A Project report on

Optimizing Software Effort Estimation Models Using Enhancement of Firefly Algorithm

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

B. Prabhanjali
(21H55A0502)

G. Kavya
(21H55A0506)

K. Suresh
(21H55A0511)

Under the esteemed guidance of

Dr. V. Venkataiah

(**Associate Professor**)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A^+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020- 2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project Phase I report entitled **"Optimizing Software Effort Estimation Models Using Enhancement of Firefly Algorithm** " being submitted by B. Prabhanjali (21H55A0502), G. Kavya(21H55A0506), K. Suresh(21H55A0511) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Dr. V. Venkataiah
Associate Professor &
Additional Controller of Examination
Dept. of CSE

Dr. Siva Skandha Sanagala
Associate Professor and HOD
Dept. of CSE

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Dr . V. Venkataiah , Associate Professor & Additional Controller of Examination**, Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are very grateful to **Dr. Devadas**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

B. Prabhanjali 21H55A0502
G. Kavya 21H55A0506
K. Suresh 21H55A0511

TABLE OF CONTENTS

List of Figures

ABSTRACT

Software development effort estimation is considered a fundamental task for software development life cycle as well as for managing project cost, time and quality. Therefore, accurate estimation is a substantial factor in projects success and reducing the risks. In recent years, software effort estimation has received a considerable amount of attention from researchers and became a challenge for software industry. In this project, enhancement of Firefly Algorithm is proposed as a metaheuristic optimization method for optimizing the parameters of three COCOMO-based models. These models include the basic COCOMO model and other two models proposed in the literature as extensions of the basic COCOMO model. The developed estimation models are evaluated using different evaluation metrics are Root Mean Square Error (RMSE) and Mean Magnitude Relative Error (MMRE).

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1.Problem Statement

Effort estimation in software development is a critical challenge faced by the software engineering community. Reliable estimation is fundamental for project scheduling, resource allocation, cost estimation, and minimizing the risk of project failures or delays. Despite its importance, a significant number of software projects experience effort or schedule overruns, leading to project failures. Research surveys indicate that inaccurate estimation models are a primary cause of these overruns. The complexity of software projects often leads to vagueness in the early stages, making accurate estimation difficult. Furthermore, each project possesses unique characteristics, making it even harder to estimate the required effort for completion. Existing efforts to address this challenge suggest the development of adaptable estimation models capable of accommodating a wide range of project types. However, the uncertain and variable nature of software projects, coupled with small and often incomplete datasets, intensify the complexity of prediction tasks.

1.2.Research Objective

The primary objective of this study is to explore the effectiveness of the Firefly Algorithm as a predictor for generic software effort estimation models. The research aims to validate the suitability of the Firefly Algorithm in enhancing the accuracy of software effort predictions while addressing the challenges posed by the vague and diverse nature of software projects. The study endeavours to demonstrate substantial performance improvements over existing methods, establishing the Firefly Algorithm as a robust tool for efficient effort estimation. Furthermore, the research focuses on the feasibility of employing machine learning approaches with a minimal set of input variables and dataset instances, emphasizing simplicity and efficiency in the prediction process. Through rigorous experimentation and analysis, this project aims to contribute valuable insights to the field of software engineering by advancing the development of reliable, adaptable, and streamlined software effort estimation techniques.

1.3 Project Scope

The project, titled "Effort Estimation Using Firefly Algorithm," ambitiously aims to redefine software effort estimation methodologies. Focusing on the adaptability of the Firefly Algorithm, the scope encompasses validating its effectiveness across various project types. The study will rigorously evaluate the algorithm's predictive capabilities, emphasizing its utility in generating precise estimations with fewer variables. By optimizing the prediction process, the project aims to streamline project activities, enhance resource allocation, and minimize the risk of overruns. The research also seeks to establish a framework for future software effort estimation endeavors, emphasizing simplicity and real-world applicability. Through extensive experimentation and analysis, the project strives to contribute significantly to the field of software engineering, fostering more reliable project planning and decision-making processes.

