JANOLI International Journal of
Computer Science And Engineering (JIJCSE)
(A Peer Review Journal)
journal homepage: https://janolijournals.com/index.php/jijcse

Volume 1, Issue 1, 2025 ISSN:3107-4537

Evaluating Machine Intelligence Techniques for Software Effort Estimation with Failure Patterns and Time Constraints

Narendra Kumar
NIET NIMS University, Jaipur, India

ARTICLE INFO

Article History:
Received December 15, 2024
Revised December 30, 2024
Accepted January 12, 2025
Available online January 25, 2025

Keywords:
Software Effort Estimation
Hierarchical Temporal Memory (HTM)
Auditory Machine Intelligence (AMI)
Failure Patterns
Time Constraints
Optimization Techniques
Machine Intelligence.

Correspondence:
E-mail: drnk.cse@gmail.com

ABSTRACT

Software effort estimation plays a critical role in project management, especially in the face of failure patterns and time constraints that complicate predictive accuracy. This study evaluates the application of machine intelligence techniques, specifically Hierarchical Temporal Memory (HTM) and Auditory Machine Intelligence (AMI), to improve estimation accuracy. Employing a quantitative methodology, the research investigates five core aspects: comparative effectiveness of HTM and AMI, the impact of failure patterns on estimation accuracy, the influence of time constraints on technique performance, complexity trade-offs, and optimization strategies for real-time applications.

Findings reveal HTM's superior accuracy compared to AMI, though its complexity poses challenges for real-time usability. Failure patterns significantly influence estimation accuracy, emphasizing the need for adaptive models. Time constraints were found to affect the performance of both techniques, with HTM displaying greater sensitivity. Optimization techniques demonstrated the potential to mitigate complexity and enhance real-time applicability.

This research contributes to the evolving field of software effort estimation by providing a nuanced understanding of the interplay between machine intelligence, failure patterns, and time dynamics. While limitations include reliance on historical datasets and restricted real-time validation, future research can address these gaps by exploring diverse datasets and advanced optimization techniques, offering practical implications for more accurate and efficient software project planning.

1.Introduction

This chapter will introduce the use of machine intelligence techniques for the estimation of effort in software, particularly under the theme of failure patterns and time bounds in simulation settings. The key research question aims to determine if two neural machine intelligence techniques, namely Hierarchical Temporal Memory (HTM) and Auditory Machine Intelligence (AMI), will help improve the accuracy of the estimation of software effort. The five sub-research questions involve: How well do HTM and AMI compare in the aspect of estimation accuracy? To what extent does a failure pattern have an effect on estimation accuracy? What is the effect of the time constraint upon the performance of HTM and AMI? What would be the complexity trade-off in HTM and AMI? How to optimize these techniques for real-time application? The study adopts a quantitative approach and centres on the interaction between machine intelligence techniques (independent variables) and estimation accuracy (dependent variable), with control variables being failure patterns and time constraints. The article is set up to shift from literature review to methodology, results, and conclusion, analysing in a systematic manner how machine intelligence can be applied to improve software effort estimation.

2.Literature Review

This section critically reviews existing research on machine intelligence applications in software effort estimation, structured around the five sub-research questions. It explores the comparative effectiveness of HTM and AMI, the influence of failure patterns and time constraints, the complexity trade-offs, and optimization strategies for real-time applications. This section also addresses gaps in current research, such as the limited exploration of time boundaries and the need for comprehensive complexity assessments. The paper outlines five hypotheses that are based on the interrelation of the variables.

2.1 Comparative Effectiveness of HTM and AMI

Initial studies had suggested that both HTM and AMI could enhance estimation accuracy; however, such basic comparative studies did not gain further momentum. Subsequent studies have provided a detailed comparison but fail to assess applicability in the real world. Current researches attempt to fill this gap with simulation-based comparisons; however, it still raises questions of scalability and robustness. Hypothesis 1: HTM outperforms AMI in achieving better accuracy for estimating software effort is proposed.

2.2 Impact of Failure Patterns on Estimation Accuracy

The first investigations explored the use of failure patterns in estimation and utilized static models for most cases. Intermediate investigations focused on dynamic analysis that produced some relationships but failed to generate sufficient real-time details. Current investigations elaborate these relationships through detailed simulation models but comprehensive relationships among a wide variety of patterns remain unobserved. Hypothesis 2: The existence of significant interactions between failure patterns and effort estimation model accuracy is advanced.

