International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
DOI: https://doi.org/10.17762/ijritcc.v11i11s.8102
***Article Received:** 20 June 2023 **Revised:** 16 August 2023 **Accepted:** 02 September 2023*

A Review: Effort Estimation Model for Scrum Projects using Supervised Learning

Neelam Sunda[1], Ripu Ranjan Sinha[2]
[1]Department of Computer Science, RTU, Kota, Rajasthan, India, research.neelam@gmail.com
[2]Department of Computer Science, RTU, Kota, Rajasthan, India, drsinhacs@gmail.com

Abstract— Effort estimation practice in Agile is a critical component of the methodology to help cross-functional teams to plan and prioritize their work. Agile approaches have emerged in recent years as a more adaptable means of creating software projects because they consistently produce a workable end product that is developed progressively, preventing projects from failing entirely. Agile software development enables teams to collaborate directly with clients and swiftly adjust to changing requirements. This produces a result that is distinct, gradual, and targeted. It has been noted that the present Scrum estimate approach heavily relies on historical data from previous projects and expert opinion, while existing agile estimation methods like analogy and planning poker become unpredictable in the absence of historical data and experts. User Stories are used to estimate effort in the Agile approach, which has been adopted by 60–70% of the software businesses. This study's goal is to review a variety of strategies and techniques that will be used to gauge and forecast effort. Additionally, the supervised machine learning method most suited for predictive analysis is reviewed in this paper.

Keywords- Agile Methodologies, SCRUM, User Story, Effort Estimation, Supervised Learning.

I. INTRODUCTION

The majority of software development businesses utilize Scrum, the most popular Agile technique, yet there are issues with the method's ability to accurately estimate effort. In a software project, estimating effort is a crucial activity since it aids in the development of workable implementation strategies and it significantly affects whether the project succeeds or fails. As part of the agile development process, a user story (or group of user requirements) is assigned a value (story point) to represent the amount of effort anticipated for developing that story. The efficacy of project planning is increased by accurate effort estimates, which is advantageous to the company in many ways. The chance that a project will be successfully finished is increased by an effective work plan.

It has been said that the current Scrum estimating method heavily relies on past data from previous projects and expert opinion, hence no technique is effective in the absence of historical data or experts. These estimates are frequently produced by several team members, some of whom may assign low efforts according to their own personal experiences. These guesstimates might lead to inconsistency and estimation discrepancies. For the software project's labor to be calculated precisely and effectively, an algorithmic approach is required. Accurate software predictions are now achievable with the use of ML methods, which enable learning algorithms based on previously finished projects. ML algorithms are designed in such a manner that they can learn from data and predict the future. There are two types of machine learning algorithms: supervised and unsupervised. A method called as supervised learning uses labelled training data to learn how to predict outcomes from unlabeled data. When employing supervised learning, you use carefully "labelled" data to teach the computer. It demonstrates that some information has already undergone accurate labeling. Unsupervised learning makes use of unlabeled data as opposed to supervised learning. These data

are used to create patterns that can help with clustering or association issues.

Agile estimating has always been difficult for IT professionals throughout the world, and many scholars have regularly covered this topic in their works. Figure 1 displays a typical estimating architecture used by most IT organizations. The requirements, or desired user stories, are stacked up in the product backlog and further tagged with their appropriate sizes. The most popular unit of measurement for sizing a user story is a story point.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
DOI: https://doi.org/10.17762/ijritcc.v11i11s.8102
***Article Received:** 20 June 2023 **Revised:** 16 August 2023 **Accepted:** 02 September 2023*

