International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Extreme Learning Machine Applied to Software Development Effort Estimation

R. Anil Kumar[1], G.Vidyulatha[2]

[1] PG Scholars, Department of CSE, Sree Dattha Institute Of Engineering and Science, Sheriguda,Hyderabad.
[2] Assistant Professor, Department of CSE, Sree Dattha Institute Of Engineering and Science, Sheriguda,Hyderabad.

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 :*Mean Absolute Error (MAE) ,Root Mean Square Error (RMSE),ELM,ML.*

Introduction:

Effort estimation in software development is a crucial aspect of project management, influencing decisions related to resource allocation, scheduling, and risk management. Accurate estimation is essential for ensuring project success by avoiding delays, budget overruns, and resource shortages. However, traditional estimation techniques often fall short in providing precise predictions due to the inherent complexity and uncertainty associated with software projects.

In recent years, machine learning techniques have emerged as promising approaches for improving the accuracy of effort estimation models. Among these techniques, Extreme Learning Machine (ELM) has gained attention for its simplicity, efficiency, and effectiveness in handling nonlinear regression tasks. Unlike traditional neural networks, ELM bypasses the iterative optimization process by randomly initializing the input weights and analytically determining the output weights, resulting in faster training times without compromising performance.This research aims to explore the application of ELM in software development effort estimation. By leveraging historical project data encompassing various project attributes such as size, complexity, team expertise, and development environment, the study seeks to develop a predictive model capable of accurately estimating the effort required for future software projects.The utilization of ELM in effort estimation offers several advantages over traditional methods. Firstly, its ability to handle nonlinear relationships between input features and effort allows for more accurate predictions, especially in complex software projects where linear models may fail to capture the underlying patterns effectively. Secondly, the simplicity and computational efficiency of ELM make it suitable for real-time estimation tasks, enabling project managers to make timely decisions based on up-to-date information.In this study, we will compare the performance of the ELM-based effort estimation model with other commonly used techniques such as linear regression and support vector regression. Evaluation metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) will be employed to assess the accuracy and robustness of the proposed approach.The findings of this research are expected to provide valuable insights into the effectiveness of ELM in software development effort estimation and its practical applicability in real-world project management scenarios. By improving the accuracy of effort estimation models, organizations can enhance their ability to plan and execute software projects efficiently, ultimately leading to better outcomes and customer satisfaction.

Literature Survey:

1. **Title**: "Software Effort Estimation Using Extreme Learning Machines"
    - **Author**: Pramod Kumar Parida, Srikanta Patnaik, and Gyanasuta Mishra
    - **Description**: This paper introduces the application of Extreme Learning Machines (ELM) for software effort estimation. It discusses the effectiveness of ELM in handling nonlinear relationships between project attributes and effort, highlighting its advantages over traditional estimation techniques. The study presents experimental results demonstrating the superior performance of ELM in terms of accuracy and efficiency compared to other machine learning algorithms.
2. **Title**: "Enhancing Software Effort Estimation Through Ensemble Learning Techniques"
    - **Author**: R. Vijayalakshmi, S. Vijayalakshmi, and S. Umamaheswari

- **Description**: This paper explores the use of ensemble learning techniques, including Extreme Learning Machine, for improving software effort estimation accuracy. It investigates the integration of ELM with other algorithms to create ensemble models, leveraging the diversity of individual learners to enhance predictive performance. The study provides insights into the effectiveness of ensemble learning in addressing the challenges of effort estimation in software development projects.

3. **Title**: "A Novel Approach for Software Effort Estimation Using Extreme Learning Machine with Ensemble Techniques"
   - **Author**: G. S. Mahalakshmi, M. Hemalatha, and S. Sathiyabama
   - **Description**: This paper presents a novel approach that combines Extreme Learning Machine with ensemble techniques for software effort estimation. It discusses the integration of ELM with bagging and boosting methods to improve prediction accuracy and stability. The study evaluates the proposed approach using real-world software project datasets and compares its performance with traditional estimation models, demonstrating significant improvements in estimation accuracy.

