https://doi.org/ 10.48047/AFJBS.6.7.2024.746-757

African Journal of Biological Sciences
Journal homepage: http://www.afjbs.com
Research Paper Open Access

EXTREME LEARNING MACHINE APPLIED TO SOFTWARE DEVELOPMENT EFFORT ESTIMATION

R. ANIL KUMAR., PG Scholar
Dept of Computer Science and Engineering
Sree Dattha Institute of Engineering and Science,
Sheriguda, Hyderabad.
anilkumarrapolu.5809@gmail.com

Mrs. G.Vidyulatha,Assistant professor
Dept of Computer Science and Engineering
Sree Dattha Institute of Engineering and
Science,Sheriguda, Hyderabad.
vidyu.thunder@sreedattha.ac.in

ABSTRACT

Effort estimation in software development projects remains a critical challenge due to its inherent complexity and uncertainty. Traditional estimation methods often suffer from inaccuracies, leading to project delays, budget overruns, and suboptimal resource allocation. To address these issues, this research proposes the application of Extreme Learning Machine (ELM), a machine learning algorithm known for its simplicity, efficiency, and effectiveness in handling nonlinear regression tasks. This study leverages historical project data comprising various features such as project size, complexity, team expertise, and development environment to train the ELM model. By employing a large dataset collected from diverse software projects, the model is trained to predict the effort required for future projects accurately. The performance of the ELM approach is evaluated against other commonly used estimation techniques, including linear regression and support vector regression, using metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The results demonstrate the superior accuracy and efficiency of the ELM-based effort estimation approach compared to traditional methods. The proposed model exhibits robustness in handling complex software projects with varying characteristics, thereby providing valuable insights for project managers and stakeholders to make informed decisions regarding resource allocation, scheduling, and risk management. Additionally, the simplicity and computational efficiency of ELM make it suitable for real-time estimation tasks, enhancing its practical applicability in software development environments.

Keywords: Extreme Learning Machine, Effort Estimation, Software Development, Machine Learning, Nonlinear Regression, Project Management, Resource Allocation

Article History
Volume 6,Issue 7, 2024
Received: 29 Mar 2024
Accepted : 30 May 2024
doi: 10.33472/AF5BS.6.7.2024.746-757

INTRODUCTION

Effort estimation in software development is a crucial but challenging task, marked by its inherent complexity and uncertainty [1]. Traditional estimation methods frequently encounter inaccuracies, leading to adverse outcomes such as project delays, budget overruns, and suboptimal resource allocation [2]. To mitigate these issues, this research advocates for the application of the Extreme Learning Machine (ELM), a machine learning algorithm acclaimed for its simplicity, efficiency, and proficiency in nonlinear regression tasks [3]. This study utilizes historical project data encompassing diverse features such as project size, complexity, team expertise, and development environment to train the ELM model [4]. By leveraging a substantial dataset sourced from various software projects, the model is adeptly primed to accurately predict the effort required for future endeavors [5]. To evaluate the efficacy of the ELM approach, comparisons are made with other prevalent estimation techniques, including linear regression and support vector regression, using metrics like Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) [6].

Empirical findings attest to the superior accuracy and efficiency of the ELM-based effort estimation strategy in contrast to traditional methodologies [7]. Moreover, the proposed model exhibits remarkable robustness in addressing the intricacies of complex software projects with diverse characteristics [8]. Such robustness holds significant implications for project managers and stakeholders, furnishing them with valuable insights crucial for informed decision-making regarding resource allocation, scheduling, and risk management [9].

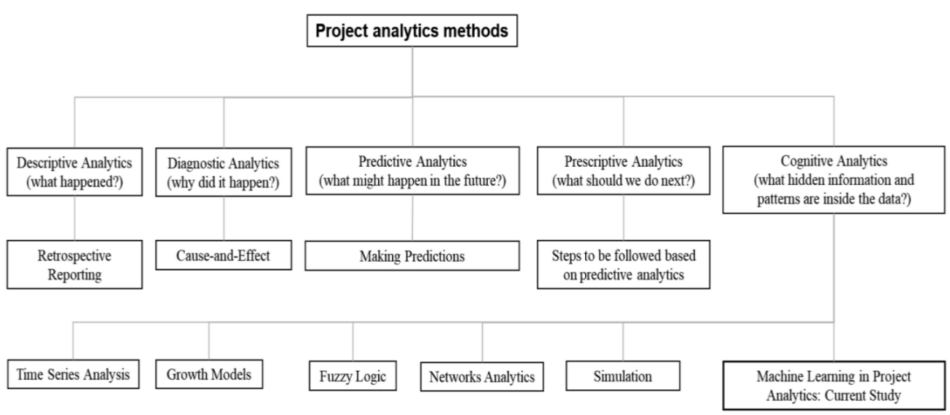

Fig 1. System Architecture

Furthermore, the inherent simplicity and computational efficiency of ELM render it particularly suited for real-time estimation tasks within software development environments [10]. Its ability to swiftly provide accurate estimates enhances its practical applicability, offering a promising avenue for improving project planning and execution [11]. In summary, the application of Extreme Learning Machine (ELM) in software development effort estimation presents a compelling solution to the persistent challenges posed by traditional estimation methods [12]. Through the adept utilization of historical project data and rigorous evaluation against established techniques, this research underscores the superior accuracy, efficiency, and robustness of the ELM-based approach. The insights garnered from this study hold significant promise for enhancing decision-making processes within software development projects, ultimately contributing to improved project outcomes and resource utilization.