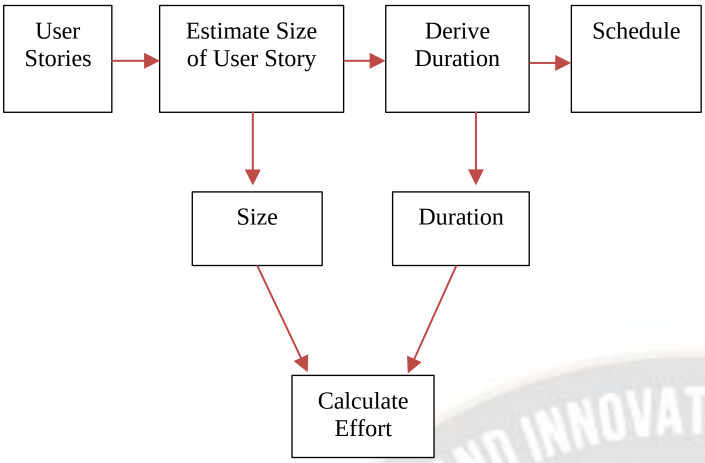

Figure 1: Estimation process in Scrum

II. RELATED WORK

The author conducted a poll with agile experts in a scrum setting to determine the variables impacting the accuracy of a sprint effort estimate. The survey was built on 15 small-scale agile activities from a well-known Mauritius firm that were completed successfully. Only 12 of the 18 identified components, such as communication, team experience, configuration, security, and demand volatility, were deemed essential by the author, and not all of them had an impact on work estimation. The three criteria that have the most influence on all efforts are technical proficiency, team experience, and the complexity of the requirements. The author suggested utilizing machine learning methods to build a prediction model based on the factors found for evaluating sprint effort [1].

Provides a machine learning (ML) model that predicts and estimates sprint effort automatically. The regression approaches used to test the model, which was based on small-scale agile projects, included linear regression, K-Nearest Neighbor, decision trees, and multi-layer perception (MLP). Since the MLP strategy outperforms other regression approaches, the author uses it for the model. Low error values and excellent prediction accuracy were provided by the model, which produced a reasonable and accurate estimate. In order to identify crucial elements for precise sprint projection, the author advised performing a survey on moderate to large-scale projects made up of cross-functional distributed teams. The author also recommended that the model be trained and evaluated using data acquired from actual projects [2].

Due to the fewer contributors, lower turnover, and faster iteration rates, the author prefers utilizing commercial projects over open-source ones. For the suggested Deep-SE model, the author employed a sizable dataset from a project created by the healthcare data science firm IQVIA. In comparison to the author's other three baselines, the model created using the commercial dataset fared better. A classification-based model for effort estimation was put out by the author [3].

Using regression-based machine learning techniques as Multi-Layer Perception, Support Vector Regression, Gradient Boosting, and Random Forest Regression, the author created a dynamic effort estimation model to evaluate the efforts. The Gradient Boosting Algorithm had the highest performance when used with the MongoDB dataset when the author evaluated all these methods on five different datasets with the same parametric structure. The Fibonacci sequence was utilized by the author to represent narrative points, and additional tale point representations were advocated for use in future studies [4].

A thorough review of the literature revealed that overspending and underutilizing resources result in 43% of projects frequently delivering late and having problems. Software projects fail as a result of inaccurate project estimates. The project, people, and resistance factors; improper application of cost drivers; disregard for the time needed for regression testing; and comprehension of user narrative size and associated complexity are just a few of the elements the author identified as the primary causes of the discrepancy between estimated and actual work. The author came to the conclusion that machine learning models significantly outperformed traditional and non-machine learning techniques of estimating after carefully analyzing the work of numerous authors and upcoming researchers who were attempting to close the gap between real and estimated effort. For dependable and accurate software project job predictions, the author advised employing ML techniques [5].

To determine how much work is required for agile software development, use narrative points. In order to determine an appropriate projection of effort, the author looked at performance measures such as MMRE, MMER, and PRED. A variety of machine learning approaches were then used to improve the findings. The author of the paper forecasted the software effort using three machine learning approaches. Radial Basis Function Network, Generalized Regression Neural Networks, and Adaptive Neuro-Fuzzy Modeling are the three techniques. The anfis and newgrnn functions of the MATLAB software are used to construct adaptive neuro-fuzzy modeling, while the newrb and newrbe algorithms are utilized to construct GRNN and RBFNs, respectively. The author advised carrying out more research utilizing the Fireworks Algorithm (FA), Random Forest, etc. [6].

The Neuro Fuzzy Inference System (NFIS), a cutting-edge estimation technique, is studied by the authors of this paper.

