Using Data Mining to Identify COSMIC Function Point Measurement Competence

Selami Bagriyanik[1], Adem Karahoca[2]
[1] Digital Learning Solutions Technology Department, Turkcell Technology, Turkey
[2] Department of Software Engineering, Bahcesehir University, Turkey

Article Info

Article history:

Received Feb 27, 2018
Revised Jun 18 2018
Accepted Jul 29, 2018

Keyword:

COSMIC funtion point
Educational data mining
Machine learning algorithms
Requirements analyst
Requirements artifacts

ABSTRACT

Cosmic Function Point (CFP) measurement errors leads budget, schedule and quality problems in software projects. Therefore, it's important to identify and plan requirements engineers' CFP training need quickly and correctly. The purpose of this paper is to identify software requirements engineers' COSMIC Function Point measurement competence development need by using machine learning algorithms and requirements artifacts created by engineers. Used artifacts have been provided by a large service and technology company ecosystem in Telco. First, feature set has been extracted from the requirements model at hand. To do the data preparation for educational data mining, requirements and COSMIC Function Point (CFP) audit documents have been converted into CFP data set based on the designed feature set. This data set has been used to train and test the machine learning models by designing two different experiment settings to reach statistically significant results. Ten different machine learning algorithms have been used. Finally, algorithm performances have been compared with a baseline and each other to find the best performing models on this data set. In conclusion, REPTree, OneR, and Support Vector Machines (SVM) with Sequential Minimal Optimization (SMO) algorithms achieved top performance in forecasting requirements engineers' CFP training need.

Corresponding Author:

Adem Karahoca,
Department of Software Engineering,
Bahcesehir University,
Faculty of Engineering, Besiktas, Istanbul, 34349, Turkey.
Email: adem.karahoca@eng.bau.edu.tr; akarahoca@gmail.com

1. INTRODUCTION

Requirements engineers are one of the key profiles within software development teams. They balances all project stakeholders' expectations from idea to post production phases. Requiremens engineering is not solely a technical discipline. Additionally, it also has an inter-disciplinary nature that concerns Cognitive Psychology, Anthropology, Sociology, Linguistics and Philosophy aspects of the subject [1]. Thus, they have a dramatical impact on the success of the software products and their continuous competence development is critical.

Functional size measurement (FSM) is an important task that is used for scoping, budgeting, managing outsourcing contracts, effort estimation, etc. This task is generally under the responsibility of system analysts or requirements engineers (REs). CFP is one of the recent FSM methods. Function Point variants are mainly used in software cost estimation [2] and productivity of the development organisations [3]. Function Point is also a good indicator in identifying business complexity of the software [4]. Additionally It's CFP variant is also a strong tool for requirement quality and process improvement [5]. CFP measurement errors, made by requirements engineers, leads budget, schedule and quality problems in software projects.

Therefore, it’s crucial to foresee and plan requirements engineers’ CFP training need in a quick and correct manner. A recent paper points out that CFP training need should be represented more in higher education [6]. We think training is also critical in the workplace setting and REs should be continually developed in CFP competence when need arises. Training is the dominating factor for quality improvement of FSM [7]. Factors that cause inconsistent and inaccurate CFP measurements might be improved by training [8]. In this study, requirements engineers CFP training need has been forecasted by using the artifacts they produced in the workplace and machine learning algorithms.

Data mining ors software analytics studies that use requirements engineering artifacts are scarce [9]. For example, one of these rare studies aims early test effort prediction by using UML diagrams [10]. On the other hand, software data mining studies which are based on sofware code and code change artifacts are common [9]. For instance, code smells in the source code have been investigated using Neural Network Models in a recent study [11]. We observe that using CFP data and data mining for Educational purposes is even more rare in the literature. As far as we know, this research is the first study in the literature that uses CFP data and educational data mining to improve REs’ CFP measurement capabilities. The rest of the paper is organized as follows: in the 2nd section, a background on Data Mining, Machine Learning Algorithms, CFP and study details are provided. Results are presented in the 3rd section and is followed by conclusions in the 4th section.

