Asian Journal of Information Technology 18 (3): 124-132, 2019
ISSN: 1682-3915

Performance Calculation and Benchmarking using the ISBSG Dataset Release 12 Data Repository: Empirical Study

[1]Shadi Mohammad Alkhatib, [1]Hala Hani Abuawwad, [1]Khaled Almakadmeh,
[1]Khalid Al-Sarayreh and [2]Ahed J. Alkhatib
[1]Department of Software Engineering,
Faculty of Prince Al-Hussein Bin Abdullah II for Information Technology,
Hashemite University, Zarqa, Jordan
[2]Department of Legal Medicine, Toxicology of Forensic Science and Toxicology,
School of Medicine, Jordan University of Science and Technology, Irbid, Jordan

Abstract: The International Software Benchmarking Standards Group (ISBSG) maintains a software development repository with 6,006 software projects. The definition of productivity is a single ratio of output to input and then combined with various cost factors leading to a single value. Of these values we have dataset makes it possible to calculate the productivity of projects, effort, size and quality. By contrast, the concept of performance is more comprehensive than productivity. This study explores a comparison between performance and productivity and how it can affect projects by several other factors that affect its using ISBSG dataset V.12. In this research, tree data analysis techniques were applied: data clustering, neural network. SPSS was used to conduct statistical analysis and data visualization.

Key words: ISBSG, performance, productivity, effort, size, quality, neural network, data clustering, data visualization

INTRODUCTION

Currently, the science of software engineering is one of the most important sciences because it uses systematic and experimental research to make software engineering experts and stakeholders able to make decisions until the confidence stage is reached in software products. Example productivity of projects, effort and size are all these in the collection leading to the purposes of measuring productivity and performance (Sameh and Al-Masri, 2019).

In companies that have a good program for measuring software, the basic productivity level is usually used to estimate the effort, costs and size required for any future project. Productivity is defined as the amount of output produced per unit used because there is more than one unit to measure productivity and size and each has a different measurement unit such as IFPUG (IFPUGI., 2004) and the COSMIC function points (Ferrucci *et al*., 2014), for example are used to measure productivity. However, the concept of performance is more comprehensive than the concept of productivity; performance has been defined as “the degree to which a system or component achieves its specific functions within certain constraints”. Performance is not the main output of the project but takes into account several other outcomes of the project. This performance concept is included in some Software Process Improvements (SPI) forms such as CMMI (Team, 2006).

In addition, to what was mentioned earlier, work focused on data characterization of ISBSG projects using data analytics (Buglione and Abran, 2002; Abran *et al*., 2005, 2015) and using the same techniques as in this study. More specifically, many research works such as by Buglione and Abran (2001, 2005), Meridji *et al*. (2017), Fernandez-Diego and Gonzalez-Ladron-De-Guevara (2014) proposed several models in this domain to conduct a systematic and empirical research using ISBSG to improve the software product quality with the minimum time and cost and find the difference between performance and quality.

Literature review: Top *et al*. (2011) presents the observed difficulties in the utilization of external and multi-organizational software benchmark repositories for effort estimation model construction for a software organization in the finance domain. ISBSG, Albrecht, China, Desharnais, Finnish, Maxwell and Kemerer repositorie’s data were utilized in this study. The approach was the utilization of these repositories and organization’s own repository to estimate the software development effort and evaluate whether external and

Corresponding Author: Ahed J. Alkhatib, International Mariinskaya Academy, Department of Medicine and Critical Care, Department of Philosophy, Academician Secretary of Department of Sociology Tel: 00962795905145

multi-organizational data be used for effort estimation. In addition, Quesada-Lopez and Jenkins (2014) reported on a replicated study carried out on a subset of the ISBSG dataset to evaluate the structure and applicability of function points.

The goal of this replication was to aggregate evidence and confirms results reported about internal issues of FPA as a metric using a different set of data. First, we examined FPA counting in order to determine which Base Functional Components (BFC) were independent of each other and thus, appropriate for an additive model of size. Second, we investigated the relationship between size and effort.

