Research Trends in Software Development Effort Estimation

Yulia Swandari
Department of Electrical and Information Engineering
Universitas Gadjah Mada
Yogyakarta, Indonesia
yuliaswandari@mail.ugm.ac.id

Ridi Ferdiana
Department of Electrical and Information Engineering
Universitas Gadjah Mada
Yogyakarta, Indonesia
ridi@ugm.ac.id

Adhistya Erna Permanasari
Department of Electrical and Information Engineering
Universitas Gadjah Mada
Yogyakarta, Indonesia
adhistya@ugm.ac.id

2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI) | 979-8-3503-0686-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/EECSI59885.2023.10295716

***Abstract*— Developing a software project without the appropriate amount of effort would significantly impede and even fail the project, putting the software developer's quality at risk. Therefore, software development effort estimation (SDEE) is the most critical activity in software engineering. SDEE has seen extensive research, resulting in a massive rise in the literature in a relatively short period. In this regard, it is crucial to identify the significant study topics in software development effort estimation that will assist researchers in understanding and recognizing research trends. This research applied a systematic literature review (SLR) to compile all journals from the predefined search directory about software development effort estimation thoroughly and unbiasedly from 2018 to 2022. This review was a prelude to further research activities in software development effort estimation. Five research topics out of 71 papers have been revealed, including the machine learning approach, algorithmic technique, expert judgement, dataset analysis, and evaluation metric. With 27 journals, deploying a machine learning approach for SDEE is the most discussed research topic. The potential research described in this study can be investigated further in software development effort estimation field.**

Keywords*— *software development, effort estimation, literature review, research trends

I. Introduction

Software development effort estimation (SDEE) predicts the effort needed to develop a software application [1]. SDEE can also be shortened to software effort estimation (SEE) or several synonymous words: cost estimation, cost prediction, time estimation, and effort prediction. The number of man-hours dedicated to software development over time, from specification up to delivery, is generally measured by SDEE [2]. But not limited to that, SDEE can also define the skill requirements that must be possessed and the number of costs, time, and assets needed to work on the project system. Overestimation wastes resources, whereas underestimation results in schedule or budget overruns and quality compromises [2]. Software professionals and researchers highly value accurate estimation of software effort.

Many effort estimation methods have been suggested in the SDEE field since the 1980s [1]. Both benefits and drawbacks of the models and approaches created by researchers are well acknowledged. The vast availability of published literature on the domain challenged researchers to analyze and determine the optimal path for their research [3].

The review article on expert estimation of SDEE conducted by Jørgensen [4] reviewed 15 studies for validating the conformance to twelve expert estimation 'best practices' and found expert estimation to be the dominant approach for SDEE, with no confirmation to support model estimates above expert estimations.

Machine learning (ML) technique in SDEE was investigated by Wen et al. [1] who determined different ML techniques, their prediction precision, and the comparison between other models and estimation contexts from 84 primary studies. Based on their research, they disclosed that eight different ML techniques had been used in SDEE and concluded that ML models offer more precise predictions than non-ML models.

Effort estimation in agile software development (ASD) was reviewed by Usman et al. [5] based on 20 papers selected. The study showed that most methods rely on expert judgement; extreme programming (XP) and scrum are the only two agile methods identified in the primary analysis.

Idri et al. [6] reviewed 65 papers published on analogy-based software effort estimation (ASEE) from 1990 to 2012 and revealed that the research's primary focus is feature and case subset selection. They identified that ASEE methods outperform eight strategies and produce more accurate results when combined with fuzzy logic (FL) or genetic algorithms (GA).

Research patterns in SDEE were summarized by Sehra et al. [3] through a generative statistical method called Latent Dirichlet Allocation (LDA) from a large set of SDEE research articles published from 1996 to 2016. They found twelve core research areas and sixty research trends based on a library of 1178 articles.

This research used a systematic literature review (SLR) to update the summarization of all papers from the predefined search directory concerning software development effort estimation from 2018 to 2022. An SLR is an approach to discovering, analyzing, and comprehending all accessible research on a specific topic area [7]. Research trends identified through this review can assist early researchers in finding and updating potential research areas in the software development effort estimation.

The rest of this research is structured as follows. Section 2 describes the research method, including the planning and conducting stage. Section 3 discusses key findings, and in Section 4 challenges to validity are presented. Finally, Section 5 provides a conclusion.

II. Research Method

We planned and conducted the SLR following Kitchenham & Charter [7] guidelines. All the steps are explained in the following subsections.

A. Planning Stage

The most significant action during planning is developing the research questions (RQ) [7]. This research addressed the following inquiries.

RQ1: What are the research trends in SDEE for the last five years, and what topic is the most discussed? This question was intended to identify the SDEE research trends from 2018 until 2022 and analyze the most discussed topic in SDEE.

RQ2: What are the potential future research topics in SDEE? This question was intended to examine the potential for future research in the SDEE field.

B. Conducting Stage

As seen in [7], "the aim of a systematic review is to find as many primary studies relating to the research question as possible using an unbiased search strategy". One characteristic that differentiates a systematic literature review (also called a systematic review) from a traditional review is the rigour of the search procedure. The search procedure used in this study involves search strategy, inclusion and exclusion criteria, data extraction, and data synthesis.