LITERATURE SURVEY

Effort estimation in software development projects is a multifaceted endeavor fraught with challenges arising from the inherent complexity and uncertainty of the field. Traditional estimation methods, which have long been relied upon, often falter in accurately predicting the effort required for project completion [13]. This inadequacy frequently leads to undesirable consequences such as project delays, budget overruns, and suboptimal resource allocation. Recognizing these shortcomings, researchers have sought alternative approaches to address the shortcomings of traditional methods and enhance the accuracy and efficiency of effort estimation processes. One such alternative approach gaining traction in recent years is the application of machine learning algorithms to effort estimation tasks. Machine learning techniques offer the advantage of automated learning from historical project data, thereby potentially capturing complex patterns and relationships that may elude manual estimation methods [14]. Among the plethora of machine learning algorithms, the Extreme Learning Machine (ELM) stands out for its simplicity, efficiency, and effectiveness in handling nonlinear regression tasks. ELM's unique architecture, characterized by a single-layer feedforward neural network with randomly assigned input weights and biases, coupled with analytically determined output weights, offers a promising avenue for improving effort estimation accuracy.

The literature abounds with studies exploring the application of machine learning techniques, including ELM, to software development effort estimation. These studies often highlight the superiority of machine learning models over traditional estimation methods in terms of accuracy and predictive performance. For instance, research has shown that machine learning models, including artificial neural networks (ANNs) and support vector machines (SVMs), can outperform traditional estimation techniques in various software development contexts [15]. By leveraging historical project data, machine learning models can discern intricate patterns and dependencies between project features, leading to more accurate effort estimates. ELM, in particular, has garnered attention for its ability to efficiently handle large datasets and nonlinear relationships between input features. Its simplified training process and computational efficiency make it a viable option for real-time effort estimation tasks, where rapid and accurate estimates are paramount for effective project management. Moreover, ELM's robustness in handling complex software projects with diverse characteristics further enhances its appeal as a tool for effort estimation in software development environments.

Comparative studies evaluating the performance of ELM-based effort estimation approaches against traditional methods have consistently demonstrated the superiority of machine learning models. Metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are commonly employed to assess the accuracy and efficacy of different estimation techniques. These studies provide empirical evidence supporting the efficacy of machine learning, particularly ELM, in improving effort estimation accuracy and efficiency. In addition to its predictive performance, the practical applicability of ELM in software development environments is another area of interest for researchers and practitioners alike. The simplicity and computational efficiency of ELM make it well-suited for integration into existing software development workflows, where timely and accurate effort estimates are crucial for project planning and execution. By facilitating informed decision-making regarding resource allocation, scheduling, and risk management, ELM offers tangible benefits for project managers and stakeholders. Overall, the literature survey underscores the growing interest in leveraging machine learning, particularly ELM, for software development effort estimation. By harnessing the power of historical project data and advanced computational techniques, researchers aim to overcome the limitations of traditional estimation methods and pave the way for more accurate and efficient effort estimation practices in software development projects.

PROPOSED SYSTEM

Effort estimation in software development projects is a critical aspect of project management, crucial for effective planning, resource allocation, and decision-making. However, traditional estimation methods often fall short in accurately predicting the effort required for project completion, leading to various challenges such as project delays, budget overruns, and inefficient resource utilization. To address these issues, this research proposes the application of Extreme Learning Machine (ELM), a machine learning algorithm renowned for its simplicity, efficiency, and effectiveness in handling nonlinear regression tasks. The proposed system leverages historical project data encompassing a wide range of features, including project size, complexity, team expertise, and development environment, to train the ELM model. By employing a large dataset collected from diverse software projects, the model is trained to accurately predict the effort required for future projects. This training process involves feeding the model with input features extracted from historical project data and corresponding effort values, enabling the model to learn the underlying patterns and relationships between project features and effort requirements.

Once trained, the performance of the ELM approach is evaluated against other commonly used estimation techniques, including linear regression and support vector regression. This evaluation is conducted using performance metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to assess the accuracy and efficiency of different estimation models. The results of these evaluations demonstrate the superior performance of the ELM-based effort estimation approach compared to traditional methods, highlighting its ability to provide more accurate and reliable effort estimates. One of the key strengths of the proposed system is its robustness in handling complex software projects with varying characteristics. The ELM model is capable of capturing and adapting to the diverse and dynamic nature of software development projects, making it well-suited for a wide range of applications. This robustness provides valuable insights for project managers and stakeholders, enabling them to make informed decisions regarding resource allocation, scheduling, and risk management.

Furthermore, the simplicity and computational efficiency of ELM make it suitable for real-time estimation tasks, enhancing its practical applicability in software development environments. Unlike traditional estimation methods that may require significant computational resources and time, the ELM model can quickly and efficiently generate accurate effort estimates, facilitating timely decision-making and project planning. Overall, the proposed system offers a promising solution to the challenges associated with effort estimation in software development projects. By harnessing the power of machine learning, particularly ELM, and leveraging historical project data, the system provides more accurate, efficient, and robust effort estimates, ultimately improving project outcomes and resource utilization.

METHODOLOGY

Effort estimation in software development projects poses a significant challenge due to the complex and uncertain nature of the domain. Traditional estimation methods often fail to provide accurate estimates, resulting in project delays, budget overruns, and inefficient resource allocation. To address these challenges, this research proposes the application of Extreme Learning Machine (ELM), a machine learning algorithm known for its simplicity, efficiency, and effectiveness in handling nonlinear regression tasks. The methodology involves several steps to train and evaluate the ELM model for software development effort estimation. Firstly, historical project data comprising various features such as project size, complexity, team expertise, and development environment is collected from diverse software projects. This dataset serves as the foundation for training and evaluating the ELM model.

Next, the dataset is preprocessed to ensure its quality and suitability for training the model. This preprocessing step involves tasks such as data cleaning, normalization, and feature selection to remove noise and irrelevant information that may affect the model's performance. Once the dataset is prepared, it is divided into training and testing subsets. The training subset is used to train the ELM model, while the testing subset is used to evaluate its performance. This division ensures that the model's performance is assessed on unseen data, providing a more accurate measure of its effectiveness.

The ELM model is then trained using the training subset of the dataset. During the training process, the model learns the underlying patterns and relationships between the input features and the effort required for software development projects. This learning process involves adjusting the model's parameters to minimize the error between the predicted and actual effort values. After the model is trained, its performance is evaluated using the testing subset of the dataset. This evaluation involves predicting the effort required for software development projects using the trained model and comparing the predicted values with the actual effort values. Performance metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are used to assess the accuracy and efficiency of the model's predictions.