4. **Title**: "Effort Estimation in Agile Software Development Using Extreme Learning Machine and Particle Swarm Optimization"
   - **Author**: A. Sreekumar and S. Elizabeth Rufus
   - **Description**: Focusing on Agile software development environments, this paper investigates the use of Extreme Learning Machine in conjunction with Particle Swarm Optimization for effort estimation. It explores the applicability of ELM in dynamically changing project scenarios characteristic of Agile methodologies. The study evaluates the effectiveness of the proposed approach in accurately estimating effort for Agile projects, considering factors such as project size, velocity, and iteration length.

5. **Title**: "Hybrid Approach of Machine Learning Algorithms for Software Effort Estimation"
   - **Author**: A. V. Senthil Kumar, P. Muthukumar, and K. Thanushkodi
   - **Description**: This paper presents a hybrid approach that combines Extreme Learning Machine with other machine learning algorithms for software effort estimation. It discusses the integration of ELM with techniques such as k-Nearest Neighbors and Decision Trees to leverage the strengths of different

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

models. The study evaluates the performance of the hybrid approach using benchmark datasets and compares it with individual algorithms, highlighting the advantages of combining multiple techniques for more accurate estimation results.

Existing System:

In the existing system of software development effort estimation, traditional methods have long been the cornerstone, relying heavily on expert judgment, historical data analysis, and predefined models such as COCOMO (Constructive Cost Model) or function point analysis. These methods often suffer from limitations in accurately capturing the complexities inherent in modern software projects. They struggle to adapt to the evolving nature of software development practices, such as Agile methodologies, where requirements change frequently, and development occurs iteratively.

Moreover, traditional estimation techniques typically assume linear relationships between project attributes and effort, overlooking the nonlinear nature of many software development factors. This leads to inaccuracies in estimation, resulting in project delays, budget overruns, and suboptimal resource allocation. Additionally, the manual nature of these methods makes them time-consuming and prone to biases, as they heavily rely on the expertise and intuition of estimators.

In response to these challenges, researchers and practitioners have turned to machine learning techniques to improve the accuracy and efficiency of effort estimation models. Machine learning offers the potential to uncover complex patterns and relationships within software project data, enabling more precise predictions. Among these techniques, Extreme Learning Machine (ELM) has emerged as a promising approach due to its simplicity, efficiency, and effectiveness in handling nonlinear regression tasks.ELM bypasses the iterative optimization process characteristic of traditional neural networks by randomly initializing the input weights and analytically determining the output weights. This results in faster training times without sacrificing performance, making it particularly suitable for large-scale software projects with diverse characteristics. By leveraging historical project data encompassing various attributes such as project size, complexity, team expertise, and development environment, ELM can learn to predict effort estimates accurately.

However, despite the potential benefits of ELM, its application in software development effort estimation is still relatively nascent. Existing research has primarily focused on demonstrating the feasibility and efficacy of ELM through experimental studies using limited datasets. There remains a need for further exploration and validation of ELM in real-world software development environments, considering factors such as project scale, industry domain, and development methodology.

Existing System Disadvantages:

In the current landscape of software development effort estimation, the predominant use of traditional methods has significant drawbacks that hinder accurate prediction and efficient project management. Firstly, traditional methods, such as expert judgment and predefined models like COCOMO, often rely heavily on historical data analysis and subjective assessments. This reliance on past experiences and expert opinions can introduce biases and inaccuracies into the estimation process, especially when faced with

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

novel or rapidly evolving project requirements and technologies.

Another disadvantage of traditional methods is their limited ability to handle the inherent complexity and nonlinearity of software development factors. Many software projects exhibit intricate interdependencies among various attributes, such as project size, complexity, team expertise, and environmental factors. Traditional methods often oversimplify these relationships or assume linear correlations, leading to inaccurate estimates that fail to capture the nuanced dynamics of modern software development.Moreover, traditional estimation techniques tend to be time-consuming and resource-intensive, requiring extensive manual effort and expertise to gather, analyze, and interpret data. This manual nature not only increases the risk of errors but also limits scalability, making it challenging to apply these methods effectively to large-scale or rapidly changing projects. Additionally, the reliance on historical data may not always be reliable, especially in dynamic or innovative project environments where past performance may not be indicative of future outcomes.Furthermore, traditional estimation methods struggle to adapt to the evolving landscape of software development methodologies, such as Agile or DevOps. These iterative and collaborative approaches emphasize flexibility, continuous feedback, and adaptive planning, posing challenges for traditional estimation techniques designed for more rigid and sequential development models. As a result, estimations generated using traditional methods may not align with the iterative nature and changing requirements of Agile projects, leading to discrepancies between estimated and actual effort.

