (DevType), organization type (OrgType), development platform (DevPlat), and normalized effort (NorEffort). An appropriate subset of ISBSG dataset was selected for this research. In the first step, the project with quality rates other than A and B were removed; therefore there was no doubt in the accuracy of the data. Then, the projects were filtered by some resource levels other than development, so that the learning effort and alike are not considered (resource level≤ 1). Finally, the projects that measurement metric of their sizes were other than IFPUG were removed. In the end, by following the above-mentioned filters, 448 software projects were obtained and they are used as the sample of the analysis.

Here, the coefficient of variation (CV) is computed through dividing the standard deviation by the mean [25] to show the distribution of effort in each dataset. Table 6 shows the values of CV on these datasets. Higher CV values show greater imbalance, on the other hand, lower CV values indicate better distribution of effort. Table 6 presents the value of CV for Desharnais dataset indicates a higher level of imbalance in the distribution of effort as compared to CF and a lower level as compared to DPS. In addition, the value of CV for Maxwell shows the highest non-normality in distribution of effort as compared to the other datasets.

Table 4
Description of Maxwell dataset

Attribute	Description	Min	Max	Mean	Median	Std Dev
Time	Time	1	9	5.58	6	2.13
App	Application type	1	5	-	-	-
Har	Hardware platform	1	5	-	-	-
Dba	Database	0	4	-	-	-
Ifc	User interface	1	2	-	-	-
Source	Where developed	1	2	-	-	-
Telonuse	Telon use	1	4	-	-	-
Nlan	Number of development languages	0	1	0.24	3	0.43
T01	Customer participation	1	5	3.05	3	1
T02	Development environment adequacy	1	5	3.05	3	0.71
T03	Staff availability	2	5	3.03	3	0.89
T04	Standards use	2	5	3.19	3	0.70
T05	Methods use	1	5	3.05	3	0.71
T06	Tools use	1	4	2.90	3	0.69
T07	Software's logical complexity	1	5	3.24	3	0.90
T08	Requirements volatility	2	5	3.81	4	0.96
T09	Quality requirements	2	5	4.06	4	0.74
T10	Efficiency requirements	2	5	3.61	4	0.89
T11	Installation requirements	2	5	3.42	3	0.98
T12	Staff analysis skills	2	5	3.82	4	0.69
T13	Staff application knowledge	1	5	3.06	3	0.96
T14	Staff tool skills	1	5	3.26	3	1.01
T15	Staff team skills	1	5	3.34	3	0.75
Duration	Duration (months)	4	54	17.21	13.5	10.65
Size	Application size (FP)	48	3643	673.31	385	784.08
Effort (h)	Work carried out	583	63694	8223	5189	10500

Table 5
Description of ISBSG dataset

Attribute	Description	Min	Max	Mean	Median	Std Dev
InpCont	Input count	3	2221	152.20	72	226.96
OutCont	output count	4	2455	141.25	65.50	210.06
EnqCont	enquiry count	3	1306	115.81	64.50	155.39
FileCont	file count	7	1732	130.82	68.50	184.11
IntCont	Interface count	5	1572	70.87	30	147.63
AFP	adjusted function point	29	7633	625.66	380	770.13
NF	NorEffort (h)	64	60826	5588.65	3216	7095.64

Table 6
CV values for different datasets

Dataset	CV	Comment
DPS	1.26	High level of imbalance in the distribution of effort
CF	0.46	Relatively normal distributed
Desharnais	0.87	Normal distributed
Maxwell	1.28	Highest non-normality in distribution
ISBSG	1.27	Includes a large number of samples