Fuzzy logic and artificial neural networks are combined in this hybrid approach to produce a more precise approximation [7].

This paper introduces a novel model that combines the satin bower bird optimization algorithm (SBO) and adaptive neuro-fuzzy inference system (ANFIS) in order to predict software development effort more precisely. Using a unique optimization technique known as SBO, it has been recommended that the ANFIS variables be altered in order to adapt the system's constituent pieces. The proposed hybrid model is an improved neuro-fuzzy-based estimating model that might provide accurate estimates for a range of software applications. The suggested optimization strategy is compared to numerous bio-inspired optimization methods using 13 typical test functions, including unimodal and multimodal functions. Utilizing three actual data sets, the suggested hybrid model is also assessed. The data gathered suggests that the proposed technique may significantly improve the performance indicators [8].

This article aims to bridge the gap between the state-of-the-art in research and corporate implementations by describing effective and useful machine learning deployment and maintenance approaches that draw on research discoveries and industry best practices. Cross validation, an ensemble average of three machine learning methods, the ISBSG dataset, and intelligent data preparation were all employed to accomplish this. It is predicted that companies who develop or deploy software systems would use the available models for effort and duration prediction as a decision support tool [9].

In this comparative investigation, support vector regression, an adaptive neurofuzzy inference system, and four neural network techniques were all used. The results suggest that most soft computing technologies can be used to this issue accurately and successfully. In several accuracy tests, the general regression neural network consistently comes out on top, making it the most effective. Furthermore, it has been found that the accuracy and stability may be increased by using basic UCP variables alone or in conjunction with adjustment factors [10].

The suggested model, which is based on Feedforward Artificial Neural Network, was trained, and tested using NASA projects dataset in order to improve the precision of time prediction in the software industry. A more sophisticated and accurate software estimating model was developed as a result of utilizing the Dragonfly Algorithm to deliver the finest training. The suggested model performed much better than comparable estimate techniques in tests utilizing project datasets, it was found. The core claim of the research, that the suggested model may be used to calculate the quantity of work necessary for different kinds of software projects, was evaluated and approved using a range of performance criteria [11].

Nothing we examined for the COCOMO data sets surpassed Boehm's initial approach, which is the research's basic flaw. Thus, we draw the following conclusions: (i) it is strongly advised to use the data; and (ii) COCOMO should be used to make predictions when COCOMO-style features are available. We make this claim since the investigations for this work demonstrate that, at least for effort estimate, the data collection method is more significant than the learning strategy used [12].

In the Veins simulator, a detailed simulation is done to test the effectiveness of the fuzzy evaluation and examine the Markov chain driver behaviour model while changing the starting trust score for all or select drivers. According to a comparison of the fuzzy and fixed RSU assessment systems [13], the fuzzy assessment scheme can motivate drivers to behave better. For agile users to comprehend the most recent advancements in cost estimation in ASD, this article presents a thorough review of cost estimation in Agile Software Development (ASD) [14].

The accuracy (the proportion of correctly predicted occurrences over the total number of instances), mean absolute error, root mean squared error, relative absolute error, and root relative squared error are some of the statistics reported in this work that are used to evaluate the model's accuracy. High prediction accuracy is provided by the findings [15].

In this paper, using the COSMIC functional size evaluation approach, we compile and analyze the results from three case studies that compared the effectiveness of COSMIC-based and narrative point-based task estimation in an agile environment. Utilizing COSMIC size and actual effort, models for predicting effort were developed, and their efficacy was then evaluated [16].

III. METHODOLOGY

This section has addressed the several research areas, review inclusion and exclusion criteria, explanations of the data sources, and the study selection process.

A. *Research Questions*

An overview of the state of machine learning models' application to Scrum-based projects is the goal of this review article. The following research questions have been created and are offered in this context:

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
DOI: https://doi.org/10.17762/ijritcc.v11i11s.8102
***Article Received:** 20 June 2023 **Revised:** 16 August 2023 **Accepted:** 02 September 2023*

- RQ1: What elements should be considered when assessing the work of a user story?
- RQ2: Which Supervised ML approach will accurately predict the labour required to complete a user story?
- RQ3: Can a model be created to predict effort with accuracy using recognized variables and supervised machine learning?