Fernandez-Diego *et al*. presents a snapshot of the existing usage of ISBSG in software development research. ISBSG offers a wealth of information regarding practices from a wide range of organizations, applications and development types which constitutes its main potential. However, a data preparation process is required before any analysis. Lastly, the potential of ISBSG to develop new research is also outlined. However, Meridji *et al*. (2017) explore empirically only the software Development Projects of Renewable Energy Applications in the ISBSG dataset V.13 based on software project factors such as effort and teamwork size to define the correlations between them. In this research, three data analysis techniques were applied: statistical analysis, data clustering and datavisualization. Both SPSS and RapidMiner are used to conduct statistical analysis and data visualization. Stroian *et al*. (2014) used a white-box tool, “MultiPERF”, based on the international ISBSG repository of software project data is proposed for setting performance targets in software organizations. Gallego and Sicilia (2012) used black box estimation tools to use projects from the ISBSG dataset. To achieve that goal three steps, the data set were analyzed, estimation was made, then tested using a software estimation tool. As a result for the majority of project, the black box estimation was far to be accurate. Gencel and Buglione (2016) conducted an empirical study using the data in the International Software Benchmarking Standards Group (ISBSG) database.

However, Cheikhi and Abran (2013) proposed an approach to software estimation based on productivity models with fixed/variable costs and economies/ diseconomies of scale. The study looks first at productivity alone as a single variable model and then discusses multi-variable models for estimation in specific contexts. An empirical study in a Canadian organization that illustrated the contribution of these concepts from economics in developing tailor-made estimation models based on the performance of the organization studied is presented as well as the use of the SWEBOK Guide for the identification of process improvements areas. In addition, Abran *et al*. (2015) provided additional information on these datasets by identifying the topics addressed, highlighting the availability of the data file and of the description of attributes related to the datasets and indicating their usefulness for benchmarking studies.

MATERIALS AND METHODS

Research objectives and methodology: The International Software Benchmarking Standards Group (ISBSG) maintains a software development repository more than 6,000 software projects developed from 32 different countries for 7 major industry types, different IT and metrics organizations submit their software project data to the ISBSG. It contains the following data such as productivity of projects, effort, size and quality. Data can be used for estimation, benchmarking, improvement and management of projects.

Research objectives: This research aimed to find the relationship between performance and productivity and how it can affect projects. On the basis of this relationship, projects were classified into two categories, first category development project and second category enhancement project and based on this classification, the relationship is going to be found using ISBSG dataset V.12.

Research methodology: The proposed research methodology used in this study is composed of eight main steps as follows.

Step 1: Selection of project applications of new development project from ISBSG, more specifically, IFPUG method.

Step 2: Selection of attributes performance, productivity, size and more attributes.

Step 3: Filtering data through ISBSG V12.

Step 4: Data is analyzed on the types of models and through the results, we can establish a relationship between performance and productivity.

Step 5: Analysis of data based on data clustering model.

Step 6: Analysis of data based on the neural network model.

Step 7: Presentation of research results and discussions.

Figure 1 illustrates the detailed research methodology conducted to achieve the research objectives as follows: read the ISBSG V12 Excel sheet, select the required attributes are performance and productivity, filter with projects can be divided into development or enhanced projects, then detect the extreme programs based on selected features will be used techniques in the end to obtain results, for example, data analysis based on data clustering, neural network.

Table 1: Quantitative fields from ISBSG R12

Groups/Measures	Descriptions	Id.	#
Sizing			
Functional size	Unadjusted function points	UFP	M1
Effort			
Normalised work effort	Full life-cycle effort for all teams reported	NEW	M2
Productivity			
Normalised PDR (ufp)	Normalised productivity delivery rate in hours per functional size unit	NPDR	M3
Schedule			
Project elapsed time	Total elapsed time for the project in calendar-months	PET	M4
Quality			
Total defects delivered	Total number of defects reported in the first month of use of the software	TDD	M5
Effort attributes			
Max team size	The maximum number of people who worked at any time on the project (peak team size)	MTS	M6
Average team size	The average number of people who worked on the project	ATS	M7
Size attributes (IFPUG/NESMA/mark II)			
Input count	The unadjusted size of external inputs	EI	M8
Output count	The unadjusted size of external outputs	EO	M9
Enquiry count	The unadjusted size of external inquiries	EQ	M10
File count	The unadjusted size of internal logical files	ILF	M11
Interface count	The unadjusted size of external interface files	EIF	M12
Size attributes (all methods)			
Added	The unadjusted size of additions	UFPA	M13
Changed	The unadjusted size of changes	UFPC	M14
Deleted	The unadjusted size of deletions	UFPD	M15
Size (other than FSM)			
Lines of code	The no. of the Source Lines of Code (SLOC) produced by the project	LOC	M16