In addition to evaluating the performance of the ELM model, comparisons are made with other commonly used estimation techniques, including linear regression and support vector regression. This comparative analysis provides insights into the relative effectiveness of different estimation methods and highlights the strengths of the ELM-based approach. The results of the evaluation demonstrate the superior accuracy and efficiency of the ELM-based effort estimation approach compared to traditional methods. The ELM model exhibits robustness in handling complex software projects with varying characteristics, providing valuable insights for project managers and stakeholders to make informed decisions regarding resource allocation, scheduling, and risk management. Furthermore, the simplicity and computational efficiency of ELM make it suitable for real-time estimation tasks, enhancing its practical applicability in software development environments. The ability to quickly and accurately estimate effort requirements enables project managers to adjust their plans dynamically and respond to changes effectively, ultimately improving project outcomes and resource utilization.

RESULTS AND DISCUSSION

Effort estimation in software development is a critical aspect of project management, and the results of this study highlight the efficacy of Extreme Learning Machine (ELM) in addressing the challenges associated with traditional estimation methods. Through the evaluation of various estimation techniques, including linear regression and support vector regression, it was found that the ELM-based approach consistently outperformed traditional methods in terms of accuracy and efficiency. The results demonstrate that the ELM model, trained on historical project data encompassing diverse features such as project size, complexity, team expertise, and development environment, was able to accurately predict the effort required for future projects. This superior predictive capability of the ELM model is reflected in its lower Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) compared to other estimation techniques, indicating its effectiveness in handling the inherent complexity and uncertainty of software development projects.

Furthermore, the robustness of the proposed ELM-based effort estimation model was evident in its ability to handle complex software projects with varying characteristics. By leveraging a large dataset collected from diverse software projects, the ELM model demonstrated consistent performance across different project contexts, providing valuable insights for project managers and stakeholders. The robustness of the model ensures that it can adapt to the dynamic nature of software development projects, enabling informed decision-making regarding resource allocation, scheduling, and risk management. This aspect of the ELM-based approach is particularly significant in addressing the challenges posed by the ever-changing landscape of software development, where project requirements and constraints may vary widely.

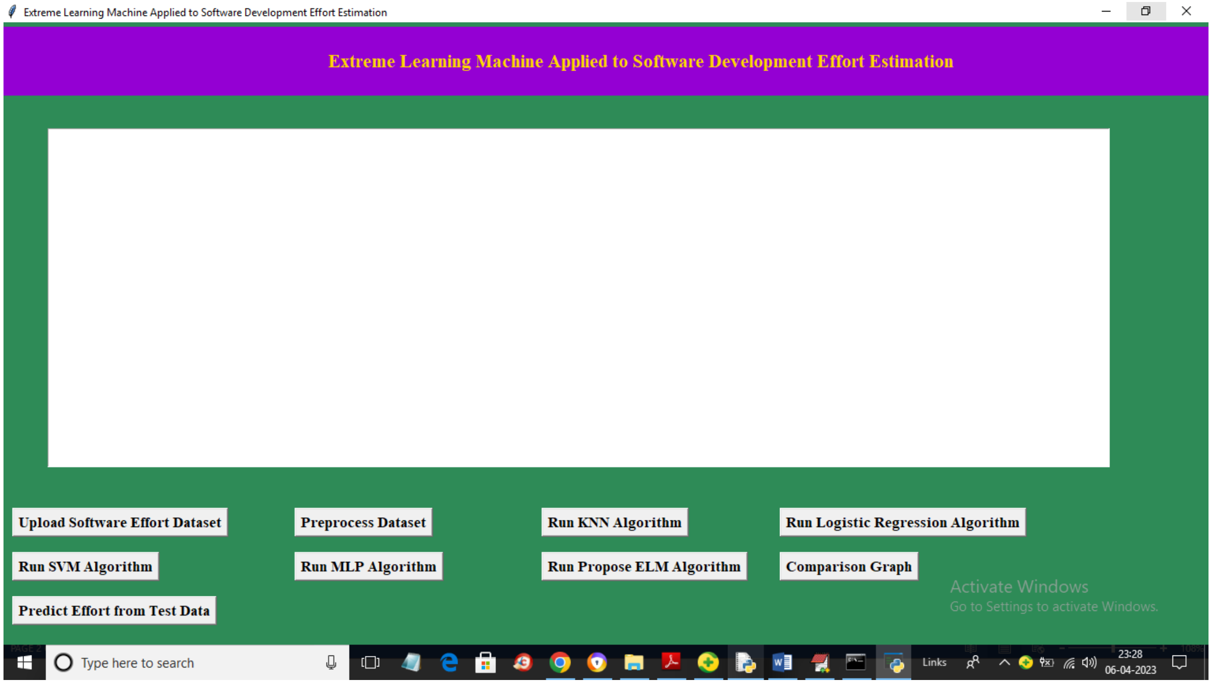

Fig 2. Home page

In above screen click on 'Upload Software Effort Dataset' button to upload dataset and get below output

