iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** XI **Month of publication:** November 2021

DOI: https://doi.org/10.22214/ijraset.2021.38977

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

Estimation Approaches of Machine Learning in Scrum Projects: A Review

Chitrak Vimalbhai Dave[1], Abhishek Patel[2], Utkarsh Keshri[3]
[1, 2, 3]*Department of Computer Science and Engineering, Lovely Professional University, Phagwara, India*

Abstract: It is inevitable for any successful IT industry not to estimate the effort, cost, and duration of their projects. As evident by Standish group chaos manifesto that approx 43% of the projects are often delivered late and entered crises because of over budget and less required functions. Improper and inaccurate estimation of software projects leads to a failure, and therefore it must be considered in true letter and spirit. When Agile principle-based process models (e.g. Scrum) came into the market, a significant change can be seen. This change in culture proves to be a boon for strengthening the collaboration between developer and customer. Estimation has always been challenging in Agile as requirements are volatile. This encourages researchers to work on effort estimation. There are many reasons for the gap between estimated and actual effort, viz., project, people, and resistance factors, wrong use of cost drivers, ignorance of regression testing effort, understandability of user story size and its associated complexity, etc. This paper reviewed the work of numerous authors and potential researchers working on bridging the gap of actual and estimated effort. Through intensive and literature review, it can be inferred that machine learning models clearly outperformed non-machine learning and traditional techniques of estimation.
Keywords: Machine Learning, Scrum, Scrum Projects, Effort Estimation, Agile Software Development

I. INTRODUCTION

Agile estimation has always been challenging for IT experts across the globe, and this issue has been constantly put on by various researchers in their literature. A typical estimation framework opted by most of the IT industries is given in Fig. 1, wherein requirements *aka* the desired user stories are being stacked in the product backlog and further tagged with their respective sizes. Story point is most used unit to size a user story, i.e., 61.67% of industries employing it.

As per ISPA [1], two-thirds of software projects neglect to be conveyed on time and inside budget. There are two principle reasons for software project disappointments: One is improper estimation as far as task size, cost, and staff required, and second being the uncertainty of system and software requirements. The major challenges for estimating of Scrum-based projects are change and sprint-wise estimation. Most of the IT industries have adopted hybrid process models which are mostly driven by Agile umbrella methodologies. As per [2] the transition of process models from heavyweight like iterative waterfall to lightweight like Agile, a change can also be seen in effort estimation approaches. All the tradition estimation approaches [3] like expert judgement, top-down estimation, and Delphi cost estimation are well suited in one or other form for heavyweight process models but lack in bridging the estimated and actual effort gap of Agile methodologies. Thus, due to volatile nature of Agile-based project requirements, researchers started exploring alternatives and end up at soft computing techniques [4]. A standout among the most widely recognized uses of neuro-fuzzy frameworks [5] is delivering rules for unpredictable issues. On alternate hands, soft- ware projects are characteristically uncertain and complex with the goal that the accessible data is not sufficient at the beginning time of task and the issue of effort estimation is totally unclear. In this circumstance, fuzzy and neuro-fuzzy models can deal with the vulnerability and increment the estimation exactness. Also, encouraging outcomes has been accounted from fuzzy-based models connected to the field of software effort estimation.

