Hybrid Model for Improving the Accuracy of Software Development Team Effort Estimation with Genetic Algorithm-Optimized Regression Ensembles

Wilamis Kleiton Nunes da Silva [1]

[1]Cesar School, Recife, Pernambuco, Brazil
Caixa Postal 50030-220 –Avenida Cais do Apolo –77 – Recife –PE – Brazil

[2]Professional Doctorate in Software Engineering - DPES

{wkns}@cesar.school

Abstract. *This thesis plan aims to propose a Hybrid Model for Software Development Team Effort estimation by integrating Regression Ensembles and GA. The methodology will consist of an experimental evaluation of the proposed machine learning model using multiple datasets from the PROMISE repository, employing the evaluation metrics MAE, RMSE, MdMRE, PRED(25), and R^2 score to analyze predictive performance.*

1. Introduction and Main Challenges

Problem: Software development team effort estimation refers to predicting the amount of work required to successfully complete a software development project. The accuracy of these estimates is crucial to ensure that projects are delivered on time, within budget, and with the expected quality. However, this task remains one of the most challenging activities in the software development industry due to the inherent complexity and unpredictability of software projects [Goyal 2022].

According to the CHAOS Report 2020, published by the Standish Group, software development projects continue to experience high failure rates. In a study involving approximately 50,000 projects worldwide, the report revealed that only 31% of the projects were successful, that is, delivered on time, within budget, and with all originally specified functionalities. In contrast, 50% of the projects faced significant challenges, resulting in schedule delays, budget overruns, or partial delivery of functionalities, while 19% were canceled or abandoned before completion [STANDISH GROUP 2020]. These figures indicate the persistence of structural problems in software project management and execution. Among the main factors contributing to software project failure, the CHAOS Report highlights inaccuracies in effort estimation and misunderstandings of requirements. Inadequate estimates frequently lead to substantial deviations in schedules and budgets, thereby compromising planning feasibility. In this context, the application of machine learning (ML) methods has shown promise in improving estimation accuracy by leveraging historical project data and adapting predictive models to the specific characteristics of each new software development project [Wen et al. 2012]. Complementarily, a recent survey conducted by the Boston Consulting Group in 2024 revealed that approximately 30% of software development projects experience delays or budget overruns, and around 20% of leaders report that half of their projects fail to deliver satisfactory outcomes, underscoring the persistence of challenges in software development team effort estimation [Boston Consulting Group 2024].

Therefore, software development team effort estimation remains a challenging task and is of critical importance to the success of software projects. The pursuit of more accurate and adaptable methods, such as ML algorithms, emerges as a potential solution to overcome the limitations of traditional approaches and to ensure that software projects are completed more efficiently and within the established parameters.

Motivation: Producing consistent and accurate estimates remains a significant challenge for software development project managers [STANDISH GROUP 2020]. The selection of techniques for predicting these estimates involves several factors that must be considered and is guided by organizational needs and capabilities. In general, the primary objective is to maximize estimation accuracy, in addition to other relevant factors [de Barcelos Tronto et al. 2006]. However, due to the different characteristics of the datasets employed, there is still no consensus on which techniques should be preferentially adopted for predicting software development team effort [Hameed et al. 2023]. Therefore, it is essential to investigate new techniques with higher accuracy to ensure the success of software development projects.

In recent years, a number of studies have been conducted with the aim of developing, evaluating, and recommending different ML techniques for software development team effort prediction, as reported in [Mohsin 2021, Reddy and Thinakaran 2022, Abnane et al. 2023, Beesetti et al. 2023, Khan et al. 2021, Rajput et al. 2025]. Nevertheless, despite the progress achieved, several gaps and challenges remain to be addressed. The wide variety of available techniques and the diversity of datasets require careful analysis to determine the most suitable approach for each specific context. Moreover, the continuous evolution of the software project development landscape and the rapid expansion of data-driven technologies highlight the ongoing importance of research and innovation in this area.

Contribution: The main expected contributions of this thesis plan focus on the proposal, implementation, and evaluation of a hybrid model for software development team effort estimation based on regression ensembles optimized using GA, aiming to overcome potential limitations identified in 15 traditional approaches. It is expected that the proposed model will yield statistically significant gains in terms of improved effort estimation accuracy and enhanced generalization capability across multiple datasets of software development projects.