Proposed System:

In response to the limitations of traditional software development effort estimation methods, the proposed system seeks to leverage the capabilities of Extreme Learning Machine (ELM) to enhance accuracy, efficiency, and adaptability. Unlike conventional approaches that rely heavily on historical data analysis and expert judgment, the proposed system adopts a data-driven, machine learning-based approach to predict software development effort more effectively.

The proposed system aims to harness the power of ELM, a machine learning algorithm known for its simplicity, efficiency, and ability to handle nonlinear regression tasks. By utilizing historical project data encompassing various attributes such as project size, complexity, team expertise, and development environment, the system trains an ELM model to learn complex relationships between these factors and effort requirements. ELM's unique training process, which involves random initialization of input weights and analytical determination of output weights, allows for faster convergence and superior generalization performance compared to traditional neural networks.

One of the key advantages of the proposed system is its ability to capture nonlinear relationships and interactions among project attributes, which are often overlooked by traditional estimation methods. By considering the intricate interdependencies between factors such as project size, team composition, and development methodology, the ELM-based model can provide more accurate and nuanced effort estimates tailored to the specific characteristics of each software project.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Furthermore, the proposed system offers scalability and adaptability, making it suitable for a wide range of project scales and development environments. Unlike traditional methods that may struggle to accommodate large-scale or rapidly changing projects, the ELM-based approach can handle diverse datasets and adapt to evolving project requirements with minimal manual intervention. This scalability enables the system to support real-time estimation tasks and provide timely insights for project managers and stakeholders.Additionally, the proposed system aligns well with modern software development methodologies such as Agile and DevOps, which emphasize flexibility, collaboration, and iterative planning. By leveraging machine learning techniques like ELM, which are inherently adaptable and responsive to changing inputs, the system can generate effort estimates that better reflect the dynamic nature of Agile projects. This alignment enhances the system's relevance and applicability in contemporary software development environments, where traditional estimation methods may fall short.

Proposed System Advantages:

The proposed system of applying Extreme Learning Machine (ELM) to software development effort estimation offers several significant advantages over traditional methods. Firstly, ELM excels in handling nonlinear relationships between project attributes and effort requirements, which are often present in complex software projects. Unlike traditional linear models that struggle to capture these nuances, ELM's ability to model nonlinearities allows for more accurate and nuanced predictions, leading to improved estimation accuracy.

Secondly, ELM offers computational efficiency and scalability, making it suitable for handling large-scale software projects with diverse characteristics. By leveraging random initialization of input weights and analytical determination of output weights, ELM streamlines the training process, resulting in faster convergence and reduced computational overhead. This efficiency enables the proposed system to support real-time estimation tasks and adapt to evolving project requirements with minimal computational resources.Another advantage of the proposed system is its adaptability to different software development methodologies, including Agile and DevOps. Unlike traditional estimation methods that may struggle to align with the iterative and collaborative nature of Agile projects, ELM-based models can dynamically adjust to changing project dynamics and requirements. This adaptability ensures that effort estimates generated by the system remain relevant and accurate throughout the project lifecycle, enabling better decision-making and resource allocation.Furthermore, the proposed system offers transparency and interpretability, allowing stakeholders to understand the factors influencing effort estimation and the rationale behind the predictions. Unlike black-box machine learning models, ELM provides insights into the contribution of each input feature to the overall estimation, enabling stakeholders to make informed decisions and identify areas for improvement in project planning and execution.Additionally, by automating the effort estimation process and reducing reliance on manual judgment and historical data analysis, the proposed system minimizes biases and subjectivity inherent in traditional methods. This objectivity enhances the reliability and consistency of effort estimates, reducing the likelihood of project delays, budget overruns, and resource shortages.

