IITM Journal of Management and IT
Vol. No. 15 Issue No. 1
June 2024

Agile Effort Estimation Using Machine Learning – A Systematic Review

PRABHNEET KAUR*
ANKUR SHARMA**

This review paper examines the integration of machine learning techniques into Agile software development, primarily focusing on effort estimation. It evaluates existing methodologies for Effort Prediction (EP) in Agile Software Development (ASD) projects, emphasizing the Evolutionary Cost-Sensitive Deep Belief Network (ECS-DBN) model's ability to predict task effort during the early stages of Agile projects. The model's efficacy is assessed using real-world data from 160 tasks in Agile projects. Furthermore, the paper explores the applications of machine learning in various project management aspects within Scrum, such as sprint planning, backlog prioritization, and team performance prediction, as well as within Kanban, including workflow visualization, workload balancing, and lead time prediction. Emphasis is placed on the significance of data quality, algorithm selection, and the need for explainable AI. The paper concludes with a review of studies on software effort estimation in agile methodologies, highlighting the importance of machine learning algorithms in optimizing estimation formulas. Suggestions for future research include exploring additional metrics and applying machine learning techniques to industrial projects.

Keywords: *Machine Learning Techniques, Agile Software Development, Effort Estimation, Velocity, Story Points, Effort Prediction (EP), Evolutionary Cost-Sensitive Deep Belief Network (ECS-DBN), Real-World Data, Sprint Planning, Backlog Prioritization, Team Performance Prediction, Scrum, Workflow Visualization, Workload Balancing, Lead Time Prediction, Kanban, Data Quality, Algorithm Choice, Explainable AI, Swarm Optimization Algorithms, Additional Metrics, Industrial Projects*

* Assistant Professor, IITM prabhneet.it@iitmipu.ac.
** Student, GTBIT, ankur.as502@gmail.com.

0. INTRODUCTION

In the rapidly evolving field of software development, this review synthesizes various elements to explore key topics like effort estimation, regression testing, the integration of machine learning, and Agile methodologies. As the software development paradigm continuously evolves, the paper strategically places emphasis on Agile methodologies, propelling the discourse forward by advocating for the indispensable nature of accurate effort estimation within Agile projects. In addressing this critical need, the paper introduces an innovative formula that utilizes velocity and story points, aimed at transforming and improving the complexities involved in agile software effort estimation. A substantial section of the paper focuses on investigating the relationship between machine learning methodologies and Agile software development.. Within this narrative, the authors articulate a compelling vision, advocating for the utilization of real-time data, domain-specific models, continuous learning mechanisms, and collaborative artificial intelligence to transcend traditional project management and decision-making paradigms.

Delving into the detailed facets of the software development process, the paper meticulously categorizes and explores different dimensions. It covers deep learning methods, features used as model inputs, evaluation techniques for deep learning models, and the crucial datasets and systems used in training of these sophisticated models.

0.1. BACKGROUND

AI encompasses a wide range of technologies aimed at developing intelligent machines (Figure 1), ML, a subset of AI, centers on deriving insights and making predictions from data, and DL, a branch of machine learning, utilizes deep neural networks to emulate human cognitive functions such as reasoning, problem-solving, natural language understanding, and pattern recognition patterns and representations from the source. These activities include learning, problem-solving, understanding, and decision-making. Its objective is to develop machines capable of emulating human cognitive abilities such as reasoning, problem-solving, understanding natural language, and recognizing patterns.

Figure 1. Hierarchy of disciplines of artificial intelligence

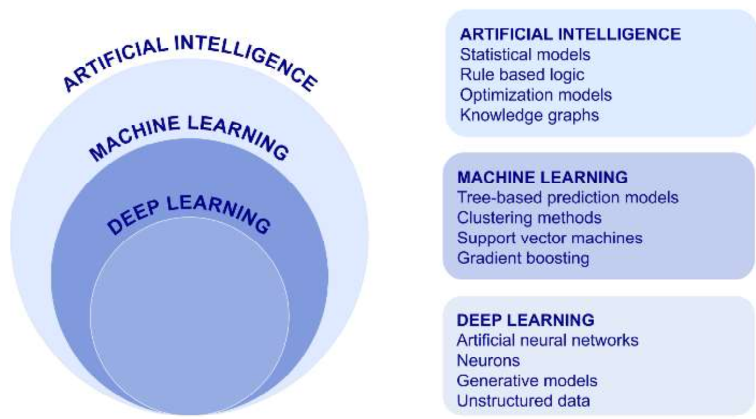

Fig. 1. AI HIERARCHY

AI applications span virtual assistants like Siri and Alexa, recommendation systems as seen on Netflix and Amazon, and autonomous vehicles. This subset concentrates on creating algorithms and models enabling computers to learn from data and make decisions or predictions without detailed programming". At the same time, a lot of information is transferred to improve performanceMachine learning techniques encompass supervised learning, unsupervised learning, and incremental learning. Supervised learning entails training a model on labeled data. Unsupervised learning focuses on detecting patterns within unlabeled data, whereas reinforcement learning is based on a trial-and-error approach.