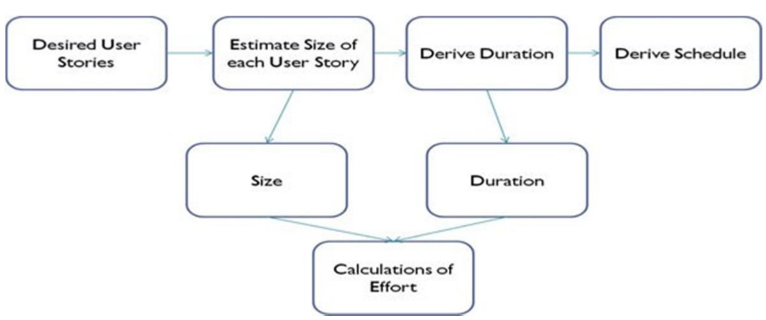

Fig. 1 Typical estimation process in Scrum

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

Because of unpredictability of effort estimation issue and trouble of project and people attribute relational analysis, the optimization procedure assumes an essential job here. The optimization [6] can specifically be connected to effort estimation process like quality weighting in analogy-based estimation or in a roundabout way connected to machine learning strategies, for example, ANN and ANFIS. It can be further extended to attribute weighting, tuning ANN adjustment (weight and bias), ANFIS adjustment, structure configuration, variable positioning.

To the best of our information, there is no current review that centers around ML models of Scrum-based projects, which rouses our work in this paper. This paper contains technical abbreviated terms which can be viewed in Table 1. The upcoming section, viz., Sect. 2, highlights a collaborated context of ML impact in ASD, Sect. 3 explains the review method, Sect. 4 discusses the review results, and Sect. 5 provides conclusion and future research directions.

Table 1 List of abbreviations

SDEE	Software development effort estimation	ANFIS	Adaptive neuro-fuzzy inference system
ML	Machine learning	ANN	Artificial neural network
ASD	Agile software development	CBR	Case-based reasoning
DT	Decision tree	BN	Bayesian network
SVR	Support vector regression	GA	Genetic algorithm
GP	Genetic programming	AR	Association rule
ISBSG	International Software Benchmarking Standards Group	MMRE	Mean magnitude of relative error
PRED	Percentage relative error deviation	LR	Linear regression
RF	Random forest	MLP	Multilayer perceptron
SGB	Stochastic gradient boosting	RBF	Radial basis function
ABC	Artificial bee colony	PSO	Particle swarm optimization
CART	Classification and regression tree	TLBO	Teaching–learning-based optimization
TLBABC	Teaching–learning-based artificial bee colony	DABC	Directed artificial bee colony
LM	Levenberg–Marquardt	ISPA	International Society of Parametric Analysis
NB	Naïve Bayes	KNN	K-nearest neighbor
EJ	Expert judgement	FPA	Function point analysis
MRE	Magnitude of relative error	NF	Neuro-fuzzy

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

II. LITERATURE REVIEW

Jorgensen and Shepperd presented a core review in [7] which recognizes more than 10 estimation methods in 80 s used for effort estimation, wherein regression-based techniques are better as compared to empirical techniques of estimation. In spite of the expansive number of exact investigations on machine learning models in the estimation of software projects irrespective of the process model approach, conflicting outcomes have been accounted for with respect to the estimation exactness of these models. For instance, it was accounted that estimation exactness shifts under a similar machine learning model when it is developed with various datasets [3, 8] or scenarios [9].

With respect to the correlation between ML model and regression model, thinks about in [3] announced that ML model is better than regression model, while examines in [10] reasoned that regression model beats ML model. ANN and case based reasoning techniques outperformed each other when applied on different datasets in [8] and [18].

The difference in the current empirical examinations on ML models has not yet completely comprehended and may keep experts from embracing ML models by comparing with different areas in which ML systems have been connected effectively. Besides, the hypothesis of ML systems is greatly entangled than that of traditional estimation procedures. To encourage the uses of ML procedures in SDEE area, it is pivotal to deliberately condense the empirical proof on ML models in ongoing research and practice. Industry experts use expert judgement and Delphi cost estimation techniques more as compared to ML. Some of the ML strategies that have been utilized for SDEE are [11–14] CBR, ANN, DT, BN, SVR, GA, GP, AR, etc., and most of them are not yet applied in Agile estimation.

