Classifier Ensembles for Software Development Team Effort Estimation: A Rapid Systematic Literature Review

Wilamis Kleiton Nunes da Silva[1], Rafael Batista Duarte[1], Bernan Rodrigues Nascimento[2]

[1]Cesar School, Recife, Pernambuco, Brazil

Caixa Postal 50030-220 –Avenida Cais do Apolo –77 – Recife –PE – Brazil

DPES

[2]Postgraduate Program in Computer Science

Universidade Federal do Piauí (UFPI) – Teresina, PI – Brazil

{wkns,rbd}@cesar.school, bernanr7@gmail.com

Abstract. *This work presents a systematic literature review (SLR) on the use of classifier ensembles for software team effort estimation. The study analyzes the advantages of homogeneous and heterogeneous ensembles, highlighting how the combination of algorithms improves prediction accuracy. A total of 27 relevant studies were examined, using metrics such as MAE, RMSE, and PRED to assess model performance. The results indicate that heterogeneous ensembles are more effective in environments with high variability, while homogeneous ensembles excel in specific domains. The work suggests the exploration of hybrid metrics and dynamic optimization for future research.*

1. Introduction

Software development team effort estimation aims to calculate the amount of effort required to successfully complete a project [Goyal 2022a]. According to [Nassif et al. 2013], software projects may fail for several reasons; however, imprecise estimates and the misunderstanding or incompleteness of requirements are particularly noteworthy. The accuracy of estimates, preferably performed in the early stages of the software life cycle, is crucial for improving software project management. Software effort estimation remains a challenging task in the software development industry, and machine learning methods have been increasingly applied to improve estimation accuracy [Wen et al. 2012]. Predicting software development team effort estimation is a fundamental task for planning and executing software development projects. The ability to accurately and reliably estimate the effort required to complete specific activities is essential for strategic decision-making and effective management of available resources [Sakhrawi et al. 2022a].

According to the taxonomy proposed by [Boehm et al. 2000], software development team effort estimation techniques can be classified into: Empirical and Composite approaches, which encompass traditional techniques based on data from previous projects; Dynamic approaches, which consider changes in estimates over the project timeline; Expert judgment-based approaches, which focus on expert analysis and are often used in conjunction with other techniques to correct discrepancies; and Machine Learning (ML)-oriented approaches, which exploit historical domain data by using algorithms to formulate or infer rules and/or models capable of predicting future values.

According to [Hosni et al. 2019], a promising approach to improving collaboration and efficiency in software development is the use of classifier ensembles. In the literature, classifier ensembles are defined as an ML technique that combines the predictions of multiple individual models to obtain more accurate predictions [Araújo 2019]. This approach has gained prominence compared to traditional ML techniques [Cabral and Oliveira 2021]. Several studies have investigated the use of ensemble methods for software development team effort estimation [Senevirathne and Wijayasiriwardhane 2020], [Benaroch and Lyytinen 2023], [Venson 2020], [Mahmood et al. 2020a], among others. These studies include comparisons among models based on different techniques, evaluated using performance metrics and specific datasets to measure estimation accuracy. The joint use of multiple classifiers allows the exploitation of diversity in opinions and approaches across different prediction algorithms. This collaborative and integrated approach can significantly contribute to improving the accuracy and reliability of software development team effort estimations [Seni and Elder 2010].

In this context, the following research question emerged: What evidence is available in the literature regarding the use of classifier ensembles for predicting software development team effort estimation? The objective of this study was to conduct a systematic literature review (SLR) on the use of classifier ensembles for predicting software development team effort estimation, aiming to identify research gaps and opportunities for future investigations in this field. This study is organized as follows: Section 2 details the applied protocol; Section 3 presents the study results; and finally, Section 4 provides the concluding remarks.

The main contributions of this study include: i) a detailed analysis of the advantages and limitations of homogeneous and heterogeneous ensembles in software development team effort estimation; ii) the identification of research gaps, such as the lack of dynamic model selection strategies, the impact of feature diversity, and the absence of practical validation in real-world environments; iii) suggestions for future research directions, highlighting the use of hybrid metrics, evolutionary techniques, multi-objective optimization, and metric standardization to improve model robustness and generalization.

2. Applied Protocol

For conducting the SLR methodology, the study by [Kitchenham et al. 2015] was adopted, which is a prominent reference in the field of computer science for the development of SLRs. This study provides robust guidelines, ensuring the accuracy and reproducibility of the results. The following steps were carried out to achieve the objectives of this study:

Research Planning: In this phase, the research questions were defined and a review protocol was developed. The protocol included inclusion and exclusion criteria, data sources, search strategies, and methods for data extraction and synthesis; **Research Execution:** During this phase, relevant studies were selected. Specific search strings were applied to the selected digital libraries. The studies were filtered based on the previously defined inclusion and exclusion criteria. Subsequently, data from the selected studies were extracted and synthesized. **Results Synthesis:** Finally, the studies that met the established criteria were considered in the SLR. In this phase, the extracted data were analyzed and synthe-

sized to answer the research questions. These steps ensured that the SLR was conducted in a systematic manner, facilitating the reliable aggregation of evidence.