III. Correlation Between Variables

When studying the many features of a population, we may want to know whether these features are related to each other. Correlation coefficient is a statistical tool for determining the type and the degree of relationship between two quantitative variables, measuring the strength of this relationship. This tool is also used to show the type of the relationship, that is whether it varies from -1 to +1 (inverse or direct relationship) or zero (no relationship) between two variables [26]. Correlation coefficient is a symmetrical relationship and the closer it is to 1, the stronger the dependence between the two variables will be. Correlation coefficient between the two variables X and Y is defined in the following equation:

$$Corr(X,Y) = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)}{\sigma_X \sigma_Y} \quad (1)$$

In Equation 1, E is the expected value operator, Cov is the covariance, $Corr$ is the usual symbol for correlation, and σ is the symbol for standard deviation. Tables 7 to 11 show the correlation coefficients for the DPS, CF, Desharnais, Maxwell and ISBSG respectively (for datasets with a large features, or categorical features, only the numerical and important features are considered). The best answer for effort is highlighted. As shown in the tables, the highest correlation in DPS (0.935) is that between the variables of effort and FP. This is also true for other repositories (with some difference in the case of ISBSG). Therefore, the most influential parameter in effort estimation is the FP variable. The lowest correlation belongs to ISBSG and the highest to DPS (which shows more accurate effort estimation in DPS dataset). Moreover, CF is the only dataset with negative values. Later in the article, this analysis is used to study the most important feature of the datasets: the FP value and its relationship with Effort.

Table 7
Correlations between different variables of DPS

Features	IC	OC	QC	FC	FP	Effort
IC	1.000	0.437	0.518	0.329	0.670	0.628
OC		1.000	0.678	0.734	0.906	0.876
QC			1.000	0.578	0.776	0.841
FC				1.000	0.822	0.761
FP					1.000	0.935
EF						1.000

Table 8
Correlations between different variables of CF

Features	EIF	ILF	IC	OC	IQC	FP	Effort
EIF	1.000	-0.596	-0.599	-0.260	0.162	-0.137	-0.058
ILF		1.000	0.349	-0.22	-0.262	0.044	-0.173
IC			1.000	-0.321	-0.112	0.270	0.155
OC				1.000	-0.502	-0.095	-0.009
IQC					1.000	-0.002	0.127
FP						1.000	0.882
EF							1.000

Table 9
Correlations between different variables of Desharnais

Features	Transactions	FP	Length	Entity	Effort
Transactions	1.000	0.883	0.671	0.176	0.596
FP		1.000	0.734	0.589	0.735
Length			1.000	0.476	0.657
Entities				1.000	0.500
Effort					1.000

Table 10
Correlations between different variables of Maxwell

Features	Duration	FP	Effort
Duration	1.000	0.521	0.656
FP		1.000	0.841
Effort			1.000

Table 11
Correlations between different variables of ISBSG

Features	FP	IntCont	InpCont	OutCont	EnqCont	FileCont	Effort
FP	1.000	0.556	0.867	0.832	0.708	0.846	0.529
IntCont		1.000	0.225	0.423	0.171	0.495	0.150
InpCont			1.000	0.659	0.632	0.669	0.544
OutCont				1.000	0.505	0.595	0.364
EnqCont					1.000	0.499	0.459
FileCont						1.000	0.427
Effort							1.000

IV. Data Visualization

There are many various graphic tools for displaying information related to data instances. Among the most important graphic tools are histograms, box plots, and scatter diagrams and these tools show the way data is distributed, the presence of outlier values, and possible relationships between the variables [26].

In this section, visualization of the datasets with the help of box plots is explained first. In descriptive statistics, the box plot is a diagram used for describing data changes. Box plots, also called box-and-whisker plots (which means box and vertical lines), can give us useful information on how the data is distributed, and on outlier data related to a quantitative variable. The length of the box is important, and its width bears no meaning. A box is used to display the interval between the first and third quartiles. A line inside the box, called the median, represents the second quartile; and if it is in the middle of the box, the data distribution is normal. The two lines outside the box show the minimum and maximum values of the data and, finally, the outlier instances are represented as red points. In Figure 2, data distributions for DPS, CF, Maxwell, Desharnais, and ISBSG are shown, respectively (in all of these diagrams; the interval for each feature is normalized between zero and one).

