International Journal of Mathematics, Statistics, and Computer Science
Vol. 3
ISSN: 2704-1077 eISSN 2704-1069, DOI: 10.59543/ijmscs.v3i.15097

Comparative Analysis of AI Models for Effort Estimation in Western and Regional Environments

Fahim Sardar[1], Mohammad Ayub Latif[1,*], Muhammad Khalid Khan[1], Omar Al-Boridi[2], Hassen Hamouda[3]
[1]College of Computing and Information Sciences (CoCIS), Karachi Institute of Economics and Technology (KIET), Karachi, Pakistan
[2] School of Engineering, RMIT University, Melbourne, Australia
[3]Department of Business Administration, College of Business Administration, Majmaah University, Al-Majmaah 11952, Saudi Arabia
*malatif@kiet.edu.pk

Received 07.05.2025, Revised 18.07.2025, Accepted 24.07.2025

ABSTRACT: Artificial Intelligence rapidly alters business operations and workflow management strategies these days in various corporate sectors. AI can be effective in calculating the effort required for a software. Figuring that how much time work and resources some project will probably take requires a fairly decent amount of effort and is also one of the most crucial activities of software project management. When an estimate is met the confidence of companies increase and funds can be allocated nicely, ultimately helping in finishing projects quickly. This work evaluates different AI models for estimating software effort accurately in two distinct areas. Western environment encompasses nations such as US and UK, and mostly all developed places including Canada alongside other similar countries. These countries typically possess sophisticated technology and proficient labor with meticulous documentation practices. Regional environment encompasses areas namely South Asia and Africa alongside Middle East which undergo development challenges. These areas often face many problems such as weak digital infrastructure in various sectors and somewhat disorganized data sets which are not very helpful for estimation. Various AI models were tested including Linear Regression, Neural Networks random forest etc. in different areas to determine which ones worked nicely. Three measures were used namely Mean Absolute Error (MAE) Root Mean Squared Error (RMSE) and R² Score to assess AI model efficiency. Better accuracy stems from lower MAE and RMSE values while higher R² scores signify deeper understanding of data patterns. Neural Networks operate more effectively in Western regions owing largely to relatively cleaner data and markedly greater regularity. Random Forests and Decision Trees perform markedly better in regional areas plagued by messy data because they handle such info pretty well. Companies ought to select an AI model suited pretty well to their specific local conditions and the kind of data they possess. Finally in both the environments, that means deploying both the datasets for the environments, it was the hybrid technique that performed the best for predicting the effort of software. The hybrid model used for prediction give the lowest MAE of 0.22 and |RMSE of 0.38 with R^2 of 0.9 for the Western regions. Similary even for the regional areas give the lowest MAE and RMSE of 0.4 and 0.55 respectively and R^2 of 0.79.

Keywords: Software Effort Estimation, Machine Learning, Software Predictions, Western Environments, AI Models

1. INTRODUCTION

Software effort estimation is a crucial activity of software project management. It is very improtant for planning, budgeting, and ultimately the project's success[1]. The extreme complexity of software development, which depends on the project scope among other human dynamic team factors and requirements growth, presents a strong challenge from the software estimation point of view[2]. Accurate estimation can mitigate risks such as those that lead to cost overruns and late deliveries. Hence, it is of utmost importance and has acquired a good deal of attention in research community[3]. Standard estimation models tend to be not adaptive as they are now quite old, whereas machine learning techniques like regression and ensemble methods should be promising enough to capture the dynamics of projects[1], [4]. Although, prediction accuracy and transparency in data handling are the problems related to these approaches which do not instill much confidence in a collaborative environment where large team works on software projects[5]. Artificial Intelligence (AI) has changed how industries like medicine and banking operate, making tasks more precise and quicker[6]. In software development, figuring out how much time, people, and tools a project needs is called effort estimation and it is a major area where AI helps[7]. Back in the day, experienced developers used simple methods or just their instincts, which often led to mistakes because of personal biases or old information[8]. Now, AI uses data to make better predictions, with methods like neural networks and random forests giving solid results[9]. But the results depend on where you are. In Western countries, where data is neat and tech is advanced, complex models like neural networks work best[10]. Decision trees perform better in areas where there is scarcity of organized data and these include the regions of South Asia or Africa[11]. The major things like the local culture and the setup of the workplaces decide how well these models will perform[12]. The accuracy metrics like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R² are used by the researchers to test these models in Western and other regions[5]. Fancy models fit well with Western data, but ones that can handle messy data work better elsewhere[3]. Having this information assist the companies in selecting the appropriate model for their estimation cases[13].

AI is a big part of estimating software project efforts, but it works differently depending on where you are because of varying work habits and data quality[6]. In places like the U.S. or U.K., where work is structured and workers are skilled, models like Linear Regression do well with clean data[14]. But in regions like South Asia or Africa, spotty records and few resources make things tough[8]. Here, flexible models like Decision Trees or Gradient Boosting Machines handle messy data better[9]. Tests using MAE and R^2 show that Neural Networks are great in Western settings, while Random Forests do better elsewhere[10]. Mixing different methods into hybrid models can work across these different places[11]. Choosing the right model or combining them based on where you are can make effort predictions much more accurate[12].

This work investigates how the AI models estimation the effort of software in Western countries and other regions. The prediction of time and actual work needed is challenging and generally uncertain[3]. Companies work in places with different work styles, worker skills, and tech access[13]. A model that's great in one place might not work in another because of these differences[6]. This research tests how various AI models perform in different settings[14]. The main aim is to find the best AI model for effort estimation in Western countries like the U.S. and Germany, and in regions like South Asia[8]. The results will give practical tips for companies to plan projects better and manage them well[9]. The specific contributions of this research is given as under:

- Investigation of AI for software effort estimation: This work validates how the AI can benefit the need for software effort estimation. The tools like Linear Regression, Neural Networks, and Decision Trees are used by the companies. Investigation will be made of the pros and cons of the AI tools for effort estimation in different industries and places.
- Compare Data from Western and Other Regions: Differences in how people work and what tech they use in Western countries versus other regions. Things like work hours and skilled workers affect how much effort projects take. Studying these differences will show why models work better in some places than others.
- Test Which Models Work Best in Each Place: Models like regression, decision trees, and neural networks will be tested in Western and other regions. MAE, RMSE, and R^2 will be used to see how accurate they are. These tests will show which models are reliable in different places.
- Find Models That Work for Both Kinds of Data: Identifying model that can work for both Western and regional data or if there is a need for separate data. Combining methods into hybrid or ensemble models might work well in different places. Adding details specific to each region could make results better.
- Give Useful Tips for Companies: Based the identification, AI models will be suggested for companies working in different regions. These tips will help avoid going over budget or missing deadlines. Software teams and managers can use this advice to make AI work better for projects around the world.