First, search strategy consisted of keywords and search strings used in the electronic source to acquire the relevant literature. The electronic source used in this research was SCOPUS because it is the most popular and widely accepted database by significant institutions worldwide and provides the most comprehensive coverage of reputable scientific literature among other databases.

We used the following search string to include the keywords and filters applied: TITLE (software AND (effort* OR cost* OR siz*) AND (estimat* OR predict* OR forecast*) AND NOT defect*) AND (LIMIT-TO (OA , "all")) AND (LIMIT-TO (PUBYEAR , 2022) OR LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018)) AND (LIMIT-TO (LANGUAGE , "English")). This search string was required to restrict the search results based on the following selection parameters:

- the publications on effort estimation with the option of using another relevant phrase and different spellings in the titles
- not in the defect prediction domain
- only open-access documents
- published during the five years from 2018 to 2022
- written in English

The search process resulted in 136 publications, and then we exported the results in a CSV file to facilitate further elimination of literature. When compiling this review using Zotero [8], these results were exported into RIS file format (.ris extension) to manage bibliographic references.

As indicated in Table 1, the search results from the search strategy were manually reselected by title and abstract using the inclusion and exclusion selection criteria. Only journal papers were selected because they had undergone peer review by the journal's publisher.

Table 2 displays the number of papers based on the search results and the manual selection procedure. The final steps are data extraction and data synthesis. The goal of data extraction is to gather all the information required to respond to the research questions. Data synthesis entails compiling and analyzing the findings of the included main papers, in this case, to categorize studies into significant study topics and year-wise publications.

TABLE I. INCLUSION AND EXCLUSION CRITERIA

Selection	Criteria
Inclusion	Journal paper Publicly available In the case of a duplication paper, the publication stage is chosen as the final one
Exclusion	Conference paper, review, survey, lecture note, book chapter, poster, erratum, replication, comparative study The topic outrange of SDEE

TABLE II. NUMBER OF PAPERS CHOSEN THROUGH THE SELECTION PROCESS

Search Results	Filtered Results	Selected Papers
136	65	71

III. RESULTS AND DISCUSSIONS

This section presents findings from reviewed papers according to two research questions. All the figures were visualized using MATLAB from the data synthesis results. Fig. 1 displays the year-by-year publishing of selected SDEE papers from 2018 to 2022. The total of 71 documents on SDEE shows an average upward trend every year.

Intending to answer **RQ1**, we have identified the five research topics presented in Table 3, and the most widely discussed research topic is the machine learning approach, with 27 journals.

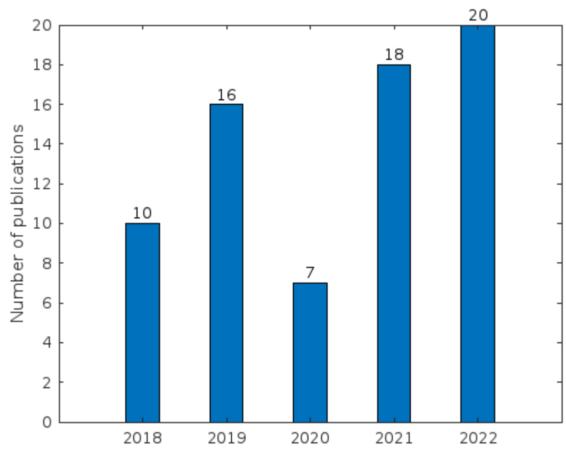

Fig. 1. Paper publication year-wise.

TABLE III. FIVE RESEARCH TOPICS

Id	Research Topics	Paper Count	Refs
T1	Machine learning approach	27	[9]–[35]
T2	Algorithmic technique	22	[36]–[57]
T3	Expert judgement	11	[58]–[68]
T4	Dataset analysis	10	[69]–[78]
T5	Evaluation metric	1	[79]