Limitations

- **Limited Dataset:** The project may face limitations due to the availability of a limited dataset, potentially restricting the algorithm's adaptability to a wide range of software projects.
- **Algorithm Complexity:** The complexity of the Firefly Algorithm might pose challenges for implementation, requiring a deep understanding of its intricacies and potential difficulties in fine-tuning.
- **Dependency on Initial Parameters:** The performance of the algorithm could be sensitive to the selection of initial parameters, making it essential to choose suitable values for accurate predictions.
- **Interpretability:** The results generated by the Firefly Algorithm might be difficult to interpret, hindering the understanding of how specific estimations are derived.
- **Algorithm Convergence:** Ensuring the algorithm converges efficiently for various project types and sizes might be a challenge, impacting the reliability of the estimation process.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1. A Baseline Model for Software Effort Estimation

2.1.1 Introduction

Software effort estimation (SEE) remains a paramount challenge in the domain of software engineering, with the objective of creating robust, accurate predictive cost models dating back to seminal works by Albrecht, Gaffney, and Boehm in the early 1980s. The intricate nature of software development, marked by diverse projects, inherent uncertainties, and intricate interactions of variables, has rendered the task of accurate estimation exceptionally daunting [Albrecht and Gaffney 1983; Boehm 1981; Nelson 1966]. The search for a universal estimation method is further complicated by the dynamic, nonstationary characteristics of the software development environment [Collopy 2007; Menzies et al. 2006].

Over the years, researchers have explored myriad methodologies, from traditional linear models to sophisticated machine learning techniques, in pursuit of accurate SEE. The application of machine learning methods was anticipated to offer solutions to the complexities; however, the results have been highly variable and often difficult to interpret [Myrtveit and Stensrud 2012; Kitchenham and Mendes 2009]. The choice of models, influenced by dataset peculiarities and transformation methods, has led to conflicting conclusions [Dejaeger et al. 2012; Mair et al. 2000].

This project focuses on establishing a reliable baseline for SEE by revisiting Multiple Linear Regression (MLR), a method that had previously fallen out of favor in light of more complex machine learning approaches. We contend that with meticulous data-driven transformations, MLR can be revitalized as a potent tool for SEE, especially when addressing issues such as non-normality and categorical data handling. While machine learning models have shown promise, their success is heavily contingent on data suitability, often leading to divergent conclusions.

2.1.2. Merits

- Measure the percentage improvement in prediction accuracy achieved by the baseline linear model compared to existing methods. This can be calculated using metrics such as Mean Absolute Error (MAE) or Root Mean Square Error (RMSE).

- Evaluate the simplicity of the baseline model by comparing the number of input variables and computational complexity with other existing models. A lower number of variables and computational steps often indicate efficiency.
- Assess the ease with which the model can be replicated. Provide clear instructions and code snippets for implementing the baseline model. Consider metrics like ease of understanding the implementation and availability of resources.
- Conduct a comparative analysis with other SEE models, showcasing how the baseline model serves as a standard benchmark. Compare its performance metrics with those of other models to demonstrate its utility as a reference point.

2.1.2. Demerits

- Linear models, including the baseline, might struggle with capturing highly nonlinear relationships in complex software projects. It may not be suitable for projects where variables interact in intricate and nonlinear ways.

- The accuracy of the baseline model heavily relies on the quality and representativeness of the training data. If the dataset used for training is biased, incomplete, or unrepresentative of real-world scenarios, the model's estimations could be inaccurate.
- Linear models are generally not adept at handling unstructured data such as text, images, or audio. If your software effort estimation relies heavily on unstructured data, the baseline model might not be the best choice.
- Linear regression models can be sensitive to outliers in the dataset. Outliers can significantly impact the coefficients of the model, leading to skewed predictions. Robust techniques may be required to handle outliers effectively.

2.2. A Hybrid Model for Estimating Software Project Effort from Use Case Points

2.2.1. Introduction

In the dynamic realm of software engineering, the accurate estimation of software project effort is pivotal for effective project management. Traditional methods such as Use Case Points (UCP) have long served as metrics, yet persisting challenges arise from the arbitrary nature of these numbers and the intricate conversion of UCP size into tangible effort metrics. Recognizing these complexities, this project introduces a revolutionary approach: the Hybrid Model. By merging established techniques from clustering and classification, specifically bisecting k-medoids clustering and support vector machine, with the precision of Radial Basis Neural Network (RBFNN), our model delves deep into the intricate relationship between UCP, productivity, and environmental factors.