2.1. Research Questions

The objective of this study was to conduct an SLR on the use of classifier ensembles for predicting software team effort estimation, aiming to identify research gaps and opportunities for future studies in this area. To achieve this objective, the following research questions were formulated:

Q1) Which datasets are most commonly used in the studies?

Motivation: Identifying the datasets most frequently reported in the literature makes it possible to understand which data sources are considered standard or reliable for software effort estimation prediction.

Q2) Which evaluation metrics and machine learning (ML) techniques are most commonly applied in the studies?

Motivation: Understanding the evaluation metrics and classifier ensemble techniques most frequently employed allows the identification of the main criteria used to assess the accuracy of software team effort estimation. This supports the definition of benchmarks and facilitates performance comparison across different approaches, promoting the selection of more effective methods.

2.2. Digital Libraries

The digital libraries used in this study were selected based on their scientific relevance in the fields of computer science and informatics. These sources are widely recognized by the scientific community for their high quality standards and ease of access to indexed publications. The digital libraries adopted in this research for study collection were: IEEE Xplore[1] , Scopus[2] , ACM Digital Library[3] , and Web of Science[4].

2.3. Search Strategy

The selected digital libraries were used to conduct the study, covering works published between 2019 and 2025, written in English. The choice of English as the inclusion language is justified by its universality in scientific communication. The search string defined for this SLR was:

*(“Heterogeneous Ensemble” **OR** “Homogeneous Ensemble”) **AND** (”Software Development Effort” **OR** ”Software Project Effort”) **AND** (”Machine Learning” **OR** ”Artificial Intelligence”)*

Figure 1 illustrates the methodological process of this study. The search across the digital libraries resulted in the identification of 635 studies. IEEE Xplore was the digital library with the highest number of results, totaling 495 studies, followed by Scopus with 133 studies. The ACM Digital Library and Web of Science returned 3 and 4 studies,

[1] http://www.ieeexplore.ieee.org/Xplore
[2] http://www.scopus.com/
[3] https://dl.acm.org/
[4] https://www.webofscience.com/

respectively. After applying the exclusion criteria and analyzing the titles, abstracts, and, when necessary, the full text of the studies, from the 608 initially identified studies, of which 34 were duplicates, only 55 were selected for the full-text reading phase. This stage was conducted by two independent reviewers in order to minimize potential biases in the analyses. In cases of disagreement, a third reviewer was consulted to resolve conflicts. After completing the full-text reading and applying the inclusion criteria, 27 studies were selected for the data extraction and results synthesis phase, with 6 studies from the *IEEE Xplore* digital library, 19 from *Scopus*, and 2 from *Web of Science*.

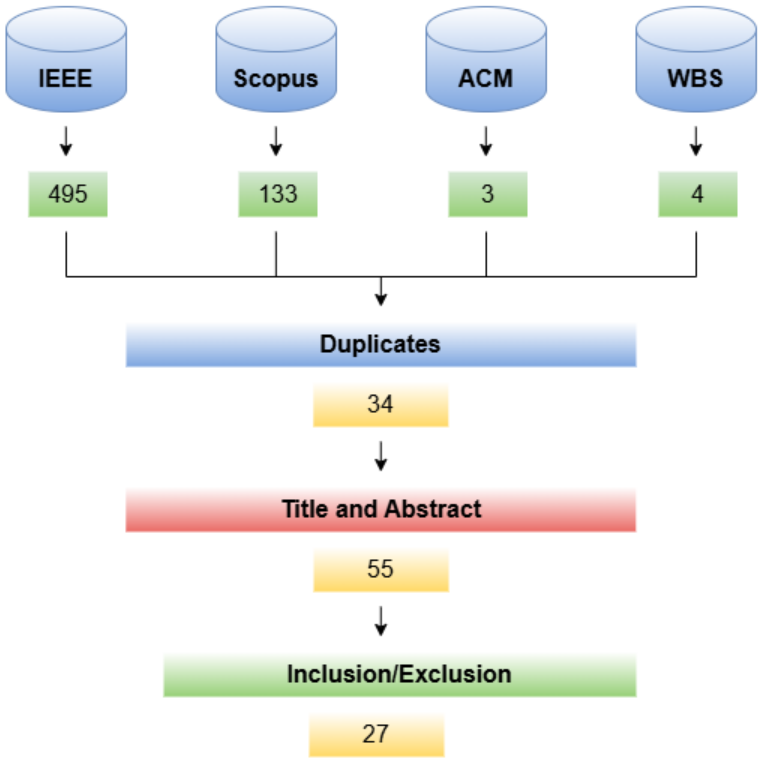

Figure 1. Methodological process of the SLR

2.4. Inclusion and Exclusion Criteria

To address the research questions, criteria were defined to determine the inclusion or exclusion of studies.

Inclusion Criteria: 1) Studies fully published between 2019 and 2025; 2) Studies published in peer-reviewed conferences, journals, and workshops; 3) Studies reporting the use of classifier ensembles for predicting software team effort estimation; 4) Studies written in English.