We will investigate and respond to the following research issues during our study.

The strategy to be employed to develop a model that satisfies the user requirements is described in this section.

B. Collection of Vital factors

The Product Owner, Scrum Master, and Development Team of the Jaipur-based firm will prepare a list of variables that influence effort estimation based on current research and a survey. Twelve factors have been determined, including staff experience, technical aptitude, prior project experience, extracurricular activities, stakeholder communication, configuration, security, flaws and changes in previous implementations, the quality and complexity of the need, and the level of risk involved [17–19].

C. Construction of Dataset

To estimate the effort, we shall utilize specific datasets. The factors that emerged as the most prevalent, as well as those that are highly supported by agile practitioners, will be taken into consideration to simulate the dataset. These open-source projects include Apache, JBoss, MongoDB, Spring for SCRUM, and prevailing datasets like NASA and Desharnais which consist of dataset extensively available for effort estimation [21] [28] [36]. The chosen elements will then be utilized to gather data from previous projects of an IT company that creates Scrum-based projects for the proposed model.

D. Application of ML algorithm

Several research have used machine learning (ML) approaches to get reliable approximations [22] [30] [33-35]. Machine learning algorithms fall into two major categories: supervised and unsupervised. Models are trained to produce the desired outputs using a training set in supervised learning. This training dataset has the right inputs and outputs, enabling the model to develop over time. Naive Bayes, Linear Regression, K Nearest Neighbors, Decision Trees, Support Vector Regression, and many other methods are examples of supervised learning techniques. The dataset will be split into two subsets for the experiment: a training data set and a testing data set. While the testing data set will be used for testing or validating purposes, our model will be trained using the training data set.

E. Model Selection

In order to determine which Supervised method offers the best prediction values based on the data set and parameters, the various algorithms will be compared and assessed using the cross-validation approach. The method used to generate the forecast values for the model will be the one that most closely matches the actual effort levels.

F. Performance Evaluation

In order to evaluate the performance of our model, we will use performance measures like Mean of Magnitude of Error Relative (MMER), Prediction Accuracy (PRED), Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These matrices are widely used to judge if ML algorithms are correct or not.

IV. RESULT ANALYSIS

Agile software development and the practices it encompasses have made substantial use of a wide range of ML models. Machine learning methods and scrum framework estimation are shown in Table 1.

TABLE I. MACHINE LEARNING APPROACHES & SCRUM FRAMEWORK

S. No.	Technique	Authors	Year
1	Proximity-Based Classifiers	Ramessur, M. A., & Nagowah, S. D.	2021
2	DEEP-SE Algorithm	M. Abadeer and M. Sabetzadeh	2021
3	Regression-Driven Machine Learning	Gultekin, M., & Kalipsiz, O.	2020
4	Fireworks-Inspired Neural Network	Thanh Tung Khuat and My Hanh Le	2018
5	Tree of Decisions	Shashank Mouli Satapathy et al.	2017
6	Probabilistic Graph Models	Dragicevic Srdjana et al.	2017
7	Hybrid ABC–PSO Algorithm	Thanh Tung Khuat and My Hanh Le	2017
8	Random Forests	Shashank Mouli Satapathy et al.	2017
9	Multiagent Systems	Muhammad D Adnan et al.	2017
10	Mamdani Fuzzy Inference Engines	Jasem M. Alostad et al.	2017
11	Gradient Boosting with Stochastic	Shashank Mouli Satapathy et al.	2017

A trend that academics are currently using ML approaches to develop an auto-estimate environment can be deduced from the following table, which shows that most of the authors have employed various ML techniques, Scrum framework, and their respective year of publication. There is a comparison analysis in the section that follows.