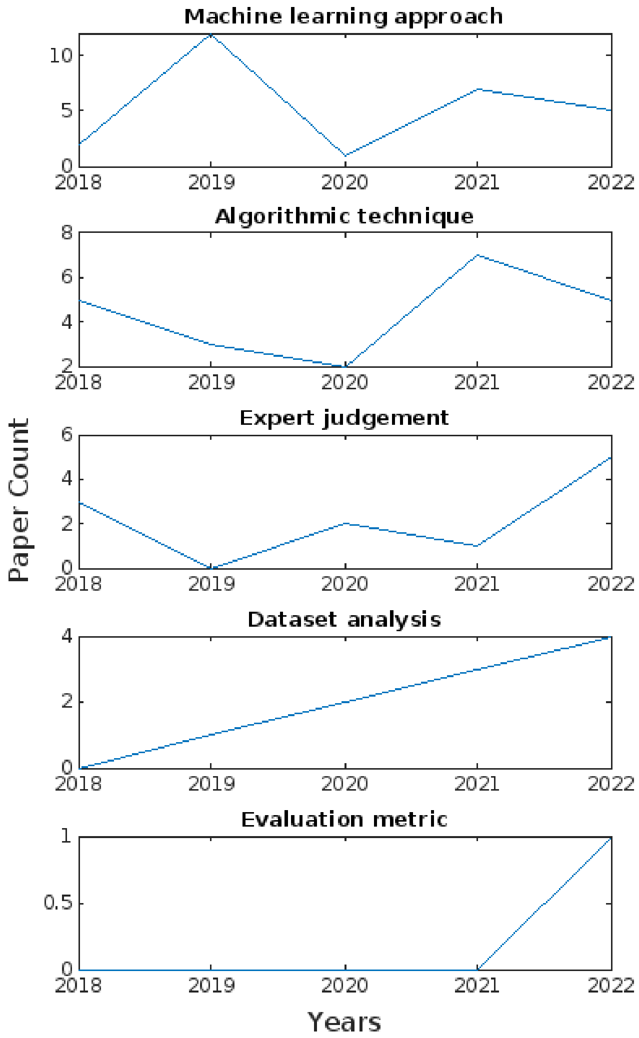

Fig. 2. Research trends of five topics.

Further, Fig. 2 highlights the research trends for each topic. The trends in research topics and the potential for further research are described in the subsequent section to answer **RQ2**.

A. Machine Learning Approach

Applying "machine learning approach" (T1) to estimate software effort has garnered numerous studies and uncovered several research trends. The paper counts from 2018 to 2022 for this research topic were 2, 12, 1, 7, and 5, respectively. The most widely explored are artificial neural networks (ANN), deep learning (DL), regression models, and ensemble methods. Some ANN, DL, and regression models combined with several techniques then become a hybrid or ensemble method, such as genetic algorithm and neural network in test effort estimation [11]; deep learning and metaheuristic algorithm [12], [19]; bayesian optimization and ensemble learning [13]; deep learning and random forest [14]; fuzzy-neural network and metaheuristic algorithm [15]; ANN and Taguchi method [20]; support vector regression (SVR) and feature selection [22]; metaheuristic algorithm and regression model [26]; SVR and metaheuristic algorithm [27]; SVR and neural network [28]; feature selection and multilayer perceptron [30]; bayesian and synthetic bootstrap [33]; and deep learning and function point [18].

Fuzzy logic is also applied in single and hybrid methods: fuzzy regression tree [9], [32], neuro-fuzzy [25], fuzzy delphi in test effort estimation [29], and fuzzy-neuro-genetic [35]. Furthermore, researchers also explored the techniques in the context of open-source software [10], dynamic cross-company mapped model learning (Dycom) [23], and unstructured software project descriptions [21]. Aside from the research trends stated above, some contemporary trends have evolved from the study, including adaptive neuro-fuzzy inference system (ANFIS) [16], extreme learning machine (ELM) [17], feature engineering [24], deep belief network [31], and random forest [34].

Carvalho et al. [17] applied the ELM technique and compared it with the literature models, which resulted in the error estimate rates tending to be reduced. For future studies, ELM and other machine learning methods can be optimized with metaheuristic algorithms, such as particle swarm optimization (PSO), to enhance their accuracy. Besides, there is the opportunity for additional study of:

- Deep learning models improve with fully automatic element integration.
- Design the transformation of the artificial neural network model to a web service and deploy it to a cloud computing platform.

B. Algorithmic Technique

Various trends have emerged in the "algorithmic technique" (T2), with the number of papers for the last five years being 5, 3, 2, 7, and 5. One such trend is the metaheuristic algorithm, namely particle swarm optimization [36], [39], [48], [56], differential evolution [41], antlion optimization [42], firefly algorithm [43], dolphin algorithm [49], artificial bee colony algorithm [57], and hybrid bat algorithm [55]. Other trends identified are the morphological approach [54], local data approach [37], regression analysis [38], use case point-based method [40], [44], global software development project [45], class diagram-based estimation [46], software rework index [47], use case reuse [50], safety-critical software [51], function point analysis [52], and linear programming [53].

Four papers improved and modified PSO to enhance its accuracy for effort estimation. Ardiansyah et al. [36] proposed modified chaotic particle swarm optimization with uniform particle initialization (MUCPSO) to improve the comprehensive performance of standard PSO by presenting three additional schemes. Venkataiah et al. [39] proposed chaotic linear increasing inertia weight and diversity-improved mechanism to enhance the variety of PSO. Alanis-Tamez [48] et al. presented the application of PSO for improving the parameters of statistical regression equations (SRE). Khuat et al. [56] proposed an improved algorithm combining the advantages of the artificial bee colony (ABC) and PSO algorithms. The variant improvements of PSO can be compared and combined with another method, such as a machine learning approach, in future work to enhance the effort estimation accuracy. Another research can be conducted:

- Investigate the impact of different methodologies, languages, datasets, and environments on algorithm performance.

C. Expert Judgement

"Expert judgement" (T3) reveals the research trends, including analogy-based estimation [58], [59], [63]–[65], [67], estimation in agile projects [61], [62], [66], [68] and blockchain-based models [60]. The paper counts for this topic in 2018, 2020, 2021, and 2022 were 3, 2, 1, and 5, respectively, and none in 2019.