Exclusion Criteria: 1) Duplicate studies; 2) Studies not available for online access; 3) Articles published as short papers and/or secondary studies.

2.5. Qualitative Analysis

The quality assessment of the studies aimed to determine the relevance, rigor, and credibility of the 27 studies selected for results synthesis. Each study was read and analyzed

in depth and assessed against eight specific questions evaluating different methodological and result-related aspects. Each study's score was assigned based on its compliance with these questions, as detailed below: 1) Does the study address the use of classifier ensembles for predicting software team effort estimation? 2) Are the study objectives clearly defined? 3) Is the study context adequately described? 4) Is the research design appropriate to achieve the stated objectives? 5) Are the research results rigorously validated? 6) Does the study significantly contribute to software effort estimation prediction? 7) Are the variables impacting software effort estimation prediction clearly identified? 8) Does the study include statistical tests?

The studies were then classified according to the scores obtained for the questions above, using a binary evaluation scale: positive (1) and negative (0). Each row represents a primary study, and columns 'Q1' to 'Q8' correspond to the assessment questions. Only studies that achieved a score equal to or greater than 4 were selected for the data extraction phase. Table 1 presents the results of the primary studies' quality assessment.

Table 1. Quality analysis of the primary studies

Study	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Total
[Rai et al. 2021]	1	1	1	1	1	1	1	1	8
[Goyal 2022b]	1	1	1	1	1	1	0	1	7
[Cabral and Oliveira 2021]	1	1	1	1	1	1	0	1	7
[Abnane et al. 2023]	1	1	1	1	1	1	0	1	7
[Ali et al. 2023]	1	1	1	1	1	1	1	0	7
[Hosni et al. 2021]	1	1	1	1	1	1	0	1	7
[Shukla and Kumar 2023a]	1	1	1	1	1	1	1	0	7
[Rhmann et al. 2022]	1	1	1	1	1	1	1	0	7
[Sakhrawi et al. 2022a]	1	1	1	1	1	1	1	0	7
[Marapelli et al. 2021]	1	1	1	1	1	1	1	0	7
[Shukla and Kumar 2023b]	1	1	1	1	1	1	0	1	7
[Abnane et al. 2021]	1	1	1	1	1	1	0	1	7
[Shahpar et al. 2022]	1	1	1	1	1	1	0	1	7
[Charmanas et al. 2020]	1	1	1	1	1	1	0	1	7
[Hosni et al. 2019]	1	1	1	1	1	1	0	1	7
[Hosni 2024]	1	1	1	1	1	1	0	1	7
[Marco et al. 2022]	1	1	1	1	1	1	0	0	6
[Beesetti et al. 2023]	1	1	1	1	1	1	0	0	6
[Srivastava et al. 2021]	1	1	1	1	1	1	0	0	6
[Thakur and Dutta 2025]	1	1	1	1	0	1	1	0	6
[Javdani Gandomani et al. 2024]	1	1	1	1	0	1	1	0	6
[Raghu Raman et al. 2024]	1	1	1	1	0	1	1	0	6
[Labidi and Sakhrawi 2023]	1	1	1	1	0	1	0	0	5
[Mahmood et al. 2020b]	1	1	1	1	1	0	0	0	5
[Nosrati and Rahmani 2023]	1	1	1	1	1	0	0	0	5
[Sakhrawi et al. 2022b]	1	1	1	1	0	0	0	0	4
[BN and Suresh 2023]	1	1	1	1	0	0	0	0	4
Total	**27**	**27**	**27**	**27**	**21**	**23**	**9**	**11**	

It can be observed that all 27 studies address questions Q1 to Q4. However, six studies do not report how the research results were validated (Q5); four studies do not present significant contributions to software effort estimation prediction, failing to demonstrate relevant advances compared to previously reported approaches in the literature (Q6); nine studies do not identify the variables with the greatest impact on software effort

estimation prediction (Q7); and only eleven studies report statistical tests (Q8). Among all studies, [Rai et al. 2021] stands out with the highest score, while [Sakhrawi et al. 2022b] and [BN and Suresh 2023] obtained the lowest scores.

3. Results and Discussion

This section presents and discusses the results obtained from the analysis of the selected studies, organized according to the defined research questions.

3.1. Which datasets are most commonly used in the studies? (RQ1)

Datasets play a fundamental role in the evaluation of ML models, as inconsistent or unreliable datasets hinder fair and consistent analyses and limit the ability to draw conclusions from the obtained results. Table 2 presents the datasets used in the analyzed studies. Among them, the Desharnais dataset stands out as the most frequently cited, appearing in 48.1% of the studies, which indicates its widespread use and recognition in the literature. This dataset focuses on analyzing the relationship between software project size and the effort required for project completion, being considered relevant for providing detailed information on how project size influences the productivity of software development teams. Therefore, the Desharnais dataset constitutes a significant resource for investigations aimed at efficiency and productivity in software development.

The ISBSG dataset appears as the second most frequently cited dataset, being present in 40.7% of the studies. Its popularity can be attributed to the diversity of projects included in the dataset, encompassing different domains, sizes, and organizational contexts. Moreover, it is a continuously updated dataset, which contributes to its relevance in the literature. However, its use requires caution, as project characteristics may vary significantly across organizations.