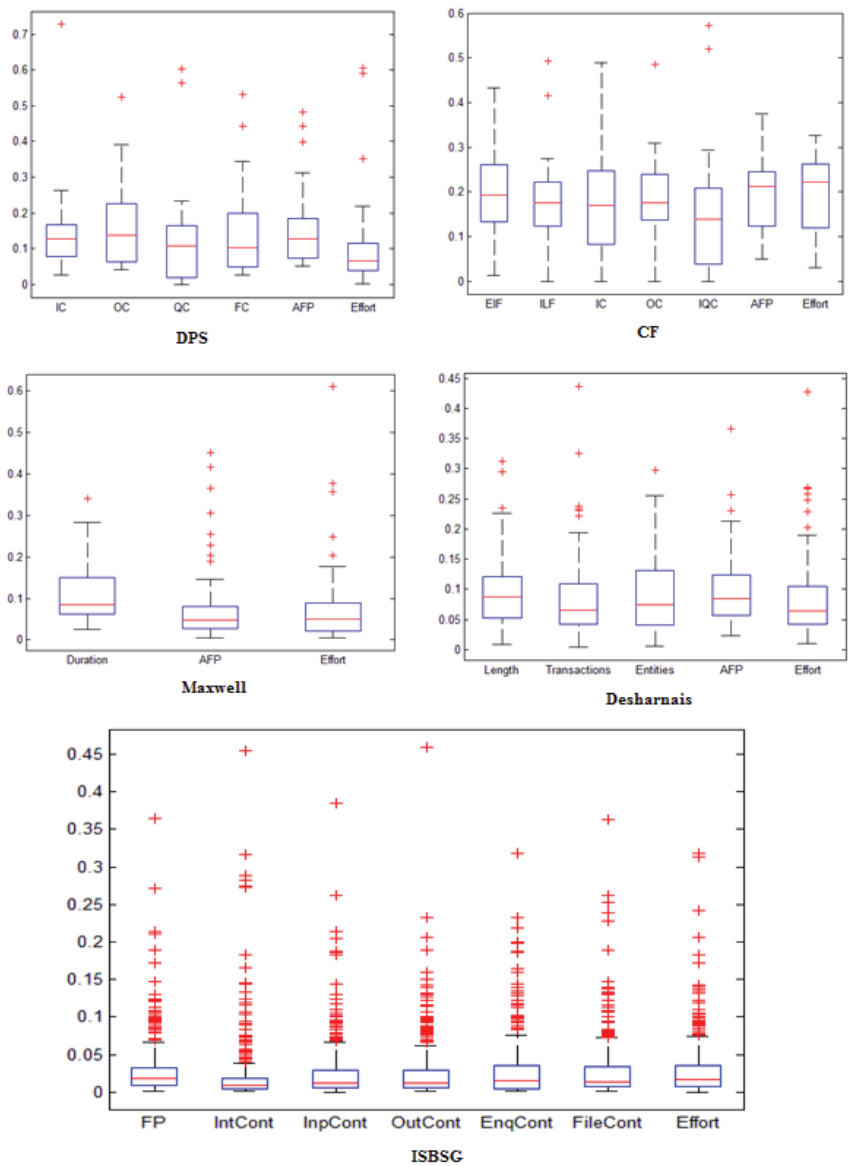

Figure 2: Boxplot diagrams for different effort estimation datasets

As shown in the diagrams, ISBSG has the maximum number of outliers and the greatest variety (which is expected given its large number of data instances). Contrary to ISBSG, the DPS and CF datasets enjoy a good balance, and their values are located in suitable intervals. Maxwell dataset possesses the most categorical features that are not very usable in visualization or statistical operations (there are only three important numerical variables in this dataset). Figure 2 shows the median is not in the middle of the box and; therefore, distribution of values is by no means normal in CF, while the other datasets have almost normal distributions (considering the locations of the medians).