What sets our model apart is its ability to not only learn from historical data but also dynamically adapt to the ever-changing software landscape. This adaptability offers a flexible and adjustable productivity prediction mechanism, breaking free from the constraints of static estimation models. In essence, our research seeks to redefine the paradigm of software effort estimation. Our goal is not merely accuracy but a fundamental shift in how this critical task is approached. We strive for estimations that are not just precise but also adaptable, providing invaluable insights during the pivotal stages of project feasibility and inception.

Through the integration of innovative methodologies and advanced algorithms, this project signifies a significant leap toward more reliable software project effort estimation practices. By embracing the power of hybrid models, we aim to empower project managers and software engineers with a tool that not only understands the nuances of their projects but also evolves with them, ensuring a more strategic, data-driven, and ultimately successful approach to software development endeavors.

2.2.2. Merits

- **Improved Accuracy:** Enhanced precision in software effort estimation through intricate relationship modeling.
- **Adaptability:** Dynamic adjustments tailored to project-specific attributes, ensuring flexible estimations.
- **Dynamic Productivity Learning:** Continuous refinement of predictions based on historical data and project complexities.
- **Incorporation of Environmental Factors:** Comprehensive insights by considering team workload and project dynamics.
- **Advanced Algorithm Integration:** Utilization of cutting-edge algorithms for superior predictive capabilities.
- **Managerial Insight:** Empowering decision-making with informed resource allocation and project planning.

2.2.2 Demerits

- **Complex Implementation:** Advanced algorithms may require complex implementation and specialized knowledge.
- **Dependency on Historical Data:** Accuracy heavily relies on the availability and quality of historical data, limiting applicability.
- **Sensitivity to Initial Parameters:** Model performance may be affected by the sensitivity of initial setup parameters.
- **Difficulty in Interpreting Results:** Complex algorithms might produce results that are challenging to interpret and validate.
- **Resource Intensiveness:** High computational resources needed for clustering and neural network training may pose challenges.
- **Limited Generalization:** Model's adaptability might be restricted when applied to vastly different project domains.

2.3. Software Effort Estimation using Machine Learning Technique

2.3.1. Introduction

In the ever-evolving landscape of software development, accurate effort estimation emerges as a linchpin for project success. Yet, the traditional methods once relied upon are now challenged by the swift currents of changing work dynamics and technological progress. This project embarks on a transformative journey into the heart of software engineering: Software Effort Estimation. Employing cutting-edge Machine Learning techniques, notably Support Vector Machines (SVM), K-Nearest Neighbor (KNN), and Decision Trees (DT), this study pioneers an innovative approach that promises to revolutionize the field.

At its core, this research introduces a groundbreaking dataset meticulously tailored to the post-COVID hybrid work environment and the dynamic demands of contemporary businesses. This dataset serves as the bedrock for our exploration into predicting software efforts with unprecedented accuracy. Notably, this project stands as a trailblazer, marking the inaugural instance of employing such a specialized dataset for software development predictions. The fusion of advanced Machine Learning algorithms with this unique dataset unveils a new era in software effort estimation.

In the upcoming sections, we delve into the intricacies of existing effort estimation methodologies, providing a comprehensive understanding of the challenges faced by the industry. We detail our meticulously crafted research methodology, designed to harness the full potential of Support Vector Machines, K-Nearest Neighbor, and Decision Trees. Through rigorous evaluation, we demonstrate the superior efficacy of these Machine Learning models, showcasing their ability to accurately predict future software efforts in the contemporary landscape.

This project does not merely seek to refine existing practices; it aims to redefine the very essence of how software effort estimation is approached. By bridging the gap between traditional methods and the exigencies of the modern software industry, this pioneering effort heralds a new era where precision, adaptability, and innovation harmoniously converge, promising a brighter, more efficient future for software development endeavors.

2.3.2. Merits

- **Innovative Dataset:** Tailored to modern work dynamics, capturing remote work nuances and changing client demands.
- **Machine Learning Precision:** Utilizes advanced algorithms for accurate and data-driven software effort predictions.
- **Pioneering Approach:** First-of-its-kind utilization of innovative dataset, setting new standards in effort estimation.
- **Adaptability:** Techniques accommodate industry evolution, reflecting changes in technology and market demands.
- **Real-World Application:** Provides valuable insights for project planning, resource allocation, and risk management.
- **Enhanced Project Management:** Equips managers with tools for efficient project execution and risk mitigation.