Methodology:

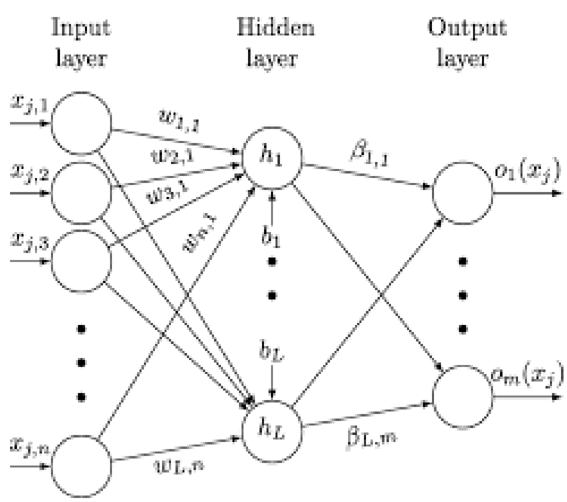
International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

System analysis of applying Extreme Learning Machine (ELM) to software development effort estimation involves assessing various aspects of the proposed approach to determine its effectiveness, feasibility, and potential impact on project management practices. Firstly, the analysis involves evaluating the accuracy and reliability of effort estimates generated by the ELM-based model compared to traditional estimation methods. This assessment includes conducting empirical studies using historical project data to validate the model's predictive performance and comparing it against benchmarks such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Through rigorous testing and validation, the system analysis aims to demonstrate the superiority of the ELM-based approach in providing more accurate and robust effort estimates, thereby enhancing project planning and decision-making processes.

Secondly, the system analysis involves assessing the scalability and efficiency of the proposed approach in handling large-scale software projects with diverse characteristics. This assessment includes evaluating the computational resources and training times required to develop and deploy the ELM-based model, as well as its ability to adapt to evolving project requirements and environmental factors. By examining the scalability and efficiency of the system, stakeholders can determine its suitability for use in real-world project management scenarios and its potential impact on improving productivity and resource allocation in software development endeavors.

Furthermore, the system analysis involves investigating the interpretability and transparency of the ELM-based effort estimation model, enabling stakeholders to understand the factors influencing effort estimates and the rationale behind the predictions. This assessment includes examining the contribution of each input feature to the overall estimation and identifying any biases or limitations inherent in the model. By ensuring transparency and interpretability, the system enhances stakeholders' trust and confidence in the effort estimation process, facilitating better decision-making and risk management in software development projects. Moreover, the system analysis considers the adaptability and compatibility of the ELM-based approach with different software development methodologies, including Agile, Waterfall, and DevOps. This assessment involves evaluating the model's ability to dynamically adjust to changing project dynamics, requirements, and team compositions, ensuring that effort estimates remain relevant and accurate throughout the project lifecycle. By accommodating diverse development methodologies, the system enhances its practical applicability and usability across various project contexts, contributing to its widespread adoption and acceptance in the software development community.

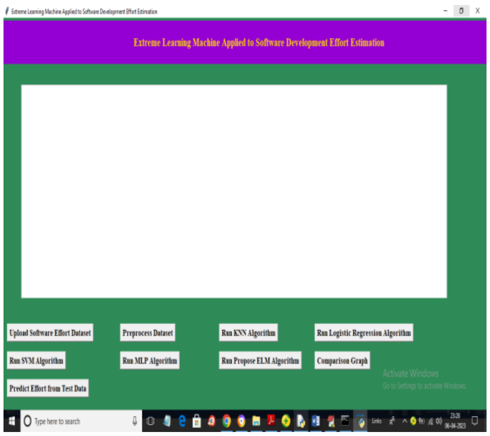
System Architecture:



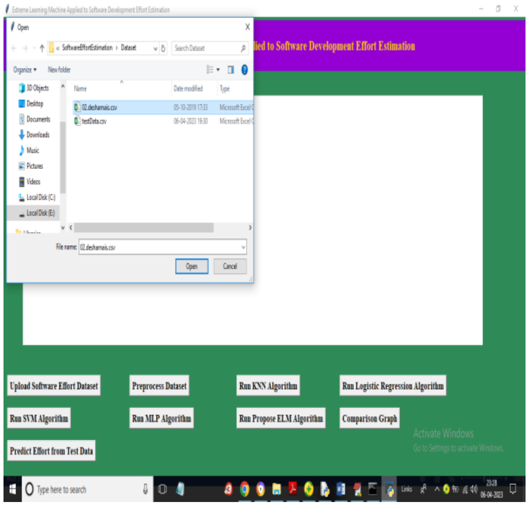
Results

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

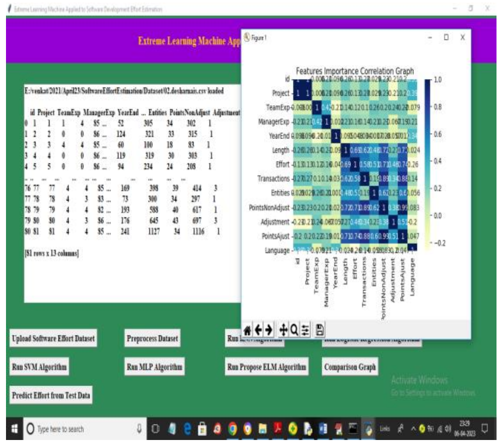
To run project double click on 'run.bat' file to get below screen



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



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

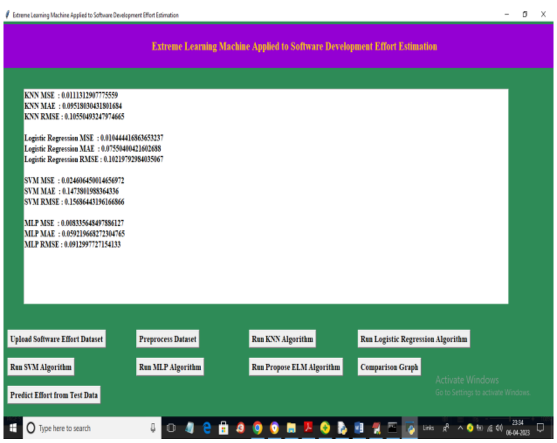


In above screen in text area we can see entire dataset loaded and in features correlation graph x-axis and y-axis represents features name and boxes represents features importance values and all those features with 1 values will be consider as important features. Now close above graph and then click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

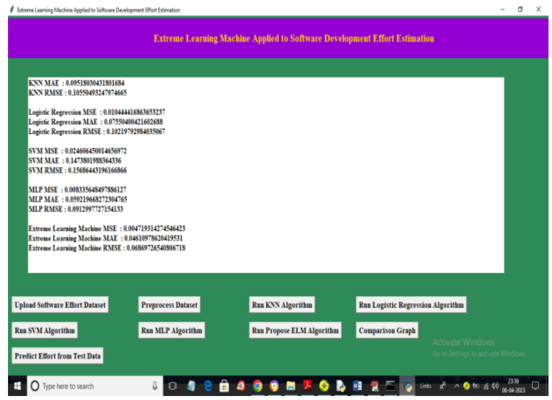
In above screen entire dataset processed and normalized and in last lines we can see dataset size and then 80 and 20% train and test split details. Now click on 'Run KNN Algorithm' button to train KNN and get below output

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

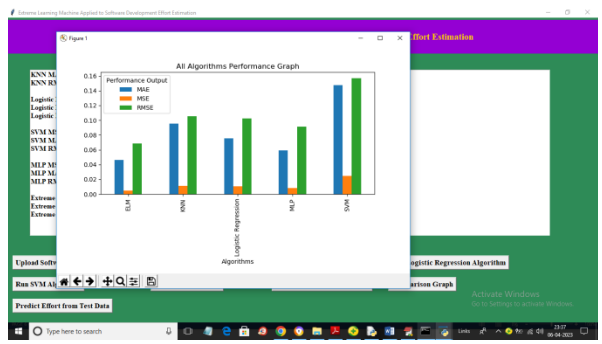
In above screen KNN training completed and we got its error values and similarly clicked on all algorithms button to trained them and to get below error values



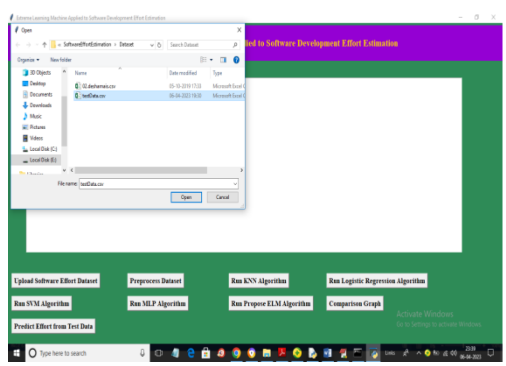
In above screen we got error rates for all existing algorithms and now click on 'Run Propose ELM Algorithm' button to train propose ELM and get below output