Figure 3 shows scatter plot diagrams of the effort values based on the values of FP (it was mentioned in section 3 that the variable effort had the highest correlation with this feature). Identification of the dependent factors in a process is usually necessary for controlling the process. If one of these factors is controlled, the other one will also be under control because the two factors are correlated [26]. That is why scatter diagrams are used. Scatter diagrams are employed for understanding potential relationships between two variables. To plot these diagrams, the data is prepared in pairs such as Y_i and X_i. The value of Y_i is plotted against that of X_i in these diagrams. The way the points are plotted in the diagrams shows the type of relationship between the two variables and determines the degree of correlation between them. The blue points represent data values and the red line is the regression line (that is used for estimation). Section 6 is wholly devoted to regression. Here again DPS has the best distribution and the maximum correlation, with most points located around, and very close to, the regression line. This is less observed in ISBSG considering the larger number of points and the fact that they are much more scattered.

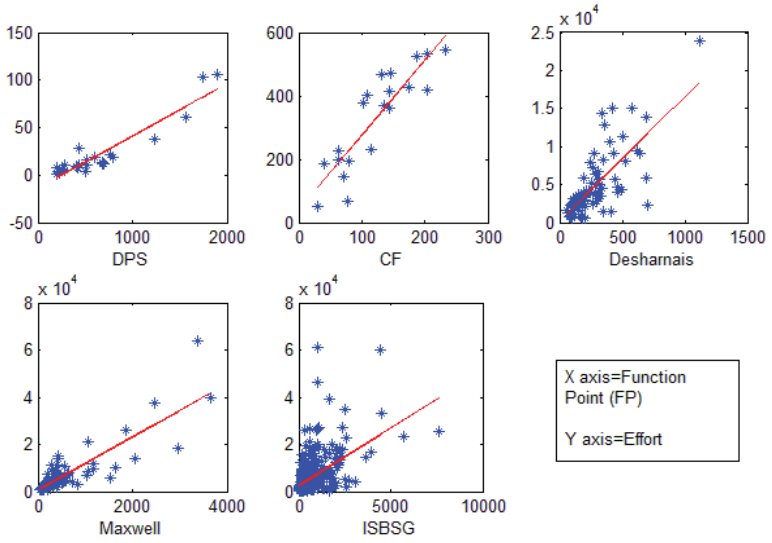

Figure 3: Scatter plot diagrams and regression line for different effort estimation datasets

V. Data Distribution

A typical style is the bell–shaped curve called the normal distribution. In a normal distribution, elements are being expected to arise in one area of the average as on the other. Remember, nevertheless, that different distributions appear like the normal distribution. Statistical computations have to be employed to show a normal distribution. Normality test should be conducted before assuming data distribution is normal. Histograms are a special form of bar chart and help in describing data. In a histogram, a large amount of data is classified in a special format so that it can be understood and analyzed more easily. One way of displaying how the values are distributed is to use a histogram in which the y-axis shows the frequencies of the values, and the x-axis shows the ranges. If the diagram is bell-shaped, the distribution is called normal. Figure 4 and 5 show the histograms of FP values and the development efforts of the various datasets, respectively. As shown in these figures, CF has the most normal frequency distribution and ISBSG the least. In fact, each histogram becomes more abnormal the less similar it is to the bell shape. Statistical analysis of datasets that lacks normal distribution is more difficult compared to those with normal distribution.