The above ML systems are utilized either alone or in blend with other ML or non-ML methods. For example, GA has been utilized with CBR, ANN, and SVR for highlight weighting and choice. Fuzzy logic [15] is utilized with CBR, ANN, and DT for execution. Different datasets have been utilized for estimation, viz., ISBSG, JIRA, PROMISE data repository, and so on. W.R.T approval techniques, holdout, n times overlay cross-validation ($n > 1$), and leave-1-out cross-validation [3, 16] are the predominant ones. MMRE, PRED (25) (percentage of forecasts that are inside 25% of the real estimate), and MdMRE [17] are the three most well-known precision measurements.

Out of all ML techniques, BN [3, 18, 19] found to have most exceedingly bad MMRE in contrast to CBR (51%), ANN (37%), DT (55%), SVR (34%), and GP and AR (49%) separately for estimating projects includes both traditional and lightweight methodologies, but in some cases it did not. Research demonstrates ANN and SVR [9] beat other ML models, yet it does not mean that we can utilize them without confinement as to expand precision, expanding the number of concealed layers will build the preparation time and may create over-fitting issues [3]. Examination of ML models with regression models, COCOMO estimation, EJ, and FPA [20] has also been carried out. Studies indicate CBR and ANN are more exact than regression models. GP is less precise than regression. So, based on the stats we have concluded generally that ML models outflank non-ML strategies. Distinctive estimation settings are made with reference to in writing, for example, little informational collection, anomalies, absolute highlights, and missing qualities.

Analysts recommend [21, 22] that it is more productive to decide the best model in a specific setting instead of deciding the best single model, since estimate models carry on uniquely in contrast to one dataset to other, which makes them precarious. Studies directed on information mining report that group strategies furnish exact outcomes in examination with single strategies as every strategy has quality and shortcoming so joining will moderate the shortcoming. Outfit effort estimation systems might be gathered into two noteworthy classes [22–24, 16]: homogeneous (e.g., bagging and SVR, RF, MLP, LR, RBF, ANFIS, CBR, RF, SGB, CART, and so forth) and heterogeneous perceived by their base models and blend rules. ANN was utilized most with outfits. Studies demonstrate that solitary ML procedures are the predominant methodology used to develop ensembles. It has been discovered that homogeneous troupes dependent on DT are the most exact, trailed by homogeneous one's dependent on CBR, and from there on came SVR homogeneous development.

ANN, DT, CBR, SVR, regression, and neuro-fuzzy [5] are most utilized for group, wherein request of best outcomes pursues DT, regression, CBR, SVR, and afterward NF. Mix rules have additionally been extricated for consolidating endeavors of base models and are partitioned into two sections such as linear and nonlinear. Mean, mean weighted, and middle are most utilized straight mix rules. MLP, SVM, CART, and FIS utilizing c imply subtractive grouping are most utilized non-straight principles. All the techniques mentioned and discussed in this section are derived from general estimation approaches to demonstrate a trail of estimation trends. The next section will include some research questions which will be revolve around Scrum-based project estimation only.

III. METHODOLOGY

In this section, we have discussed the various research questions, review inclusion and exclusion criterion, data source description, and study select process.

A. Research Questions

This review paper aims to summarize the present status of implication of machine learning models in Scrum-based projects. The following research questions have been framed in this context and are given as follows:

1) *RQ1:* Which ML models have been used for Scrum estimation?
2) *RQ2:* Do ML models distinctively outperform other ML models for Scrum estimation?
3) *RQ3:* What is the overall estimation accuracy of ML techniques used in Scrum-based projects?
4) *RQ4:* Does estimation accuracy of Scrum-based projects increase by using meta- heuristic algorithms?
5) *RQ5:* What are the various Scrum project datasets available on Web?
6) *RQ6:* Are ensemble estimation methods better than single estimation for Scrum projects?
7) *RQ7:* What are the various significant factors affecting effort of Scrum projects?

B. Include and Exclude Criterion

This study incorporates the papers which have connected the diverse soft computing techniques for estimation in Agile software development. Papers are incorporated from different online sources, journals, conferences, and so forth distributed till date. A few papers which are not explicitly based on ASD are also likewise included because of some essential data. Papers and data which are not important to the exploration subject are excluded from the examination.