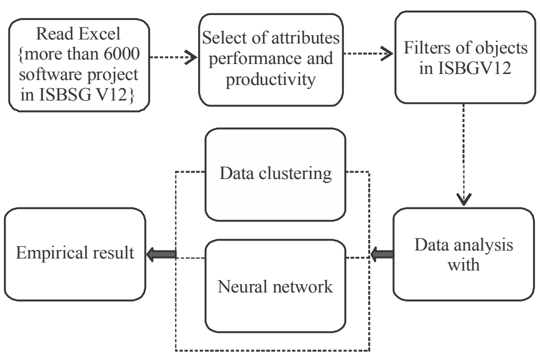

Fig. 1: Detailed methodology

ISBSG R12 data repository: In this study, we explain how key factors were taken to reach performance. All the factors were taken from ISBSG R12.

Selection of data samples in the ISBSG repository: R12 get a comparison in the performance of projects in the ISBSG repository, a common set of indicators affect performance (defined in terms of ratios) was selected from such list of measures to represent the three perspectives (E, S, T) (IFPUGI., 2004). Table 1 lists the quantitative fields from the ISBSG R12 repository that can be useful for creating sets of indicators. For the purposes of this study, the projects selected had to meet the following criteria:

- A class “A” or “B” data quality rating
- The functional size value considered was UFP

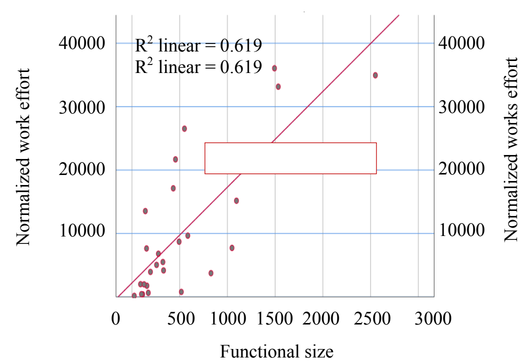

Fig. 2: Size vs. effort (new dev. projects, n26); Dual Y axis scale X axis of normalised work effort value of normalised work effort by function size

- Non-blank fields for the number of defects detected within one month of the delivery of the software product
- Non-blank fields for the detail on the No. of BFC (Base Functional Components) according to the functional size measurement method chosen

Table 2 shows the results from the data preparation process used to determine the samples on which the performance values were calculated according to ISBSG R10 data calculation rules. Now we will present the results that their factors were previously selected from the SPSS common measures (Fig. 2).

Table 2: Preparation of ISBSG R12 data: IFPUG projects

Steps	Attributes	Filters
1	Functional sizing approach13	= IFPUG
2	Data Quality Rating (DQR)	= {A | B}
3	UFP rating	= {A | B}
4	Total defects delivered	= {non-blanks}
5	IFPUG BFC (EI, EO, EQ, ILF, EIF)	= {non-blanks}
6	Project elapsed time	= {non-blanks}
7	Max team size	{non-blanks}
8	Average team size	= {non-blanks}
9	Development type	= {New development}
		= {Enhancement}
		= {Re-development}

Table 3: Samples from the ISBSG R12 repository

Dev. type/FSM methods	IFPUG (projects)
New development	26
Enhancement	52

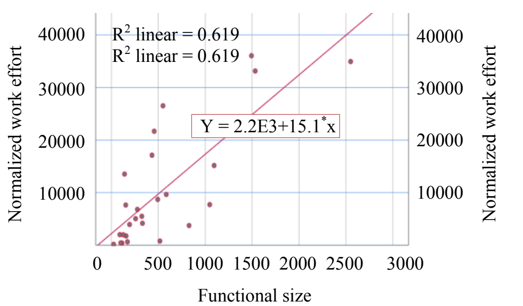

Fig. 3: Size vs. effort (new dev. projects, n26); Dual Y axis with scale X axis of normalised work effort value of normalised work effort by function size

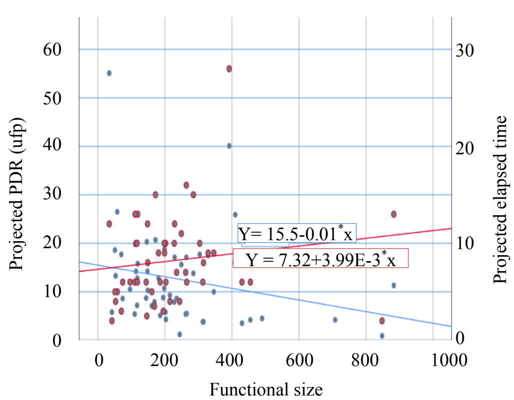

Fig. 4: Size vs. performance (Enh. Projects, n52); Dual Y axis with scale X axis of normalised PDR (ufp), value of project elapsed time by function size

The first relationship to investigate is the one between functional size and effort. The strongest relationship is in the sample of new development projects measured (n = 26) with $R^2 = 0.619$ (Table 3).