0.1.1. MACHINE LEARNING (ML)

Machine learning is a subset of artificial intelligence that focuses on developing algorithms and models that allow computers to Machine learning involves deriving predictions or decisions from data without relying on explicit programming. Machine learning algorithms analyze both labeled and unlabeled data to identify underlying patterns, relationships, and insights. Over time, performance improves as more data becomes accessible. Machine learning techniques encompass supervised learning, unsupervised learning, and reinforcement learning. Supervised learning entails training a model with labeled data, unsupervised learning involves uncovering patterns within unlabeled data, and reinforcement learning centers on learning via iterative trial and error processes.

0.1.2. AGILE TRENDS 2024: THE NEXT WAVE OF AGILE TRANSFORMATION

The significance of company resilience saw a dramatic shift in 2022 due to the impact of Covid-19. Enterprises were compelled to streamline production operations and processes. Looking at the looming uncertainties, organizations are increasingly turning to agile project management processes and ways to maintain competitiveness and adaptability. A study by Delta Matrix states that Agile teams are 25% productive! Agile practices target optimizing intricate production processes, empowering project managers to deliver projects in a functional state. This approach facilitates rapid improvements throughout the project. Agile testing prompts project teams to detect issues and deploy solutions throughout the growth cycle, with a strong emphasis on meeting customer needs.

0.1.3. AGILE DESIGN THINKING

Design thinking emphasizes creating customer-centric products that align with users' end needs and requirements. This recent Agile trend ensures that products under development empathize with consumers, thereby enhancing customer value and satisfaction

Design thinking centers around identifying and resolving issues and tasks by iteratively refining through consistent feedback loops. Integrating Agile methodologies early in design thinking-based initiatives enables businesses to reduce risks and costly redesign cycles. This integration also leads to shorter development and testing phases.

0.1.4. DOMINANCE OF SCALED AGILE AND SCRUM

Scaled Agile is a framework adopted by organizations aiming to facilitate Agile development practices across their teams and departments. It comprises workflow patterns primarily implemented at the enterprise level. This approach is instrumental in monitoring progress and achieving enhanced efficiency, collaboration, and resilience on a larger scale, presenting significant advantages.

Unlike the past, when Agile practices were primarily used by smaller teams on smaller projects, today, larger organizations are increasingly adopting these practices for complex projects. This shift enables them to deliver intricate software projects efficiently while fostering improved collaboration and coordination among larger teams.

The Scrum creation method promotes efficient organization of cross-functional team activities, aiming to produce functional code at each iteration's end or sprint. Software QA companies are increasingly aiming to expand their Scrum operations to drive greater value and foster improved collaboration.

0.1.5. AGILE AI AND MACHINE LEARNING INTEGRATION

Despite implementing agile approaches, project teams like product developers and testers still undertake substantial theoretical work. Machine learning algorithms and Artificial intelligence plays a vital role in analyzing growth data and projections, providing real-time insights and rapid advanced analytics. For instance, they enable precise forecasts regarding the completion timelines of project phases. This becomes especially pertinent as projects approach the release phase, attracting significant attention from executives monitoring schedules closely.

0.1.6. MACHINE LEARNING IN AGILE

AI and machine learning present supplementary advantages within agile methodologies, including:

- Offering more precise insights and streamlined processes for testing and developing programming code.
- Conducting thorough code checks with increased accuracy to detect and eliminate bugs.
- Incorporating cutting-edge smart technologies such as robotic process automation (RPA) and Internet of Things (IoT) devices, quantum computing, and other cutting-edge technologies to accelerate development timelines and expedite product launches to market.

1. RELATED WORK

The researched papers collectively offer comprehensive insights into the advancements in agile software development, particularly in the domains of effort estimation, regression testing, and the integration of machine learning techniques. Effort estimation holds crucial importance in Agile software development because of its iterative and customer-centric nature. The papers discuss various techniques employed for accurate estimation, such as Story Points, Person-Hours, COSMIC Function Points, and the integration of machine learning-based methodologies. The importance of maintenance in Agile projects, often overlooked, is highlighted, with suggestions for revised Agile methodologies or hybrid approaches as projects mature. Regression testing in Agile is also extensively covered, emphasizing its role in maintaining software quality and the need for efficient test selection algorithms, AI integration, and ongoing research to accommodate evolving software development practices. The comparison between Traditional Agile Methods and Machine Learning-Enhanced Agile Methods underlines the significant advancement represented by the latter, with a focus on data-driven decision-making, predictive analytics, automated testing, continuous improvement, and personalized task assignment.

Machine learning techniques are integrated into various facets of Agile processes, including effort estimation, sentiment analysis, defect prediction, risk assessment, resource allocation, user story prioritization, automated testing, and team productivity analysis. Emphasis is placed on optimizing strategies for these techniques, encompassing data quality and quantity, algorithm selection, explainable AI, real-time data incorporation, domain-specific models, continuous learning, and collaborative AI. The synthesis of the papers also addresses the significance of accurately estimating software development efforts, the application of machine learning methodologies for effort prediction in both Agile and non-Agile software development, and the benefits of applying historical data and data-driven strategies for estimation accuracy, particularly in agile projects. The collective research papers not only provide a comprehensive overview of the subject matter but also offer a comparative analysis of the techniques and methodologies discussed. They highlight the advantages and limitations of the various estimation and testing techniques, emphasizing the significance of accurate estimation in software project planning, monitoring, and development within complex and dynamic environments. The papers also underscore the need for further research to address gaps, improve predictive accuracy, and develop innovative approaches for effort estimation and regression testing in Agile software development.