The China and COCOMO81 datasets were cited in 37.0% of the analyzed studies, indicating their well-established importance in the field of software effort estimation. Subsequently, the Miyazaki94, Albrecht, and Kemerer datasets appear with moderate but consistent usage in 25.9% of the studies. The Maxwell dataset was cited in 22.2% of the studies. In contrast, the Kitchenham, COCOMONASA V1, COCOMONASA V2, and PROMISE datasets were cited in 7.4% of the studies, while the JM1 dataset was cited in only 3.7% of the studies. This suggests that these datasets have been less explored or may present limited relevance to the research domain.

Although these datasets are widely accepted and used in the software team effort estimation literature, they present significant limitations when analyzed in the context of contemporary software development. Many of these datasets were collected in environments that no longer reflect modern development practices, which may impact the validity of the obtained results. The discussion of Table 2 shows that some datasets are extensively used and highly cited, indicating their relevance, while others exhibit fewer citations. Considering this information when selecting datasets for future research is essential. In this regard, the use of more up-to-date datasets that are representative of real-world software projects is recommended, as well as the creation of new datasets aligned with contemporary development demands.

Table 2. Datasets used in each study

Dataset	Study	Total
DESHARNAIS	[Marco et al. 2022], [Goyal 2022b], [Cabral and Oliveira 2021], [Abnane et al. 2023], [Hosni et al. 2021], [Beesetti et al. 2023], [Rhmann et al. 2022], [Abnane et al. 2021], [Shahpar et al. 2022], [Hosni et al. 2019], [BN and Suresh 2023], [Hosni 2024],[Thakur and Dutta 2025]	13
ISBSG	[Marco et al. 2022], [Ali et al. 2023], [Hosni et al. 2021], [Labidi and Sakhrawi 2023], [Rai et al. 2021], [Shukla and Kumar 2023a], [Sakhrawi et al. 2022a], [Charmanas et al. 2020], [Hosni et al. 2019], [Hosni 2024], [Raghu Raman et al. 2024]	11
CHINA	[Marco et al. 2022], [Goyal 2022b], [Cabral and Oliveira 2021], [Abnane et al. 2023], [Hosni et al. 2021], [Beesetti et al. 2023], [Abnane et al. 2021], [Hosni et al. 2019], [Javdani Gandomani et al. 2024],[Thakur and Dutta 2025]	10
COCOMO81	[Marco et al. 2022], [Goyal 2022b], [Cabral and Oliveira 2021], [Abnane et al. 2023], [Hosni et al. 2021], [Beesetti et al. 2023], [Abnane et al. 2021], [Hosni 2024], [Javdani Gandomani et al. 2024],[Thakur and Dutta 2025]	10
MIYAZAKI94	[Goyal 2022b], [Cabral and Oliveira 2021], [Abnane et al. 2023], [Hosni et al. 2021], [Abnane et al. 2021], [Hosni et al. 2019], [Hosni 2024]	7
ALBRECHT	[Marco et al. 2022], [Hosni et al. 2021], [Beesetti et al. 2023], [Shahpar et al. 2022], [Hosni et al. 2019],[Hosni 2024],[Javdani Gandomani et al. 2024]	7
KEMERER	[Marco et al. 2022], [Abnane et al. 2023], [Beesetti et al. 2023], [Abnane et al. 2021], [Shahpar et al. 2022], [Hosni 2024],[Javdani Gandomani et al. 2024]	7
MAXWELL	[Marco et al. 2022], [Goyal 2022b], [Cabral and Oliveira 2021], [Beesetti et al. 2023], [Shahpar et al. 2022], [Javdani Gandomani et al. 2024]	6
PROMISE	[Shukla and Kumar 2023a], [Charmanas et al. 2020]	2
KITCHENHAM	[Marco et al. 2022], [Cabral and Oliveira 2021]	2
COCOMONASA_V2	[Cabral and Oliveira 2021], [Rhmann et al. 2022]	2
COCOMONASA_V1	[Cabral and Oliveira 2021], [Rhmann et al. 2022]	2
JM1	[Thakur and Dutta 2025]	1

3.2. Which evaluation metrics and ML techniques are most commonly applied in the studies? (RQ2)

Table 3 details the main evaluation metrics used in software team effort estimation studies, highlighting their relevance in the scientific literature by enabling a comprehensive analysis of the accuracy and reliability of predictive models. First, MAE, reported in 55.6% of the analyzed studies, stands out due to its simplicity and ease of interpretation, facilitating comparisons across models by computing the mean absolute error in the problem's original units. In contrast, PRED, used in 33.3% of the studies, measures the percentage of predictions that fall within an acceptable error threshold, such as PRED(25), and is therefore crucial for assessing model reliability. Additionally, SA, also reported in 29.6% of the studies, measures the average accuracy of predictions, consistently capturing model performance across different samples.