C. Data and Literature Sources Description

This data source used in the study includes papers from TOSEM (ACM), IEEE transactions, ScienceDirect, Google Scholar, Springer, etc. Some search strings have been used to search papers from aforementioned online databases, viz.
Software AND (effort OR cost) AND (estimate) AND (learning OR “machine learning”) OR “machine” OR “case-based reasoning” OR “decision tree” OR “regression analysis” OR “neural net” OR “Bayesian network” OR “Bayesian net” OR “support vector machine” OR “support vector regression” OR “deep” OR “learning” OR “fuzzy” OR “neuro-fuzzy” OR “ANFIS” OR “meta-heuristic” OR “Scrum” OR “Agile” AND “software” AND “development” OR “genetic algorithm” OR analogy OR “expert judgement” OR “planning poker.”

D. Study Selection Process

After applying the include and exclude criterion, the selection has been primarily carried out in two steps:

1) Choosing abstract and title: The review procedure is brought through a few research papers where some of them were chosen by looking on to their titles and modified works.
2) Choosing complete article. A good number of papers and articles are reviewed and thoroughly analyzed, and the same has been discussed in Sect. 4 research questions.

IV. RESULTS AND DISCUSSION

The various research questions mentioned in Sect. 3.1 will be answered here.

A. Which ML Models Have Been Used for Scrum Estimation (RQ1)?

A wide variety of ML models has been extensively used in Agile software development and its associated methodologies under its umbrella. Table 2 contains ML techniques used in Scrum estimation with their frequency and year of publication.

Table 2 ML techniques used in Scrum estimation

ML techniques	Use in paper	YOP
Fireworks algorithm optimized neural network	Thanh Tung Khuat and My Hanh Le in [25]	2018
Multiagent techniques	Muhammad D Adnan et al. in [26]	2017
Mamdani fuzzy inference systems	Jasem M. Alostad et al. in [15]	2017
General regression neural networks	Aditi Panda et al. in [27]	2015
Probabilistic neural networks	Aditi Panda et al. in [27]	2015
GMDH polynomial neural network	Aditi Panda et al. in [27]	2015
Cascade correlation neural network	Aditi Panda et al. in [27]	2015
Stochastic gradient boosting	Shashank Mouli Satapathy et al. in [28]	2017
Random forest	Shashank Mouli Satapathy et al. in [28]	2017
Decision tree	Shashank Mouli Satapathy et al. in [28]	2017
Bayesian networks	Dragicevic Srdjana et al. in [19]	2017
Hybrid ABC–PSO algorithm	Thanh Tung Khuat and My Hanh Le in [29]	2017
SVM, NB, KNN, DT	Simone Porru et al., in [30]	2016
Naïve Bayes (NB)	K Moharreri et al. in [31]	2016
Deep learning—long short-term memory (recurrent neural networks)	M. Choetkiertikul et al. in [32]	2015
Particle swarm optimization (PSO)	Manga I, et al. in [33]	2014
SVR kernel methods	Shashank Mouli Satapathy et al. in [9]	2014

From the above table, it can be seen that most of the authors have used different ML techniques and as per their respective year of publication a trend can be inferred that researchers are now shifted to ML techniques to create an auto-estimate environment. In the subsequent section, a comparative analysis has been carried out

B. Do ML Models Distinctively Outperform Other ML Models for Scrum Estimation? (RQ2)

It has been given in Sect. 2 that ML techniques outperform non-ML techniques. Moreover, expert-based estimations are suffered from individual bias. In this research question, a comparative analysis of all the ML techniques applied for Scrum-based project estimation is mentioned in Table 3.

As an accuracy parameter, various metrics like MRE and PRED have been mentioned as per the availability of data in the literature. Various ML techniques outperform other ML techniques by applying on either same dataset or different datasets.