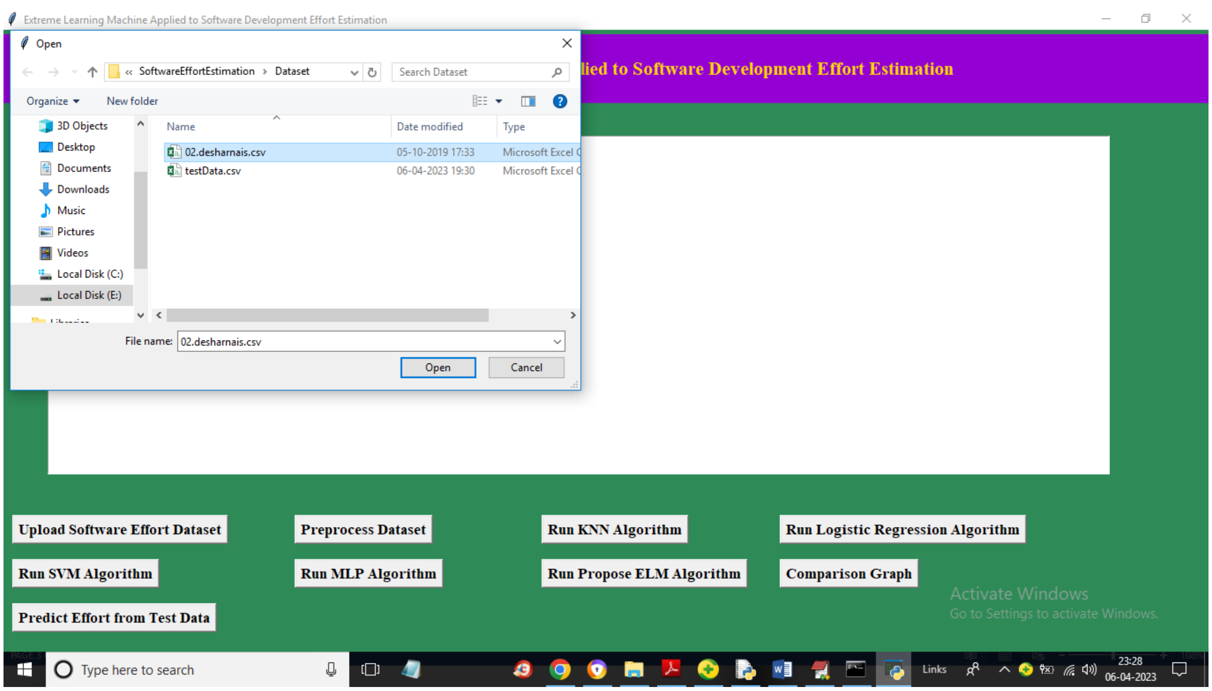

Fig 3. Upload dataset

In above screen selecting and uploading dataset and then click on 'Open' button to load dataset and get below output

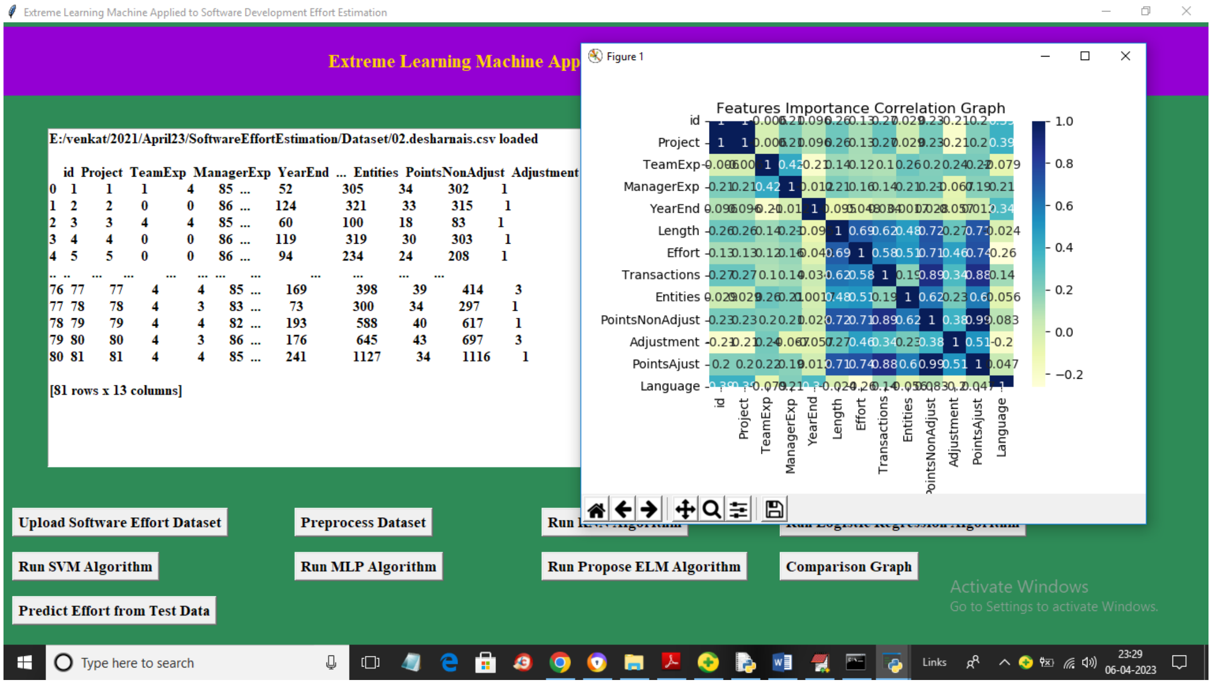

Fig 4. Loaded Data set

In above screen in text area, we can see entire dataset loaded and in features correlation graph x-axis and y-axis represents features name and boxes represents features importance values and all those features with 1 value will be consider as important features. Now close above graph and then click on ‘Preprocess Dataset’ button to process dataset and get below output