2.3 Influence of Time Constraints on Technique Performance

Initial studies about time constraints are based on theoretical issues. Subsequent studies incorporated the time factor in models and it was found that performances were different for different models but not robust enough. Recent attempts use sophisticated simulation to measure such effects but miss the real picture of real-time applications. Hypothesis 3: Time constraints considerably affect the performance of HTM and AMI in the estimation of effort in software is presented.

2.4 Trade-offs between complexity of HTM and AMI

Early works focused on the complexity of HTM and AMI by analysing them case-by-case. Mid-term research had added the aspect of computational cost analyses to demonstrate the trade-off between these, yet there is a lack of general metrics for all analyses. Current works will attempt to detail complexity, yet it has the problems with real-time application. Hypothesis 4: HTM complexity could hinder its use in real time over AMI.

2.5 Optimization towards Real-Time Application

The initial focus of optimization was given to theoretical concepts. Later work considered practical approaches toward estimation, while preliminary results lack real-time experimentations. Simulations are found in recent research towards optimizations, while the complete test with real time validation is awaited. Hypothesis 5. Optimization techniques increase the real time usability of HTM and AMI in Software effort estimation will be proposed through the methodology

This section describes the quantitative research approach used to test the hypotheses, detailing the data sources, variables, and statistical methods applied. The methodology ensures accurate and reliable findings, providing insights into the application of HTM and AMI for software effort estimation.

3.1 Data

The data for this study were obtained from a publicly available software failure dataset. It was collected based on comprehensive assessments of software projects over a defined period. The dataset included variables such as failure patterns, time constraints, and performance metrics. Sampling was conducted to ensure diverse representation across project types and sizes. The criteria for sample selection included projects with varied complexity levels and failure occurrences, ensuring robust analysis of estimation techniques under different conditions.

3.2 Variables

Independent variables of the study are the machine intelligence techniques HTM and AMI, and the dependent variable is the accuracy of software effort estimation. The failure patterns and time constraints are considered as control variables since these are significant to isolate the effect of the techniques on estimation accuracy. Other variables that have been taken into account for the further fine-tuning of the analysis are project size and complexity. All the measurement methods for such variables are supported by literature to be reliable. The regression analysis was further used in this study to examine relationships between different variables, based on testing the hypotheses.

4.Results

The results section starts with reporting of descriptive statistics of the dataset. Descriptive statistics are reported for independent variables (HTM and AMI), dependent variables (estimation accuracy), and for control variables including failure patterns, and time constraints. Regression analyses validate the hypotheses, which affirm HTM's superior accuracy, the effect of failure patterns and time constraints, and the trade-offs in complexity. Findings indicate the need for optimization strategies for real-time application, thereby filling the gap in the present research and showcasing the applicability of machine intelligence techniques in the estimation of software effort.

4.1 HTM's Superior Accuracy in Effort Estimation

This finding supports Hypothesis 1, which demonstrates the superior accuracy of HTM as compared to AMI in software effort estimation. Through analysis of the dataset, HTM will be seen to have a constant superiority over AMI in the estimation error. The independent variables are all the applications of HTM and AMI while concentrating on the dependent variable as the estimation accuracy metrics. The correlation presents the view that the high-order pattern recognition of HTM explains its performance advantage. Empirical significance The value of HTM over improvement in estimation accuracy is suggested by empirical significance, which also aligns with theories of neural intelligence in software engineering. By filling gaps in previous comparative analyses, this finding points out the promise of HTM for bettering software effort estimation.

4.2 Influence of Failure Patterns on Estimation Models

This finding supports Hypothesis 2, pointing to the significant influence of failure patterns on software effort estimation models. Data analysis reveals that varying failure patterns lead to changes in estimation accuracy, with certain patterns posing greater challenges for both HTM and AMI. Key independent variables include failure pattern characteristics, while dependent variables focus on estimation accuracy. This relationship emphasizes the need for models to account for diverse failure scenarios to enhance accuracy. The empirical implication is that knowing the failure patterns is important for the development of estimation models that are robust, consistent with the theories of software reliability engineering. The identification of gaps in the analysis of failure patterns indicates the need for the inclusion of different patterns in the estimation models.