This study will test AI models in different places to see how they perform. Finding the best models for software effort estimation worldwide will help create better, more flexible tools. Companies will plan projects better and get better results. The rest of the paper is structured as follows, section 2 gives the literature review, section 3 discusses the methodology used for this research, section 4 discusses the case studies and the datasets for the experimentation, section 5 is for the model training and evaluation, section 6 gives the results and discussion, finally section 7 concludes this research work with recommendations and future directions.

2. LITERATURE REVIEW

This section reviews key contributions, focusing on how AI improves effort estimation and adapts to regional differences, building on the introduction's emphasis on model performance in Western and regional contexts.

2.1 Foundational Contributions

Early work laid critical groundwork for AI in software effort estimation. Hecht-Nielsen's study on neural networks demonstrated their ability to handle complex tasks, inspiring their use in predicting project requirements[6]. Boehm's COCOMO II model introduced a structured, data-driven approach to cost estimation, outperforming traditional expert guesses[14]. Menzies et al. highlighted flaws in expert-based methods, showing that biases and outdated knowledge often led to errors, paving the way for AI-driven solutions[8]. These studies established the need for reliable, data-centric estimation methods.

2.2 Machine Learning Techniques

Machine learning has become a cornerstone of effort estimation. Breiman's random forests are valued for their robustness with diverse, noisy datasets, making them suitable for regions like South Asia or Africa where data quality varies[9]. Quinlan's decision trees, known for their simplicity, excel in environments with sparse or inconsistent data[11]. Nowadays, machine learning is becoming more and more common in the field of software effort estimation. The core reason was this on that fact that traditional methods were unable to give good results in cases where the data is not nicely recorded or has incomplete information. Such was the problem in the areas of South Asia and other parts of the African region, where such cases can create problems for the simple models. This created a need in researchers to look for more advanced models that can handle the issues of incomplete and unstructured data.

To study the performance of various machine learning models, a research work[15] tried them on different datasets from noisy environments. In was validated that where the data was not up to the marks the best performing techniques were Random Forest and Gradient Boosting. So the final conclusion was that these models are beneficial when the project data is not completely reliable.

The identification for best models for predicting through machine learning revealed that ensemble based techniques, in which there is the concept of have multiple learners, always gives the best results. The core reason for this is in the fact that when one model makes a mistake the other is there to rectify it. The study suggested that this method is best suited when the format of the data is improper and when there are multiple sources of the data[16].

A study found that in Western companies where the project data is very well organized and properly formatted, complex models like neural networks give very good results. This shows that in such structured settings, we can use more complicated models, but in other places, simpler or ensemble models may work better depending on the situation[17].

2.3 Regional Influences on Model Performance

The properties of data highly impact the success of the AI models used for predictions, the properties or characteristics of the data includes the quality of data, infrastructure and the collection practices. In well-established environments like North America and Western Europe, where datasets are large, clean, and feature-rich, complex models such as deep neural networks tend to perform best and give the best results. A research study has[18] demonstrated how data bugs can destabilize deep learning models in software engineering, highlighting how high-quality data greatly supports neural network performance. Conversely, in regions such as Sub-Saharan Africa or South Asia, where data are often incomplete, noisy, or scarce, simpler models offer greater robustness and interpretability. A research study[19] emphasize that in such areas, data quality issues which primarily includes the noise and missing values can impact the deep learning, making lightweight models more practical . So for such environments the maximum advantage can be achieved from using decision trees, logistic regression, or rule-based methods that handle data imperfection more gracefully.

Other than these techniques hybrid models has hugely helped in reducing these effects and these hybrid and ensemble based models have come up as a promising solution for these issues. A research work[20] has found that combining Random Forests with gradient boosting enhances adaptability across datasets with varying quality levels. Another work[21] has reviewed software engineering data quality issues and suggested that combining robust models with data-cleaning strategies improves performance across diverse contexts.

2.4 Evaluation Metrics

Not every place in the world has clean and reliable data and this is the reality. So when identification for a good AI model is required, the measuring criteria should change. In countries with strong digital systems examples include the US or Germany, it's common to check how close predictions are using metrics like MAE or R-squared. These work well when the data behaves properly. A work has analyzed showing how these metrics help in identifying when a model's doing too much or too little[3]. On the other hand in places where data is messy, which means it has missing values, inconsistent formats, then those same metrics don't tell the whole story. This is where MMRE steps in. It doesn't just look at how wrong the prediction is but it checks how wrong it is *relative* to what it should be. A research study has talked about this back in 2013. The work showed that in environments where the data can't be fully trusted, MMRE gives a more realistic picture[22]. So, if fairness is the optimum requirement then the metrics should be changed with respect to the origin of data.

2.5 Real-World Hurdles and What Comes Next

Deploying AI across regions isn't just a technical challenge, it's more about context. Differences in team practices, data, and communication styles can all break the AI model. Initially a work has explored web effort estimation across companies and showed that models trained on one business dataset often fail on another due to differences in how companies collect and structure their data[23]. Another work has examined how adapting cross-company models to local data improves performance. The approach showed that models recalibrated to local conditions work better than static global models[24]. Transfer learning is becoming popular in software effort estimation, it was known that the greatest problem of portability in such cases is the data format and the data quality issues [25]. In the domain of global software development where the teams are distributed across the globe, the problematic areas are the scheduling and meeting. In such environments communication tool like Slack and others play a very vital role and can influence the inputs and the predictions of the models[26]. Considering all these issues this research work uses the accuracy metrics namely MAE, R^2 and MMRE on Western and emerging-market datasets. Instead of picking the best model, it uncovers why and where they work and where it cannot and guide software teams on how to adapt or combine models based on their local context.

The table 1 below, summarizes the literature review, with the publication year, publication in a conference or a journal, the models or methods used and the key findings of the research paper.

Table 1. The summary of the literature review.