Ahmed et al. [62] presented a blockchain-based software effort estimation (BBSEE) to handle disadvantages of analogy-based estimation, such as a scarcity of historical data and experts. In future research, the suggested method is compatible with a machine learning-based method that experts might employ when making choices. Another elaborative analysis can be carried out on:

- Exploring other factors influencing agile effort estimation, such as formal training in estimations and learning structures.
- Combining machine learning approach with various analogy-based estimations and expert checklists.

D. Dataset Analysis

In "dataset analysis" (T4), paper counts from 2019 until 2022 were 1, 2, 3, and 4, respectively, and none in 2018. The trends identified are size analysis [69], [73], weight values [70], outlier analysis [71], [75], clustering dataset [72], [74], replicate dataset [76], volatile requirements [77], and dataset quality [78].

The pace of change in requirements, not only in the development phase but also in the operating phase, is measured by requirement volatility. It exists in practically all software projects; hence, solutions for volatile requirements benefit practitioners and the software community. The future research path could be the classification of volatile needs concerning different types of software. Other potential topics for further exploration are:

- Elaborative studies on clustered or segmented data methods.
- Design and develop clear correction solutions for the identified outliers.

E. Evaluation Metric

The research area "evaluation metric" (T5) has not been discussed much in the last five years. The review results found only one paper [79] regarding evaluation metrics in 2022.

The paper offered formulas related to the PRED(x) and MMRE for expressing the degree of scatters of forecast values versus actual values on the left (sig left), on the right (sig right), and the mean of the scatters (sig). The authors indicated whether the model is over or under the baseline based on the sig left, sig right, and sig values. However, quantifying this overestimation or underestimation has not yet been explored. As this area is still understudied, the research community should prioritize investigating evaluation metrics to find future difficulties.

IV. Challenges To Validity

Two challenges to the validity of our research—first, the efficacy of the used search string. Due to the limits of search term selection and synonyms, search string strategies, and databases, an incomplete literature dataset may arise. Because of our keyword selection and search string, there is a chance that related papers will be missed. We rigorously examined in two steps to ensure the relevance of the selected publications in this study. Another threat to validity refers to the naming of research topics. It is an issue because of the subjectivity and bias involved. To get around this constraint, we read the journal's full text to determine appropriate and significant research topics.

V. Conclusions

This study applied a manual systematic review of the literature on SDEE research trends. It identified research trends by examining 71 journals published by researchers between 2018 and 2022. Five research topics, sorted from the most widely discussed, are (1) machine learning approach with 27 papers; (2) algorithmic technique with 22 papers; (3) expert judgement with 11 papers; (4) dataset analysis with 10 papers; and (5) evaluation metrics with 1 paper.

This review's research trends can help early researchers identify and update potential study areas in software development effort estimation. Future research trends in the SDEE field are on machine learning approaches, especially deep learning, and artificial neural networks, to produce a high level of accuracy in estimating work with efficient use of resources. Furthermore, researchers might evaluate any of the five research subjects indicated in the current research for additional investigation.

References

[1] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic Literature Review of Machine Learning Based Software Development Effort Estimation Models," *Inf. Softw. Technol.*, vol. 54, no. 1, pp. 41–59, Jan. 2012, doi: 10.1016/j.infsof.2011.09.002.

[2] M. O. Elish, T. Helmy, and M. I. Hussain, "Empirical Study of Homogeneous and Heterogeneous Ensemble Models for Software Development Effort Estimation," *Math. Probl. Eng.*, vol. 2013, p. 312067, Jul. 2013, doi: 10.1155/2013/312067.

[3] S. K. Sehra, Y. S. Brar, N. Kaur, and S. S. Sehra, "Research Patterns and Trends in Software Effort Estimation," *Inf. Softw. Technol.*, vol. 91, pp. 1–21, Nov. 2017, doi: 10.1016/j.infsof.2017.06.002.

[4] M. Jørgensen, "A Review of Studies on Expert Estimation of Software Development Effort," *J. Syst. Softw.*, vol. 70, no. 1, pp. 37–60, Feb. 2004, doi: 10.1016/S0164-1212(02)00156-5.

[5] M. Usman, E. Mendes, F. Neiva, and R. Britto, "Effort Estimation in Agile Software Development: A Systematic Literature Review," in *10th International Conference on Predictive Models in Software Engineering*, ACM, Torino, Italy, Sep. 2014, pp. 82–91. doi: 10.1145/2639490.2639503.

[6] A. Idri, F. azzahra Amazal, and A. Abran, "Analogy-Based Software Development Effort Estimation: A Systematic Mapping and Review," *Inf. Softw. Technol.*, vol. 58, pp. 206–230, Feb. 2015, doi: 10.1016/j.infsof.2014.07.013.

[7] B. Kitchenham and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software Engineering Version 2.3," *Engineering*, vol. 45, no. 4ve, p. 1051, 2007.

[8] "Zotero." Corporation for Digital Scholarship, Oct. 07, 2014. [Online]. Available: https://www.zotero.org/

[9] A. Najm, A. Zakrani, and A. Marzak, "Cluster-Based Fuzzy Regression Trees for Software Cost Prediction," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 27, no. 2, pp. 1138–1150, 2022, doi: 10.11591/ijeecs.v27.i2.pp1138-1150.