Complementarily, RMSE, likewise employed in 29.6% of the studies, penalizes large errors more heavily by squaring deviations, making it essential for identifying significantly inaccurate estimates. Subsequently, MMRE, present in 25.9% of the studies, remains a classical metric in the field, despite criticism regarding its sensitivity to extreme values, as it measures the mean relative error and provides an overall view of prediction accuracy. MSE, used in 18.5% of the studies, further emphasizes the penalization of large errors without applying the square root, being useful for detecting outliers and severe deviations. In turn, R^2, adopted in 22.2% of the studies, measures how much of the data variability is explained by the model, offering a clear view of the overall goodness of fit. Finally, MdMRE, which is less sensitive to outliers, provides greater reliability in scenarios where extreme values may influence the results.

Other metrics appear sporadically in the analyzed studies. MBRE and Effect Size,

each used in 11.1% of the studies, provide complementary analyses of relative error. Metrics such as MRE, BMMRE, BMMBRE, MIBRE, and LSD, present in only 3.7% of the studies, tend to be less frequently adopted, as they are typically proposed to address specific objectives or to overcome limitations observed in more traditional metrics.

Thus, the diversified use of these metrics demonstrates the need for multifaceted approaches to evaluate effort estimation models. For future work, we recommend exploring hybrid metrics that combine traditional approaches with ML-based techniques in order to capture nonlinear patterns present in effort estimations. In addition, sensitivity analyses of evaluation metrics with respect to noisy or imbalanced data are suggested, as well as the application of multi-criteria optimization techniques to select balanced models. Such future investigations may provide significant advances by promoting effort estimation methods that are more accurate, robust, and applicable in practice.

Table 3. Evaluation metrics used in each study

Evaluation metrics	Study	Total
MAE	[Marco et al. 2022], [Goyal 2022b], [Ali et al. 2023], [Rai et al. 2021], [Mahmood et al. 2020b], [Shukla and Kumar 2023a], [Beesetti et al. 2023], [Rhmann et al. 2022], [Sakhrawi et al. 2022a], [Sakhrawi et al. 2022b], [Shukla and Kumar 2023b], [Hosni et al. 2019],[Hosni 2024],[Thakur and Dutta 2025], [Raghu Raman et al. 2024]	15
PRED	[Goyal 2022b], [Abnane et al. 2023], [Ali et al. 2023], [Mahmood et al. 2020b], [Shukla and Kumar 2023b], [Shahpar et al. 2022], [BN and Suresh 2023],[Javdani Gandomani et al. 2024], [Hosni 2024]	9
SA	[Abnane et al. 2023], [Hosni et al. 2021], [Shukla and Kumar 2023a], [Sakhrawi et al. 2022a], [Shukla and Kumar 2023b], [Abnane et al. 2021], [Hosni et al. 2019],[Hosni 2024]	8
RMSE	[Marco et al. 2022], [Labidi and Sakhrawi 2023], [Rai et al. 2021], [Beesetti et al. 2023], [Rhmann et al. 2022], [Sakhrawi et al. 2022a], [Sakhrawi et al. 2022b],[Raghu Raman et al. 2024]	8
MMRE	[Goyal 2022b], [Ali et al. 2023], [Mahmood et al. 2020b], [Shahpar et al. 2022], [Srivastava et al. 2021], [BN and Suresh 2023], [Javdani Gandomani et al. 2024]	7
MSE	[Rai et al. 2021], [Shukla and Kumar 2023a], [Marapelli et al. 2021], [Shukla and Kumar 2023b], [Hosni et al. 2019]	5
R^2	[Marco et al. 2022], [Labidi and Sakhrawi 2023], [Sakhrawi et al. 2022a], [Marapelli et al. 2021],[Thakur and Dutta 2025],[Raghu Raman et al. 2024]	6
MDMRE	[Ali et al. 2023], [Mahmood et al. 2020b], [Shahpar et al. 2022], [Javdani Gandomani et al. 2024]	4
MBRE	[Abnane et al. 2023], [Shukla and Kumar 2023b],[Hosni 2024]	3
EFFECT SIZE	[Abnane et al. 2021], [Hosni et al. 2019],[Hosni 2024]	3
MRE	[Javdani Gandomani et al. 2024]	1
BIMMRE	[Javdani Gandomani et al. 2024]	1
BMMRE	[Javdani Gandomani et al. 2024]	1
MIBRE	[Hosni 2024]	1
LSD	[Hosni 2024]	1

Table 4 provides an overview of the classifier ensembles employed across different studies, categorized into homogeneous and heterogeneous ensembles. This distinction is important because it directly influences model behavior in terms of accuracy, robustness, and flexibility in the software team effort estimation process.