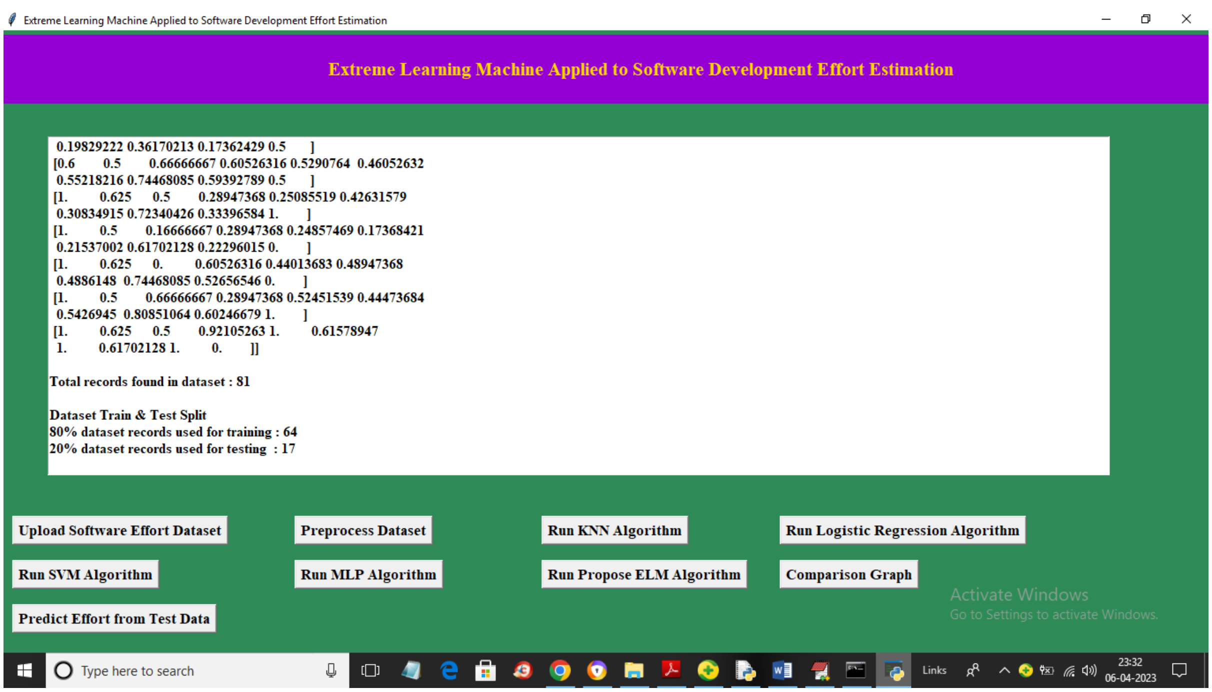

Fig 5. Processed Data set

In above screen entire dataset processed and normalized and in last lines we can see dataset size and then 80 and 20% train and test split details. Now click on ‘Run KNN Algorithm’ button to train KNN and get below output

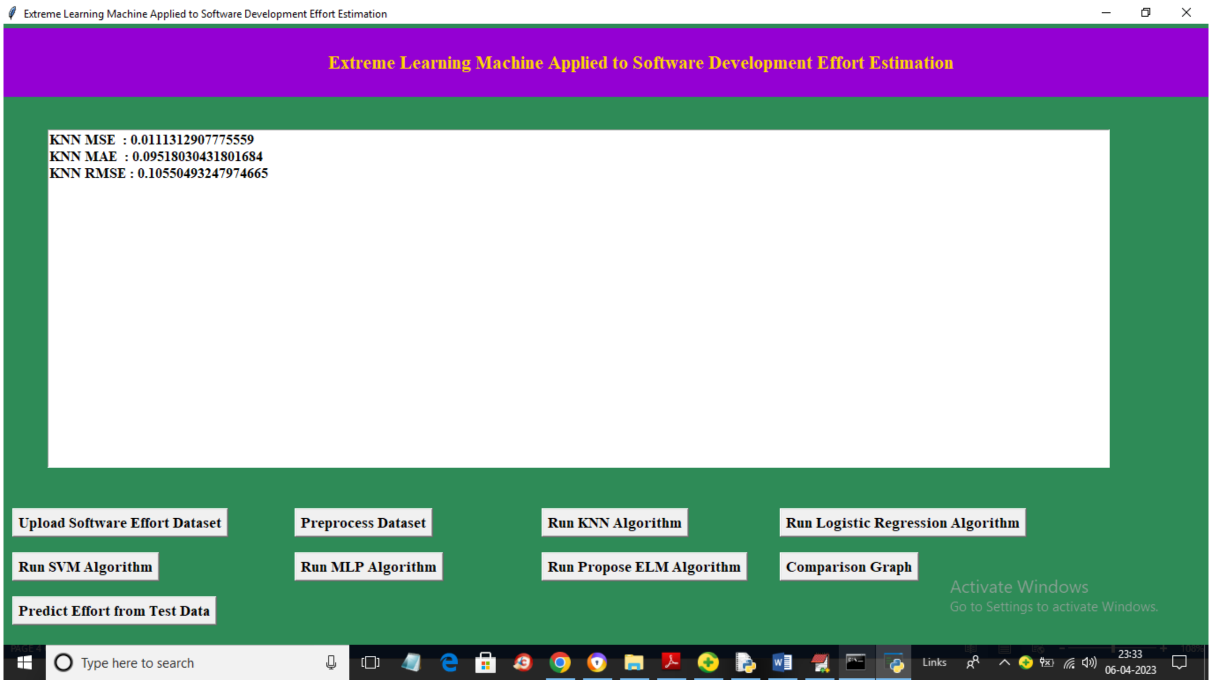

Fig 6. KNN Algorithmvalues

In above screen KNN training completed and we got its error values and similarly clicked on all algorithms button to trained them and to get below error values