The second relationship to investigate is between functional size and effort. The lowest relationship, here, exists in the enhancement project sample (n = 52) with $R^2 = 0.217$ (Fig. 3 and 4).

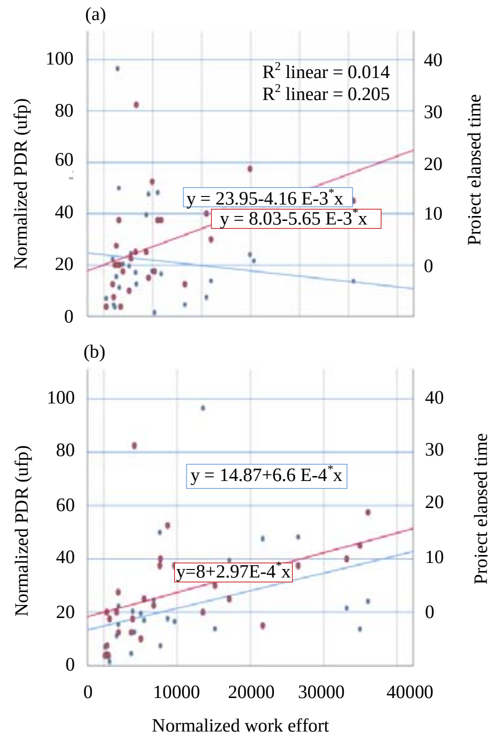

Fig. 5: Effort vs. performance (new dev. projects, n26): (a) and (b) Dual Y axis with scale X axis of normalised PDR (ufp), value of project elapsed time by function size

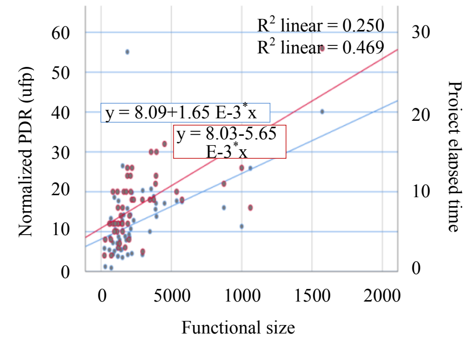

Fig. 6: Effort vs. performance (Enh. projects , n52); Dual Y axis with scale X axis of normalised PDR (ufp), effort value of project elapsed time by normalised work

The third relationship to investigate is between functional size and effort. The lowest relationship, here, exists in the new. dev project sample (n = 26) with $R^2 = 0.205$ and $R^2 = 0.014$ (Fig. 5 and 6).

The fourth relationship to investigate is between functional size and effort. The lowest relationship, here, exists in the enhancement project sample (n = 52) with $R^2 = 0.055$ and $R^2 = 0.027$.