Homogeneous ensembles are composed of classifiers of the same type or with similar characteristics, which promotes greater specialization in a specific learning approach. In this context, ensembles based on Random Forest [Marco et al. 2022] and Support Vector Regression [Sakhrawi et al. 2022b] stand out, as they are known for consistent performance in scenarios where the data exhibit regularity and low variability. In addition,

Table 4. Homogeneous and heterogeneous classifier ensembles based on ML algorithms used in the studies

Classifier Ensemble	Ensemble Type	Study
Homogeneous Ensembles		
Random Forest	Specialized homogeneous	[Marco et al. 2022]
Support Vector Regression	Specialized homogeneous	[Sakhrawi et al. 2022b]
Multilayer Perceptron forming a neural-network-based ensemble	Specialized homogeneous	[Rai et al. 2021]
LinearSVR forming a specific linear-regression ensemble	Specialized homogeneous	[Sakhrawi et al. 2022a]
Dynamically adapted evolutionary ensemble	Evolutionary homogeneous	[Shahpar et al. 2022]
Data Envelopment Analysis forming a specific ensemble	Statistical homogeneous	[Charmanas et al. 2020]
Neural Network forming a deep learning–based ensemble	Deep homogeneous	[BN and Suresh 2023]
Support Vector Regression with different kernels (Linear, Polynomial, RBF, and Sigmoid)	Specialized homogeneous	[Hosni 2024]
Heterogeneous Ensembles		
AdaBoost, Random Forest, and Bayesian Optimization forming an adaptive ensemble	Adaptive heterogeneous	[Marco et al. 2022]
Artificial Neural Networks and Support Vector Regression combined	Classical heterogeneous	[Goyal 2022b]
AdaBoost, Bagging, Best First Tree, Decision Stump, IBk, J48, JRip, LMT, Logistic Regression, Multilayer Perceptron, Random Forest, SVM, and OneR forming a large ensemble	Large multi-algorithm heterogeneous	[Cabral and Oliveira 2021]
K-Nearest Neighbor, Support Vector Regression, and Decision Tree	Basic heterogeneous	[Abnane et al. 2023]
Support Vector Machine, Linear Regression, K-Nearest Neighbor, XGBoost, and Artificial Neural Networks	Multi-algorithm heterogeneous	[Ali et al. 2023]
Use Case Points, Expert Judgment, and Case-Based Reasoning	Analogy-based heterogeneous	[Mahmood et al. 2020b]
K-Nearest Neighbors, Support Vector Regression, Multilayer Perceptron, and Decision Tree	Balanced heterogeneous	[Hosni et al. 2021]
Multilayer Perceptron, Support Vector Regression, and Decision Tree	Classical and deep heterogeneous	[Labidi and Sakhrawi 2023]
AdaBoost, Gradient Boosting, Extreme Gradient Boosting, Linear Regression, Random Forest, and Voting	Advanced heterogeneous	[Beesetti et al. 2023]
Random Forest, Multilayer Perceptron, SVM, Linear Regression, Bagging, Stacking, and Voting	Robust heterogeneous	[Rhmann et al. 2022]
LinearSVR, Gradient Boosting Regressor, and Random Forest	Regressive heterogeneous	[Sakhrawi et al. 2022a]
Voting Regressor, Random Forest Regressor, and Gradient Boosting Regressor	Voting-based heterogeneous	[Marapelli et al. 2021]
Linear Regression, K-Nearest Neighbor, Decision Tree, Support Vector Machine, and Multilayer Perceptron	Classical multi-algorithm heterogeneous	[Shukla and Kumar 2023b]
K-Nearest Neighbors, Expectation Maximization, Support Vector Regression, and Decision Tree	Mixed heterogeneous	[Abnane et al. 2021]
Ensemble model combining Linear Regression, Random Forest, and Support Vector Machine	Multi-algorithm heterogeneous	[Nosrati and Rahmani 2023]
Fuzzy analogy forming an approximate ensemble	Approximation-based heterogeneous	[Hosni et al. 2019]
Distributed homogeneous feature selection framework forming an optimized ensemble	Optimized heterogeneous	[Rhmann et al. 2022]
Random Forest, Decision Tree, Support Vector Regression, and Linear Regression	Weighted adaptive heterogeneous	[Javdani Gandomani et al. 2024]
Use Case Point, Artificial Neural Network, and Expert Judgment	Hybrid heterogeneous	[Raghu Raman et al. 2024]
Linear Regression, Support Vector Regression, Random Forest, and Gradient Boosting	Stacking-based heterogeneous	[Thakur and Dutta 2025]

ensembles structured with Multilayer Perceptron models [Rai et al. 2021] or deep neural networks [BN and Suresh 2023] are widely adopted when the objective is to capture complex and nonlinear patterns. However, the homogeneity of these ensembles may limit their adaptability in dynamic environments, since they do not exploit diverse learning perspectives.

In contrast, heterogeneous ensembles combine different algorithms, leveraging

their complementary strengths to provide greater flexibility and adaptability. A relevant example is the ensemble composed of AdaBoost, Random Forest, and Bayesian Optimization [Marco et al. 2022], whose adaptive approach allows dynamic adjustments in response to data changes. Likewise, broader heterogeneous ensembles, such as those integrating AdaBoost, Bagging, SVM, Multilayer Perceptron, and Random Forest [Cabral and Oliveira 2021], employ diverse methods to ensure more robust predictions.