It can be inferred from Table 3 that best current ML technique as per the accuracy metric MMRE for 21 project data is fireworks algorithm optimized neural network with 2.93% MMRE. It cannot be deduced exactly as the projects/datasets used by other authors are different and may have less or more predication. An improved predication can also be seen in others with different datasets. The compiled MMRE results can be seen in Fig. 2.

C. What Is the Overall Estimation Accuracy of ML Techniques Used in Scrum-Based Projects? (RQ3)

To the best of our knowledge, 16 ML techniques have been used for Scrum-based project estimation till date. From Table 3, the average mean magnitude of relative error for the same dataset ML techniques comes out to be 0.2822.

D. Does Estimation Accuracy of Scrum-Based Projects Increase by Using Meta-Heuristic Algorithms? (RQ4)

In the literature, very less, empirical evidence can be seen in the context of inclusion of meta-heuristic algorithms in Scrum-based projects. As mentioned in Table 3, only two such papers have been refereed, i.e., fireworks algorithm [25] and ABC– particle swarm optimization (PSO) [29]. Among these two, fireworks algorithm has good estimation accuracy as compared to all ML techniques used for Scrum. In this context, we can deduce that estimation accuracy does increase with the inclusion of meta-heuristic algorithms.

Table 3 Comparative accuracies of different ML estimation techniques

Estimation techniques	Use in paper	Accuracy parameters	Dataset used	Outperformed
Fireworks algorithm optimized Neural network	[25]	MMRE-0.0293	21 projects developed by six software companies presented in Zia's work	TLBO, TLBABC, DABC, LM
Multiagent techniques	[24]	MMRE—0.1	12 Web projects	Delphi and planning poker
Mamdani fuzzy inference systems	[14]	MMRE (sprint1)—0.28 MMRE (sprint2)—0.15 MMRE (sprint3)—0.09	Three sprints of real software projects	Comparison with actual est.
General regression neural network (GRNN)	[25]	MMRE—0.3581	21 projects developed	Regression (Zia's work) and PNN
Probabilistic neural network (PNN)	[25]	MMRE—1.5776	21 projects developed	Zia's work
GMDH polynomial neural network. (GMDHPNN)	[25]	MMRE—0.1563	21 projects developed	GRNN and PNN
Cascade correlation neural network (CCNN)	[25]	MMRE—0.1486	21 projects developed	GRNN, PNN, GMDHPNN
Stochastic gradient boosting (SGB)	[26]	MMRE—0.1632	21 projects developed	RF and DT
Random forest (RF)	[26]	MMRE—0.2516	21 projects developed	DT
Decision tree (DT)	[26]	MMRE—0.3820	21 projects developed	Zia's work
Bayesian networks	[18]	Accuracy—above 90% for six Datasets	160 tasks in real Agile projects	Comparison with actual estimate
Hybrid ABC–PSO algorithm	[27]	MMRE—0.0569	21 projects developed	ABC, PSO, GRNN, PNN, GMDHPNN, CCNN

(continued)

Table 3 (continued)

Estimation techniques	Use in paper	Accuracy parameters	Dataset used	Outperformed
SVM, NB, KNN, DT	[28]	SVM MMRE—0.50 NB MMRE—0.85 KNN MMRE—0.70 DT MMRE—0.98	699 issues of inventive s/w designers 5607 issues from 8 open-source projects	Comparison with actual estimates
Naïve Bayes (NB)	[29]	MMRE—2.044	10 teams in IBM rational team concert	None
Deep learning	[30]	Improved MMRE	23,313 issues from 16 projects	Empirical estimation technique like educated guess
Particle swarm optimization (PSO)	[31]	MMRE—0.1988	21 projects	Zia's work
SVR kernel methods SVR linear kernel SVR polynomial kernel SVR RBF kernel SVR sigmoid kernel	[11]	MMRE—0.1492 MMRE—0.4350 MMRE—0.0747 MMRE—0.1929	21 projects	SVR linear, polynomial, and sigmoid kernel

E. What Are the Various Scrum Project Datasets Available on the Web? (RQ5)

Datasets for Scrum projects can be found on various online repositories and are shown in the table either as a Web link or as a paper link. Some of the repositories like ISBSG also contain Agile data that may be present as such, and an appropriate data cleaning and filtering techniques need to be applied to get the same (Table 4).