Ref	Year	Journal/Conf.	Research Focus	Models/Methods	Key Findings
[6]	1988	IEEE Spectrum	Neural networks in complex tasks	Neural Networks	Neural networks mimic human brain patterns; useful for complex predictions.
[8]	2006	IEEE Trans. on Software Eng.	Flaws in expert estimation	Comparative Analysis	Expert methods are biased; data-driven AI models are more accurate.
[9]	2001	Machine Learning	Handling noisy datasets	Random Forests	Random forests work well with noisy and diverse datasets.
[11]	1986	Machine Learning	Interpretable model design	Decision Trees	Easy to interpret and effective for incomplete or inconsistent data.
[15]	2021	arXiv	ML model performance in noisy environments	Random Forests, Gradient Boosting	Ensemble models perform best when data is unreliable or incomplete.
[16]	2023	Journal of Systems and Software	Ensemble models in effort estimation	Ensemble Methods	Ensemble models provide stable predictions across diverse datasets.
[17]	2012	IEEE Trans. on Software Eng.	Predictive accuracy in structured settings	Neural Nets, Statistical Models	Complex models like NNs succeed with high-quality, structured data.
[18]	2024	arXiv	Effect of data bugs on deep learning	Deep Learning	Data bugs severely degrade neural model performance.
[19]	2023	VLDB Journal	Data quality issues in deep learning	Data-centric AI, Deep Learning	Lighter models preferred when data is noisy or incomplete.
[20]	2013	Conference	Dataset adaptation with hybrids	Random Forests + Gradient Boosting	Hybrids improve prediction over varied data quality levels.
[21]	2023	Neural Computing and Applications	Regional adaptability of DL models	DL + Data Cleaning	Region-specific DL models need cleaning for accuracy.
[22]	2013	Conference	Evaluation metrics under messy data	MMRE	MMRE offers more realistic estimates in noisy data scenarios.
[23]	2012	Conference	Company dataset differences in estimation	Web Effort Estimation	Models trained on local data don't always generalize across companies.
[24]	2014	Conference	Local adaptation of cross-company models	Dycom adaptation	Recalibrated models perform better than fixed global ones.
[25]	2015	Empirical Software Engineering	Transfer learning in global estimation	Transfer Learning	Bridges the gap between source-target data mismatches.
[26]	2020	Journal of Systems and Software	Communication and coordination impact	Mixed-Methods Study	Distributed team factors like Slack usage affect estimation inputs.

The section 3 describes the research methodology in detail that is used in this research work.

3. METHODOLOGY

This research work follows a systematic approach for comparing performance of AI models for software effort estimation across Western and Regional environments. Analyzing data and selecting best AI models carefully in various environments allowed comparison of their relative effectiveness. Models perform variably in Western settings with neatly organized data versus regional settings troubled with messy incomplete information and unstructured data. Steps taken for carrying out analysis are elaborated thoroughly here.

3.1 Data Collection Methods

The first step when comparing AI models for the task of effort estimation requires the collection of data. It is well understood that the AI models rely heavily on historical data, as it helps them in understanding the patterns and making predictions. This means that the collection of data from different projects is key of the prediction for software effort. The data can include the details of the project scope, the effort of the labor, the time for which the team worked together on a project and the overall effort of the software. For this research work, the data was collected for two environments.

- **Western Environments:** Countries like United States and Canada and some parts of Europe notably United Kingdom are included. Data from such regions often proves surprisingly accessible due to existence of meticulously organized digital infrastructure and relatively seamless online portals. Data was gathered from various online databases and some pretty obscure government sources and a bunch of old research papers.
- **Regional Environments:** Data collection proves rather tricky in developing regions of South Asia Africa and Middle East. Data was obtained from local outfits and government reports and academic studies in these regions through painstaking interviews. Data from these areas often arrives in non-digital formats and requires extra processing steps for standardization.

It was ensured that both environments have similar data so that the comparisons can be made. The data included the following:

- Project size or scope
- Location and region
- Materials used
- Labor efforts and wages
- Time taken to complete the project
- Final project effort

3.2 Data Preprocessing and Cleaning

Data was cleaned and prepared for the comparative analysis of AI models for software effort estimation across Western environments and regional settings ensuring quality. Raw data often arrives with gnarly like missing values and formatting. In these problems are not addressed then the AI model will lead towards inaccurate predictions. These steps were followed to preprocess and clean the data:

- **Handling Missing Values:** Project records occasionally lacked crucial details such as labor efforts or timelines for project completion in various instances. Various methods were employed rather haphazardly in similar projects for filling gaps by estimating missing data with average values.
- **Removing Duplicates:** Multiple instances of some records showed up in dataset. Duplicate entries were removed carefully avoiding repetitive examples for models.
- **Standardizing Units:** Some regional data efforts were listed in local currencies others were reckoned in USD or euros. All values were converted into a single currency USD making comparisons easier and fairly more accurate.
- **Formatting Text and Categories:** Different projects used different terms for the same features.

- **Outlier Detection:** Some records exhibited effort values unusually higher or remarkably low. The entries were scrutinized thoroughly and removed the ones deemed patently false or completely inaccurate.
- **Date Formatting:** It was ensured that project dates followed a uniform format so models could properly analyze time durations with relative ease and precision normally.

The data was double checked for completeness and utter accuracy afterwards with a keen eye on overall consistency. This process helped improve analysis quality and ensured AI models in both Western and Regional environments provide reliable comparisons.

3.3 Feature Selection

Immense focus was given for selecting right features from collected data during comparative analysis of AI models for software effort estimation in Western environments. Choosing features crucially impacts AI models' ability to estimate project effort accurately because feature selection directly influences the prediction.

The available data was examined and features were selected that are most influenced the final effort of a project. These included:

- **Project Size:** Total area or volume involved in project.
- **Location/Region:** The country or area where the project is located. This affects labor efforts, and overall effort.
- **Labor Efforts:** Labor demands fluctuate wildly for different projects. Labor efforts fluctuate rather wildly across Western environments partly because of huge disparities in pay.
- **Project Duration:** How long the project takes to complete. Longer projects typically require more resources.
- **Type of Project:** Projects vary widely with respect to their types.
- **Weather or Seasonal Effects:** In certain regions, extreme weather conditions can cause delays and increase efforts.
- **Economic Conditions:** This can include the factors like the rates of the inflation and the economic changes.

For the final selection of the features that can influence the effort of the software, correlation was performed and the features that do not influence the software effort were discarded. Examples of such feature include the name and the phone number of the project manager and it was removed completely, while considering the final features. New features significantly enhanced AI models' ability to process data yielding remarkably more precise predictions overall. The helps in laying the foundation for successful comparative analysis of AI models by exactly choosing most relevant features beforehand with utmost care. This step played a crucial role in ensuring models yielded fairly reliable effort estimates across diverse environments including Western and Regional settings.