[10] R. Kapur and B. Sodhi, "OSS Effort Estimation Using Software Features Similarity and Developer Activity-Based Metrics," *ACM Trans. Softw. Eng. Methodol.*, vol. 31, no. 2, 2022, doi: 10.1145/3485819.

[11] V. Chahar and P. K. Bhatia, "Performance Analysis of Software Test Effort Estimation Using Genetic Algorithm and Neural Network," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 10, pp. 376–383, 2022, doi: 10.14569/IJACSA.2022.0131045.

[12] C. A. Ul Hassan *et al.*, "Optimizing Deep Learning Model for Software Cost Estimation Using Hybrid Meta-Heuristic Algorithmic

Approach," *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/3145956.
[13] R. Marco, S. S. S. Ahmad, and S. Ahmad, "Bayesian Hyperparameter Optimization and Ensemble Learning for Machine Learning Models on Software Effort Estimation," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 3, pp. 419–429, 2022, doi: 10.14569/IJACSA.2022.0130351.
[14] A. G. Priya Varshini, K. Anitha Kumari, and V. Varadarajan, "Estimating Software Development Efforts Using A Random Forest-Based Stacked Ensemble Approach," *Electron. Switz.*, vol. 10, no. 10, 2021, doi: 10.3390/electronics10101195.
[15] A. Karimi and T. J. Gandomani, "Software Development Effort Estimation Modeling Using A Combination of Fuzzy-Neural Network and Differential Evolution Algorithm," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 1, pp. 707–715, 2021, doi: 10.11591/ijece.v11i1.pp707-715.
[16] Z. R. Mohsin, "Investigating the Use of An Adaptive Neuro-Fuzzy Inference System in Software Development Effort Estimation," *Iraqi J. Comput. Sci. Math.*, vol. 2, no. 2, pp. 18–24, 2021, doi: 10.52866/ijcsm.2021.02.02.003.
[17] H. D. P. De Carvalho, R. Fagundes, and W. Santos, "Extreme Learning Machine Applied to Software Development Effort Estimation," *IEEE Access*, vol. 9, pp. 92676–92687, 2021, doi: 10.1109/ACCESS.2021.3091313.
[18] K. Zhang, X. Wang, J. Ren, and C. Liu, "Efficiency Improvement of Function Point-Based Software Size Estimation With Deep Learning Model," *IEEE Access*, vol. 9, pp. 107124–107136, 2021, doi: 10.1109/ACCESS.2020.2998581.
[19] M. S. Khan, F. Jabeen, S. Ghouzali, Z. Rehman, S. Naz, and W. Abdul, "Metaheuristic Algorithms in Optimizing Deep Neural Network Model for Software Effort Estimation," *IEEE Access*, vol. 9, pp. 60309–60327, 2021, doi: 10.1109/ACCESS.2021.3072380.
[20] N. Rankovic, D. Rankovic, M. Ivanovic, and L. Lazic, "A New Approach to Software Effort Estimation Using Different Artificial Neural Network Architectures and Taguchi Orthogonal Arrays," *IEEE Access*, vol. 9, pp. 26926–26936, 2021, doi: 10.1109/ACCESS.2021.3057807.
[21] T. Kangwantrakool, K. Viriyayudhakorn, and T. Theeramunkong, "Software Development Effort Estimation From Unstructured Software Project Description by Sequence Models," *IEICE Trans. Inf. Syst.*, vol. E103D, no. 4, pp. 739–747, 2020, doi: 10.1587/transinf.2019IIP0014.
[22] A. Zakrani, M. Hain, and A. Idri, "Improving Software Development Effort Estimation Using Support Vector Regression and Feature Selection," *IAES Int. J. Artif. Intell.*, vol. 8, no. 4, pp. 399–410, 2019, doi: 10.11591/ijai.v8.i4.pp399-410.
[23] L. L. Minku, "A Novel Online Supervised Hyperparameter Tuning Procedure Applied to Cross-Company Software Effort Estimation," *Empir. Softw. Eng.*, vol. 24, no. 5, pp. 3153–3204, 2019, doi: 10.1007/s10664-019-09686-w.
[24] S. P. Pillai, T. Radharamanan, and S. D. Madhukumar, "Feature Engineering for Enhanced Model Performance in Software Effort Estimation," *Int. J. Recent Technol. Eng.*, vol. 8, no. 3, pp. 6053–6063, 2019, doi: 10.35940/ijrte.C5602.098319.
[25] S. Rama Sree, S. N. S. V. S. C. Ramesh, and C. Prasada Rao, "An Effective Software Effort Estimation Based on Functional Points Using Soft Computing Techniques," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 10, pp. 3729–3733, 2019, doi: 10.35940/ijitee.J9674.0881019.
[26] S. W. Ahmad and G. R. Bamnote, "Whale–Crow Optimization (WCO)-Based Optimal Regression Model for Software Cost Estimation," *Sadhana - Acad. Proc. Eng. Sci.*, vol. 44, no. 4, pp. 1–15, 2019, doi: 10.1007/s12046-019-1085-1.
[27] R. Marco, N. S. Herman, and S. S. S. Ahmad, "Optimizing Software Effort Estimation Models Based on Metaheuristic Methods: A Proposed Framework," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 1.5 Special Issue, pp. 294–304, 2019, doi: 10.30534/ijatcse/2019/5181.52019.
[28] V. K. Attri and J. S. Bal, "An Advanced Mechanism for Software Size Estimation Using Combinational Artificial Intelligence," *Int. J. Intell. Eng. Syst.*, vol. 12, no. 4, pp. 255–263, 2019, doi: 10.22266/ijies2019.0831.24.
[29] M. M. Rejab, N. F. M. Azmi, and S. Chuprat, "Fuzzy Delphi Method for Evaluating HyTEE Model (Hybrid Software Change Management Tool With Test Effort Estimation)," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 4, pp. 529–535, 2019, doi: 10.14569/ijacsa.2019.0100465.