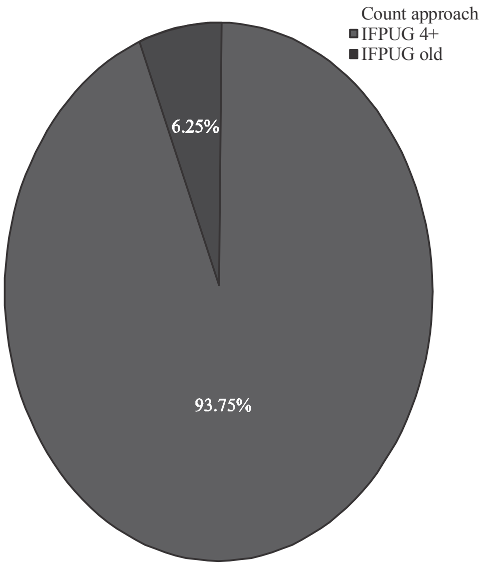

Fig. 7: Type of count approach

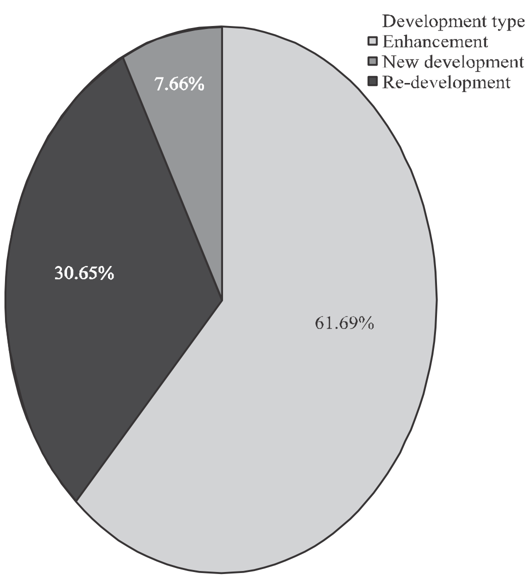

Fig. 8: Type of count approach

The fifth relationship to investigate is between functional size and effort. The lowest relationship, here, exists in the new. dev project sample (n = 26) with $R^2 = 0.130$ and $R^2 = 0.209$.

The sixth relationship to investigate is between functional size and effort. The lowest relationship, here, exists in the enhancement project sample (n = 52) with $R^2 = 0.250$ and $R = 0.469$ (Fig. 7).

In this Fig. 8 shows the distribution of count approach of all projects and distributed ratio of 100%. In this Fig. 9 shows the distribution of count approach of all projects and distributed ratio of 100%.

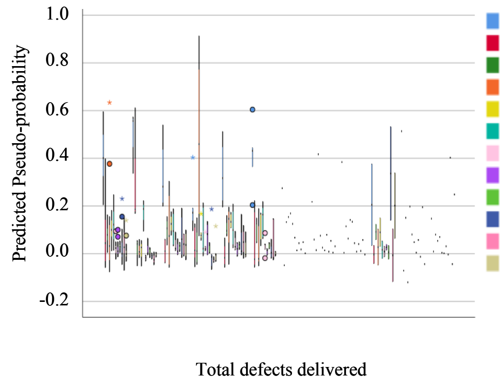

Fig. 9: The relationship between total defects delivered and predicted

RESULTS AND DISCUSSION

Experimental results using different technique

First technique using neural network: The data should be analyzed to determine the main factor influencing performance calculation. To calculate performance you must specify a set of factors to calculate such as sizing, effort, productivity and size attributes, etc. The name of the data set used should be the International Software Benchmarking Standards Group (ISBSG), the analysis process using a multi-layered algorithm.

Select one of output is total defects delivered and all this under is input: In this study we need to analyze the International Software Benchmarking Standards Group (ISBSG) data. The analysis process will be based on one output, total defects delivered. After the analysis, we will discover that this factor is the factor in the performance calculation and is there any other factor or not. The analysis process uses a multi-layered algorithm (Table 4).

In the previous Table 4, we provide a complete summary of the samples and the status taken from the International Software Benchmarking Standards Group (ISBSG) conducted on the total operations: training, testing and results can be tracked in percentages.

In the previous Table 5, we provide a complete summary of the information of the International Software Benchmarking Standards Group (ISBSG). In this table, we summarize the process used, i.e., input, processing and output process. In the first stage, the inputs used by age attributes, messages, friends, etc. In the second stage, this stage divides the data into layers and is divided into 6a layers. In the last stage, this stage is called output and this stage depends on the main factor in the classification total defects delivered (Fig. 10).

Table 4: Case processing summary

Case processing	No. of tests	Percentage
Sample		
Training	38	70.7
Testing	11	29.3
Valid	49	100.0
Excluded	5960	-
Total	6009	-

Table 5: Network information (Complete summary of samples)

Variable	Results	Information processing
Input layer	**Factors**	
	1	Max team size
	2	Interface count
	3	Deleted count
	4	Data quality rating
	5	UFP rating
	6	Count approach
	Covariates	
	1	Functional size
	2	Normalised work effort
	3	Project elapsed time
	4	Average team size
	5	Input count
	6	Output count
	7	Enquiry count
	8	File count

No. of units: 41, Rescaling method for covariates: Standardized, Hidden layer (s), No. of units: 6a, Activation function: Softmax, Output layer, Dependent variables: 1, Total defects delivered, Number of units: 12, Activation function: Identity, Error function: Sum of squares

Table 6: Model summary

Information processing	Results
Training	
Sum of squares error	13.110
Percent incorrect predictions	57.9%
Training time	0:00:00.05
Testing	
Sum of squares error	3.763a
Percent incorrect predictions	45.5%

Dependent variable

Total defects delivered: The No. of hidden units is determined by the testing data criterion: The "best" number of hidden units is the one that yields the smallest error in the testing data (Table 6).