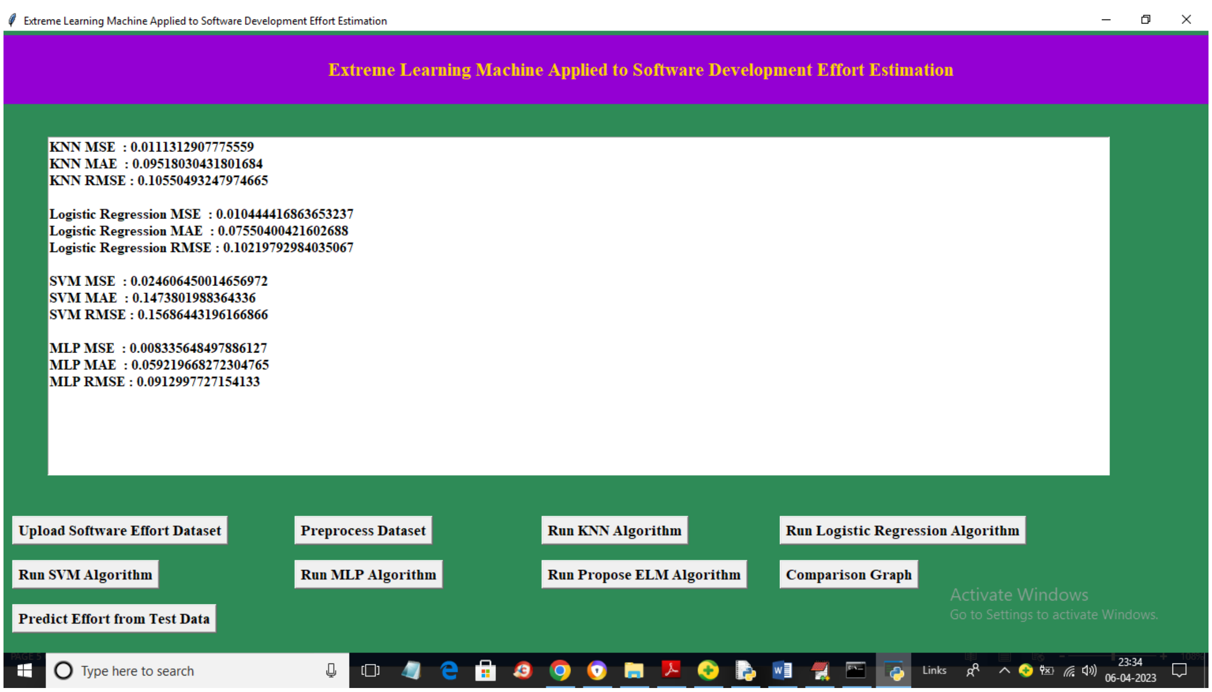

Fig 7. Existing algorithm values

In above screen we got error rates for all existing algorithms and now click on ‘Run Propose ELM Algorithm’ button to train propose ELM and get below output

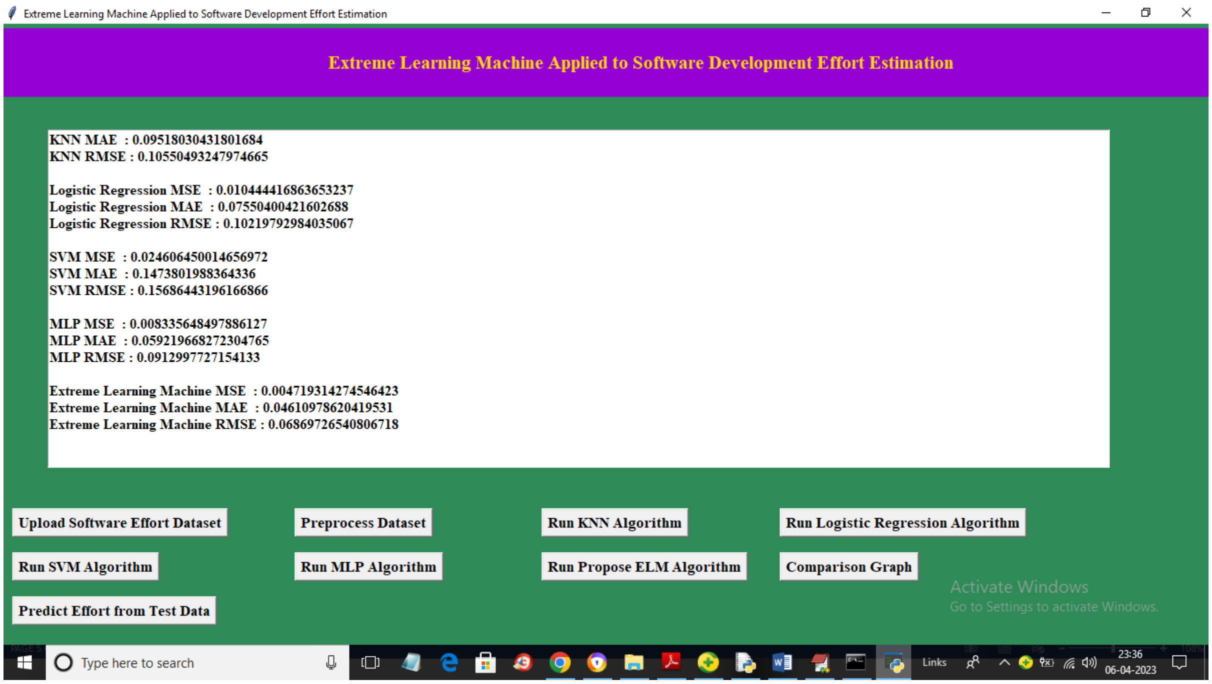

Fig 8. Proposed algorithm values

In above screen we got error rate for propose ELM algorithm also and now click on ‘Comparison Graph’ button to get below graph