2. RESEARCH METHOD

In this section, first of all, machine learning and CFP methods are explained briefly in the subsections 2.1 and 2.2. Second, CFP training need prediction usecase, feature set design and data gathering and preparation phases of the study is presented in 2.3, 2.4 and 2.5. Finally, models training details and evaluation results are given in 2.6.

2.1. Data Mining and Machine Learning Algorithms

Data Mining is defined as “the process of discovering patterns, automatically or semi-automatically, in large quantities of data” [12]. Knowledge discovery from data (KDD) is another common term used in the literature [13]. Following algorithms which were implemented in Weka [14] are used in this study:

- ***Random Forest (RF):*** This is an ensemble learning method consisting of a set of decision tree classifiers. Each tree in the forest is triggered by an independently created random number vector [15].
- ***Naïve Bayes (NB):*** This method uses Bayes’ rule to do the classification by computing class probabilities and using observed attribute values. The method is called “naïve” since it has two basic assumptions: attributes are conditionally independent and no hidden factor impacts on the prediction process [16].
- ***REPTree:*** This is a fast decision tree algorithm that generates a decision tree using information gain method to split [17]. Missing values are managed as in C4.5 algorithm [18].
- ***J48:*** It is a Java implementation of a slightly different version of C4.5 [17].
- ***LMT:*** Logistic Model Trees are standard decision trees which use logistic regression functions at their leaves [19].
- ***Multilayer Perceptron (MLP):*** MLP is a feed-forward artificial neural network which uses back propagation training algorithm. It is a system of interconnected nodes or neurons which maps an input vector into an output vector to maintain a nonlinear relation [20]. The neurons are connected via weights and output signals [20].
- ***Support Vector Machines (SVM) and Sequential Minimal Optimization (SMO):*** In linear case, an SVM is a hyperplane that set a boundary between some positive instances and negative instances [21]. It can also be further extended to non-linear cases [21]. Training an SVM requires quadratic programming (QP) optimization problem solving which is a very time and memory consuming operation and SMO is a substantial improvement on the original training algorithm [21].
- ***K-nearest Neighbour Classifier (IBk):*** It classifies a data point based on its k most similar other data points [22].
- ***ZeroR***: It predicts the majority class of nominal test data while it predicts the average value if numeric class is the case [12]. In this study, it will be used as a baseline for the performance of machine learning algorithms.
- ***OneR:*** This method classifies instances based on a one rule which is extracted from a single attribute [23].

2.2. COSMIC Function Point (CFP)

Software functional size measurement (FSM) has been in use for more than forty years [24]. There are many FSM methods [25]. COSMIC Function Point is a new generation software functional size measurement method. First version of the method was published in 1988 [26]. It's one of four ISO certified FSM methods which are dominating in the industry: IFPUG, COSMIC, NESMA and Mark II [25]. CFP measurement is a three-step process: measurement strategy, mapping and measurement steps. Purpose, scope, and level of granularity of the measurement are determined in the first step; in mapping phase, functional processes and data groups in the requirements are determined; in the final stage, data movements are specified and counted, for all functional processes [26]. CFP is the functional sizing measurement method that is used in the company under study [3,7,27].

2.3. CFP Requirement Ontology

Requirement artefacts that will be used in training the machine learning models are instances of requirement and CFP ontologies designed in [3]. Currently, this requirement ontology and CFP measurements are standard methods used by requirements engineers within the same telecommunications company in which this study is conducted. Periodically, a subset of all requirements documents are randomly selected and examined by internal audit team manually to identify errors in CFP measurements. After each audit, problematic CFP measurements are identified, recorded and potential learning needs are reported to requirements engineering management. By this data mining research, the manual examination process by audit team is intended to be semi-automated and learning opportunities will automatically be extracted from requirements documents.

2.4. Feature Set Extraction from CFP Requirement Ontology

CFP Ontology concepts are shown in the second column of the Table 1. Related concepts are categorised into concept categories to specify data indicators that will be used in data mining process. Ontology concept categories are shown in the first column of Table 1. As a result, features of the data and the predicted outcome (Class) of the classification process is shown Table 2. In Table 2, the first seven attributes are input attributes and the last one, "CFP Training Need", is class or Classification result.