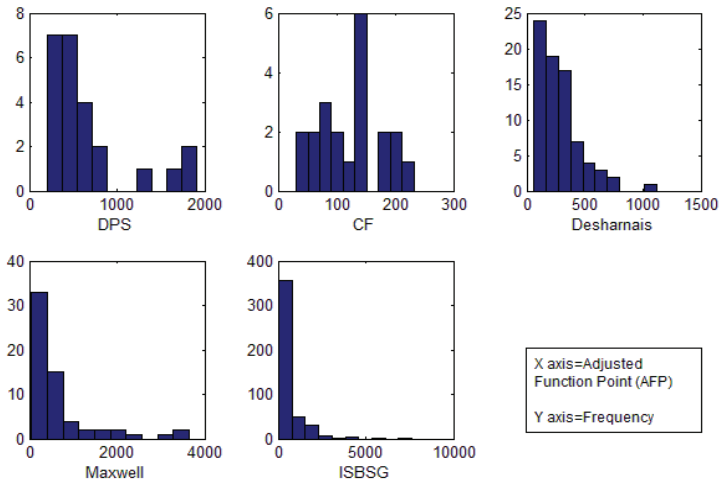

Figure 4: FP values histograms for different effort estimation datasets

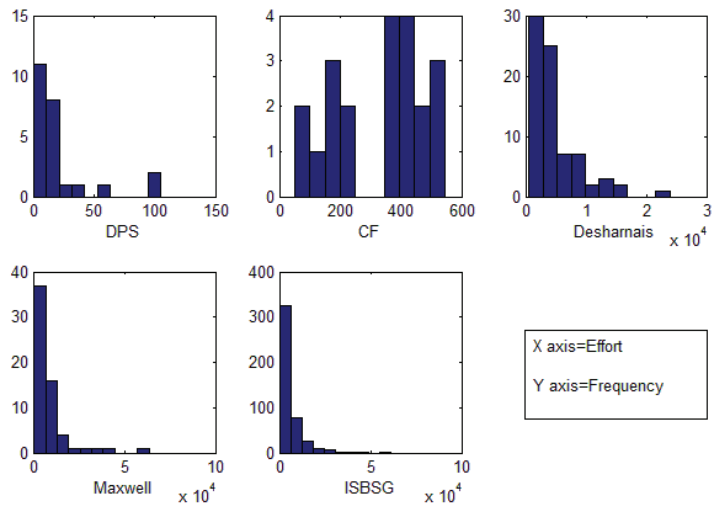

Figure 5: Development effort values histograms for different effort estimation datasets

VI. Regression Analysis

Regression analysis is used to find meaningful relationships between variables. It consists of a number of methods for modeling or evaluating various factors, while the target is on the correlation between a based factor and several free factors. Regression analysis is very popular for forecast and also predicting, in which its apply possesses significant overlap with the area of machine learning. In short, regression analysis assists an understanding of how the standard value of the based factor adjustments when there are some unbiased factors. Regressions are a statistical model in which a dependent variable is estimated by using several independent variables [27]. Equation 2 shows the general form of linear regression models. In this formula, Y is the dependent variable, X_i the independent variables, B_i the variable coefficients, and e is the amount of error.

$$Y = B_1X_1 + B_2X_2 + \cdots + B_nX_n + e \quad (2)$$

The purpose in a regression model is to determine those coefficients of B_i that minimize the amount of error (e). After finding the suitable coefficients, the obtained model is used to estimate the dependent variable using the independent ones. In effort estimation, the software project the independent variables are the features and the dependent variable is the effort values. MLR, ROR, and SWR are among the most popular regression models in effort estimation [2-4, 27]. Considering the calculated correlations in section 3, effort values are estimated here using three different regression methods. Table 12 shows the suggested coefficients and the amount of error in each method used for the various datasets. The best answer for residual in each

line is highlighted. The distribution diagram and the regression line of each dataset are presented in Figure 6 to better understand them. The figure shows the different regression methods, the coefficients, and the slopes of the different lines (in three different colors). As can be seen in the Table 12, the MLR method gives better answers than the other two and has fewer errors.