1.1. OVERVIEW OF AGILE SOFTWARE DEVELOPMENT

The research paper "Optimizing Regression Testing in Agile Software Development: Techniques, Effort Estimation, and Future Directions"[1] provides a comprehensive overview of Agile software development, focusing on regression testing, effort estimation, and the integration of machine learning techniques. It discusses various effort estimation methods like Story Points and Person-Hours, emphasizing the importance of accurate estimation for resource allocation. The role of regression testing in Agile is highlighted, together with approaches for maximizing it such as test automation and Continuous Integration. Future prospects include AI integration and ongoing research to improve Agile practices. Integration of machine learning in Agile spans various areas like effort estimation, defect prediction, and automated testing. Overall, the paper emphasizes Agile's adaptability and the need for innovative techniques to ensure software quality and agility. Effort Estimation in Agile Software Development: A Systematic Literature Review [2] The systematic review of literature (SLR) on effort estimation in Agile Software Development (ASD) collected insights from 25 primary studies. It identified that subjective estimation methods such as expert judgment and planning poker are commonly used in Agile environments. Additionally, Story Points and use case points are frequently employed as size metrics. The review also highlighted Mean Magnitude of Relative Error (MMRE) and Magnitude of Relative Error (MRE) as commonly used accuracy metrics, while team skills, prior experience, and task size were noted as significant cost drivers. Only Extreme Programming (XP) and SCRUM were identified as Agile methods in the primary studies.Effort estimation is crucial in Agile Software Development, with the shift towards ASD emphasizing iterative planning at release, iteration, and daily levels. Planning Poker emerged as a useful technique, with other methods like incremental prediction models and Constructive Agile Estimation Algorithm (CAEA) showing promise in enhancing estimation accuracy. The review highlighted research gaps and future directions, underscoring the need for further exploration in effort estimation for ASD. It aimed to bridge literature from both effort estimation and Agile software development perspectives, providing a comprehensive synthesis of evidence and guiding future research endeavors. The systematic literature review process was detailed, outlining steps, research questions, and the search strategy employed. The review's conclusion stressed its significance in consolidating evidence, identifying gaps, and setting forth future research trajectories in effort estimation for AS. Empirical assessment of machine learning models for agile software development effort estimation using story points [3] The study aimed to enhance the accuracy of effort estimation in Agile software development through the story point approach (SPA). It evaluated various machine learning techniques such as decision trees, stochastic gradient boosting, and random forest, comparing them with traditional estimation methods. Agile software development is known for its iterative process and rapid delivery, but effort estimation using SPA typically incurs an error rate of 20-30%, indicating room for enhancement. Thus, the study aims to boost prediction accuracy by applying machine learning to the story point dataset and comparing these techniques with traditional methods.

In the empirical assessment, decision tree, stochastic gradient boosting, and random forest techniques were evaluated using a dataset that included twenty-one software projects. According to the findings, stochastic gradient boosting demonstrated superior performance compared to existing techniques and other machine learning methods. The study utilized statistical analysis to evaluate criteria such as Mean Magnitude of Error Relative to the estimate (MMER) and Prediction Accuracy (PRED(x)), to gauge performance and compare with existing work. The conclusion emphasizes the necessity for additional analysis and enhancements in effort estimation for Agile software development, acknowledging limitations like the small dataset size. It underscores the effectiveness of machine learning techniques, especially stochastic gradient boosting, in enhancing prediction accuracy using the story point approach. Additionally, it suggests exploring other machine learning methods in future research to refine the Effort estimation processes for Agile software development: A review on software cost and effort estimation techniques for the Agile development process [4]. The paper discusse*d* the importance of accurate software estimation, especially in agile development, and categorizes estimation techniques into basic, agile-specific, and machine learning-based methods. It surveys basic techniques like SLIM, function point-based models, COCOMO, expert judgment, analogy-based methods, and Planning Poker. It also covers the application of traditional techniques in agile contexts, highlighting factors like developer experience and system changes. Additionally, it explores Machine learning techniques such as genetic algorithms and support vector regression are explored for optimizing effort estimation accuracy. The paper concludes with a comparative analysis of various estimation techniques, stressing the necessity for further research to improve accuracy in dynamic development environments.