Table 1. CFP Ontology Concept Categories

Ontology Concept Category	Ontology Concept
Use case	Use case
Use case	Application Interaction Diagram
Interaction	Interaction
Evolution Type	Add Evolution Type
Evolution Type	Modify Evolution Type
Evolution Type	Delete Evolution Type
Application	Application Business Module
Application	Application Database Module
Application	Application Service
Application	Application Service Boundary
Use case	Use case Actor
Use case	Use case Event
Information	Information Asset
Interaction	Integration Entry Interaction
Interaction	Integration Exit Interaction
Interaction	User Interface Entry Interaction
Interaction	User Interface Exit Interaction
Interaction	Database Write Interaction
Interaction	Database Read Interaction
Scope	Project Scope
Scope	Application Service Scope
Not Applicable	Productivity Measurement
Business Logic	Use case Business Logic
Business Logic	Interaction Business Logic

Table 2. CFP Ontology Indicator set and Their Possible Values

Ontology Concept Category	Value
Use case	Yes, No, Partial
Interaction	Yes, No, Partial
Information	Yes, No, Partial
Evolution Type	Yes, No, Partial
Business Logic	Yes, No, Partial
Application	Yes, No, Partial
Scope	Yes, No, Partial
CFP Training Need	Yes, No

2.5. Data Gathering and Preparation

First seven data attributes shown in Table 2 are obtained from requirements documents by checking whether each concept category value is existing or not. If their values are partially existing then the concept category value is recorded as “Partial”. The last attribute is captured from audit examination results. If the difference between the actual measurement done by requirements engineer and correct measurement result identified by audit team is greater than 5 % then this case is recorded as a learning opportunity by recording “CFP Training Need” value as “Yes”. 101 data points have been collected and results have been recorded in a comma-delimited values (.csv) file. Next, this file has been converted into the attribute-relation file format (.arff) which is the standard file format used by The Waikato Environment for Knowledge Analysis (WEKA) data mining software [14]. The conversion tool used for .csv to .arff is an online web tool [28].

2.6. Model Training and Evaluation

To train and evaluate the machine learning models, Weka Experimenter [29] has been used. Two experiments have been done in Weka. In the first experiment, “Data sets first” parameter checked and number of repetitions has been set as 100 in iteration control parameters panel. Experiment type is selected as Cross validation. Number of folds attribute is set to 10. Dataset has been selected as the .arff file which is created as described in section 2.5. All algorithms which have been explained in section 2.1have been selected in Algorithms panel. Next, the experiment with this configuration has been run on an Intel Core i7-5600U CPU, 2.6 GHz, 8 GB RAM and 64-bit Windows Operating System machine.

The total execution time was 194 seconds and MLP had the slowest running time. Finally, in analyse tab of Weka Experimenter user interface, all algorithms have been selected as test base separately and test is performed for each algorithm. Test has been repeated for three evaluation metrics: Accuracy (Number of Correct Classifications), F-Measure (FM), and Kappa statistic. In the second experiment, experiment type was set to Train/Test Percentage Split (data randomized) and train percentage was set to 66%. All other configuration remained the same as in Experiment 1. In this case, the total execution time was 74 seconds and MLP had the slowest running time, again.

3. RESULTS AND ANALYSIS

Table 3 designates the algorithm performances in terms of accuracy, FM, and Kappa metrics for both experiments. Metric values are shown as averages with standard deviations. All algorithms seem meaningfully better than ZeroR baseline performance. Support Vector Machines and OneR algorithms have the largest average accuracy values. However evaluating the algorithms solely based on the average values and standard deviations wouldn’t be sufficient since differences between results might not be statistically significant. Therefore, in Weka Experimenter Analyse interface, Significance has been set to 0.05, all algorithms have been selected as test base separately and tests have been performed. Statistically significant differences have been recorded during tests. Statistical significance is denoted by “v” and “*” symbols in the Weka interface. Former means statistically significant better performance while latter implies statistically significant worse performance [29].