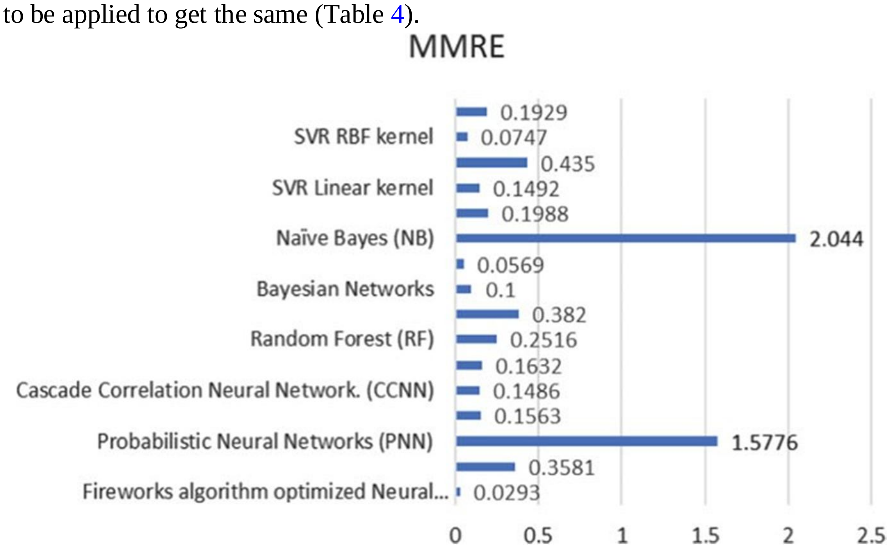

Fig. 2 Comparative accuracies of ML techniques used in Scrum

Table 4 Agile projects dataset links

Dataset name	Dataset links
ISBSG datasets	[6]
Three sprints of real software projects	[14]
Twelve Web projects data for an e-commerce site	[26]
699 issues from industrial projects and 8 open-source projects	[28]
Story point dataset	[32]
Twenty-One projects developed by six software companies presented in Zia's work	[34]

F. Are Ensemble Estimation Methods Better Than Single Estimation in Scrum Projects? (RQ6)

Yes, it can be inferred from the review that ensemble estimation techniques yield better results that is just single estimation method. The estimation accuracy of par- ticle swarm optimization alone is less than artificial bee colony–PSO. On the sim- ilar grounds, when we backtrack our literature for estimation techniques used for heavyweight process models, ensemble wins in majority.

Table 5 Factors affecting Scrum-based project effort [35, 36]

Project-related factors	People-related factors	Resistance factors
Project domain	Communication skills	Perfect team composition and defects in third-party tools
Quality requirement	Familiarity in team	Working place un-comfort and stakeholder response
Hardware and software requirements	Managerial skills	Drifting to Agile, lack of clarity in requirements, volatility of requirements
Operational ease	Security	Team dynamics and change in working environment
Complexity	Working time	Expected team changes and other project responsibilities
Data transaction	Past project experience	Introduction to new technology and prerequisite availability of resources
Multiple sites	Technical ability	Usability

G. What Are the Various Significant Factors Affecting Effort of Scrum Projects (RQ7)?

Effort in Scrum projects has been widely affected by various people, resistance, and project factors. Many authors have proposed various factors in this context, and the same can be viewed in Table 5.

V. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In this review paper, the following research gaps have been identified that open up an opportunity for all the potential researchers across the globe.