Among the main expected outcomes, the following stand out:

- The development of a hybrid model that integrates regression ensemble techniques with GA for the dynamic selection and weighting of the most suitable regression algorithms for each software development project context;
- The ability of the model to provide automated recommendations regarding the most promising regression algorithms based on the characteristics of the input data, contributing to the construction of customized predictive solutions;
- The incorporation of population evolution strategies, including selection, crossover, and mutation, aiming to optimize not only individual hyperparameters but also the architecture and composition of regression ensembles;
- The definition of a reproducible experimental protocol grounded in controlled experimentation and robust statistical analyses in order to ensure the internal and external validity of the results;

2. Research Questions and Objectives

Research Question: The present thesis plan proposal seeks to address the following research question: how can software development team effort estimation be improved through the use of regression ensembles and Genetic Algorithms (GA)? This study is grounded on the premise that the combination of these techniques has the potential to outperform individual predictive models that are well established and widely used in the literature. The originality of the proposal lies in the integrated application of the aforementioned techniques for software development team effort estimation, an approach that has been relatively underexplored in the literature and shows promising potential to improve model accuracy across different software development project contexts.

According to [Easterbrook et al. 2008], this research question is classified as a comparative causal question. This type of inquiry aims to understand causal relationships between variables through the comparison of two or more groups or conditions. Such questions play a fundamental role in investigating the impact of specific interventions, practices, or independent variables on observed outcomes within controlled experimental research methods. The proposed thesis plan adopt

General and Specific Objectives: The overall objective of this thesis plan is to develop and empirically validate a hybrid machine learning ML model based on the integration of regression ensembles and GA, aiming to improve the prediction accuracy and adaptability of software development team effort estimation across different software development projects.

The specific objectives are as follows: to identify approaches for constructing regression ensembles and strategies for applying GA to regression-based predictive models; to develop a hybrid model combining regression ensembles and GA with the goal of improving the accuracy of software development team effort estimation, and to evaluate its performance using accuracy metrics and statistical tests applied to multiple datasets of software development projects; and to comparatively assess the proposed methods against traditional machine learning regression techniques for software development team effort estimation, identifying the strengths and limitations of the proposed model.

3. Methodology

The methodology was structured to ensure the validity and reproducibility of the experiments, as well as to enable a comprehensive analysis of the ML techniques. The main methodological steps include the selection and preparation of the *datasets*, the definition of evaluation metrics, and the configuration of the experiments. The methodological process adopted for the proposed model is divided into five main stages, as illustrated in Figure 1.

The initial stage of the proposed model begins with data import, which will be obtained from different datasets available in the PROMISE repository. These data will undergo a rigorous preprocessing phase, which is essential to ensure data integrity, standardization, and quality. This procedure will include noise removal, handling of missing values, and data preparation for the application of ML algorithms. In the second stage, the data will be partitioned into training and testing sets using cross-validation techniques, enabling a more robust evaluation of the predictive capability of the models and minimizing the risk of overfitting.