Table 12
The results of different regression methods

Dataset	MLR		ROR		SWR	
	Residual	Coeff	Residual	Coeff	Residual	Coeff
DPS	250.71	0.0411	210.44	0.0320	349.47	0.544
CF	1225	2.6317	1387	2.3470	1355	2.3750
Desharnais	152640	16.1674	157800	13.0228	153130	16.3673
Maxwell	228480	11.6757	291310	5.4000	224700	11.272
ISBSG	1864700	25.3235	2000100	10.3000	1856900	16.4764

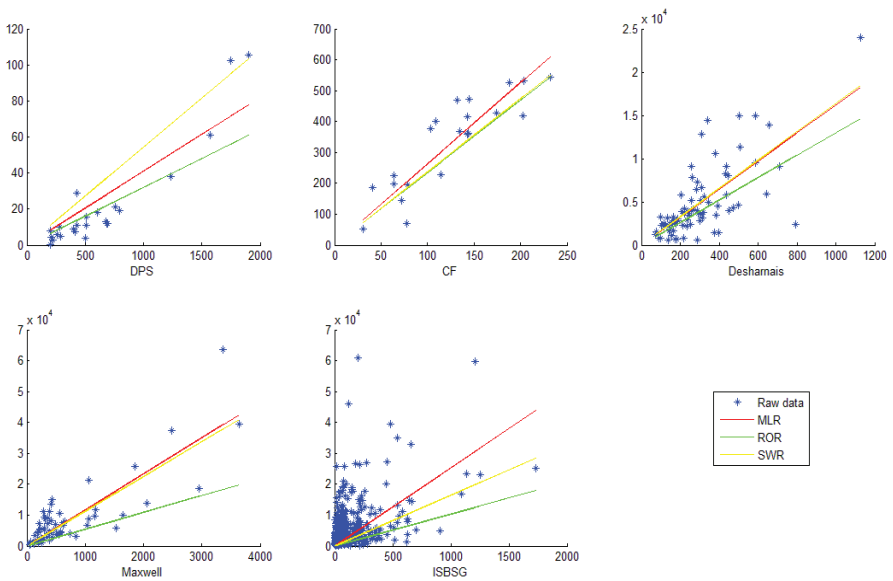

Figure 6: Regression methods for different effort estimation datasets

Since a linear regression method is possibly not suitable for the computer data, it is advisable to evaluate the appropriateness of the strategy by showing residuals or analyzing residual plots. The difference between the realized value of the based variable (Actual Effort) and estimated value is known as the residual. Every record includes one residual. Equation 3 shows the formula for calculating the residual amount.

$$Residual = Actual\ value - Estimated\ value \quad (3)$$

A residual chart is a graph that displays the residuals on the vertical axis and unbiased factor on the horizontal axis. A residual chart is used in Figure 7 to indicate the amount of residual for every project in each dataset. The residuals appear in the plot in case order. Obviously, when the size of the dataset increases, the amount of error rises, and the regression line efficiency decreases. That is why the small size of real datasets in comparison to artificial ones that have adjustable sizes, is a major and important reason for their weakness. The results of applying Wilcoxon test to the regression model' absolute residuals are shown in Table 13. Since the p-value obtained from the Wilcoxon signed rank test is less than 0.05 for all the comparisons, the existence of a significant difference between the regression estimates and those achieved by the other estimators is substantially proved.

Table 13
P-values of Wilcoxon test on absolute residuals

Dataset	MLR	ROR	SWR
DPS	0.02	0.01	0.03
CF	0.04	0.00	0.02
Desharnais	0.04	0.01	0.02
Maxwell	0.03	0.00	0.04
ISBSG	0.03	0.01	0.04