The paper "Effort Prediction in Agile Software Development with Bayesian Networks"[5] delves into the advantages and hurdles of agile software development, focusing on effort estimation. It notes that while agile methodologies have boosted project success rates due to their adaptability and client collaboration, inaccurate effort estimation remains a significant cause of project failure. To tackle this issue, the paper introduces a Bayesian Network (BN) model tailored for agile projects to enhance effort estimation accuracy. The paper outlines the shift in software development towards agility, emphasizing continuous value delivery over predictability. It explores various agile frameworks like Scrum, eXtreme Programming (XP), and Lean Software Development (LSD) for their potential in reducing development time and costs.While agile principles are beneficial, the paper acknowledged that they don't guarantee success. It highlights common project failure factors such as evolving requirements and inadequate user involvement. Effort estimation, often done subjectively through practices like Planning Poker and story points, is identified as a major challenge. The paper introduces a BN model that factors in teamwork quality and user story characteristics to enhance effort estimation accuracy. These factors are represented as nodes in a graph, aiming to capture the intricate relationships influencing effort estimation. The model's construction, validation, limitations, and future potential are thoroughly discussed. In essence, the paper provides valuable insights into the challenges of effort estimation in agile software development. It introduces a novel approach based on Bayesian Networks (BN) to tackle these challenges, aiming to enhance accuracy and project success. The paper titled "Effort Estimation in Agile Software Development using Evolutionary Cost-Sensitive Deep Belief Network (ECS-DBN)" [6] addresses the specific challenges associated with effort estimation in Agile Software Development. (ASD) and introduces the ECS-DBN method for effort prediction.The paper outlines the difficulties in effort estimation within ASD and introduces the ECS-DBN method. This method is designed to assist project managers by predicting effort even with missing data and capturing causal relationships. It aimed to be agile-friendly and is applied during the planning stage of software development projects.The paper underscores the importance of software reliability in quality estimation and addresses the importance of precise software effort estimation across three dimensions: size estimation, effort estimation, and cost estimation.

The ECS-DBN method is compared with existing techniques, demonstrating a significant improvement in accuracy, reaching nearly 99%. Various statistical measures like MMRE, Pred. (m), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Relative Absolute Error (RAE), and Accuracy metrics are employed to assess the performance of the method. The method employs an adaptive differential evolution algorithm and cost-sensitive learning to enhance prediction accuracy and reduce misclassification costs. The paper concluded by highlighting the ECS-DBN method's effectiveness in predicting effort in ASD and its superior performance compared to existing methods. It suggested potential future extensions of the method to other deep learning approaches for further improvements. Overall, the paper introduces the ECS-DBN method as a promising solution for achieving accurate effort estimation in Agile Software Development, contributing to better project planning and management.The research paper "Machine Learning models to predict Agile Methodology adoption"[7] focuses on utilizing machine learning to create predictive models for the adoption of the Scrum Agile methodology in software development projects. The paper emphasizes the prevalence of the Scrum framework in Agile software development and the importance of enhancing project outcomes. It explores the application of machine learning to predict Scrum adoption, detailing the research findings problem, relevant literature on combining machine learning with Agile software development, and the employed research methodology, including statistical analysis techniques. The paper presents the outcomes of three machine learning predictive models: the full feature set adoption model, the transformed logarithmic adoption model, and the transformed logarithmic with omitted features adoption model. It identified the model with the highest prediction accuracy and discusses its findings.. The transformed logarithmic with omitted features adoption model is highlighted as the most accurate, achieving an R2 value of 0.527 and a Mean Squared Error (MSE) of 0.038. Conversely, the full feature set model exhibited lower accuracy, with an R2 of 0.564 and MSE of 0.507. The paper concluded by offering recommendations for future research endeavors. It acknowledged limitations and suggests utilizing additional model validation techniques to enhance predictive accuracy. The research outcomes have implications for both researchers and practitioners in software development, providing insights into significant factors for predicting Scrum adoption and identifying avenues for further exploration, such as leveraging larger datasets and employing additional model validation methods.The International Journal on "Technical and Physical Problems of Engineering" (IJTPE)[8] published a paper focusing on predicting effort in Agile software projects using linear regression, ridge regression, and logistic regression techniques. The paper focuses on software estimation techniques within Agile methodology, particularly addressing challenges related to effort estimation in Agile projects. It discussed the common use of story point-based techniques and introduced a novel effort prediction method based on predictors extracted from user stories. The novel effort prediction method utilizes regression techniques and is compared against existing literature to showcase its superior performance. The paper discusses the challenges of effort estimation in Agile projects due to iterative development and evolving requirements.An experimental evaluation and comparative analysis of the proposed technique are conducted, comparing it against existing models including Planning Poker, mathematical models, and machine learning-based techniques. The regression-based model demonstrates better performance in the comparative analysis.The research includes a comprehensive comparative analysis of various existing models, both machine learning and statistical regression-based, highlighting their prediction accuracy and performance. The proposed regression-based model, based on user stories, shows competitive advantages over existing models.The paper outlines the steps for model building, data partitioning, cross-validation, and calculation of prediction accuracy and error rates. It stresses the significance of precise estimation for software project planning and monitoring.The comparative performance analysis shows that the proposed regression-based model outperforms current models based on metrics such as mean magnitude of relative error, root mean square error, and prediction accuracy at various levels.This comparison underscores the advantages of machine learning techniques for software project estimation. In conclusion, The paper underscores the critical importance of accurate effort estimation in Agile software development and presents a detailed evaluation of the proposed regression-based model. It acknowledges the contributions of previous research, particularly Mr. Satapathy's doctoral thesis, in shaping the development of the paper. The research paper titled "Software Effort Estimation in Agile Methodology" [9] examines the challenges and techniques involved in effort estimation within Agile software development projects. It aims to review effort estimation practices in Agile development, highlighting the widespread adoption and effectiveness of Agile methodologies compared to traditional methods. The study identifies machine learning, expert judgment, and algorithmic approaches as the primary estimation methods in Agile, with machine learning being the most commonly used. Effort estimation is crucial for achieving project success and preventing budget overruns. The study identified twenty-four attributes used in effort estimation in Agile, with complexity, experience, size, and time being the most common. Precision measurement of effort estimates is challenging, and the paper suggests presenting estimates as intervals to address uncertainty. The Agile methodology's emphasis on user involvement and iterative development affects effort estimation approaches. The paper evaluated various effort estimation methods in Agile, including Planning Poker, COSMIC, Function Point Analysis, Use Case Point, artificial neural networks, and fuzzy models. It also discussed non-hybrid and hybrid implementation approaches for effort estimation, considering attributes like volatility and customer requirement changes. In conclusion, the paper provided insights into effort estimation challenges and techniques in Agile software development. It identifies commonly used estimation methods, attributes affecting estimation, and implementation approaches in Agile projects, contributing to the development of suitable effort estimation methods tailored for Agile environments. Overall, the paper provides valuable insights into the intricacies and strategies of effort estimation in Agile software development, emphasizing the necessity for standardized techniques and precise estimation to guarantee project success. The systematic literature review focuses on data-driven effort estimation techniques for Agile user stories [10] provides a detailed analysis of various aspects related to this topic. The primary studies emphasized estimating the individual effort of user stories within Agile methodologies rather than estimating effort for the entire project. Data-driven methods, including Decision Trees, Bayesian networks, Support Vector Machines, and advanced techniques like deep learning architectures, were found to be effective in effort estimation. The review identified several performance evaluation methods used in the studies, such as performance measures, baseline benchmarks, comparison against existing techniques, comparison against human estimation, statistical tests, and distribution of estimates. Data-driven techniques were shown to outperform human estimation in some cases, with fast estimation times suitable for Agile meetings. The main effort drivers for agile user stories effort estimation were categorized into personnel factors (team characteristics and experience), product factors (priority and complexity of user stories), and estimation-related factors (SP, person-hours, person-days). The datasets used for effort estimation varied in characteristics, constructed from issue tracking systems, software companies, or houses, with varying numbers of user stories ranging from a few to tens of thousands. Effort metrics included SP, person-hours, person-days, and COSMIC Function Point (CFP). The review conducted an in-depth analysis of data-driven techniques, including technique types, performance evaluation methods, accuracy, independent factors affecting effort estimation, and dataset characteristics. It highlighted the need for additional research into independent factors that influence effort estimation. The paper noted the limited availability of datasets for user story effort estimation from industry sources and emphasized the importance of constructing more datasets. It also highlighted opportunities for further research in improving accuracy and exploring additional independent factors. The review concluded by summarizing the effectiveness of data-driven effort estimation techniques, the need for more datasets, and potential areas for future research, including accuracy analysis, dataset construction, and exploration of additional factors influencing effort estimation. The article “Recent Advances in Software Effort Estimation using Machine Learning” [11] highlights significant developments in effort estimation methodologies, with a particular emphasis on machine learning approaches in both non-agile and agile software development. It explores recent advancements in utilizing machine learning for software effort estimation, especially within agile environments. The article underscores the importance of historical data, mentions specific datasets used, and discusses the advantages of agile methodologies for achieving accurate predictions. Key machine learning models, such as neural networks, are highlighted for their role in enhancing estimation accuracy. Additionally, the article identifies areas for future research, including tracking individual engineers’ patterns and addressing challenges related to extrapolation.