[30] E. Ertuğrul, Z. Baytar, Ç. Çatal, and C. Muratli, "Performance Tuning for Machine Learning-Based Software Development Effort Prediction Models," *Turk. J. Electr. Eng. Comput. Sci.*, vol. 27, no. 2, pp. 1308–1324, 2019, doi: 10.3906/elk-1809-129.
[31] H. M. Premalatha and C. V. Srikrishna, "Effort Estimation in Agile Software Development Using Evolutionary Cost- Sensitive Deep Belief Network," *Int. J. Intell. Eng. Syst.*, vol. 12, no. 2, pp. 261–269, 2019, doi: 10.22266/IJIES2019.0430.25.
[32] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software Development Effort Estimation Using Regression Fuzzy Models," *Comput. Intell. Neurosci.*, vol. 2019, 2019, doi: 10.1155/2019/8367214.
[33] L. Song, L. L. Minku, and Y. A. O. Xin, "Software Effort Interval Prediction via Bayesian Inference and Synthetic Bootstrap Resampling," *ACM Trans. Softw. Eng. Methodol.*, vol. 28, no. 1, 2019, doi: 10.1145/3295700.
[34] A. Zakrani, M. Hain, and A. Namir, "Software Development Effort Estimation Using Random Forests: An Empirical Study and Evaluation," *Int. J. Intell. Eng. Syst.*, vol. 11, no. 6, pp. 300–311, 2018, doi: 10.22266/IJIES2018.1231.30.
[35] P. Rijwani and S. Jain, "Level of Confidence in Software Effort Estimation by An Intelligent Fuzzy - Neuro - Genetic Approach," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 9, pp. 568–576, 2018, doi: 10.14569/ijacsa.2018.090972.
[36] A. Ardiansyah, R. Ferdiana, and A. E. Permanasari, "MUCPSO: A Modified Chaotic Particle Swarm Optimization With Uniform Initialization for Optimizing Software Effort Estimation," *Appl. Sci. Switz.*, vol. 12, no. 3, 2022, doi: 10.3390/app12031081.
[37] Y. Alqasrawi, M. Azzeh, and Y. Elsheikh, "Locally Weighted Regression With Different Kernel Smoothers for Software Effort Estimation," *Sci. Comput. Program.*, vol. 214, 2022, doi: 10.1016/j.scico.2021.102744.
[38] A. Sharma and N. Chaudhary, "The Combined Model for Software Development Effort Estimation Using Polynomial Regression for Heterogeneous Projects," *Radioelectron. Comput. Syst.*, vol. 2022, no. 2, pp. 75–82, 2022, doi: 10.32620/reks.2022.2.06.
[39] V. Venkataiah, M. Nagaratna, and R. Mohanty, "Application of Chaotic Increasing Linear Inertia Weight and Diversity Improved Particle Swarm Optimization to Predict Accurate Software Cost Estimation," *Int. J. Electr. Electron. Res.*, vol. 10, no. 2, pp. 154–160, 2022, doi: 10.37391/IJEER.100218.
[40] H. L. T. K. Nhung, V. Van Hai, R. Silhavy, Z. Prokopova, and P. Silhavy, "Parametric Software Effort Estimation Based on Optimizing Correction Factors and Multiple Linear Regression," *IEEE Access*, vol. 10, pp. 2963–2986, 2022, doi: 10.1109/ACCESS.2021.3139183.
[41] S. P. Singh, V. P. Singh, and A. K. Mehta, "Differential Evolution Using Homeostasis Adaption Based Mutation Operator and Its Application for Software Cost Estimation," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 33, no. 6, pp. 740–752, 2021, doi: 10.1016/j.jksuci.2018.05.009.
[42] M. A.-A. Alabajee, N. A. AL-Saati, and T. R. Alreffaee, "Parameter Tuning of Software Effort Estimation Models Using Antlion Optimization," *Telkomnika Telecommun. Comput. Electron. Control*, vol. 19, no. 3, pp. 817–828, 2021, doi: 10.12928/TELKOMNIKA.v19i3.16907.
[43] A. Moradbeiky, V. K. Bardsiri, and M. Jafari, "Open Hybrid Model: A New Esemble Model for Software Development Cost Estimation," *Comput. Inform.*, vol. 39, no. 6, pp. 1148–1171, 2021, doi: 10.31577/CAI_2020_6_1148.
[44] R. Silhavy, P. Silhavy, and Z. Prokopova, "Using Actors and Use Cases for Software Size Estimation," *Electron. Switz.*, vol. 10, no. 5, pp. 1–21, 2021, doi: 10.3390/electronics10050592.
[45] J. A. Khan, S. U. R. Khan, T. A. Khan, and I. U. R. Khan, "An Amplified COCOMO-II Based Cost Estimation Model in Global Software Development Context," *IEEE Access*, vol. 9, pp. 88602–88620, 2021, doi: 10.1109/ACCESS.2021.3089870.
[46] M. Daud and A. A. Malik, "Improving the Accuracy of Early Software Size Estimation Using Analysis-to-Design Adjustment Factors (ADAFs)," *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3085752.
[47] T. A. Khalid and E.-T. Yeoh, "Enhancing Software Development Cost Control by Forecasting the Cost of Rework: Preliminary Study," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 1, pp. 524–537, 2021, doi: 10.11591/ijeecs.v21.i1.pp524-537.
[48] M. D. Alanis-Tamez, C. López-Martín, and Y. Villuendas-Rey, "Particle Swarm Optimization for Predicting the Development Effort of Software Projects," *Mathematics*, vol. 8, no. 10, pp. 1–21, 2020, doi: 10.3390/math8101819.
[49] A. A. Fadhil, R. G. H. Alsarraj, and A. M. Altaie, "Software Cost Estimation Based on Dolphin Algorithm," *IEEE Access*, vol. 8, pp. 75279–75287, 2020, doi: 10.1109/ACCESS.2020.2988867.