1) Missing estimation factors may result in poor estimation as there are potential accelerating and decelerating factors to affect the estimate of Agile-based projects. Total effort is a result of effort of all elements of a sprint and reiterates again after the potential shippable release so there is a need of adding regression test effort to make it more accurate.
2) There are so many machine learning and optimization approaches missing in the literature, and they have not yet been applied for estimating effort of Scrum-based projects
3) No standard/generic scale for story size and story complexity found in the literature.
4) No generic or single estimation model made for Scrum estimation.
5) Non-functional requirement effort missing in calculating the total effort of the sprint or project.

REFERENCES

[1] Nassif, A.B., Azzeh, M., Capretz, L.F., Ho, D.: Neural network models for software devel- opment effort estimation: a comparative study. Neural Comput. Appl. 27(8), 2369–2381 (2016)
[2] Popli, R., Chauhan, N.: Cost and effort estimation in agile software development. In: 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), pp. 57–61 (2014)
[3] Wen, J., Li, S., Lin, Z., Hu, Y., Huang, C.: Systematic literature review of machine learning based software development effort estimation models. Inf. Softw. Technol. 54(1), 41–59 (2012)
[4] Bilgaiyan, S., Mishra, S., Das, M.: A review of software cost estimation in Agile software development using soft computing techniques. In: 2016 2nd International Conference on Computational Intelligence and Networks (CINE), pp. 112–117 (2016)
[5] Sharma, A., Ranjan, R.: Software effort estimation using neuro fuzzy inference system: past and present. Int. J. Recent Innov. Trends Comput. Commun. 5(8), 78–83 (2017)
[6] Samareh Moosavi, S.H., Khatibi Bardsiri, V.: Satin bowerbird optimizer: a new optimization algorithm to optimize ANFIS for software development effort estimation. Eng. Appl. Artif. Intell. 60, 1–15 (2017)
[7] Jorgensen, M., Shepperd, M.: A systematic review of software development cost estimation studies. IEEE Trans. Softw. Eng. 33(1), 33–53 (2007)
[8] Pospieszny, P., Czarnacka-Chrobot, B., Kobylinski, A.: An effective approach for software project effort and duration estimation with machine learning algorithms. J. Syst. Softw. 137, 184–196 (2018)
[9] Satapathy, S.M., Panda, A., Rath, S.K.: Story point approach based Agile software effort estima- tion using various SVR Kernel methods. In: International Conference on Software Engineering and Knowledge Engineering, pp. 304–307 (2014)
[10] Mendes, E., Watson, I., Triggs, C., Mosley, N., Counsell, S.: A comparative study of cost estimation models for web hypermedia applications. Empir. Softw. Eng. 8(2), 163–196 (2003)
[11] Usman, M., Mendes, E., Börstler, J.: Effort estimation in Agile software development: a survey on the state of the practice. In ACM International Conference Proceedings Series, pp. 1–10 (2015)
[12] Azzeh, M., Nassif, A.B., Banitaan, S.: Comparative analysis of soft computing techniques for predicting software effort based use case points. IET Softw. 12(1), 19–29 (2018)
[13] Yousef, Q.M., Alshaer, Y.A.: Dragonfly estimator: a hybrid software projects' efforts estimation model using artificial neural network and Dragonfly algorithm. Int. J. Comput. Sci. Netw. Secur. 17(9), 108–120 (2017)
[14] Menzies, T., Yang, Y., Mathew, G., Boehm, B., Hihn, J.: Negative results for software effort estimation. Empir. Softw. Eng. 22(5), 2658–2683 (2017)
[15] Alostad, J.M., Abdullah, L.R.A., Aali, L.S.: A fuzzy based model for effort estimation in Scrum projects. Int. J. Adv. Comput. Sci. Appl. (IJACSA) 8(9), 270–277 (2017)