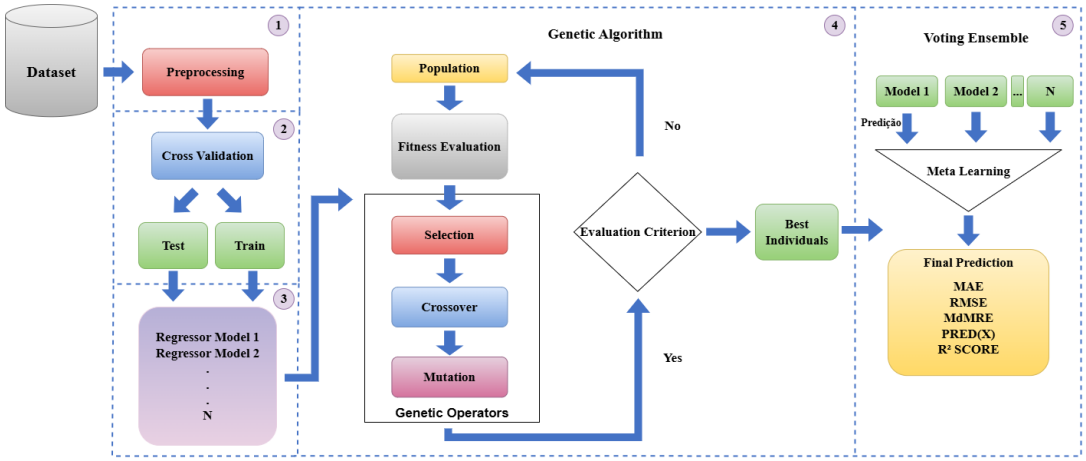

Figura 1. Proposed Model.

The third stage encompasses the training of 15 machine learning regression models, previously defined based on the literature. The use of multiple algorithms allows the exploration of different predictive characteristics, fostering a more comprehensive analysis and contributing to the construction of more robust and effective regression ensembles. In the fourth stage, a Genetic Algorithm (GA) will be incorporated as an evolutionary optimization mechanism to search for the best combinations of models and parameters. This stage will be conducted through selection, crossover, and mutation operations applied to an initial population of solutions. The objective is to maximize the fitness of individuals, which will be evaluated according to predefined performance criteria. Only the most promising individuals will be selected to compose the final regression ensemble.

Finally, the fifth stage involves the formation of the voting regression ensemble composed of the optimized models. Through a meta-learning approach, individual predictions will be intelligently combined in order to improve the final predictive performance. This aggregation will be performed using weighted averages.

3.1. Datasets

The present thesis plan aims to employ ten different datasets obtained from the PROMISE repository (`https://promise.site.uottawa.ca/SERepository/`), covering distinct application domains. The Albrecht, Cocomo81, Finnish, Kemerer, and Desharnais datasets measure effort in person-months, whereas China, Maxwell, and Telecom use person-hours as the measurement unit. These datasets encompass projects with varying sizes and levels of complexity, including attributes related to software size, productivity, team experience, project complexity, and environmental factors, enabling a comprehensive and robust evaluation of the proposed model. The selection of these datasets aims to ensure a broad, robust, and representative analysis of machine learning regression techniques applied to software development team effort estimation.

3.2. Evaluation Metrics

To evaluate the performance of predictive models applied to software development team effort estimation, several metrics can be employed. In this thesis plan, the following evaluation metrics are adopted: MAE (*Mean Absolute Error*), RMSE (*Root Mean Squared*

Error), MdMRE (*Median of the Magnitude of Relative Error*), PRED(X), and the R^2 *score*. The selection of these metrics is justified by their wide acceptance in the literature and by their complementary nature. MAE and RMSE provide a direct assessment of absolute error, with the former being less sensitive to outliers and the latter being particularly useful in contexts where larger errors are penalized more severely. The MdMRE metric provides a robust measure of relative error, while PRED(X) quantifies model reliability by indicating the proportion of estimates with an error below 25%. The R^2 *score* offers a statistical perspective on model goodness of fit by indicating the proportion of data variability explained by the predictions. Taken together, these metrics enable a comprehensive and balanced evaluation of predictive performance from both technical and statistical perspectives.

3.3. Statistical Tests

Statistical tests provide scientific support to research studies, conferring validity and credibility to their results. In statistics, a result is considered significant when the probability of its occurrence by chance is sufficiently low. Accordingly, the use of statistical tests aims to determine whether the differences observed among the evaluated systems are sufficiently relevant to indicate the superiority of one model over another. To this end, a set of statistical hypotheses is formulated to guide the analysis. The null hypothesis represents the initial assumption of no effect or difference and is maintained as valid until statistical evidence indicates otherwise. The alternative hypothesis, in turn, opposes the null hypothesis and assumes the existence of a statistically significant difference between the models. In the present study, in addition to these two hypotheses, a third hypothesis is also considered, which allows the assessment of the possibility of inferior performance of the proposed model. As these hypotheses are mutually exclusive, only one of them can be confirmed at the end of the analysis.