In the previous Table 6, we provide a complete summary of the International Software Benchmarking Standards Group (ISBSG). At this stage, it is divided into two parts: the first training in this section contains several processes including the percentage of false predictions, time factor in the division stage, etc. In the second section is the percentage of error tests and the percentage prediction of the dependent variable is total defects delivered.

In the previous Table 7 and 8, the variables used to analyze data from ISBSG are illustrated. These variables depend mainly on the analysis process and are of great importance. The preceding table shows the variables used.

In the previous Fig. 11 and 12, it is one of the most important schemes because it explains the most important variables and what is the most important factor.

Table 7: Independent variable importance

Predictor variables	Normalized importance (%)
Environmental data	
Max team size	26.0
Interface count	16.6
Deleted count	26.3
Data quality rating	30.4
UFP rating	25.4
Count approach	14.2
Functional size	45.0
Normalised work effort	72.0
Project elapsed time	10.0
Average team size	71.0
Input count	65.3
Output count	67.1
Enquiry count	53.3
File count	81.7

Table 8: Network information analyze data from ISBSG

Variables	Results	Information processing
Input layer	**Factors**	
	1	Max team size
	2	Interface count
	Count	
	3	Deleted
	4	Data quality rating
	5	UFP rating
	6	Count
Approach	**Covariates**	
	1	Functional size
	2	Normalised work effort
	3	Project elapsed time
	4	Average team size
	5	Input count
	6	Output count
	7	Enquiry count
	8	File count

Number of units: 35, Rescaling method for covariates: Standardized, Hidden layer (s), Number of units: 8a, Activation function, Softmax, Output layer, Dependent variables: 1, Normalised PDR (ufp), Number of units: 1, Rescaling method for scale dependents: Normalized, Activation function: Identity, Error function: Sum of squares

Depending on the chart, the information or factors are explained more clearly than the tables. Normalized work effort is the most important factor among all factors.

In the previous Table 8, we provide a complete summary of the information of the International Software Benchmarking Standards Group (ISBSG). In this table, we summarize the process used, i.e., input, processing and output process. In the first stage the inputs used by age attributes, messages, friends, etc. In the second stage, this stage divides the data into layers and is divided into 6a layers. In the last stage, this stage is called output and this stage depends on the main factor in the classification total defects delivered.

In the previous Table 9, we provide a complete summary of the International Software Benchmarking Standards Group (ISBSG). At this stage, it is divided into two parts: the first training in this section contains several processes including the percentage of false predictions, time factor in the division stage, etc. In the second section

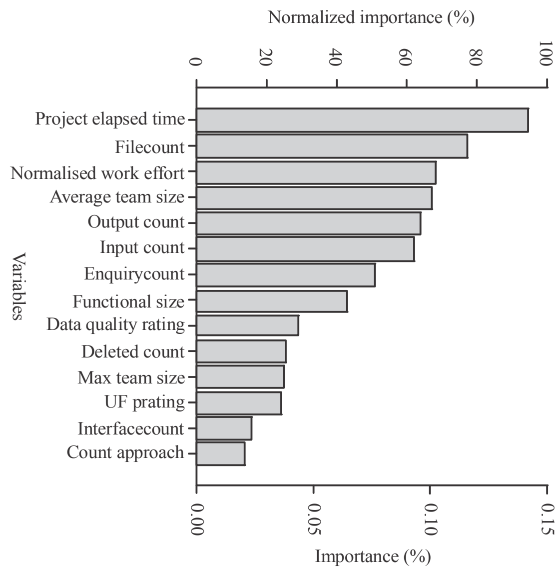

Fig. 10: Model summary (The main factors classification)

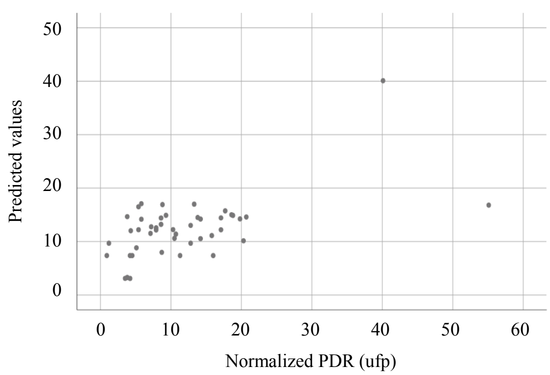

Fig. 11: The relationship between normalised PDR(ufp) and predicted

Table 9: Model summary (ISBSG)

Information processing	Results
Training	
Sum of squares error	0.426
Relative error	0.658
Training time	0:00:00.05
Testing	
Sum of squares error	0.054a
Relative error	1.169

is the percentage of error tests and the percentage prediction of the dependent variable is normalized PDR (ufp) (Table 10).