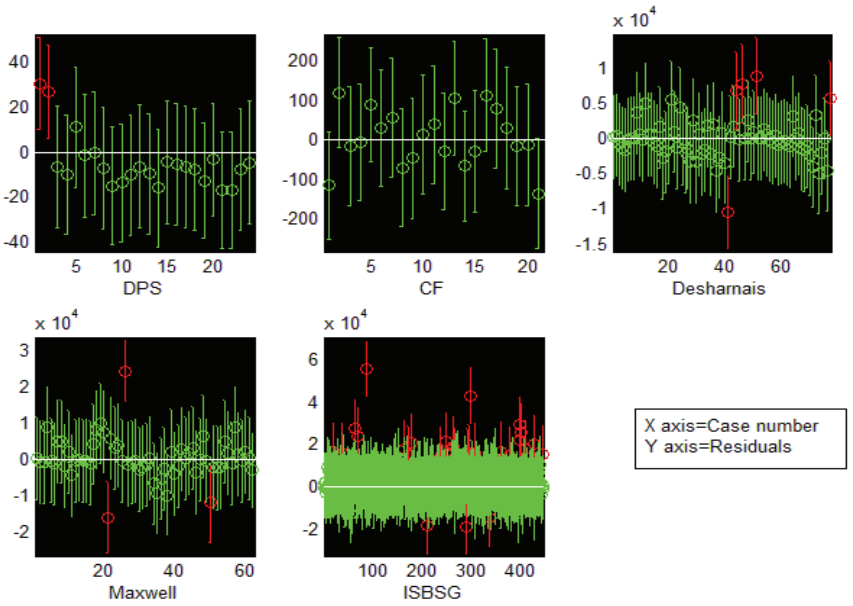

Figure 7: Residual charts of different effort estimation datasets

VII. Conclusions And Future Research

Software service development process is complex and risky due to the special features of software, and we have to use historical data and past experiences to improve this process and make it more accurate. Software datasets help researchers in making accurate estimations of the required costs and time for software development. These datasets include information on software projects that were completed in the past. Unfortunately, the large number of these datasets, and the lack of sufficient documentation related to them have made it difficult to develop estimation models. Selection of an unsuitable dataset will lead to obtaining unreal and biased results. This article performed a thorough statistical analysis of the data related to the most popular effort estimation datasets. The most important topics dealt with included descriptive statistics, correlation coefficients, data distribution, data visualization, and regression analysis.

Finally, a complete comparison was made to select a suitable dataset. Obtained figures and results showed tangible differences between the various repositories. To the best of our knowledge this is the first study that investigates the effects of dataset size, metrics set, and the feature selection techniques for software effort prediction problem. Furthermore, we employed several algorithms and views belonging to a new software engineering paradigm called effort estimation. This study showed that the most crucial component in software effort prediction is the metrics suite and not the algorithm. Future research can tackle the thematic and semantic analysis of every feature belonging to each of the studied datasets. A systematic review of the datasets can also be useful.

ACKNOWLEDGEMENT

We would like to express our gratitude to the Science and Research Branch of the Islamic Azad University of Tehran and the Tehran Polytechnic University for their spiritual support in conducting this research.

References

[1] Kocaguneli, E., T. Menzies, and J.W. Keung, *On the value of ensemble effort estimation.* IEEE Transactions on Software Engineering, 2012. **38**(6): p. 1403-1416.

[2] Dejaeger, K., et al., *Data mining techniques for software effort estimation: A comparative study.* IEEE Transactions on Software Engineering, 2012. **38**(2): p. 375-397.

[3] Bardsiri, V.K., et al., *A PSO-based model to increase the accuracy of software development effort estimation.* Software Quality Journal, 2013. **21**(3): p. 501-526.

[4] Bardsiri, V.K., et al., *A flexible method to estimate the software development effort based on the classification of projects and localization of comparisons.* Empirical Software Engineering, 2014. **19**(4): p. 857-884.

[5] Benala, T.R., et al. *Software Effort Estimation Using Data Mining Techniques*. in *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol I*. 2014. Springer.

[6] Shepperd, M., D. Bowes, and T. Hall, *Researcher bias: The use of machine learning in software defect prediction.* IEEE Transactions on Software Engineering, 2014. **40**(6): p. 603-616.