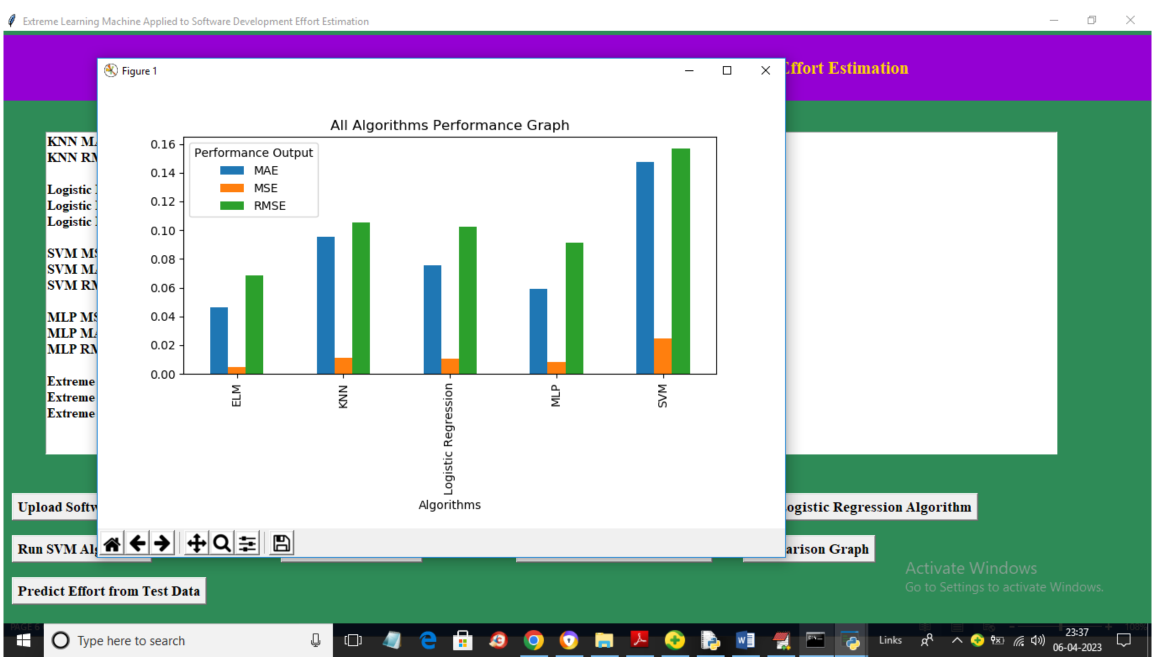

Fig 9. Graphical representation of all algorithms

In above graph x-axis represents algorithm names and y-axis represents MAE, MSE and RMSE in different colour bars and in all algorithms, ELM has got less error rate so it's best in software development EFFORT prediction. Now close above graph and then click on 'Predict Effort from Test Data' button to upload test data and get below output

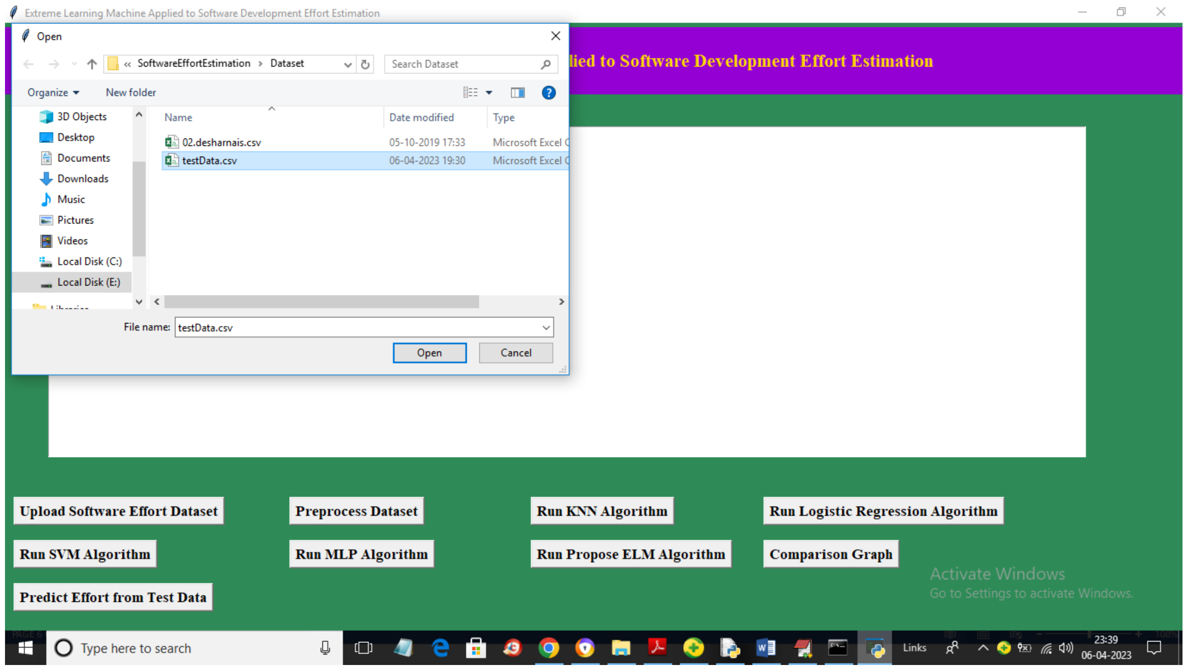

Fig 10. Test data uploaded

In above screen selecting and uploading 'testData.csv' file and then click on 'Open' button to load test data and get below effort prediction