A significant advantage of these ensembles is their ability to reduce errors in situations where individual algorithms may exhibit limitations. For instance, the ensemble combining Linear Regression, K-Nearest Neighbors, Decision Tree, SVM, and Multilayer Perceptron [Shukla and Kumar 2023b] adopts a classical multi-algorithm approach, in which each model contributes to different aspects of the estimation process. Complementarily, ensembles based on weighted voting techniques, such as those including Voting Regressor, Random Forest Regressor, and Gradient Boosting Regressor [Marapelli et al. 2021], strategically integrate predictions to achieve higher accuracy.

Despite these advantages, heterogeneous ensembles may incur higher computational costs and longer processing times. An illustrative example is the ensemble combining AdaBoost, Gradient Boosting, Extreme Gradient Boosting, Linear Regression, and Voting [Beesetti et al. 2023], which requires careful optimization to ensure an appropriate balance between diversity and performance. In practical applications, heterogeneous ensembles are often preferred in scenarios characterized by high data variability, whereas homogeneous ensembles tend to be more effective in specific and controlled domains. A notable trend in the literature is the adoption of multi-algorithm ensembles, such as those proposed by [Cabral and Oliveira 2021, Rhmann et al. 2022, Shukla and Kumar 2023b], which combine classical and modern techniques to achieve a balance between specialization and flexibility.

The identified research gaps reveal critical challenges for advancing software effort estimation. Uncertainties remain regarding the optimal choice between heterogeneous and homogeneous ensembles, reflecting the difficulty of optimizing their application across different contexts [Cabral and Oliveira 2021, Abnane et al. 2021]. The impact of feature selection on model diversity still requires deeper investigation [Hosni et al. 2021], as does the lack of dynamic strategies to adapt model selection in response to data evolution [Cabral and Oliveira 2021]. Moreover, the limited number of practical validations in real-world environments restricts the generalizability of results obtained in controlled experiments [Labidi and Sakhrawi 2023]. Other critical areas include hyperparameter tuning [Labidi and Sakhrawi 2023], the integration of incomplete data imputation techniques [Abnane et al. 2021], and the exploration of hybrid and multi-level models [Goyal 2022b]. To overcome these challenges, future research should prioritize the application of evolutionary techniques for adaptive tuning [Shahpar et al. 2022], multi-objective optimization strategies to balance accuracy and complexity [Ali et al. 2023], and the standardization of evaluation metrics [Abnane et al. 2021], thereby ensuring more robust and transferable results for real-world software projects.

4. Conclusion

The conclusion of this study highlights the importance of classifier ensembles in improving the accuracy of software team effort estimation. The results demonstrated that heterogeneous ensembles provide greater robustness and adaptability in environments characterized by high variability, whereas homogeneous ensembles stand out in more specific and controlled domains. As recommendations for future work, the exploration of hybrid metrics that combine traditional and modern approaches is suggested in order to better capture nonlinear patterns. In addition, sensitivity analyses of evaluation metrics with respect to noisy and imbalanced data are proposed, along with the development of dynamically optimized ensemble selection methods tailored to different contexts.

References

Abnane, I., Idri, A., Chlioui, I., and Abran, A. (2023). Evaluating ensemble imputation in software effort estimation. *Empirical Software Engineering*, 28(2):56.

Abnane, I., Idri, A., Hosni, M., and Abran, A. (2021). Heterogeneous ensemble imputation for software development effort estimation. In *Proceedings of the 17th International Conference on Predictive Models and Data Analytics in Software Engineering*, pages 1–10.

Ali, S. S., Ren, J., Zhang, K., Wu, J., and Liu, C. (2023). Heterogeneous ensemble model to optimize software effort estimation accuracy. *IEEE Access*, 11:27759–27792.

Araújo, D. d. C. (2019). Avaliação de comitês com classificadores tradicionais e profundos para análise de sentimentos. Master's thesis, Universidade Federal de Pernambuco.

Beesetti, K. K., Bilgaiyan, S., and Mishra, B. S. P. (2023). Software effort estimation through ensembling of base models in machine learning using a voting estimator. *International Journal of Advanced Computer Science and Applications*, 14(2).

Benaroch, M. and Lyytinen, K. (2023). How much does software complexity matter for maintenance productivity? the link between team instability and diversity. *IEEE Transactions on Software Engineering*, 49(4):2459–2475.

BN, R. K. and Suresh, Y. (2023). Software effort estimation using ann (back propagation). In *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1–2. IEEE.

Boehm, B., Abts, C., and Chulani, S. (2000). Software development cost estimation approaches—a survey. *Annals of software engineering*, 10(1):177–205.

Cabral, J. T. H. d. A. and Oliveira, A. L. (2021). Ensemble effort estimation using dynamic selection. *Journal of Systems and Software*, 175:110904.

Charmanas, K., Mittas, N., and Angelis, L. (2020). Ensemble software development effort estimation using data envelopment analysis. In *Proceedings of the 24th Pan-Hellenic Conference on Informatics*, pages 202–207.

Goyal, S. (2022a). Effective software effort estimation using heterogenous stacked ensemble. In *2022 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, volume 1, pages 584–588. IEEE.