[50] K. Rak, Ž. Car, and I. Lovrek, "Effort Estimation Model for Software Development Projects Based on Use Case Reuse," *J. Softw. Evol. Process*, vol. 31, no. 2, 2019, doi: 10.1002/smr.2119.
[51] L. P. dos Santos and M. Ferreira, "Applying COCOMO II for A DO-178C Safety-Critical Software Effort Estimation," *J. Aerosp. Technol. Manag.*, vol. 11, 2019, doi: 10.5028/jatm.v11.1031.
[52] P. Silhavy, R. Silhavy, and Z. Prokopova, "Categorical Variable Segmentation Model for Software Development Effort Estimation," *IEEE Access*, vol. 7, pp. 9618–9626, 2019, doi: 10.1109/ACCESS.2019.2891878.
[53] F. Sarro and A. Petrozziello, "Linear Programming as A Baseline for Software Effort Estimation," *ACM Trans. Softw. Eng. Methodol.*, vol. 27, no. 3, 2018, doi: 10.1145/3234940.
[54] S. Bilgaiyan, K. Aditya, S. Mishra, and M. Das, "A Swarm Intelligence Based Chaotic Morphological Approach for Software Development Cost Estimation," *Int. J. Intell. Syst. Appl.*, vol. 10, no. 9, pp. 13–22, 2018, doi: 10.5815/ijisa.2018.09.02.
[55] D. Nandal and O. P. Sangwan, "Software Cost Estimation by Optimizing COCOMO Model Using Hybrid BATGSA Algorithm," *Int. J. Intell. Eng. Syst.*, vol. 11, no. 4, pp. 250–263, 2018, doi: 10.22266/ijies2018.0831.25.
[56] T. T. Khuat and M. H. Le, "A Novel Hybrid ABC-PSO Algorithm for Effort Estimation of Software Projects Using Agile Methodologies," *J. Intell. Syst.*, vol. 27, no. 3, pp. 489–506, 2018, doi: 10.1515/jisys-2016-0294.
[57] S. Yigit-Sert and P. Kullu, "Software Cost Estimation Using Enhanced Artificial Bee Colony Algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 4, pp. 67–70, 2018, doi: 10.14569/IJACSA.2018.090412.
[58] E. M. De Bortoli Fávero, D. Casanova, and A. R. Pimentel, "SE3M: A Model for Software Effort Estimation Using Pre-Trained Embedding Models," *Inf. Softw. Technol.*, vol. 147, 2022, doi: 10.1016/j.infsof.2022.106886.
[59] A. Kaushik, P. Kaur, N. Choudhary, and Priyanka, "Stacking Regularization in Analogy-Based Software Effort Estimation," *Soft Comput.*, vol. 26, no. 3, pp. 1197–1216, 2022, doi: 10.1007/s00500-021-06564-w.
[60] M. Ahmed *et al.*, "Blockchain-Based Software Effort Estimation: An Empirical Study," *IEEE Access*, vol. 10, pp. 120412–120425, 2022, doi: 10.1109/ACCESS.2022.3216840.
[61] P. Sudarmaningtyas and R. Mohamed, "Significant Factors in Agile Software Development of Effort Estimation," *Pertanika J. Sci. Technol.*, vol. 30, no. 4, pp. 2851–2878, 2022, doi: 10.47836/pjst.30.4.30.
[62] L. Cao, "Estimating Efforts for Various Activities in Agile Software Development: An Empirical Study," *IEEE Access*, vol. 10, pp. 83311–83321, 2022, doi: 10.1109/ACCESS.2022.3196923.
[63] M. Dashti, T. J. Gandomani, D. H. Adeh, H. Zulzalil, and A. B. M. Sultan, "LEMABE: A Novel Framework to Improve Analogy-Based Software Cost Estimation Using Learnable Evolution Model," *PeerJ Comput. Sci.*, vol. 7, 2021, doi: 10.7717/PEERJ-CS.800.
[64] M. A. Shah, D. N. A. Jawawi, M. A. Isa, M. Younas, A. Abdelmaboud, and F. Sholichin, "Ensembling Artificial Bee Colony With Analogy-Based Estimation to Improve Software Development Effort Prediction," *IEEE Access*, vol. 8, pp. 58402–58415, 2020, doi: 10.1109/ACCESS.2020.2980236.