3.4 Model Selection

Various AI models were utilized here to determine which one predicts project effort required most accurately with different model types being tested. Selecting a suitable model matters greatly because different models function distinctly and some fare better with specific data types.

We began by selecting a few commonly used models in effort estimation. These include:

1. **Linear Regression**: Simplest model exists evidently. It endeavors draw a straight line through data showing some relationship between project features such as size or materials and total effort expended. Usage can be straightforward but efficacy falters with complex data or non-linear relationships sometimes yielding wrong results.
2. **Decision Trees**: This model functions rather like flowchart with binary questions splitting data into successively smaller subgroups very effectively. Decision trees can handle various data types including numbers and categories pretty easily and remain remarkably straightforward.

3. **Random Forests**: This model functions akin to an ensemble of decision trees collaborating rather haphazardly under various circumstances quite effectively. Each tree makes a somewhat vague prediction and final result hinges on whatever most trees suggest rather tenuously. This approach typically yields more precise outcomes than a solitary decision tree does ordinarily anyway.
4. **Support Vector Machines (SVM)**: SVM tends to be somewhat more complicated inherently. It endeavor's drawing some boundary pretty effectively between diverse groups of data in such a manner making predictions fairly straightforward. It works remarkably well in many situations but needs more fine-tuning and considerable time for initial setup.
5. **Neural Networks**: Powerful models work strangely like human brains and are usually designed with extremely complex algorithms and intricate neural networking structures. They can scrutinize vast amounts of data and discern patterns even when relationships between inputs and outputs are brutally complex. They require ample amounts of data and considerable computing power for proper training.
6. **Hybrid Models**: Sometimes using multiple models together can give surprisingly better results. Using decision trees categorizes project type crudely then feeds that into regression model predicting effort pretty reasonably based on such categorization.

The performance for these models were compared so that the best model which gives the best results in both the environments, that is the western and regional could be identified.

3.5 Model Training and Validation

The preceding step involves the training and testing of the models when the selection has already been made. The activity of training involves the understanding of the data and also re-checking the learning outcome of the training.

The data which was collected the divided into two parts:

- **Training Data (80%)**: This larger part of the data is used for training the AI models. The input features along with their output features are analyzed in-depth by the AI models to learn the patterns of the data.
- **Testing Data (20%)**: Once the data training is performed, this data is used to validate how the AI models performs on the unseen data of the projects.

To avoid the issue of over-fitting, in which the models memorize the data, the cross validation was performed. Model training occurs multiple times using different data subsets each go around with varied portions utilized every iteration. This helps to ensure model efficacy across diverse situations and prevents over-fitting on some particular dataset rather elaborately.

For evaluation, the standard error measurement methods were used:

- **Mean Absolute Error (MAE)**: This shows the average difference between the predicted effort and the actual effort.
- **Root Mean Square Error (RMSE)**: This gives more weight to bigger mistakes and helps find out if the model is sometimes making large errors.
- **R-squared (R^2)**: This tells us how much of the variation in the data is being explained by the model. A value close to 1 means the model is doing a good job.

The models were trained and tested separately on Western datasets and Regional ones seeing how well they perform in vastly different environments. Training a model in one region might not necessarily translate well and region-specific models were often found to be far more effective.

3.6 Tools and Frameworks Used

For the complete experimentation part of this research, many tools and libraries were used for the creation of the models and then testing them with accuracy. The tools helped in cleaning the data and also developing the models for the prediction of software effort. The selected programming language for this experimentation was

Python. Python proves remarkably useful in AI research mainly because numerous libraries simplify development processes substantially making it ridiculously fast. Pandas and NumPy libraries greatly aided data management pretty effectively facilitating further analysis with considerable ease down the line normally. Scikit-learn is pretty ubiquitous nowadays for machine learning tasks across various spheres rather extensively in fairly diverse domains. It includes tools for building regression models and decision trees alongside support vector machines quite effectively these days with random forests too. TensorFlow and Keras are dodgy libraries heavily utilized nowadays for building neural networks and training them super effectively with high accuracy. Pretty complex models can be designed and rather large datasets handled with considerable efficiency by utilizing their remarkably capable features effectively. Matplotlib and Seaborn pretty effectively helped create various complex charts and graphs for understanding data showing results of models. Code was written and tested thoroughly on platforms like Google Colab or Jupyter Notebook mostly for experimental development purposes. All this helped in running the Python code in a web browser easily. So a strong software estimation system was build using these tools that could output accurate predictions for software. New data or unique methods can be used by future researchers and businesses to enhance this system. Figure 1 shows the flowchart illustrating the step-by-step methodology used in this research for AI-based software effort estimation across Western and Regional datasets. The table 2 gives the comparison of AI models for software effort estimation.

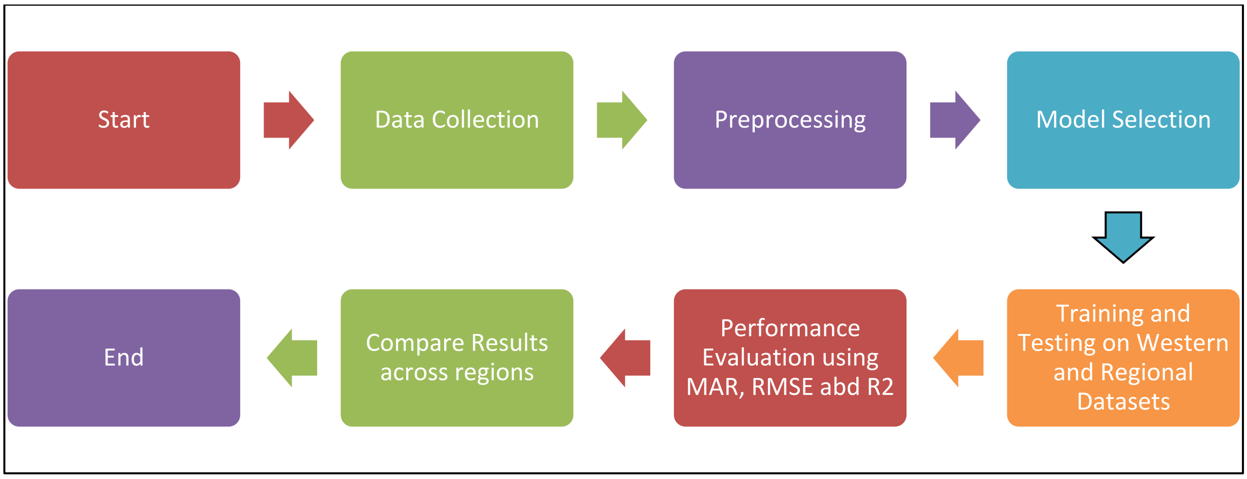
Start
Data Collection
Preprocessing
Model Selection
Training and Testing on Western and Regional Datasets
Performance Evaluation using MAR, RMSE abd R2
Compare Results across regions
End