2. UNVEILING INSIGHTS: A COLLECTIVE SUMMARY OF RECENT RESEARCH CONTRIBUTIONS TO AGILE EFFORT ESTIMATION

The research papers studied so far on agile methodologytopics related to software development, with an emphasis on agile methodology, effort estimation, and the integration of machine learning into project management practices. Effort estimation in Agile software development is a recurring theme, with a focus on techniques that utilize story points and regression models, such as Linear, Ridge, and Logistic regression, to optimize estimation processes. The research underscores the significance of accurate effort estimation in the context of evolving requirements and highlights the potential for improved prediction accuracy using machine learning models. Several papers discussed the compatibility of Agile methodology with different software project characteristics and the impact of effort estimation on project success. They identify a need for further research in this area and emphasize the importance of constructing more datasets to facilitate the improvement of estimation techniques. Additionally, there is a focus on machine learning’s potential integration with Agile methodologies, including agile project planning, backlog prioritization, and workforce balancing for continuous integration/deployment. The discussion of data-driven effort estimation techniques for agile user stories includes an assessment of the significance of accurate estimation in software project success and points towards the need for additional effort estimation studies. Furthermore, The papers explore various machine learning models used for estimating software development effort, including regression trees, multilayer perceptrons, and ensemble methods, emphasizing the potential of neural network models in effort prediction. In conclusion, the collection of research papers offers valuable insights into the significance of effort estimation in Agile software development, emphasizing the potential of machine learning. Furthermore, the papers identify areas for future research, such as exploring hybrid methodologies and utilizing historical project data to improve estimation accuracy in Agile software development.