Statistically significant superiorities between algorithms are shown in Table 4. For instance, in Experiment 1, Naïve Bayes performs better than IBk and ZeroR when Accuracy and Kappa metrics are concerned. Best performing algorithms have been determined by comparing the number of all statistically significant superiorities. We show this number as “Number of Wins” in Table 4. As a result; REPTree, OneR and SVM with SMO algorithms have the maximum “Number of Wins” values and are determined to be the top three algorithms performing best in CFP dataset of this study. As far as we know, this is the first study that use data mining on requirements and CFP measurement data. Therefore, we couldn’t compare the performance of our study with other similar research directly. However if we benchmark with some educational data mining studies in general, we see our top performing models are very good in terms of accuracy [30–33].

Table 3. Algorithm Performances for CFP Dataset

Algorithm	Experiment 1: Cross - validation			Experiment 2: 66% Split Test		
	Accuracy (%)	FM	Kappa	Accuracy	FM	Kappa
Random Forest (RF)	78.79 (11.72)	0.82 (0.11)	0.56 (0.25)	79.20 (5.54)	0.83 (0.04)	0.56 (0.12)
Naive Bayes (NB)	82.97 (11.13)	0.86 (0.09)	0.63 (0.24)	81.29 (5.46)	0.85 (0.04)	0.60 (0.12)
REPTree	84.02 (11.25)	0.86 (0.11)	0.67 (0.23)	84.27 (4.81)	0.86 (0.04)	0.68 (0.10)
J48 (Weka C 4.5 Implementation)	82.59 (11.39)	0.84 (0.11)	0.65 (0.23)	83.38 (4.58)	0.85 (0.04)	0.66 (0.09)
Logistic Model Trees (LMT)	83.84 (11.34)	0.86 (0.11)	0.67 (0.23)	83.52 (5.21)	0.86 (0.05)	0.66 (0.11)
Multilayer Perceptron (MLP)	76.74 (12.50)	0.80 (0.12)	0.52 (0.26)	75.94 (6.55)	0.80 (0.05)	0.49 (0.15)

Table 3. Algorithm Performances for CFP Dataset

Algorithm	Experiment 1: Cross - validation			Experiment 2: 66% Split Test		
	Accuracy (%)	FM	Kappa	Accuracy	FM	Kappa
Support Vector Machines with SMO	84.16 (11.25)	0.86 (0.11)	0.68 (0.23)	83.70 (4.91)	0.86 (0.04)	0.67 (0.10)
K-nearest Neighbour Classifier (IBK)	75.61 (11.56)	0.80 (0.10)	0.48 (0.25)	75.38 (5.16)	0.81 (0.04)	0.47 (0.12)
OneR	84.16 (11.25)	0.86 (0.11)	0.68 (0.23)	84.27 (4.81)	0.86 (0.04)	0.68 (0.10)
ZeroR	59.45 (01.64)	0.75 (0.01)	0.00 (0.00)	59.37 (0.63)	0.75 (0.04)	0.00 (0.00)