[16] Idri, A., Hosni, M., Abran, A.: Systematic literature review of ensemble effort estimation. J. Syst. Softw. 1, 1–35 (2016)
[17] Bilgaiyan, S., Sagnika, S., Mishra, S., Das, M.: A systematic review on software cost estimation in Agile software development. J. Eng. Sci. Technol. Rev. 10(4), 51–64 (2017)
[18] Radlinski, L.: A survey of bayesian net models for software development effort prediction. Int. J. Softw. Eng. Comput. 2(2), 95–109 (2010)
[19] Dragicevic, S., Celar, S., Turic, M.: Bayesian network model for task effort estimation in agile software development. J. Syst. Softw. **127**, 109–119 (2017)
[20] Salmanoglu, O.D.M., Hacaloglu, T.: Effort estimation for Agile software development : com- parative case studies using COSMIC functional size measurement and story points. In: ACM Mensura, pp. 1–9 (2017)
[21] Padmaja, M., Haritha, D.: Software effort estimation using meta heuristic algorithm. Int. J. Adv. Res. Comput. Sci. **8**(5), 196–201 (2017)
[22] Murillo-Morera, J., Quesada-López, C., Castro-Herrera, C., Jenkins, M.: A genetic algorithm based framework for software effort prediction. J. Softw. Eng. Res. Dev. **5**(1), 1–33 (2017)
[23] de Araújo, R.A., Oliveira, A.L.I., Meira, S.: A class of hybrid multilayer perceptrons for software development effort estimation problems. Expert Syst. Appl. **90**, 1–12 (2017)
[24] Dave, V.S., Dutta, K.: Neural network based models for software effort estimation: a review. Artif. Intell. Rev. **42**(2), 295–307 (2014)
[25] Khuat, T., Le, H.: An effort estimation approach for Agile software development using fireworks algorithm optimized neural network. Int. J. Comput. Sci. Inf. Secur. **14**(7), 122–130 (2018)
[26] Adnan, M., Afzal, M.: Ontology based multiagent effort estimation system for Scrum Agile method. IEEE Access, 25993–26005 (2017)
[27] Panda, A., Satapathy, S.M., Rath, S.K.: Empirical validation of neural network models for Agile software effort estimation based on story points. Procedia Comput. Sci. **57**, 772–781 (2015)
[28] Satapathy, S.M., Rath, S.K.: Empirical assessment of machine learning models for agile soft- ware development effort estimation using story points. Innov. Syst. Softw. Eng. **13**(2–3), 191–200 (2017)
[29] Khuat, T.T., Le, M.H.: A novel hybrid ABC-PSO algorithm for effort estimation of software projects using Agile methodologies. J. Intell. Syst. **27**(3), 489–506 (2018)
[30] Porru, S., Murgia, A., Demeyer, S., Marchesi, M., Tonelli, R.: Estimating story points from issue reports. In: Proceedings of the 12th International Conference on Predictive Models and Data Analytics in Software Engineering—PROMISE 2016, pp. 1–10 (2016)
[31] Moharreri, K., Sapre, A.V., Ramanathan, J., Ramnath, R.: Cost-effective supervised learning models for software effort estimation in Agile environments. In: 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), pp. 135–140 (2016)
[32] Choetkiertikul, M., Dam, H.K., Tran, T., Pham, T.T.M., Ghose, A., Menzies, T.: A deep learning model for estimating story points. IEEE Trans. Softw. Eng. **14**(8), 1–12 (2016)
[33] Manga, I., Blamah, N.V.: A particle swarm optimization-based framework for Agile software effort estimation. Int. J. Eng. Sci. **3**(6), 30–36 (2014)
[34] Tipu, S.K., Zia, S.: An effort estimation model for agile software development. Adv. Comput. Sci. Appl. **2**(1), 314–324 (2012)
[35] Popli, R., Chauhan, N.: Agile estimation using people and project related factors. In: 2014 International Conference on Computing for Sustainable Global Development (INDIACom), pp. 564–569 (2014)
[36] Arora, M., Chopra, S., Gupta, P.: Estimation of regression test effort in Agile projects. Far East J. Electron. Commun. **3**(II), 741–753 (2016)