For the problem addressed in this thesis plan, the following hypotheses are defined: **Null hypothesis (H_0):** The proposed model does not present a statistically significant performance difference in relation to traditional ML models; therefore, it cannot be stated that the proposed model is statistically superior or inferior to the others. **Alternative hypothesis (H_1):** The proposed model exhibits statistically superior performance compared to traditional ML models, indicating that it is statistically superior. **Alternative hypothesis (H_2):** The proposed model exhibits statistically inferior performance compared to traditional ML models, indicating that it is statistically inferior. It should be taken into account that, for the execution of statistical tests, it is necessary to define in advance the significance level or confidence level, which represents the probability that the observed statistical result is not true. In general, significance levels of $P = 0.05$ and $P = 0.01$ are widely accepted, meaning that, when rejecting the null hypothesis, the decision is made with 95% or 99% confidence, respectively. In this thesis plan, two statistical tests widely used by the scientific community are adopted: the Wilcoxon test, to verify whether there is a statistically significant difference between two related samples, and the Friedman test, to assess whether there is a statistically significant difference among three or more related samples.

According to [Wilcoxon 1992], the Wilcoxon test is a non-parametric method for comparing two paired samples. Initially, the numerical values of the differences between each pair are calculated, resulting in three possible conditions: increase (+), decrease

(–), or equality (=). Once all differences between the paired observations are computed, they are ranked according to their absolute values (disregarding the sign), and the original values are then replaced by their corresponding ranks in the ordered scale.

The Friedman test is a non-parametric alternative that uses data ranks rather than raw values to compute the test statistic. It is widely employed to compare three or more related samples. The algorithm that achieves the best performance is assigned rank 1, the second-best receives rank 2, and so forth [Friedman 1992].

3.4. Experimental Setup

For each dataset considered, the following ML regression techniques are applied as base regressors: XGBR (XGBoost Regressor), SVR (Support Vector Regressor), STR (Stacking Regressor), RANSAC (Random Sample Consensus), MLPR (Multilayer Perceptron Regressor), LASSO (Least Absolute Shrinkage and Selection Operator), KNNR (K-Nearest Neighbors Regressor), ENR (Elastic Net Regressor), BRR (Bayesian Ridge Regressor), BGR (Bagging Regressor), and DTR (Decision Tree Regressor). The selection of these techniques was based on several criteria. First, they are well-recognized and validated methods within the scientific community, with established applications across different domains, which reinforces their effectiveness in regression tasks. Each technique presents distinct and complementary characteristics, contributing to a more comprehensive and robust analysis with improved predictive accuracy. The adoption of multiple methods also enables the exploration of different predictive perspectives, fostering the construction of more reliable models. Finally, prior experience and empirical evidence of their strong performance in similar problems motivated their inclusion in this study.

The Genetic Algorithm (GA) is employed as an optimization technique to select the most suitable combinations of regression models that compose the ensemble. Each individual is represented by a binary vector, in which each gene indicates the inclusion or exclusion of a given model. The objective function aims to minimize the MAE, estimated through cross-validation, with penalties applied to infeasible individuals that do not select any model. Population evolution is conducted through crossover and mutation operators, ensuring genetic diversity and efficient exploration of the search space. Individual selection is performed using the tournament selection method, promoting controlled selective pressure.

After the selection of the best individuals by the GA, the chosen models are integrated into a regression ensemble using the Voting Regressor approach. Individual model predictions are combined through an arithmetic mean with uniform weights, following the default configuration, in order to reduce individual errors and increase the robustness of the estimates. The final ensemble composition is directly determined by the fittest solution obtained by the GA, with the number of models varying according to the selected combination.

4. Results

As preliminary results, experimental studies and a systematic literature review were conducted. As an outcome of these activities, two scientific articles were accepted for publication. The study presented in [Silva et al. 2025] aimed to evaluate, through a controlled experiment, the performance of machine learning (ML)-based regression techniques for

software development team effort estimation using datasets from the PROMISE repository. To enhance model performance, Particle Swarm Optimization (PSO) was employed for hyperparameter tuning. The results indicated that the use of PSO significantly contributed to improving predictive accuracy, highlighting XGBoost as the model with the best overall performance, while Stacked Generalization and K-Nearest Neighbors techniques proved to be promising in specific scenarios.