Dependent variable

Normalised PDR (ufp): The No. of hidden units is determined by the testing data criterion: The “best” number of hidden units is the one that yields the smallest error in the testing data.

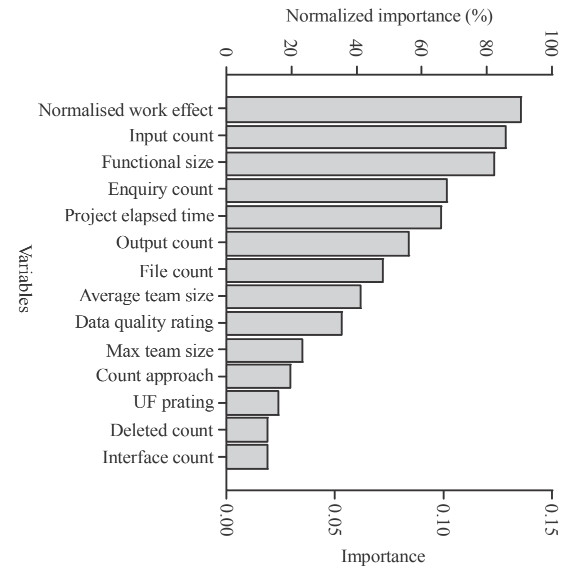

Fig. 12:Model summary

Table 10: Independent variable importance

Predictor variables	Normalized importance (%)
Environmental data	
Max team size	26.1
Interface count	13.8
Deleted count	14.1
Data quality rating	38.9
UFP rating	17.5
Count approach	21.7
Functional size	90.7
Normalised work effort	100.0
Project elapsed time	72.2
Average team size	45.3
Input count	93.8
Output count	61.8
Enquiry count	74.4
File count	53.1

In the previous Table 11, the variables used to analyze data from ISBSG are illustrated. These variables depend mainly on the analysis process and are of great importance. The preceding table shows the variables used.

In the previous Fig. 13 and 14, it is one of the most important schemes because it explains the most important variables and what is the most important factor. Depending on the chart, the information or factors are explained more clearly than the tables. Normalized work effort, the most important factor among all factors.

Second technique using data clustering with SPSS: k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into cells. In this study, the data must be analyzed using another method, namely the analysis of the k-means group (Table 12 and 13).

Table 11: Initial cluster centers

Predictors	No. of clusters 1	2	3	4	5	6	7	8	9	10
Normalised work effort	26520	18002	21684	33140	36046	13528	167	10623	7643	4516
Project elapsed time	15.0	9.0	6.0	16.0	23.0	8.0	1.5	8.0	15.0	16.0
Functional size	550	788	456	1533	1493	140	24	410	153	264
Normalised PDR (ufp)	48.2	22.8	47.6	21.6	24.1	96.6	7.0	25.9	50.0	17.1

Table 12: Final cluster centers

Predictors	No. of clusters 1	2	3	4	5	6	7	8	9	10
Normalised work effort	26520	17303	20890	33140	35497	14807	1295	10093	7924	4064
Project elapsed time	15.0	11.5	6.0	16.0	20.5	16.0	6.8	12.0	14.4	10.5
Functional size	550	547	792	1533	2022	542	192	626	503	332
Normalised PDR (ufp)	48.2	34.1	32.7	21.6	18.9	50.2	9.9	17.9	23.1	15.0

Table 13: Number of cases in each cluster

No. of clusters	Results
Clusters	
1	1.000
2	3.000
3	2.000
4	1.000
5	2.000
6	3.000
7	46.000
8	3.000
9	5.000
10	18.000
Valid	84.000
Missing	5925.000

CONCLUSION

This study analysis was completed for the new development projects enhancement based on IFPUG method. The ISBSG data set was used to extract the new development projects on the basis of selected factors and defined criteria's. Three steps methodology were applied; project applications selection, the research criteria's, data analytics was performed and results were presented. Traditional cost estimation models in software engineering are mainly based on the concept of productivity while the usage of performance could provide benefits being a more mature and comprehensive concept. About 2 data samples from IFPUG projects were used (NewDev-Enh) for calculating p values. New analysis and investigations will be performed using ISBSG data on: the impact of relationships of various variables on the performance results themselves, the same analysis by size ranges.