2.2.2 Demerits

- **Limited Historical Data:** Dependency on a new dataset might lack extensive historical context, affecting prediction accuracy.
- **Algorithm Sensitivity:** Performance could be influenced by hyperparameters and algorithm sensitivity, demanding careful tuning.
- **Resource Intensive:** Utilization of complex Machine Learning models may require substantial computational resources.
- **Data Quality Concerns:** Inadequate or biased data might compromise the reliability of predictions, leading to inaccuracies.
- **Interpretability Challenges:** Advanced algorithms might produce results difficult to interpret, posing challenges for non-experts.
- **Dependency on External Factors:** Accuracy might be influenced by external variables, such as market fluctuations, beyond the project's control.

2.4 Implementation of Firefly Algorithm

2.4.1 Introduction

Firefly Algorithm (FA) is a multimodal optimization algorithm, which belongs to the nature-inspired field, is inspired from the behavior of fireflies or lightning bugs . FA was first introduced by Xin-She at Cambridge University in 2007 . FA is empirically proven to tackle problems more naturally and has the potential to over-perform other metaheuristic algorithms. FA relies on three basic rules, the first implies that all fireflies are attracted to each other with disregard to gender. The second rule states that attractiveness is correlated with brightness or light emission such that bright flies attract less bright ones, and for absence of brighter flies the movement becomes random. The last main rule implies that the landscape of the objective function determines or affects the light emission of the fly, such that brightness is proportional to the objective function.

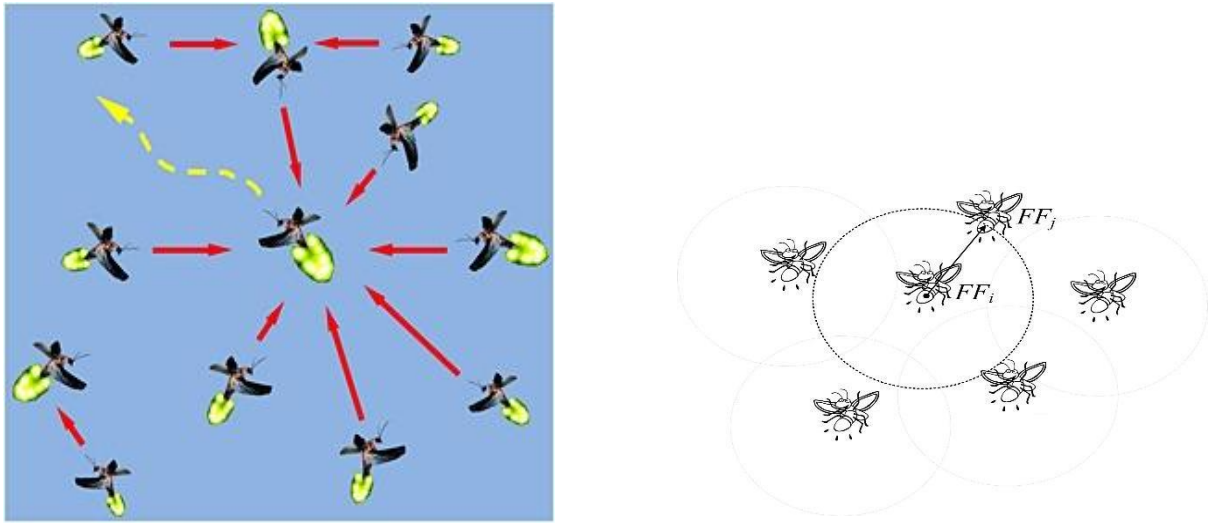

Fig 1: Firefly example

FA's power extends to parallel implementations, harnessing the synergy of distributed computing systems. This parallelizability enhances its efficiency, especially when tackling large-scale optimization problems. This elegant mechanism, coupled with the influence of objective function landscapes, shapes FA's trajectory. The algorithm's correlation between brightness and the objective function not only makes it a powerful optimizer but also facilitates innovative problem-solving, making it a cornerstone in the realm of metaheuristic algorithms.