Table 2 provides a comparison of all ML approaches used

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11s
DOI: https://doi.org/10.17762/ijritcc.v11i11s.8102
Article Received: *20 June 2023* ***Revised:*** *16 August 2023* ***Accepted:*** *02 September 2023*

for Scrum-based project estimating in this study question. According to the data that is available in the literature, several metrics, including MRE and PRED, have been listed as an accuracy parameter. When used on the same dataset or distinct datasets, some ML techniques perform better than other ML techniques. Table 2 indicates that the fireworks algorithm optimized neural network with 2.93% MMRE is the best existing ML technique according to the accuracy metric. Since the projects/datasets used by other writers are different and could contain less or greater predication, it cannot be determined precisely. Others with other datasets can also have an improved prediction. More than 10 ML approaches have currently been applied to Scrum-based project estimate.

TABLE II. COMPARATIVE ANALYSIS OF MACHINE LEARNING ESTIMATION TECHNIQUES

S. No.	Estimation Model	Comparison Metrics	Project Count	Techniques Outperforming
1	Fireworks-Enhanced Neural Network	Mean Magnitude of Relative Error (MMRE): 0.0293	21 projects	TLBO, TLBABC, DABC, LM
2	Multiagent Estimation Methods	MMRE: 0.1	12 Web projects	Delphi, Planning Poker
3	Mamdani Fuzzy Inference Systems	Sprint1 MMRE: 0.28 Sprint2 MMRE: 0.15 Sprint3 MMRE: 0.09	Three sprints of real software projects	Compared to actual estimate
4	General Regression Neural Network (GRNN)	M.M.R.E.: 0.3581	21 projects	Regression, PNN
5	Probabilistic Neural Network (PNN)	M.M.R.E.: 1.5776	21 projects	Zia’s work
6	GMDH Polynomial Neural Network (GMDHPNN)	M.M.R.E.: 0.1563	21 projects	GRNN, PNN
7	Cascade Correlation Neural Network (CCNN)	M.M.R.E.: 0.1486	21 projects	GRNN, PNN, GMDHPNN
8	Stochastic Gradient Boosting (SGB)	M.M.R.E.: 0.1632	21 projects	RF and DT
9	Random Forest (RF)	M.M.R.E.: 0.2516	21 projects	DT
10	Decision Tree (DT)	M.M.R.E.: 0.3820	21 projects	Zia’s work
11	Bayesian Networks	Accuracy: above 90% for six Datasets	160 tasks in real Agile projects	Compared to actual estimate

TABLE III. AGILE PROJECTS DATASET

S. No.	Dataset Description
1	ISBSG Dataset - Collection of software projects from various sources
2	NASA Dataset - Data from software projects undertaken by NASA
3	Three Sprints of Real Software Projects - Data from iterative development cycles
4	IQVIA Dataset - Data from projects managed by IQVIA
5	E-commerce Web Projects Dataset - Data from 12 web projects for an e-commerce site
6	Industrial and Open-Source Project Issues - Data from issue tracking in industrial and open-source projects
7	MongoDB Dataset - Data related to MongoDB usage and performance
8	Story Point Dataset - Dataset containing story point estimations
9	Desharnais Dataset - Data from Desharnais' software projects
10	Zia's Software Projects Dataset - Data from 21 projects developed by six software companies as presented in Zia's work

Table 3 lists some online sources where Scrum project datasets can be found.

Many different projects, individuals, and resistance factors have had a significant impact on effort in Scrum initiatives. Table 4 lists the numerous factors that several authors have suggested in this context.

TABLE IV. SCRUM-BASED PROJECT EFFORT FACTORS

S. No.	Project Factors	People Factors	Resistance Factors
1	Quality Requirements	Team's Familiarity	Comfort and Stakeholders' Reactions
2	Hardware and Software Requirements	Managerial Skills	Shifting to Agile, Unclear Needs, and Instability
3	Operational Ease	Security Concerns	Team Dynamics and Workplace Changes
4	Project Complexity	Working Time	Anticipated Team Adjustments and Extra Duties
5	Data Transactions	Past Project Experience	Introducing New Technology and Resource Availability