Table 4. Statistically Significant Superiorities of Algorithms for CFP Dataset

Algorithm	Experiment 1: Cross - validation			Experiment 2: 66% Split			Number
	Accuracy	FM	Kappa	Accuracy	FM	Kappa	of Wins
Random Forest (RF)	ZeroR	ZeroR	ZeroR	ZeroR	ZeroR	ZeroR	6
Naive Bayes (NB)	IBk, ZeroR	MLP, IBk, ZeroR	IBk, ZeroR	ZeroR	ZeroR	ZeroR	10
REPTree	RF, MLP, IBk, ZeroR	MLP, IBk, ZeroR	RF, MLP, IBk, ZeroR	IBk, ZeroR	IBk, ZeroR	IBk, ZeroR	17
J48 (Weka C 4.5 Implementation)	IBk, ZeroR	ZeroR	IBk, ZeroR	IBk, ZeroR	ZeroR	IBk, ZeroR	10
Logistic Model Trees (LMT)	MLP, IBk, ZeroR	MLP, IBk, ZeroR	MLP, IBk, ZeroR	IBk, ZeroR	ZeroR	IBk, ZeroR	14
Multilayer Perceptron (MLP)	ZeroR	ZeroR	ZeroR	ZeroR	ZeroR	ZeroR	6
Support Vector Machines with SMO	RF, MLP, IBk, ZeroR	MLP, IBk, ZeroR	RF, MLP, IBk, ZeroR	IBk, ZeroR	ZeroR	IBk, ZeroR	16
K-nearest Neighbour Classifier (IBk)	ZeroR	ZeroR	ZeroR	ZeroR	ZeroR	ZeroR	6
OneR	RF, MLP, IBk, ZeroR	MLP, IBk, ZeroR	RF, MLP, IBk, ZeroR	IBk, ZeroR	IBk, ZeroR	IBk, ZeroR	17
ZeroR	None	None	None	None	None	None	0

4. CONCLUSION

In this study, we conducted an educational data mining research. In the scope of this use case, a CFP dataset which was collected from a large telecommunications services and technology company has been analysed using 10 machine learning algorithms to identify CFP learning need of Requirements Engineers. After two experiments, model performances are evaluated and top performer algorithms have been identified. REPTree, OneR and SVM with SMO algorithms performed better than other algorithms in a statistically significant manner. Top performing model prediction performances are sufficient to be used in the production environment in the company. In the future, following research is planned:

- Dominating indicators in CFP measurement will be identified by using feature selection algorithms. Some new indicators from the requirements artifacts may arise in this process.
- Data points number will be increased and the study will be replicated by also adding some other algorithms such as Adaptive Neuro Fuzzy Inference System (ANFIS).

REFERENCES

[1] Nuseibeh B, Easterbrook S. “Requirements engineering: a roadmap”. In: *Proceedings of the conference on The future of Software engineering - ICSE ’00*. 2000. p. 35–46.

[2] Naik P, Nayak S. “Insights on Research Techniques towards Cost Estimation in Software Design.” *Int J Electr Comput Eng (IJECE)*. 2017;7(5):2883–94.

[3] Bagriyanik S, Karahoca A. Automated COSMIC Function Point measurement using a requirements engineering ontology. *Inf Softw Technol*. 2016;72:189–203.

[4] Fajar AN, Shofi IM. Reduced Software Complexity for E-Government Applications with ZEF Framework. *TELKOMNIKA (Telekommunication Computing, Electronics and Control)*. 2017;15(1):415–20.

[5] Trudel S, Turcotte A. “Combining Qualitative and Quantitative Software Process Evaluation : A Proposed Approach”. *Coll Econ Anal Ann*. 2017;43:135–54.

[6] Symons C, Abran A, Ebert C, Vogelezang F. “Measurement of software size : advances made by the COSMIC community”. In: *2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement. IEEE*; 2016. p. 75–86.

[7] Salmanoğlu M, Öztürk K, Bağrıyanık S, Ungan E, Demirörs O. “Benefits and challenges of measuring software size: early results in a large organization”. In: *25th International Workshop on Software Measurement and 10th*

International Conference on Software Process and Product Measurement. 2015.

[8] Ozkan B, Demirors O. “On the Seven Misconceptions about Functional Size Measurement”. In: *2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement. IEEE; 2016*. p. 45–52.

[9] Bagriyanik S, Karahoca A. “Big data in software engineering: A systematic literature review”. *Glob J Inf Technol*. 2016;6(1):107–16.

[10] Sahoo P, Mohanty JR. “Early Test Effort Prediction using UML Diagrams’. *Indones J Electr Eng Comput Sci (IJEECS)*. 2017;5(1):220–8.

[11] Kim DK. “Finding Bad Code Smells with Neural Network Models”. *Int J Electr Comput Eng*. 2017;7(6):3613–21.

[12] H. Witten I, Frank E, A. Hall M. “Data mining: Practical machine learning tools and techniques”. 3rd ed. 2011.

[13] Han J, KamberMicheline. “Data Mining Concepts and Techniques”. 2nd ed. 2006.