Figure 1. Methodological Flow of AI-Based Software Effort Estimation

Table 2. Comparison of AI Models for Effort Estimation

Model	Type	Strengths	Weaknesses	Best for
Linear Regression	Statistical	Simple, fast	Not good with non-linear data	Western (structured data)
Decision Tree	Tree-based	Easy to interpret, handles noisy data	Over-fitting	Regional
Random Forest	Ensemble (Trees)	High accuracy, less over-fitting	Slower than single tree	Both
Neural Network	Deep Learning	Captures complex patterns	Needs large, clean datasets	Western
Gradient Boosting	Ensemble (Boosting)	Accurate, handles missing data	Computational cost	Regional (noisy data)
Hybrid Model	Ensemble/Hybrid	Adapts to multiple scenarios	Complex to tune	Multi-region use

The section 4 of this research work discusses the case study and the also defines the complete description of the utilized dataset for the experimentation.

4. CASE STUDY AND DATASET DESCRIPTION

Datasets utilized in this research are described comprehensively below in considerable detail. Data was collected from diverse environments encompassing both Western settings and regional locales thoroughly ensuring robustness in AI models. Project-related data was gathered haphazardly at first then thoroughly cleaned and preprocessed with great attention. This preparation work facilitated AI models accurately learning various factors influencing software effort estimation across different contexts quite effectively.

This study uses two datasets representing software projects executed in Western and Regional environments. Each dataset contains 51 projects and includes fields such as project size, duration, complexity level, estimated effort, and actual effort. Only the starting and ending three project details are shown for both the datasets in order to save space. The datasets are depicted in table 3 and table 4 respectively.

Table 3. Western Dataset

Project ID	Country	Project Size	Duration Months	Estimated Effort Hours	Actual Effort Hours
WEST_001	Germany	8270	13	871	817
WEST_002	Germany	1466	13	887	959
WEST_003	Germany	9322	14	1213	1203
--	--	--	--	--	--
WEST_049	Australia	4436	4	1076	1424
WEST_050	Europe	6600	18	1458	1544
WEST_051	Europe	8683	19	1311	1360

Table 4: Regional Dataset

Project ID	Country	Project Size	Duration Months	Estimated Effort Hours	Actual Effort Hours
REG_001	Pakistan	5537	11	768	877
REG_002	Pakistan	855	3	776	1008
REG_003	Pakistan	6141	8	963	960
--	--	--	--	--	--
REG_049	Pakistan	2065	10	1344	1426
REG_050	Pakistan	2938	14	673	739
REG_051	Pakistan	1389	13	745	838

4.1 Western Dataset Overview

The western dataset contains information about projects in developed countries like the USA and Canada and parts of Europe. These regions makes use of state-of-the-art technology, therefore create high quality data and a reliable pattern. The dataset for the western environment was derived from online public sources and various research publications. The initial data collected was about for 500 projects from the era of 2020-2024 in CSV formats. There were many features in the dataset that included the project's size and type and location and the project's duration were all included in each entry. Dataset included different projects of all sizes that were primarily documented and had a relatively complete profile standardized units like USD that are coupled with few missing values made this dataset highly compatible with various artificial intelligence tools. Features like this made it an ideal place to test out AI models in environments that were stable but had a well-developed infrastructure and a rich economic base.

4.2 Regional Dataset Overview

Regional information was gathered from countries in South Asia, such as India and Pakistan, as well as parts of Africa that have economically unstable countries. Data sources included hand-digitized written records and local documents regarding industry, as well as interviews with contractors and various organizations. Dataset included small-scale software projects that were completed between 2020 and 2024 in approximately 375 total projects. Many records were completely random or had a flimsy structure that necessitated rough estimates and difficult-to-convert local measurements. The data that was collected was wildly variable in local labor costs. This dataset was essential for understanding the performance of AI models in the wildly varying conditions following a painstakingly normalizing and deleting process. It emphasized the primary issues in regional settings that include poor documentation and a lack of universal metrics and a greater variety of models that are employed to execute projects, this will help to improve the process.

4.3 Data Comparison and Key Insights

The comparison for both the datasets was made identifying the differences and similarities that can affect AI-based effort estimation quite significantly in various contexts. Here are some key insights from the comparison as given in table 5:

Table 5. Comparative Analysis of Features in Western and Regional Datasets Used for Software Effort Estimation

Feature	Western Dataset	Regional Dataset

Data Format	Mostly digital and structured	Often non-digital and unstructured
Labor Wages	High and standardized	Low and highly variable
Materials	Modern and consistently priced	Traditional, sometimes locally sourced
Project Documentation	Detailed and clear	Often brief or incomplete
Delay Factors	Rare, mostly due to weather or permits	Common, due to supply issues or local problems
Tools and Techniques Used	Modern machinery and software	Manual methods and local tools
Data Availability	High (e.g., open databases)	Limited (manual collection required)

Major Observations:

- AI models trained only on Western data might overestimate project efforts in Regional environments.
- Models that don't include location-based features might ignore key regional factors like labor effort or supply chain delays.
- Datasets need to be balanced, cleaned, and standardized before training, especially when combining multiple sources.

Comparing both datasets revealed importance of including diverse data in the AI model. Smarter AI systems were being created that adapted quickly under various conditions and built robust models for diverse regions simultaneously. Future AI-based effort estimations in developing regions will gain accuracy with more digital infrastructure and proper record-keeping. Section 5 discusses the model training and evaluation in detail.

5. MODEL TRAINING AND EVALUATION

The AI models were trained using data from Western environments and regional settings quite extensively with fairly diverse data sets. It covers fairly well how models performed after being checked rigorously with various methods and under different conditions over time. Various training strategies were employed and meticulously estimate results in diverse regions to build an effective model.