V. CONCLUSION AND FUTURE SCOPE

In this literature study, numerous machine learning and optimization techniques are examined. Without being specifically designed to do so, machine learning enables software firms to increase the accuracy of their cost, size, and effort estimation. In order to forecast new output values, machine learning algorithms use historical data as input. The story points are typically estimated during planning sessions or by consulting professionals, where a team comes to an agreement on how much work a user task takes in terms of story points. These assessments are largely the result of subjective judgements based on individual opinion, intuition, and sentiments. This may result in a biased and erroneous effort estimation. The software project can be planned and developed within budget and timeline constraints with our suggested model's as it is expected to provide accurate prediction and estimation of effort in a sprint utilizing supervised ML technique. The suggested model will be developed using factors that have been determined to be essential for effort estimation. For SCRUM-based projects, a novel USEEM Model will be created, and it is anticipated that the model will solve the subjectivity problem with the existing estimation techniques.

REFERENCES

[1] Ramessur, M. A., & Nagowah, S. D. (2020). Factors affecting sprint effort estimation. Advances in Intelligent Systems and Computing, 507–518. https://doi.org/10.1007/978-981-15-1483-8_43

[2] Ramessur, M. A., & Nagowah, S. D. (2021). A predictive model to estimate effort in a sprint using machine learning techniques. International Journal of Information Technology, 13(3), 1101–1110. https://doi.org/10.1007/s41870-021-00669-z

[3] Abadeer, M., & Sabetzadeh, M. (2021). Machine learning-based estimation of story points in Agile Development: Industrial Experience and lessons learned. 2021 IEEE 29th International Requirements Engineering Conference Workshops (REW). https://doi.org/10.1109/rew53955.2021.00022

[4] Gultekin, M., & Kalipsiz, O. (2020). Story point-based effort estimation model with Machine Learning Techniques. International Journal of Software Engineering and Knowledge Engineering, 30(01), 43–66. https://doi.org/10.1142/s0218194020500035

[5] Arora, M., Verma, S., & Chopra, S. (2020). A systematic literature review of machine learning estimation approaches in scrum projects. Cognitive Informatics and Soft Computing, 573-586.

[6] Prasada Rao, C., Siva Kumar, P., Rama Sree, S., & Devi, J. (2018). An agile effort estimation based on story points using machine learning techniques. In Proceedings of the Second International Conference on Computational Intelligence and Informatics (pp. 209-219). Springer, Singapore.

[7] 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)

[8] 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)

[9] 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)

[10] 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)

[11] 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)

[12] Menzies, T., Yang, Y., Mathew, G., Boehm, B., Hihn, J.: Negative results for software effort estimation. Empir. Softw. Eng. 22(5), 2658–2683 (2017)

[13] 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)

[14] 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)

[15] Dragicevic, S., Celar, S., Turic, M.: Bayesian network model for task effort estimation in agile software development. J. Syst. Softw. 127, 109–119 (2017)

[16] Soren, N. ., & Selvi Rajendran , P. . (2023). Artificial Intelligence Based - Oryza Sativa Leaf Ailment Recognition using DCT with Deep NN. International Journal of Intelligent Systems and Applications in Engineering, 11(4s), 242–253. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2650

[17] 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)

[18] Padmaja, M., Haritha, D.: Software effort estimation using meta heuristic algorithm. Int. J. Adv. Res. Comput. Sci. 8(5), 196–201 (2017)

[19] 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)

[20] 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)

[21] 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)

[22] Adnan, M., Afzal, M.: Ontology based multiagent effort estimation system for Scrum Agile method. IEEE Access, 25993–26005 (2017)

[23] Satapathy, S.M., Rath, S.K.: Empirical assessment of machine learning models for agile software development effort estimation using story points. Innov. Syst. Softw. Eng. 13(2–3), 191–200 (2017)

[24] Prof. Nitin Sherje. (2017). Phase Shifters with Tunable Reflective Method Using Inductive Coupled Lines. International Journal of New Practices in Management and Engineering, 6(01), 08 - 13. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/50

[25] 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)

[26] 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).