Table I. ML Techniques for Agile Effort Estimation

Author Name	Focus Areas	Machine Learning Technique	Effort Estimation Methods	Agile Methodologies	Evaluation Techniques	Metrics
Muhammad Usman et al.	Effort estimation in Agile software development	No	Expert judgment, planning Poker, incremental prediction models,CAEA	Extreme Programming (XP),SCRUM	Statistical analysis,MMRE,MRE, accuracy metrics	Use case points, story points, MMRE,MRE
Shashank Mouli Satapathy et al.	Machine learning,effort estimation,Agile software development	Yes(Decision Tree,stochastic gradient boosting,random forest)	Decision Tree,stochastic gradient boosting,random forest	Yes	Statisticalanalysis, MMRE,PRED(x),comparison with existing methods	function point, MMRE
Manju Vyas et al.	Software cost estimation,Agile development	Yes(Genetic algorithms,support vector regression)	SLIM,function point-based models,COCOMO,expert judgment,Planning Poker	Yes(Scrum,XP,Lean Software Development)	Comparative analysis,prediction accuracy,performance evaluation	MMRE, MRE , story point
Laura-Diana Radu	Agile software development,effort estimation	Yes(Bayesian Networks)	Expert judgment,Planning Poker,Story Points	Scrum,XP,Lean Software Development	BN model construction, validation, limitations,future potential	MMRE, MRE
Hosahalli Mahalingappa Premalatha et al.	Agile software development,effort estimation	Yes(Evolutionary Cost-Sensitive Deep Belief Network)	Story Points, Planning Poker,machine learning techniques	Agile Software Development(ASD)	Statistical measures,adaptive differential evolution algorithm,cost-sensitive learning	MMRE, MRE
Ridewaan Hanslo et al.	Agile software development,Scrum adoption	Yes(Machine learning predictive models)	Machine learning predictive models	Scrum methodology adoption	Statistical analysis,predictive accuracy,model validation techniques	R Square, s max error, mean absolute error, median squared error
M. Vyas	Agile software development, effort estimation	Yes(linear regression,ridge regression, logistic regression)	Planning Poker,mathematical models,machine learning techniques	Agile methodology	Model building steps,data partitioning,cross-validation,prediction accuracy,error rates	RMSE, MMRE
Pantjawati Sudarmaningtyas et al.	Agile software development, effort estimation	Yes(Machine learning,expert judgment,algorithmic)	Planning Poker,COSMIC, Function Point Analysis,Use Case Point,artificial neural networks,fuzzy models	Agile methodologies	Evaluation methods,future research directions,challenges in effort estimation in Agile	Code metrics
Bashaer Alsaadi et al.	Data-driven effort estimation,Agile user stories	Yes(Decision Trees,Bayesian networks,Support Vector Machines,deep learning architectures)	Decision Trees,Bayesian networks,Support Vector Machines,deep learning architectures	Agile methodologies	Performance measures,accuracy evaluation,dataset characteristics,future research directions	use case point, story point
V´ictor Uc-Cetina	Software effort estimation,machine learning	Yes(Neural networks,machine learning models)	Story Points,Person-Hours	Agile methodologies	Improvement in estimation accuracy,future research areas	

Table II. Advantages and limitations of various ML techniques

Machine Learning Technique	Advantages	Limitations
Decision Tree	- Easy to interpret and visualize	- Prone to overfitting
	- Can handle both numerical and categorical data	- Sensitive to small variations in the data
	- Requires little data preprocessing	- Not suitable for capturing linear relationships
Stochastic Gradient Boosting	- Can handle complex nonlinear relationships	- Requires careful tuning of hyperparameters to avoid overfitting
(GBM)	Robust to outliers and noisy data	Computationally intensive
	- Combines multiple weak learners to improve accuracy	- Can be sensitive to the choice of learning rate and tree depth
Random Forest	- Reduces overfitting compared to individual decision trees	- Can be slow to train on large datasets due to the ensemble nature
	- Handles high-dimensional data well	- May not perform well with noisy data or imbalanced classes without additional handling
	- Provides feature importance ranking	- Lack of interpretability compared to simpler models like decision trees
Neural Networks	- Capable of learning complex patterns and nonlinear relationships	- Requires a large amount of data for training to avoid overfitting
	- Can handle large datasets and high-dimensional features	- Complex architectures may be difficult to interpret
	- Suitable for tasks like image recognition, natural language processing, and time series prediction	- Prone to issues like vanishing gradients and training time increases with model size
Bayesian Networks	- Provides a probabilistic framework for modeling uncertainty	- Requires prior knowledge or assumptions about the network structure and conditional probabilities
	- Captures dependencies and causal relationships between variables	- Computationally intensive for large networks or when updating probabilities frequently
	- Allows for incremental learning and updating of probabilities	- May struggle with highly interconnected variables and complex dependencies
Evolutionary Cost-Sensitive	- Integrates evolutionary algorithms with deep learning for cost-sensitive learning	- Requires domain expertise for parameter tuning and model selection
Deep Belief Network (ECS-DBN)	- Handles missing data and captures causal relationships	- Computationally demanding
	- Achieves high prediction accuracy in effort estimation tasks	- Interpretability may be a challenge due to the complexity of deep learning models

3. CONCLUSION

Effort estimation in Agile software development is a challenging task, and the surveyed papers highlight the significance of accurate predictions in iterative and customer-centric environments. Various methodologies, including Story Points, COSMIC Function Points, and machine learning-based models, have been explored. The Evolutionary Cost-Sensitive Deep Belief Network (ECS-DBN) model represents a notable advancement, providing highly accurate effort predictions and seamlessly integrating with agile methodologies.

The comparison of Traditional Agile Methods and Machine Learning-Enhanced Agile Methods emphasizes the transformative potential of machine learning, introducing data-driven decision-making, predictive analytics, and automation into agile practices.