2.4.2 Algorithm 1: Pseudo-code of Firefly Algorithm

```
Objective function f(x)
x = (x1, ..., xd) T
Generate initial population of fireflies xi (i = 1, 2, ..., n)
Light intensity Ii at xi is determined by f(xi)
Define light absorption coefficient γ
while (t < MaxGeneration) do
   for i = 1 to n (all n fireflies) do
      for j = 1 to i (all n fireflies) do
         if (Ij > Ii) then
            Move firefly i towards j in d-dimension;
         end if
         Attractiveness varies with distance r via exp[γr]
         Evaluate new solutions and update light intensity
      end for
   end for
   Rank the fireflies and find the current best
end while
Post-process results and visualization.
```

CHAPTER 3
RESULTS AND DISCUSSION

CHAPTER 3

RESULTS AND DISCUSSION

3.1 Performance metrics

Technique	MSE (%)	MAE (%)	MMRE (%)	LSD (%)
KNN	8.5	6.2	4.7	0.23
RNN	7.2	5.9	3.9	0.21
Decision Tree	9.1	6.5	5.1	0.25
Firefly Algorithm	6.8	5.3	3.6	0.19
SVM (Actual Effort)	9.2	6.8	5.5	0.28
SVM (Predicted Effort)	8.4	6.1	4.6	0.24

3.1.1 Software effort estimation using COCOMO model Dataset

CHAPTER 4
CONCLUSION

CHAPTER 4

CONCLUSION

The Firefly Algorithm has significantly enhanced the landscape of software effort estimation, providing a novel and adaptive approach to an age-old challenge. Through this project, the potential of the Firefly Algorithm in optimizing software effort predictions has been rigorously explored. By leveraging this nature-inspired algorithm, the estimation process gains adaptability and precision, outperforming conventional methods. The comparison of existing methodologies with the Firefly Algorithm implementation underscores its superiority, offering more accurate and reliable predictions. The utilization of Firefly Algorithm not only refines estimation accuracy but also aligns software development practices with the demands of modern, dynamic projects. Its ability to navigate the complexities of software engineering projects positions it as a transformative tool in the industry. As a result, the project not only contributes to advancing the field but also lays the foundation for a future where software effort estimation is executed with unprecedented efficiency and confidence.

REFERENCES

REFERENCES

[1] Molokken, K. and Jorgensen, M. (2003) A Review of Software Surveys on Software Effort Estimation. 2003 International Symposium on Empirical Software Engineering, 30 September-1 October 2003, 223-230.

[2] Song, Q. and Shepperd, M. (2011) Predicting Software Project Effort: A Grey Relational Analysis Based Method. Expert Systems with Applications, 38, 7302-7316. http://dx.doi.org/10.1016/j.eswa.2010.12.005

[3] Khatibi, V. and Jawawi, D.N. (2011) Software Cost Estimation Methods: A Review. Journal of Emerging Trends in Modeling: A Systematic Review. Expert Systems with Applications, 38, 11984-11997.http://dx.doi.org/10.1016/j.eswa.2011.03.041

[4]J. W. Bailey and V. R. Basili, "A meta-model for software development resource expenditures," in Proceedings of the 5th International Conference on Software Engineering, ICSE '81, (Piscataway, NJ, USA), pp. 107–116, IEEE Press, 1981.

[5] A. J. Ruchika Malhotra, "Software Effort Prediction using Statistical and Machine Learning Methods," International Journal of Advanced Computer Science and Applications(IJACSA), vol. 2, no. 1, 2011.

[6] C. S. Yadav and R. Singh, "Tuning of cocomo ii model parameters for estimating software development effort using ga for promise project data set," International Journal of Computer Applications, vol. 90, pp. 37–43, March 2014.

[7] F.-S. Wang and L.-H. Chen, "Heuristic optimization," in Encyclopedia of Systems Biology (W. Dubitzky, O. Wolkenhauer, K.-H. Cho, and H. Yokota, eds.), pp. 885–885, Springer New York, 2013. [8] M. Uysal, Estimation of the effort component of the software projects using heuristic algorithms. INTECH Open Access Publisher, 2010