4.3 Impact of Time Constraints on Performance

This finding supports Hypothesis 3, which proves that time constraints have a very significant impact on the performance of HTM and AMI in software effort estimation. The analysis of the

dataset reveals that time constraints cause variations in the accuracy of estimation, with HTM showing more sensitivity than AMI. Independent variables are the levels of time constraints, and dependent variables are performance metrics. This relationship calls for models to take into account time dynamics in order to be accurate. Empirical significance The findings suggest that time constraints are one of the most critical factors in software effort estimation, according to the theory of time-sensitive computing. Since there are significant gaps in time constraint analysis, this finding places emphasis on how models should accommodate temporal variations.

4.4 Complexity trade-offs between HTM and AMI

This result confirms Hypothesis 4; the complexity of HTM restricts its usage in real-time applications more so than AMI. Data analysis reveals that HTM's complexity leads to increased computational costs and potential delays in real-time applications. Key independent variables include complexity metrics, while dependent variables focus on real-time performance indicators. This relationship emphasizes the need for balancing complexity and performance in model selection. The empirical significance suggests that complexity trade-offs are a critical consideration in real-time applications, aligning with theories of computational efficiency. By showing gaps in complexity analysis, this finding emphasizes the need to analyse trade-offs for real-time applicability.

4.5 Optimization Techniques for Real-time Application

This finding supports Hypothesis 5 by stating that optimization techniques can increase the real-time applicability of HTM and AMI for software effort estimation. In the analysis of the dataset, it is observed that optimization techniques result in improved performance and decreased complexity in real-time applications. Optimization techniques are the most important independent variables, whereas real-time performance metrics depend on the models. It is evident that optimization is valuable for increasing applicability from this relationship. Empirical significance indicates that targeted optimization can solve real-time challenges according to software optimization theories. It underlines the fact that adjustment of gaps in the strategies of optimization could help in improving real-time applicability with suitable techniques.

5.**Conclusion**

This study integrates findings on the application of machine intelligence techniques for software effort estimation, highlighting their effectiveness in improving accuracy, addressing failure patterns, managing time constraints, and optimizing real-time applicability. The research reveals HTM's superior performance, the significant impact of failure patterns and time constraints, and the importance of optimization strategies. However, limitations include reliance on historical data and challenges in real-time validation. Thus, future research will point toward diverse datasets and advanced optimization techniques to enhance understanding and applicability. This approach will help refine estimation models and strategies, which will have positive consequences for better software effort estimation in different situations.

References

[1] Boehm, B. W. (1981). *Software Engineering Economics*. Prentice Hall.

[2] Jørgensen, M., & Shepperd, M. (2007). A systematic review of software development cost estimation studies. *IEEE Transactions on Software Engineering*, 33(1), 33–53.

[3] Hawkins, J., & Blakeslee, S. (2004). *On Intelligence*. Times Books.

[4] George, D., & Hawkins, J. (2009). Towards a mathematical theory of cortical microcircuits. *PLoS Computational Biology*, 5(10), e1000532.

[5] Koomey, J. G. (2011). Growth in data center electricity use 2005 to 2010. *Analytics Press*

[6] Anuj Kumar, Shilpi Srivastav, Narendra Kumar and Alok Agarwal "Dynamic Frequency Hopping: A Major Boon towards Performance Improvisation of a GSM Mobile Network" International Journal of Computer Trends and Technology, vol 3(5) pp 677-684, 2012. (Scopus indexed)

[7] Anuj Kumar, Narendra Kumar and Alok Aggrawal: "Estimation of Blocking Probabilities in a Cellular Network Which Is Prone to Dynamic Losses" International Journal of Computer Trends and Technology, vol 3(5) pp 733-740, 2012.

[8] B. Srinivas, Narendra Kumar and Alok Aggrawal: "Finding Vulnerabilities in Rich Internet Applications (Flex/AS3) Using Static Techniques" International Journal of Modern Education and Computer Science, 4(1), pp 33-39, 2012

[9] Narendra Kumar and Alok Aggrawal: "Soft hand off in Mobile Network" International Journal of Engineering Trends and Technology, 3(2), 239-242, 2012.

[10] S. R. Basavala, N. Kumar and A. Agarrwal, "Authentication: An overview, its types and integration with web and mobile applications," 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, 2012, pp. 398-401, doi: 10.1109/PDGC.2012.6449853.

[11] Chandrasekaran, B., & Conrad, J. M. (2018). Optimization in machine learning. *Journal of Applied Machine Learning*, 7(3), 35–49.