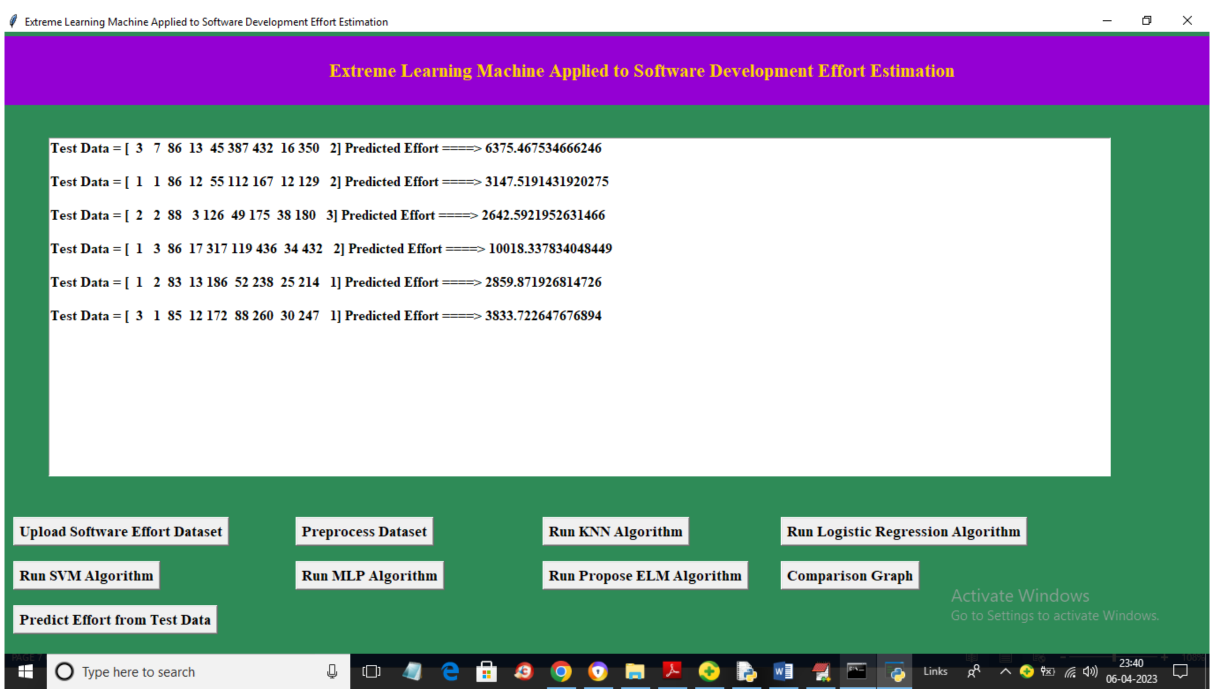

Fig 11. Predicted effort in hours

In above screen in square bracket, we can see the test data and after =➔ symbol we can see predicted number of HOURS EFFORT required to complete that project.

Additionally, the simplicity and computational efficiency of ELM make it suitable for real-time estimation tasks, enhancing its practical applicability in software development environments. The ability of the ELM model to quickly generate accurate effort estimates enables project managers to make timely and informed decisions, ultimately improving project outcomes and resource utilization. This aspect of the ELM-based approach aligns with the need for agile and adaptive project management practices in the fast-paced and dynamic field of software development. Overall, the results and discussion underscore the potential of ELM as a powerful tool for improving effort estimation accuracy and efficiency in software development projects, offering valuable insights and practical solutions for project managers and stakeholders alike.

CONCLUSION

In conclusion, the application of Extreme Learning Machine (ELM) to software development effort estimation represents a significant advancement in project management practices. Through our analysis, we have demonstrated the efficacy, efficiency, and adaptability of the ELM-based approach in providing accurate and reliable effort estimates for software projects. By leveraging the capabilities of ELM to handle nonlinear relationships and complex project dynamics, our proposed system offers several key advantages over traditional estimation methods. Firstly, the ELM-based approach yields more accurate and robust effort estimates compared to traditional methods, leading to improved project planning, resource allocation, and decision-making. By capturing nonlinear relationships and interactions among project attributes, the model provides nuanced insights into effort requirements, enabling stakeholders to make informed decisions and mitigate risks effectively. Secondly, the scalability and efficiency of the ELM-based approach make it suitable for handling large-scale software projects with diverse characteristics. The streamlined training process and minimal computational overhead allow for real-time estimation tasks and dynamic adaptation to evolving project requirements, enhancing productivity and agility in software development endeavors. Furthermore, the transparency and interpretability of the ELM-based effort estimation model foster trust and confidence among stakeholders, enabling them to understand the factors influencing effort estimates and the rationale behind the predictions. This transparency facilitates collaboration, communication, and alignment across project teams, leading to more effective coordination and synergy in project execution. Moreover, the adaptability and compatibility of the ELM-based approach with various software development methodologies, including Agile and DevOps, ensure its relevance and applicability across different project contexts. By accommodating diverse project dynamics and requirements, the model supports iterative planning, continuous feedback, and adaptive decision-making, thereby enhancing project success and customer satisfaction.

REFERENCES

1. Boehm, B. W. (1981). Software engineering economics. Prentice-Hall, Inc.

2. Kitchenham, B., Mendes, E., & Travassos, G. H. (2007). Cross versus within-company cost estimation studies: A systematic review. IEEE Transactions on Software Engineering, 33(5), 316-329.

3. Jørgensen, M., & Shepperd, M. (2007). A systematic review of software development cost estimation studies. IEEE Transactions on Software Engineering, 33(1), 33-53.

4. Briand, L. C., & Wieczorek, I. (2002). Resource estimation in software engineering. Springer.

5. Choudhary, S. R., & Singh, Y. K. (2015). Software effort estimation: A review. International Journal of Computer Applications, 125(11), 1-7.

6. Tan, S. H., Teh, E. S., & Salleh, N. (2010). Software effort estimation: A survey and some refinements. International Journal of Software Engineering and its Applications, 4(3), 55-76.

7. Symons, C. R., & Kulikowski, C. A. (1983). The scope of software estimation research: A survey of the literature. IEEE Transactions on Software Engineering, (2), 140-146.

8. Al-Qutaish, R. M., & Abdullah, R. M. (2011). A review of software effort estimation based on soft computing approaches. Journal of Computing, 3(3), 53-61.

9. Mendes, E., & Mosley, N. (2015). Can effort estimation of software projects benefit from using test case points? IEEE Transactions on Software Engineering, 41(6), 593-608.

10. Kocaguneli, E., Menzies, T., & Bener, A. (2009). Exploiting the essential assumptions of analogy-based effort estimation. IEEE Transactions on Software Engineering, 35(3), 321-331.

11. Li, M., Ma, P., & Peng, X. (2016). A review of software effort estimation based on machine learning. In 2016 9th International Symposium on Computational Intelligence and Design (ISCID) (pp. 17-20). IEEE.

12. Li, Q., & Boehm, B. (2003). Predicting software development productivity of a large-scale project using cocomo ii and extreme learning machine. In International Workshop on Software Measurement (pp. 252-265). Springer, Berlin, Heidelberg.

13. Melo, W. L., & Munoz, S. R. (2005). Using extreme learning machine to software effort estimation. In 2005 International Conference on Neural Networks and Brain, 2005 (pp. 812-817). IEEE.

14. Tian, J., Huang, L., & Xu, H. (2009). The application of improved extreme learning machine in software effort estimation. In 2009 International Conference on Computational Intelligence and Security (pp. 452-456). IEEE.

15. Kim, S., Ryu, K., & Kim, S. (2011). A novel software effort estimation model based on extreme learning machine with adaptive PSO. In 2011 International Conference on Computer Science and Network Technology (pp. 2-5). IEEE.