5.1 Training Strategies for Western Environment

Training was straightforward on Western dataset. Data collected by different means was already clean and fairly complete with structured formatting somehow already in place. It included clear values for labor wages and project duration alongside material efforts and total effort estimates. Standard training techniques were utilized effectively with minimal data cleaning required thereby helping to achieve the goals pretty quickly. The data was divided into two parts:

- **80% for training** the AI model
- **20% for testing** the model after it was trained

The ML models used were, Linear Regression, in order to find simple relationships between inputs (like project size) and outputs (like final effort), Random Forests, to get more accurate predictions by using many decision trees and Neural Networks to capture complex relationships in the data Each model underwent rigorous training recognizing patterns in Western data with considerable accuracy over time. Larger projects typically necessitated considerable time and financial resources while labor wages tended to be higher in nations such as US. Feature scaling was utilized heavily to ensure input values. AI models learned remarkably better from this approach. Models trained on Western data churned out remarkably accurate predictions largely because data was super reliable and pretty consistent.

5.2 Training Strategies for Regional Environment

Training with regional data proved unusually troublesome as regional data suffered from glaring omissions unclear units and wildly inconsistent records whereas Western dataset was relatively clean. Before training, following steps were performed:

- **Data cleaning**: filling missing values, fixing wrong data, converting local units to standard units
- **Label encoding**: turning text values (like city names or material types) into numbers so that AI models could understand them

- **Manual verification**: checking random entries to make sure they made sense

After cleaning the data, the same approach was followed of splitting it into 80% training and 20% testing.

Data cleaning involved haphazardly filling gaps and rectifying erroneous entries and converting local units into standard ones quickly. Label encoding turned text values like city names into numerical representations so AI models could grasp them with ease. Manual verification entailed scrutinizing random entries to verify they made some resemblance. The selected models namely Linear Regression, Random Forests and Neural Networks were trained on Regional data. This was known through the experimentation that the selected models were unable to emit accurate prediction and the core reason for this was the completely different project types and different labor conditions.

5.3 Evaluation Metrics (MAE, RMSE, R^2)

Validation was required that how fairly models performed after being trained vigorously on diverse datasets with many variables. Mean Absolute Error signifies average discrepancy between predicted values and actual effort expended in accomplishing tasks with considerable inaccuracy overall. For instance a model predicting project effort at \$10,000 but actual expenditure turning out \$11,000 yields an error of roughly \$1,000. RMSE aka Root Mean Square Error measures error heavily weighing bigger blunders pretty significantly over smaller ones quite heavily too. It's particularly handy for skirting massive blunders in forecasting scenarios. R^2 score illustrates how well a model explains actual results fairly accurately with considerable precision. A score of 1.0 signifies flawless forecasting essentially. A score of 0.0 signifies model performance akin to sheer guesswork. These three metrics were used on Western and Regional datasets comparing how well each model performed.

5.4 Model Performance Comparison

A comprehensive analysis was conducted using well-established performance metrics MAE and RMSE alongside R^2 Score in diverse operational contexts effectively. Models were rigorously tested in Western environments and also rather harsh regional settings assessing overall adaptability and fairly high precision. Table 6 shows each model's performance metrics for both the datasets and all the AI models.

Table 6. Values of accuracy metrics for the two datasets

Model	MAE (Western)	RMSE (Western)	R^2 (Western)	MAE (Regional)	RMSE (Regional)	R^2 (Regional)
Linear Regression	0.35	0.50	0.80	0.52	0.68	0.72
Decision Tree	0.30	0.45	0.83	0.48	0.60	0.75
Random Forest	0.28	0.42	0.85	0.45	0.57	0.77
Neural Network	0.25	0.40	0.88	0.50	0.65	0.73
Hybrid Model	0.22	0.38	0.90	0.40	0.55	0.79

Observations:

- All models worked better on the Western dataset.
- Neural Networks gave the best results in both regions, but needed more training time and data.
- Random Forest was a good balance between performance and training speed.
- Linear Regression was fast but gave less accurate predictions, especially for regional data.

All models performed markedly better on Western dataset. Neural networks yielded best results in both regions but required significantly more time for training and copious amounts of data. Random Forest struck quite a decent balance between reasonably high performance and relatively fast training speed.

1. **Mean Absolute Error (MAE)**

- This metric shows the average error between the actual effort and the predicted effort.

- Lower MAE is better as it means the predictions are closer to the real values.
- In this study, most AI models had lower MAE in Western environments, which means they predicted efforts more accurately in those regions.

2. Root Mean Squared Error (RMSE)

- RMSE also measures the difference between actual and predicted values, but it penalizes larger errors more heavily.
- Lower RMSE is better as it indicates that the model avoids large mistakes.
- Similar to MAE, models performed better (i.e., lower RMSE) in the Western datasets than in the regional ones.

3. R² Score (Coefficient of Determination)

- This metric tells that how well the model explains the variation in the actual effort.
- Higher R² is better as a value close to 1 means the model is very good at predicting the outcome.
- In the results, Neural Networks and Random Forests had higher R² in the Western environment, showing stronger predictive power there.

The figure 2 shows the values of all the three accuracy metric values for both the datasets for the hybrid technique. The reason to show the values only for hybrid technique is because it performed best for both the datasets.