Subsequently, the study entitled *A tertiary study of systematic reviews on machine learning techniques for software team effort estimation* was presented at the 51st Latin American Conference on Informatics (CLEI 2025), with publication scheduled in the IEEE proceedings. This study aimed to identify research trends, challenges, and gaps in the field. The results highlight the predominance of ensemble methods, bio-inspired algorithms, neural networks, and regression techniques, as well as challenges related to data quality, the risk of overfitting, and the scarcity of up-to-date datasets, emphasizing the need for new datasets and hybrid approaches.

5. Preliminary Conclusions

The preliminary results indicate that the integration of regression ensembles with GA presents strong potential to enhance the accuracy of software development team effort estimation. Initial evidence obtained from experimental studies and a systematic literature review suggests that hybrid approaches tend to outperform individual regression models, particularly in scenarios characterized by high data variability. It is also observed that evolutionary optimization techniques contribute significantly to the dynamic selection and combination of models, resulting in consistent predictive gains. These findings reinforce the feasibility of the proposed approach and justify the continuation of the investigation through more comprehensive experiments and rigorous statistical analyses, aiming to consolidate the effectiveness and generalization capability of the proposed model.

Referências

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

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).

Boston Consulting Group (2024). Software projects don't have to be late, costly, and irrelevant. `https://www.bcg.com/publications/2024/software-projects-dont-have-to-be-late-costly-and-irrelevant`. Acesso em: 26 jan. 2025.

de Barcelos Tronto, I. F., da Silva, J. D. S., and Sant'Anna, N. (2006). Uma investigação de modelos de estimativas de esforço em gerenciamento de projeto de software. In *Anais do XX Simpósio Brasileiro de Engenharia de Software*, pages 224–238. SBC.

Easterbrook, S., Singer, J., Storey, M.-A., and Damian, D. (2008). Selecting empirical methods for software engineering research. *Guide to advanced empirical software engineering*, pages 285–311.

Friedman, M. (1992). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. In KOTZ, S. and JOHNSON, N. L., editors, *Breakthroughs in Statistics: Methodology and Distribution*, pages 205–227. Springer, New York.

Goyal, S. (2022). 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.

Hameed, S., Elsheikh, Y., and Azzeh, M. (2023). An optimized case-based software project effort estimation using genetic algorithm. *Information and Software Technology*, 153:107088.

Khan, M. S., Jabeen, F., Ghouzali, S., Rehman, Z., Naz, S., and Abdul, W. (2021). Metaheuristic algorithms in optimizing deep neural network model for software effort estimation. *IEEE Access*, 9:60309–60327.

Mohsin, Z. R. (2021). Application of artificial neural networks in prediction of software development effort. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(14):4186–4202.

Rajput, Y., Razi, M. H., and Sharma, A. K. (2025). A comparative analysis of different machine learning techniques used in software effort estimation. In *2025 2nd International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)*, pages 385–393. IEEE.

Reddy, S. H. V. and Thinakaran, K. (2022). Novel software effort estimation method using naive bayes technique. In *2022 3rd International Conference on Smart Electronics and Communication (ICOSEC)*, pages 1600–1602. IEEE.

Silva, W. K. N. d., Nascimento, B. R. d., Miranda, P., and Vicente, E. P. (2025). Predictive regression models of machine learning for effort estimation in software teams: An experimental study. In *Anais do 27th International Conference on Enterprise Information Systems (ICEIS 2025)*, pages 219–226, Setúbal, Portugal. SciTePress.

STANDISH GROUP (2020). Chaos report 2020: Beyond infinity. Disponível em: `https://standishgroup.myshopify.com/collections/frontpage/products/copy-of-chaos-report-beyond-infinity-digital-version`. Acesso em: 26 jul. 2025.

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.

Wilcoxon, F. (1992). Individual comparisons by ranking methods. In KOTZ, S. and JOHNSON, N. L., editors, *Breakthroughs in Statistics: Methodology and Distribution*, pages 196–202. Springer, New York.