[14] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. “The WEKA data mining software: an update”. *ACM SIGKDD Explor Newsl*. 2009;11(1):10–8.

[15] Breiman L. Random forests. Mach Learn. 2001;45(1):5–32.

[16] H. John G, Langley P. “Estimating continuous distributions in Bayesian classifiers”. In: *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc.; 1995. p. 338–45.

[17] Zhao Y, Zhang Y. “Comparison of decision tree methods for finding active objects”. *Adv Sp Res*. 2008; 41(12):1955–9.

[18] Quinlan JR. C4. 5: programs for machine learning. 2014.

[19] Landwehr N, Hall M, Frank E. “Logistic model trees”. *Mach Learn*. 2005;59(1–2):161–205.

[20] Gardner M., Dorling S. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmos Environ [Internet]. 1998;32(14–15):2627–36. Available from: http://www.sciencedirect.com/science/article/pii/S1352231097004470

[21] Platt JC. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Adv kernel methods [Internet]. 1998;185–208. Available from: http://www.bradblock.com/Sequential_Minimal_Optimization_A_Fast_Algorithm_for_Training_Support_Vector_Machine.pdf

[22] Aha DW, Kibler D, Albert MK. Instance-Based Learning Algorithms. Mach Learn. 1991;6(1):37–66.

[23] Holte RC. *Very Simple Classification Rules Perform Well on Most Commonly Used Datasets*. Mach Learn. 1993;11(1):63–90.

[24] Bundschuh M, Dekkers C. “The IT measurement compendium: estimating and benchmarking success with functional size measurement”. Springer Science & Business Media; 2008.

[25] Jones C. “Applied Software Measurement Global Analysis of Productivity and Quality”. McGraw-Hill Education Group; 2008.

[26] Consortium CSMI. “The COSMIC Functional Size Measurement Method Version 4.0.1 Measurement Manual “[Internet]. [cited 2018 Jan 17]. Available from: https://cosmic-sizing.org/publications/measurement-manual-401/

[27] Bağrıyanık S, Karahoca A, Ersoy E.” Selection of a functional sizing methodology: A telecommunications company case study”. *Glob J Technol*. 2015;7(7):98–108.

[28] Ilya.kuzovkin@gmail.com. Online converter from .csv to WEKA .arff [Internet]. [cited 2018 Jan 20]. Available from: http://ikuz.eu/csv2arff/

[29] Scuse D, Reutemann P. “WEKA Experimenter Tutorial for Version 3-5-8”. 2008.

[30] Asif R, Merceron A, Ali SA, Haider NG. Analyzing undergraduate students â€™ performance using educational data mining. Comput Educ [Internet]. 2017;113:177–94. Available from: http://dx.doi.org/10.1016/j.compedu.2017.05.007

[31] Kabakchieva D. “Predicting Student Performance by Using Data Mining Methods for Classification”. *Cybern Inf Technol*. 2013;13(1):61–72.

[32] Ahmad F, Ismail NH, Aziz AA. “The Prediction of Students ’ Academic Performance Using Classification Data Mining Techniques”. *Appl Math Sci*. 2015;9(129):6415–26.

[33] Kaur P, Singh M, Singh G. “Classification and prediction based data mining algorithms to predict slow learners in education sector*”. In: P3rd International Conference on Recent Trends in Computing 2015(ICRTC-2015)* [Internet]. Elsevier Masson SAS; 2015. p. 500–8. Available from: http://dx.doi.org/10.1016/j.procs.2015.07.372

BIOGRAPHIES OF AUTHORS

Dr. Selami Bagriyanik holds a PhD in Software Engineering. He is interested in software development, requirements engineering, software measurement, advanced learning technologies, data mining and big data. He also works as the Digital Learning and Business Solutions Technology Manager in Turkcell.

Dr. Adem Karahoca holds a PhD in Software Engineering. He is interested in human–computer interaction, web based education systems, data mining, big data, and management information systems. He has published articles at prestigious journals about use and data mining applications of business information systems in health, tourism, and education.