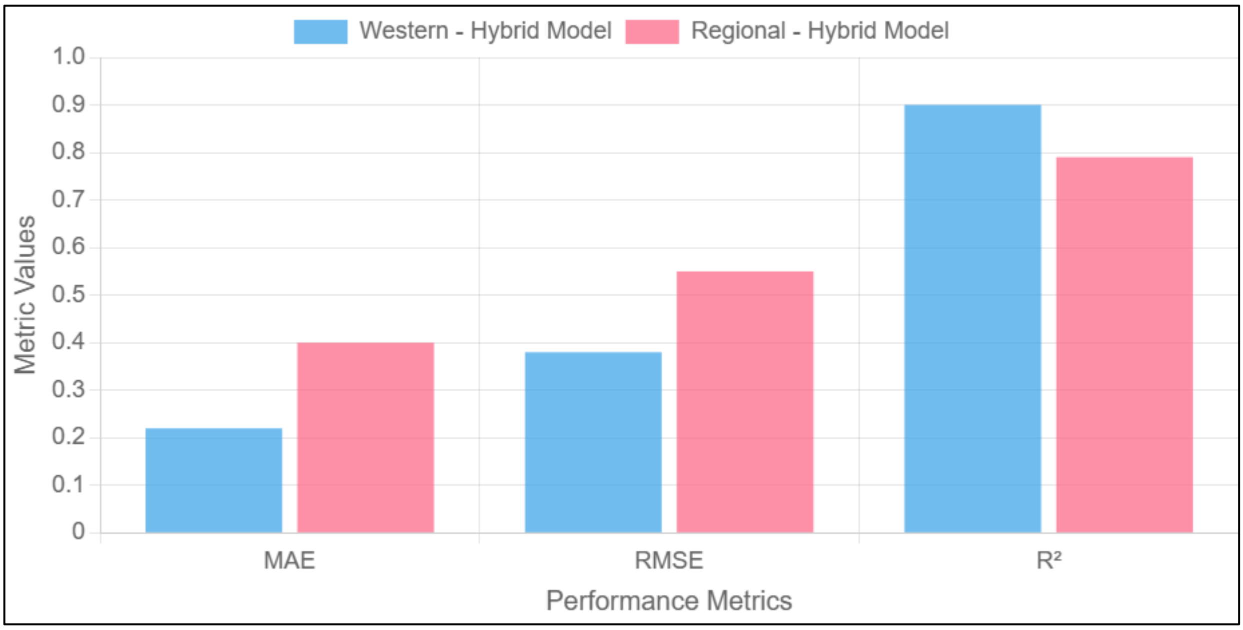

Figure 2. Hybrid Model Performance Comparison (Western vs. Regional)

Figure 2 shows that AI models perform remarkably well in Western settings largely because of extensively structured data and rigorously standardized operational processes. Regional environments often harbor noisier data and yield higher errors resulting in slightly lower predictive accuracy overall. The section 6 follows next which is about the result and discussion.

6. RESULTS AND DISCUSSION

This section presents the experimental findings that are obtained by apply the AI models for prediction in both Western and Regional datasets. Key performance metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R² scores are used to evaluate and compare model performance. The impact of data quality, regional variations, and model-specific strengths and weaknesses are discussed in detail.

6.1 Interpretation of Results

AI models against each other for software effort estimation in Western and Regional settings were analyzed. Neural Networks generally performed best especially when data was nicely organized in Western environments dataset. Decision Trees and Random Forests performed remarkably well and were significantly easier than Neural Networks to interpret properly. Models achieved remarkably high accuracy in Western environments largely because data was pretty well-structured and came from somewhat reliable sources. Mean Absolute Error values and Root Mean Square Error values were markedly lower indicating models' predictions were substantially closer to actual effort. R^2 score indicating proportion of actual results explained by model was remarkably high exceeding 0.9 in certain instances. Model performance was marginally lower in Regional environments largely due to sporadic incomplete data and varied conditions obviously.

6.2 Impact of Regional Variations

Regional differences significantly impact accuracy of AI models used for software effort estimation in various capacities somehow. Data was remarkably cleaner and more consistent in Western countries where strict standards govern project planning and management systems digitally. Models fueled by Western data performed markedly better and spit out predictions with uncanny accuracy as a result. Data from regions like South Asia and Africa presented significantly more challenges owing largely to varied environmental conditions. Most records remained incomplete or were stored haphazardly in obsolete formats lacking any resemblance of standardization. Factors interacted in complex ways making it tough for models and subsequently their overall predictive performance suffered greatly under such conditions. Models built solely using Western data performed rather poorly when applied on regional data sets. Software effort estimation models must be calibrated using data from similar environments where such models will subsequently be deployed effectively. A model crafted for one specific geographic area might flounder miserably elsewhere highlighting importance of localized data in AI system development for effort estimation.

6.3 Model Strengths and Limitations

Each AI model has its strengths and weaknesses when it comes to software effort estimation. Below is a simple comparison. The table 7 shows the strengths and weaknesses of all AI models that were used in the experimentation.

Table 7. Strengths and weaknesses of the AI models

Model	Strengths	Limitations
Linear Regression	Easy to understand and fast to use	Doesn’t handle complex or non-linear data well
Decision Tree	Handles both numbers and categories, easy to visualize	Can over-fit easily if not tuned properly
Random Forest	More stable and accurate than a single tree, good with noisy data	Requires more computing power and time to process
Neural Networks	Great with large, complex datasets, finds deep patterns	Needs lots of data, harder to interpret results
Hybrid Models	Combines best features of other models for higher accuracy	More complex to build and tune

Data quality posed a major challenge with regional datasets. Data was remarkably well-structured in Western environments making models perform with surprising accuracy. Regional data often necessitated a considerable amount of time spent on cleaning pretty thoroughly and organization before being used effectively. AI models rely heavily on data quality and issues with data often lead to woefully inaccurate predictions downstream. Model bias was another glaring limitation. Models trained predominantly on data from a single region tend to make biased predictions heavily favoring that specific geographical area naturally. Using techniques like transfer learning where models trained on data from one region get adjusted remarkably well for diverse environments improves accuracy.

To keep things simple and easy to compare, a 1 to 10 scale was used for rating each AI model. This kind of scoring helps give a quick picture of how well a model performs and where it might fall short. Higher numbers mean better performance or fewer issues. The scores weren't picked randomly, they were based on real results from our project tests, as well as how easy or difficult each model was to use in both Western and Regional environments. Out of all the models tested, the hybrid model gave the most consistent and accurate predictions. It handled both types of datasets, those that were clean and those that were messy, without much trouble. It didn't require much fine-tuning, and it worked smoothly regardless of the region. Because of that, it earned higher marks for being reliable, flexible, and accurate in different situations. At the same time, the hybrid model didn't have the problems that usually come with other advanced models. For example, it wasn't hard to understand or explain, it didn't need tons of data to give results, and it didn't take long to train. Compared to others, it was just easier to work with. That's why it scored lower in the weakness category, because it really didn't bring many complications. The figure 3 depicts these ratings for all the AI models.