[7] Trendowicz, A. and R. Jeffery, *Principles of Effort and Cost Estimation*, in *Software Project Effort Estimation*. 2014, Springer. p. 11-45.

[8] Bardsiri, A.K. and S.M. Hashemi, *Software Effort Estimation: A Survey of Well-known Approaches.* International Journal of Computer Science Engineering (IJCSE), 2014. **3**(1): p. 46-50.

[9] Turhan, B., *On the dataset shift problem in software engineering prediction models.* Empirical Software Engineering, 2012. **17**(1-2): p. 62-74.

[10] Lavazza, L. and L. Santillo. *Historical Data Repositories in Software Engineering: Status and Possible Improvements*. in *Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement (IWSM-MENSURA),*

2012 Joint Conference of the 22nd International Workshop on. 2012. IEEE.
[11] Menzies, T. and M. Shepperd, *Special issue on repeatable results in software engineering prediction.* Empirical Software Engineering, 2012. **17**(1): p. 1-17.
[12] Bardsiri, A.K. and S.M. Hashemi, *Electronic Services, the Only Way to Realize the Global Village.* International Journal of Mechatronics, Electrical and Computer Technology, 2013. **3**(6): p. 1039-1041.
[13] Hashemi, S.M., M. Razzazi ,and M. Teshnehlab, *LEVERAGING THE STREAMLINED E-GOVERNMENTS, E-COMMERCE, AND E-BUSINESSES SERVICES THROUGH ISRUP E-SERVICE FRAMEWORK*. Service Sciences, Management and Engineering, IBM, 2006.
[14] Hashemi, S.M. and M. Razzazi, *Global Village Services as the Future of Electronic Services*. 2011: Lambert Academic Publishing.
[15] Hashemi, S.M., M. Razzazi, and M. Teshnehlab, *Streamlining the Global Village Grid Services.* World Applied Sciences Journal, 2008. **3**(5): p. 824-832.
[16] Liebchen, G.A. and M. Shepperd. *Data sets and data quality in software engineering*. in *Proceedings of the 4th international workshop on Predictor models in software engineering*. 2008. ACM.
[17] Rodriguez, D., I. Herraiz Tabernero, and R. Harrison, *On software engineering repositories and their open problems.* 2012.
[18] Déry, D. and A. Abran. *Investigation of the effort data consistency in the ISBSG repository*. in *Proceedings of the 15th Intern. Workshop on Software Measurement*. 2005.
[19] Cheikhi, L. and A. Abran. *PROMISE and ISBSG Software Engineering Data Repositories: A Survey*. in *Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2013 Joint Conference of the 23rd International Workshop on*. 2013. IEEE.
[20] ISBSG. *International Software Benchmarking standard Group*. 2011; Available from: www.isbsg.org.
[21] Matson, J.E., B.E. Barrett, and J.M. Mellichamp, *Software development cost estimation using function points.* IEEE Transactions on Software Engineering, 1994. **20**(4): p. 275-287.
[22] Abran, A. and P.N. Robillard, *Function points analysis: an empirical study of its measurement processes.* IEEE Transactions on Software Engineering, 1996. **22**(12): p. 895-910.
[23] Desharnais, J., *Analyse Statistique De La Productivitie Des Projets Informatique A Partie De La Technique Des Point Des Foncti On*. 1989, University of Montreal.
[24] Maxwell, K., *Applied statistics for software managers*. 2002: Prentice Hall.
[25] Hayes, W., *Statistics (fih edition)*. 1994, New York: Harcourt Brace.
[26] la Mendes, E., *Cost Estimation techniques for web projects.* 2008.
[27] Nassif, A.B., D. Ho, and L.F. Capretz, *Towards an early software estimation using log-linear regression and a multilayer perceptron model.* Journal of Systems and Software, 2013. **86**(1): p. 144-160.