REFERENCES

Abran, A., J.M. Desharnais, M. Zarour and O. Demirors, 2015. Productivity-based software estimation models and process improvement: An empirical study. Intl. J. Adv. Software, 8: 103-114.

Abran, A., L. Buglione and D. Girard, 2005. R-LIME: Improving the risk dimension in the LIME model. Proceedings of the 3rd International World Conference on Software Quality (3WCSQ), September 26-30, 2005, Munich, Germany, pp: 25-29.

Buglione, L. and A. Abran, 2001. Multidimensionality in software performance measurement: The qest/lime models. Proceedings of the SSGRR2001 2nd International Conference on Advances in Infrastructure for Electronic Business, Science and Education on the Internet, August 10, 2001, L'Aquila, Italy, pp: 1-27.

Buglione, L. and A. Abran, 2002. QEST nD: N-dimensional extension and generalisation of a software performance measurement model. Adv. Eng. Software, 33: 1-7.

Buglione, L. and A. Abran, 2005. A model for performance management and estimation. Proceedings of the 11th IEEE International Symposium on Software Metrics, September, 19-22, 2005, Como, Italy, pp: 1-9.

Cheikhi, L. and A. Abran, 2013. PROMISE and ISBSG software engineering data repositories: A survey. Proceedings of the 23rd and 8th Joint International Conference on Software Measurement and Software Process and Product Measurement, October 23-26, 2013, IEEE, Ankara, Turkey, pp: 17-24.

Fernandez-Diego, M. and F. Gonzalez-Ladron-De-Guevara, 2014. Potential and limitations of the ISBSG dataset in enhancing software engineering research: A mapping review. Inf. Software Technol., 56: 527-544.

Ferrucci, F., C. Gravino and F. Sarro, 2014. Exploiting prior-phase effort data to estimate the effort for the subsequent phases: A further assessment. Proceedings of the 10th International Conference on Predictive Models in Software Engineering, September 17, 2014, ACM, Turin, Italy, ISBN:978-1-4503-2898-2, pp: 42-51.

Gallego, J.J.C. and M.A. Sicilia, 2012. An algorithm for the generation of segmented parametric software estimation models and its empirical evaluation. Comput. Inf., 26: 1-15.

Gencel, C. and L. Buglione, 2016. The missing links in software estimation: Team loading and team power. Proceedings of the 2016 Joint International Workshop on Software Measurement and Software Process and Product Measurement (IWSM-MENSURA), October 5-7, 2016, IEEE, Berlin, Germany, ISBN:978-1-5090-4148-0, pp: 212-212.

IFPUGI., 2004. Function point counting practices manual, Release 4.2. International Function Point Users Group, Mequon, Wisconsin, USA. https:// www.academia.edu/34111808/ Function_Point_Counting_Practices_Manual_Release_4.2.1

Meridji, K., K.T. Al-Sarayreh, M. Abu-Arqoub and W.M. Hadi, 2017. Exploration of development projects of renewable energy applications in the ISBSG dataset: Empirical study. Proceedings of the 2017 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes & Systems (IT-DREPS), December 6-7, 2017, IEEE, Amman, Jordan, ISBN:978-1-5386-1987-2, pp: 1-6.

Quesada-Lopez, C. and M. Jenkins, 2014. Function point structure and applicability validation using the ISBSG dataset: A replicated study. Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, September 18-19, 2014, ACM, Torino, Italy, ISBN:978-1-4503-2774-9, pp: 1-1.

Sameh, A. and A. Al-Masri, 2019. Smartphones network connections power-aware multiple wireless interfaces. Asian J. Inf. Technol., 18: 37-48.

Stroian, V., P. Bourque and A. Abran, 2014. A white-box tool to set performance targets in software engineering management using theISBSG repository. Proceedings of the 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, May 22-24, 2014, IEEE, Cluj-Napoca, Romania, ISBN:978-1-4799-3731-8, pp: 1-6.

Team, C.P., 2006. CMMI for development, Version 1.2: Representation, CMU/SEI-2006-TR-008, Technical Report. Software Engineering Institute, Pittsburgh, Pennsylvania.

Top, O.O., B. Ozkan, M. Nabi and O. Demirors, 2011. Internal and external software benchmark repository utilization for effort estimation. Proceedings of the 2011 Joint 21st and 6th International Conference on Software Measurement and Software Process and Product Measurement, November 3-4, 2011, IEEE, Nara, Japan, ISBN:978-1-4577-1930-1, pp: 302-307.