The decision-making framework thoroughly examines the compatibility of the Agile methodology across the Software Development Life Cycle (SDLC) phases and the Project Management Body of Knowledge (PMBOK) knowledge areas.

- Ridge Regression and ECS-DBN models have demonstrated promising results in effort estimation.
- Data-driven techniques have proven effective in estimating agile user stories effort estimation.

4. FUTURE WORK

1. *Methodology Expansion*: Future research should expand effort estimation methodologies, exploring hybrid approaches and incorporating diverse metrics beyond traditional measures.
2. *Machine Learning Integration: Expand the application of machine-learning techniques, such as support vector machines and random forest*, and stochastic gradient boosting, to further refine and enhance agile software effort estimation models.
3. *Adaptation for Heterogeneous Environments*: Investigate adaptations and refinements to effort estimation models, such as the ECS-DBN, to improve applicability in heterogeneous software development environments.
4. *Decision-Making Framework Refinement:* Refine decision-making frameworks to address gaps, particularly in assessing the impact of agile methodology on the Quality Assurance (QA) phase, scalability, and integration with agile scaling frameworks.
5. *Longitudinal Studies:* Conduct longitudinal studies to assess the sustainability and evolution of machine learning models and effort estimation methodologies over time, considering the dynamic nature of the software development landscape.
6. *User-Friendly Implementations:* Focus on developing user-friendly implementations for machine learning models and frameworks, ensuring seamless adoption and integration into industry practices.
7. *Diverse Context Evaluations*: Evaluate effort estimation models and decision-making frameworks in diverse agile contexts, ensuring their robustness and adaptability across a spectrum of software development projects.
8. *Enhanced Quality Assessment:* Enhance quality assessment criteria for decision-making frameworks, ensuring a comprehensive evaluation that considers additional factors influencing the validity of conclusions.
9. *Integration with Agile Practices:* Explore seamless integration of machine learning models with various agile practices and methodologies, aligning them with industry standards and best practices.

The combined efforts from these avenues of future research aim to advance the accuracy, adaptability, and user-friendliness of effort estimation models, decision-making frameworks, and machine learning integration in the context of agile software development. Through ongoing innovation and practical application, the field is poised for continuous improvement and refinement in addressing the unique challenges posed by agile methodologies

REFERENCES

1. USMAN (M.), MENDES (E.), WEIDT (F.) & BRITTO (R.): "Effort Estimation in Agile Software Development: A Systematic Literature Review." Department of Software Engineering, Blekinge Institute of Technology, Karlskrona, Sweden, Federal University of Juiz de Fora, Brazil. {muhammad.usman, emilia.mendes, ricardo.britto}@bth.se, fran.weidt@gmail.com.
2. KHUAT (T. T.) & LE (M. H.): "A Novel Hybrid ABC-PSO Algorithm for Effort Estimation of Software Projects Using Agile Methodologies." DOI 10.1515/jisys-2016-0294. Received November 11, 2016; previously published online March 23, 2017.
3. SATAPATHY (S. M.) & RATH (S. K.): "Empirical Assessment of Machine Learning Models for Agile Software Development Effort Estimation Using Story Points." Innovations in Systems and Software Engineering. DOI 10.1007/s11334-017-0288-z. Received: 15 September 2016 / Accepted: 1 June 2017. © Springer-Verlag London Ltd. 2017.
4. SATAPATHY (S. M.) & RATH (S. K.): "Empirical Assessment of Machine Learning Models for Agile Software Development Effort Estimation Using Story Points." Innovations in Systems and Software Engineering, 13(2017), 191–200. DOI 10.1007/s11334-017-0288-z. Received: 15 September 2016 / Accepted: 1 June 2017 / Published online: 13 June 2017. © Springer-Verlag London Ltd. 2017.
5. PREMALATHA (H. M.) & SRIKRISHNA (C. V.): "Effort Estimation in Agile Software Development using Evolutionary Cost-Sensitive Deep Belief Network." International Journal of Intelligent Engineering and Systems, Vol. 12, No. 2 (2019), pp. 261. DOI: 10.22266/ijies2019.0430.25. Received: December 19, 2018. People's Education Society University, India. People's Education Society Institute of Technology, India. Corresponding author's Email: premalathaphd2017@gmail.com.