Goyal, S. (2022b). Effective software effort estimation using heterogenous stacked ensemble. In *2022 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, volume 1, pages 584–588. IEEE.

Hosni, M. (2024). Comparative analysis of single and ensemble support vector regression methods for software development effort estimation. In *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR*, pages 509–516. INSTICC, SciTePress.

Hosni, M., Idri, A., and Abran, A. (2019). Evaluating filter fuzzy analogy homogenous ensembles for software development effort estimation. *Journal of Software: Evolution and Process*, 31(2):e2117.

Hosni, M., Idri, A., and Abran, A. (2021). On the value of filter feature selection techniques in homogeneous ensembles effort estimation. *Journal of Software: Evolution and Process*, 33(6):e2343.

Javdani Gandomani, T., Dashti, M., Zulzalil, H., and Sultan, A. B. M. (2024). Enhancing software effort estimation in the analogy-based approach through the combination of regression methods. *IEEE Access*, 12:152122–152137.

Kitchenham, B. A., Budgen, D., and Brereton, P. (2015). *Evidence-based software engineering and systematic reviews*, volume 4. CRC press.

Labidi, T. and Sakhrawi, Z. (2023). On the value of parameter tuning in stacking ensemble model for software regression test effort estimation. *The Journal of Supercomputing*, 79(15):17123–17145.

Mahmood, Y., Kama, N., Azmi, A., and Ali, M. (2020a). Improving estimation accuracy prediction of software development effort: A proposed ensemble model. In *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pages 1–6.

Mahmood, Y., Kama, N., Azmi, A., and Ali, M. (2020b). Improving estimation accuracy prediction of software development effort: a proposed ensemble model. In *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pages 1–6. IEEE.

Marapelli, B., Carie, A., and Islam, S. M. (2021). Software effort estimation with use case points using ensemble machine learning models. In *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pages 1–6. IEEE.

Marco, R., Ahmad, S. S. S., and Ahmad, S. (2022). Bayesian hyperparameter optimization and ensemble learning for machine learning models on software effort estimation. *International Journal of Advanced Computer Science and Applications*, 13(3).

Nassif, A. B., Ho, D., and Capretz, L. F. (2013). Towards an early software estimation using log-linear regression and a multilayer perceptron model. *Journal of Systems and Software*, 86(1):144–160.

Nosrati, V. and Rahmani, M. (2023). Hmde-fs: A homogeneous distributed ensemble feature selection framework based on resampling with/without replacement. *Concurrency and Computation: Practice and Experience*, 35(7):e7613.

Raghu Raman, D., Santhagarooban, D., and Sathyanarayanan, N. (2024). Optimizing software effort estimation accuracy with a machine learning model. In *2024 5th International Conference for Emerging Technology (INCET)*, pages 1–8.

Rai, P., Kumar, S., and Verma, D. K. (2021). Prediction of software effort in the early stage of software development: a hybrid model. *IEEE Canadian Journal of Electrical and Computer Engineering*, 44(3):376–383.

Rhmann, W., Pandey, B., and Ansari, G. A. (2022). Software effort estimation using ensemble of hybrid search-based algorithms based on metaheuristic algorithms. *Innovations in Systems and Software Engineering*, 18(2):309–319.

Sakhrawi, Z., Sellami, A., and Bouassida, N. (2022a). Software enhancement effort estimation using correlation-based feature selection and stacking ensemble method. *Cluster Computing*, 25(4):2779–2792.

Sakhrawi, Z., Sellami, A., and Bouassida, N. (2022b). Support vector regression for enhancement effort prediction of scrum projects from cosmic functional size. *Innovations in Systems and Software Engineering*, 18(1):137–153.

Senevirathne, D. S. and Wijayasiriwardhane, T. K. (2020). Extending use-case point-based software effort estimation for open source freelance software development. In *2020 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, pages 188–194.

Seni, G. and Elder, J. (2010). *Ensemble methods in data mining: improving accuracy through combining predictions*. Morgan & Claypool Publishers.

Shahpar, Z., Bardsiri, V. K., and Bardsiri, A. K. (2022). An evolutionary ensemble analogy-based software effort estimation. *Software: Practice and Experience*, 52(4):929–946.

Shukla, S. and Kumar, S. (2023a). Self-adaptive ensemble-based approach for software effort estimation. In *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 581–592. IEEE.

Shukla, S. and Kumar, S. (2023b). Towards ensemble-based use case point prediction. *Software Quality Journal*, 31(3):843–864.

Srivastava, D. K., Sharma, A. K., and Choudhary, D. (2021). Software development effort estimation using machine learning techniques: Multi-linear regression versus random forest. In *2021 International Conference on Computing, Communication and Green Engineering (CCGE)*, pages 1–5. IEEE.

Thakur, V. and Dutta, K. (2025). Machine learning based effort estimation models for software development projects related datasets with diverse features. In *2025 2nd International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)*, pages 807–813.

Venson, E. (2020). The effects of required security on software development effort. In *2020 IEEE/ACM 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 166–169.

Wen, J., Li, S., Lin, Z., Hu, Y., and Huang, C. (2012). Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology*, 54(1):41–59.