[65] V. Resmi and S. Vijayalakshmi, "Analogy-Based Approaches to Improve Software Project Effort Estimation Accuracy," *J. Intell. Syst.*, vol. 29, no. 1, pp. 1468–1479, 2020, doi: 10.1515/jisys-2019-0023.
[66] M. Usman, K. Petersen, J. Börstler, and P. Santos Neto, "Developing and Using Checklists to Improve Software Effort Estimation: A Multi-Case Study," *J. Syst. Softw.*, vol. 146, pp. 286–309, 2018, doi: 10.1016/j.jss.2018.09.054.
[67] Ardiansyah, M. M. Mardhia, and S. Handayaningsih, "Analogy-Based Model for Software Project Effort Estimation," *Int. J. Adv. Intell. Inform.*, vol. 4, no. 3, pp. 251–260, 2018, doi: 10.26555/ijain.v4i3.266.
[68] M. Usman, R. Britto, L.-O. Damm, and J. Börstler, "Effort Estimation in Large-Scale Software Development: An Industrial Case Study," *Inf. Softw. Technol.*, vol. 99, pp. 21–40, 2018, doi: 10.1016/j.infsof.2018.02.009.
[69] S. Sharma, P. C. Pendharkar, and Karmeshu, "Learning Component Size Distributions for Software Cost Estimation: Models Based on Arithmetic and Shifted Geometric Means Rules," *IEEE Trans. Softw. Eng.*, vol. 48, no. 12, pp. 5136–5147, 2022, doi: 10.1109/TSE.2021.3139216.
[70] V. V. Hai, H. L. T. K. Nhung, Z. Prokopova, R. Silhavy, and P. Silhavy, "A New Approach to Calibrating Functional Complexity Weight in Software Development Effort Estimation," *Computers*, vol. 11, no. 2, 2022, doi: 10.3390/computers11020015.
[71] S. S. Gautam and V. Singh, "Adaptive Discretization Using Golden Section to Aid Outlier Detection for Software Development Effort Estimation," *IEEE Access*, vol. 10, pp. 90369–90387, 2022, doi: 10.1109/ACCESS.2022.3200149.
[72] V. Van Hai, H. L. T. K. Nhung, Z. Prokopova, R. Silhavy, and P. Silhavy, "Toward Improving the Efficiency of Software Development Effort Estimation via Clustering Analysis," *IEEE Access*, vol. 10, pp. 83249–83264, 2022, doi: 10.1109/ACCESS.2022.3185393.
[73] P. Rai, D. K. Verma, and S. Kumar, "A Hybrid Model for Prediction of Software Effort Based on Team Size," *IET Softw.*, vol. 15, no. 6, pp. 365–375, 2021, doi: 10.1049/sfw2.12048.
[74] P. Silhavy, R. Silhavy, and Z. Prokopova, "Spectral Clustering Effect in Software Development Effort Estimation," *Symmetry*, vol. 13, no. 11, 2021, doi: 10.3390/sym13112119.
[75] K. Ono, M. Tsunoda, A. Monden, and K. Matsumoto, "Influence of Outliers on Estimation Accuracy of Software Development Effort," *IEICE Trans. Inf. Syst.*, vol. E104D, no. 1, pp. 91–105, 2021, doi: 10.1587/transinf.2020MPP0005.
[76] M. Gan, Z. Yücel, A. Monden, and K. Sasaki, "Empirical Evaluation of Mimic Software Project Data Sets for Software Effort Estimation," *IEICE Trans. Inf. Syst.*, vol. E103D, no. 10, pp. 2094–2103, 2020, doi: 10.1587/transinf.2019EDP7150.
[77] I. Hassan, A. Latif, K. Khan, F. Hassan, and M. Saeed, "Accurate Cost Estimation for Software With Volatile Requirements," *Int. J. Adv. Appl. Sci.*, vol. 7, no. 3, pp. 1–8, 2020, doi: 10.21833/ijaas.2020.03.001.
[78] M. F. Bosu and S. G. Macdonell, "Experience: Quality Benchmarking of Datasets Used in Software Effort Estimation," *J. Data Inf. Qual.*, vol. 11, no. 4, 2019, doi: 10.1145/3328746.
[79] H. Huynh Thai, P. Silhavy, M. Fajkus, Z. Prokopova, and R. Silhavy, "Propose-Specific Information Related to Prediction Level at x and Mean Magnitude of Relative Error: A Case Study of Software Effort Estimation," *Mathematics*, vol. 10, no. 24, 2022, doi: 10.3390/math10244649.