6. VYAS (M.), BOHRA (A.), LAMBA (C. S.) & VYAS (A.): "A Review on Software Cost and Effort Estimation Techniques for Agile Development Process." International Journal of Recent Research Aspects, ISSN: 2349-7688, Vol. 5, Issue 1, March 2018, pp. 1-5. © 2018 IJRRA All Rights Reserved, page - 1-. Manju Vyas [1], Assistant Professor, Department of Information Technology, JECRC, Jaipur. Amit Bohra [2], Assistant Professor, Department of Information Technology, GIT, Jaipur. Dr. C.S. Lamba [3], Professor, Manipal University, Jaipur. Abhilasha Vyas [4], Assistant Professor, G D Goenka University, Gurgaon.
7. RADU (L. D.): "Effort Prediction in Agile Software Development with Bayesian Networks." Department of Accounting, Business Information Systems and Statistics, Faculty of Economics and Business Administration, Alexandru Ioan Cuza University of Iasi, Blvd. Carol I, Iasi, Romania.
8. COWLESSUR (S. K.), PATTNAIK (S.) & PATTANAYAK (B. K.): "A Review of Machine Learning Techniques for Software Quality Prediction." Chapter in a Book, January 2020. DOI: 10.1007/978-981-15-1483-8_45. Citations: 5. Reads: 1,506. Authors: Sanjeev K Cowlessur, Saumendra Pattnaik, Binod Kumar Pattanayak. Affiliations: Université des Mascareignes, Siksha O Anusandhan University.
9. MENDEZ (E.), GUEVARA (F. G. L.), ABRAHAO (S.) & INSFRAN (E.): "An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review." IEEE Access, August 2020. DOI: 10.1109/ACCESS.2020.3021664. Citations: 43. Reads: 656. Authors: Erwin Mendez, Fernando González L. Guevara, Silvia Abrahão, Emilio Insfran. Affiliation: Universitat Politècnica de València.
10. HANSLO (R.) & TANNER (M.): "Machine Learning models to predict Agile Methodology adoption." Council for Scientific and Industrial Research, South Africa. Department of Information Systems, University of Cape Town, South Africa. Email: rhanslo@csir.co.za, mc.tanner @uct.ac.za.
11. VYAS (M.) & HEMRAJANI (N.): "Predicting Effort of Agile Software Projects using Linear Regression, Ridge Regression, and Logistic Regression." Department of Computer Science and Engineering, Jaipur Engineering College and Research Centre University, Jaipur, India. Email: vyas.manju@gmail.com, hod.cse@jecrcu.edu.in. Published in International Journal on "Technical and Physical Problems of Engineering" (IJTPE), June 2021, Issue 47, Volume 13, Number 2,

Pages 14-19. ISSN 2077-3528. Website: www.iotpe.com, Email: ijtpe@iotpe.com.

12. SUDARMANINGTYAS (P.) & MOHAMED (R.): "A Review Article on Software Effort Estimation in Agile Methodology." Faculty of Computing, Universiti Malaysia Pahang, 26300 UMP, Gambang, Kuantan, Pahang, Malaysia. E-mail addresses: pantja@dinamika.ac.id (Pantjawati Sudarmaningtyas), rozlina@ump.edu.my (Rozlina Mohamed). Published in Pertanika J. Sci. & Technol., Volume 29, Issue 2, Pages 837-861, April 30, 2021. ISSN: 0128-7680, e-ISSN: 2231-8526. Journal homepage: http://www.pertanika.upm.edu.my/. DOI: https://doi.org/10.47836/pjst.29.2.08. © Universiti Putra Malaysia Press.
13. ALSAADI (B.) & SAEEDI (K.): "Data-driven effort estimation techniques of agile user stories: a systematic literature review." Saudi Electronic University. DOI: 10.1007/s10462-021-10132-x. Published in Artificial Intelligence Review, January 2022. URL: https://www.researchgate.net/publication/357729732.
14. "International Journal of Research and Analytical Reviews (IJRAR) July 2022, Volume 9, Issue 3." IJRARTH00015, www.ijrar.org. "AN EXPERIMENTAL STUDY FOR SOFTWARE QUALITY PREDICTION USING MACHINE LEARNING METHODS." Regulapati Akhila, A.Anusha, Dr.P.Srinivasa Rao. 1JB Institute of Engineering and Technology, India.
15. "Empirical Software Engineering (2023) 28:105." DOI: 10.1007/s10664-023-10334-7. "Does agile methodology ft all characteristics of software projects? Review and analysis." David Itzik, Gelbard Roy. Accepted: 24 April 2023. © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023.
16. "Empirical Software Engineering (2023) 28:77." DOI: 10.1007/s10664-023-10312-z. "Deep learning approaches for bad smell detection: a systematic literature review." Amal Alazba, Hamoud Aljamaan, Mohammad Alshayeb. Accepted: 1 March 2023 / Published online: 11 May 2023. © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023.
17. "Recent Advances in Software Effort Estimation using Machine Learning." Víctor Uc-Cetina. uccetina@correo.uady.mx. Facultad de Matemáticas, Universidad Autónoma de Yucatán. March 8, 2023.

18. SUNDA (N.) & SINHA (R. R.). (2023). Optimizing Effort Estimation in Agile Software Development: Traditional vs. Advanced ML Methods. In *2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI).* IEEE.
19. ILIEVA (R. Y.) & NIKOLOV (M. A.). (2019). The Impact of AI & ML in Agile Production. In *2019 X National Conference with International Participation (ELECTRONICA).* IEEE.
20. KUMAR (S.), ARORA (M.), SAKSHI & CHOPRA (S.). (2022). A Review of Effort Estimation in Agile Software Development using Machine Learning Techniques. In *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA).* IEEE.
21. SRIKANTH (B. V.), REDDY (P. V. B.) & KAMESH (D. B. K.). (2023). Machine Learning Based Software Effort Estimation of Suggestive Agile and Scrumban Methodologies. In *Lecture Notes in Networks and Systems (LNNS, volume 648).* First Online: March 28, 2023.
22. SINHA (R. R.) & GORA (R. K.). (2020). Software Effort Estimation Using Machine Learning Techniques. In *Lecture Notes in Networks and Systems (LNNS, volume 135).* First Online: August 19, 2020