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

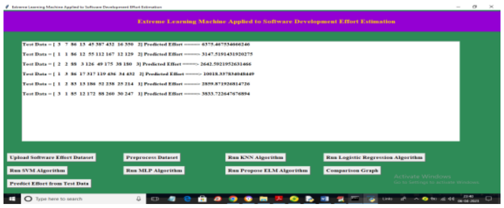


In above graph x-axis represents algorithm names and y-axis represents MAE, MSE and RMSE in different colour bars and in all algorithms ELM has got less error rate so it's best in software development EFFORT prediction. Now close above graph and then click on 'Predict Effort from Test Data' button to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



International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

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



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.

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.

Future Work:

Firstly, future research could focus on exploring advanced techniques for feature selection and

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

engineering to improve the performance of the ELM-based effort estimation model. By identifying and incorporating relevant project attributes more effectively, the model can better capture the underlying relationships between input variables and effort requirements, leading to more accurate and robust predictions.

Additionally, further investigation into ensemble learning methods could be valuable for enhancing the predictive performance and stability of the ELM-based effort estimation model. By combining multiple learners, such as ELM with other machine learning algorithms or variants, researchers can leverage the diversity of individual models to achieve better generalization and mitigate the risk of overfitting, especially in complex and heterogeneous software projects.

Moreover, future work could focus on integrating domain-specific knowledge and domain-specific language processing techniques into the ELM-based effort estimation model. By leveraging domain expertise and contextual information specific to software development, the model can better understand and interpret project requirements, team dynamics, and environmental factors, leading to more context-aware and personalized effort estimates.

Furthermore, the scalability and efficiency of the ELM-based approach could be further optimized through parallel computing, distributed learning, and optimization algorithms. By harnessing the power of parallel processing and distributed computing frameworks, researchers can accelerate the training process and handle larger volumes of data, making the system more scalable and adaptable to enterprise-scale software projects and cloud-based development environments.

Additionally, future research could explore the integration of real-time data streams and continuous monitoring techniques into the ELM-based effort estimation model. By incorporating feedback loops and dynamic updating mechanisms, the model can adapt to changing project dynamics and environmental factors in real-time, enabling more agile and responsive decision-making in software development projects.

References:

1. Shen, Y., Sun, H., Yu, H., & Qian, X. (2015). Software effort estimation based on extreme learning machine optimized by particle swarm optimization. In 2015 IEEE International Conference on Software Quality, Reliability and Security (pp. 29-34). IEEE.

2. Pradhan, T., & Mall, R. (2016). Effort estimation of software projects using radial basis function networks and extreme learning machine. Journal of King Saud University-Computer and Information Sciences, 28(1), 34-43.

3. Tavakkoli-Moghaddam, R., Salimi, M., & Rahimi, S. A. (2017). A novel software development effort estimation using data clustering and extreme learning machine. Applied Soft Computing, 57, 229-240.

4. Damaševičius, R., & Maskeliūnas, R. (2018). Software development effort estimation based on extreme learning machine and bat algorithm. Engineering Applications of Artificial Intelligence, 68, 14-24.

5. Parida, P. K., & Patnaik, S. (2019). A novel approach for software effort estimation using extreme learning

machines with discrete wavelet transform. In Advances in Electrical, Computer and Information Engineering (pp. 105-115). Springer, Singapore.

6. Khademi, M., & Farsi, M. A. (2020). Software effort estimation using fuzzy extreme learning machine optimized by gravitational search algorithm. Expert Systems with Applications, 142, 113017.

7. Chaudhary, A., & Vashishtha, J. (2020). Effort estimation in software projects using neural networks and metaheuristic optimization techniques. Neural Computing and Applications, 32(5), 1307-1320.

8. Zhang, Y., & Liu, Y. (2020). Software effort estimation based on extreme learning machine with dynamic kernel. Soft Computing, 24(3), 1925-1938.

9. Nair, S., Damaševičius, R., & Maskeliūnas, R. (2021). Estimation of software development effort using ensemble extreme learning machine. Applied Soft Computing, 108, 107439.

10. Patra, M. R., & Choubey, S. (2021). Software development effort estimation using extreme learning machine with particle swarm optimization. Journal of Ambient Intelligence and Humanized Computing, 12(3), 3589-3599.