[27] A. Przybylek, D. Kotecka, “Making agile retrospectives more awesome,” In Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, Czech Republic, 2017. https://doi.org/10.15439/2017F423

[28] A. Przybylek, M. Zakrzewski, “Adopting Collaborative Games into Agile Requirements Engineering,” In 13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE'18), Funchal, Madeira, Portugal, 2018. https://doi.org/10.5220/0006681900540064

[29] A. Przybyłek, M. Olszewski, "Adopting collaborative games into Open Kanban," In Federated Conference on Computer Science and Information Systems (FedCSIS), Gdansk, Poland, 2016. https://doi.org/10.15439/2016F509

[30] Ahammad, D. S. K. H. (2022). Microarray Cancer Classification with Stacked Classifier in Machine Learning Integrated Grid L1-Regulated Feature Selection. Machine Learning Applications in Engineering Education and Management, 2(1), 01–10. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/18

[31] K. Schwaber and J. Sutherland, “The Scrum Guide,” scrum.org, https://www.scrum.org/index.php/resources/scrum-guide. 2020.

[32] CollabNet VersionOne, “13th Annual State of Agile Report,” collab.net, https://www.stateofagile.com. 2020.

[33] A. Jung, “Machine Learning: Basic Principles,” arXiv: 1805.05052v11[cs.LG], https://arxiv.org/pdf/1805.05052.pdf. 2019.

[34] The Standish Group, “CHAOS Report 2015,” The Standish Group, https://www.standishgroup.com/sample_research_files/CHAOS Re port2015-Final.pdf. 2020.

[35] Vitality Chicago, “Agile Project Success Rates are 2X Higher than Traditional Projects,” VitalityChicago https://vitalitychicago.com/blog/agile-projects-are-more- successful-traditional-projects/. 2020.

[36] J. Kahles, J. Torronen, T. Huuhtanen and A. Jung, “Automating Root Cause Analysis via Machine Learning in Agile Software Testing Environments,” In Proceedings - 2019 IEEE 12th International Conference on Software Testing, Verification and Validation, ICST 2019, pp. 379-390. [8730163] Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ICST.2019.00047

[37] S.M. Satapathy and S.K. Rath, “Empirical assessment of machine learning models for agile software development effort estimation using story points,” Innovations in Systems and Software Engineering, 13(2-3), pp.191-200. 2017.

[38] L. Butgereit, 2019, “Using Machine Learning to Prioritize Automated Testing in an Agile Environment,” In 2019 Conference on Information Communications Technology and Society (ICTAS), pp. 1-6. IEEE. 2019.

[39] R. Hanslo and E. Mnkandla, “Scrum Adoption Challenges Detection Model: SACDM,” In Federated Conference on Computer Science and Information Systems (FedCSIS), Poznan, Poland: IEEE: 949–957. 2018.

[40] R. Hanslo, E. Mnkandla and A. Vahed, “Factors that contribute significantly to Scrum adoption,” In Federated Conference on Computer Science and Information Systems (FedCSIS), Leipzig, Germany: IEEE: 821–829. 2019.

[41] R. Hanslo, A. Vahed and E. Mnkandla, “Quantitative Analysis of the Scrum Framework,” In: Przybyłek A., Morales-Trujillo M. (eds) Advances in Agile and User-Centred Software Engineering. LASD 2019, MIDI 2019, Lecture Notes in Business Information Processing, vol 376. Springer, Cham. 2020. https://doi.org/10.1007/978-3-030-37534-8_5

[42] A. Gelman, B. Goodrich, J. Gabry and A. Vehtari, “R-squared for Bayesian regression models,” The American Statistician, 73(3), pp.307-309. 2019.

[43] Bhawana Verma, S. K.A. (2019). Design & Analysis of Cost Estimation for New Mobile-COCOMO Tool for Mobile Application. International Journal on Recent and Innovation Trends in Computing and Communication, 7(1), 27–34. https://doi.org/10.17762/ijritcc.v7i1.5222

[44] S. K.A., Raj, A. ., Sharma, V., & Kumar, V. (2022). Simulation and Analysis of Hand Gesture Recognition for Indian Sign Language using CNN. International Journal on Recent and Innovation Trends in Computing and Communication, 10(4), 10–14. https://doi.org/10.17762/ijritcc.v10i4.5556.