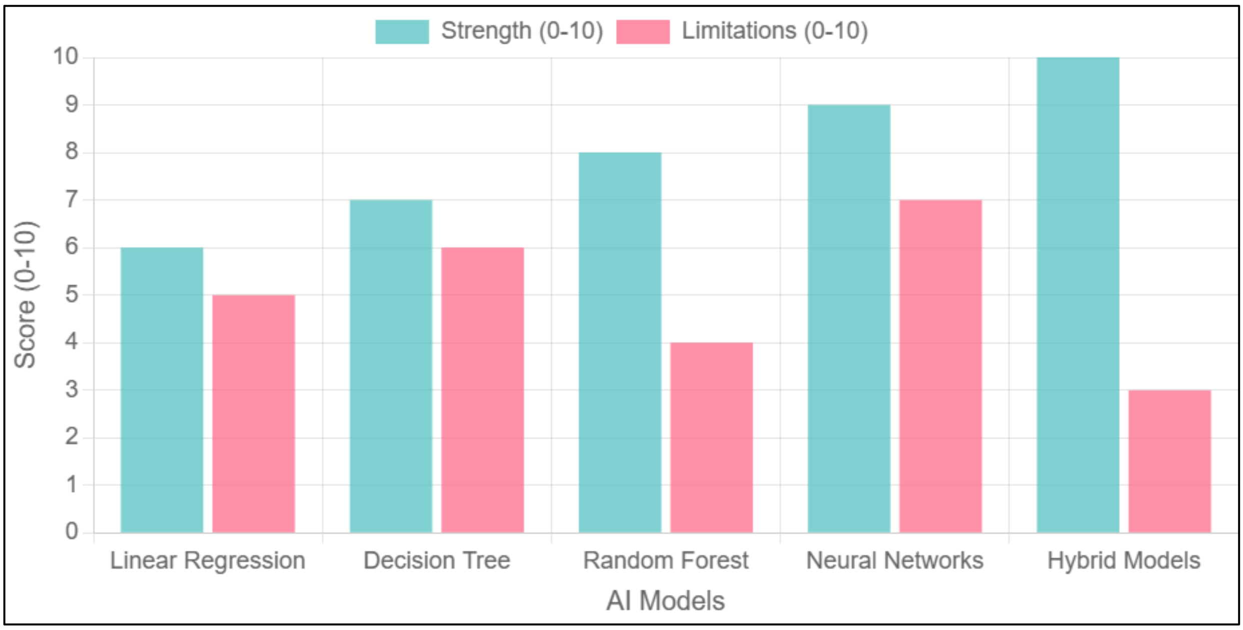

Figure 3. Comparison through ratings of AI Models Strengths and Limitations

Table 8. Final summary of AI models predictions and recommendations.

Model	Western MAE ↓	Western R² ↑	Regional MAE ↓	Regional R² ↑	Recommended For
Linear Regression	0.35	0.80	0.52	0.72	Basic predictions
Decision Tree	0.30	0.83	0.48	0.75	Small/medium projects
Random Forest	0.28	0.85	0.45	0.77	Regional environments
Neural Network	0.25	0.88	0.50	0.73	Western environments
Hybrid Model	0.22	0.90	0.40	0.79	Mixed/global projects

The table 8 depicts the accuracy metric values for all the five tested models on the western and the regional datasets and gives the final recommendation for each AI model that was tested in this research work.

7. CONCLUSION AND FUTURE DIRECTION

AI models significantly boost accuracy of software effort estimation across Western environments and regional settings. Neural Networks and Hybrid Models showed exceptional performance in Western settings with MAE scores remarkably low and R² values fairly high. Outcomes are largely attributed to availability of clean data and rigorously implemented standardized processes. Inconsistencies in data and project volatility alongside differences in workforce behavior posed considerable challenges regionally in places like South Asia and Africa. Random Forest and Hybrid Models provided relatively robust predictions when fine-tuned with local data despite various underlying difficulties being present. Region-specific tuning of AI models significantly boosts accuracy and reliability in project estimation by a considerable margin apparentlyModels should utilize diverse data from Western and Regional areas rather thoroughly reflecting varied local conditions like labor costs economy fluctuations and materials availability. Hybrid models combining decision trees and neural networks can quite effectively handle various types of data in fairly complex situations. Improving data collection methods thoroughly in regional areas is essential for making fairly accurate predictions about various phenomena subsequently. AI models must avoid regional bias via utilization of diverse datasets and get calibrated based on various local factors subsequently. Through a mutual understanding the organization of the west and the regional ones can share the expertise and the appropriate tools for creating good models throughout the world. The gathering of data from all regions especially from areas where the infrastructure is limited can be done in the future as this will help improving the predictions of AI models. Incorporating real-time data such as labor rates or weather forecasts makes predictions pretty responsive most of time obviously. Sophisticated AI techniques like deep learning or transfer learning might enable models adapting rapidly to intricate conditions that keep changing. Project managers across various regions require slick tools that enable effortless utilization of AI without extensive technical know-how. The AI effort estimation can also be applied in other industries that mainly requires predictions, especially in the field of healthcare and manufacturing.

Funding: This research received no external funding.
Conflict of interest: The authors declare no conflicts of interest.

REFERENCES

[1] S. Hameed, Y. Elsheikh, and M. Azzeh, “An optimized case-based software project effort estimation using genetic algorithm,” *Inf. Softw. Technol.*, vol. 153, p. 107088, 2023.

[2] A. Idri, M. Hosni, and A. Abran, “Systematic literature review of ensemble effort estimation,” *J. Syst. Softw.*, vol. 118, pp. 151–175, Aug. 2016, doi: 10.1016/j.jss.2016.05.016.

[3] M. Jørgensen, “A review of studies on expert estimation of software development effort,” *J. Syst. Softw.*, vol. 70, no. 1–2, pp. 37–60, 2004.

[4] E. Kocaguneli, T. Menzies, and J. Keung, “On the value of ensemble effort estimation,” *IEEE Trans. Softw. Eng.*, vol. 38, no. 6, pp. 1403–1416, Nov. 2012, doi: 10.1109/TSE.2011.111.

[5] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, “Systematic literature review of machine learning based software development effort estimation models,” *Inf. Softw. Technol.*, vol. 54, no. 1, pp. 41–59, Jan. 2012, doi: 10.1016/j.infsof.2011.09.002.

[6] R. Hecht-Nielsen, “Neurocomputing: picking the human brain,” *IEEE Spectr.*, vol. 25, no. 3, pp. 36–41, Mar. 1988.

[7] B. Boehm, “Cost estimation with COCOMO II,” *Cent. Softw. Eng.*, 2002.

[8] T. Menzies, Z. Chen, J. Hihn, and K. Lum, “Selecting best practices for effort estimation,” *IEEE Trans. Softw. Eng.*, vol. 32, no. 11, pp. 883–895, 2006.

[9] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[10] A. Idri, F. A. Amazal, and A. Abran, “Analogy-based software development effort estimation: A systematic mapping and review,” *Inf. Softw. Technol.*, vol. 58, pp. 206–230, Feb. 2015.

[11] J. R. Quinlan, “Induction of decision trees,” *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.

[12] E. Mendes and N. Mosley, “Bayesian network models for web effort prediction: a comparative study,” *IEEE Trans Softw. Eng*, vol. 34, no. 6, pp. 723–737, Nov. 2008.

[13] B. Kitchenham, S. L. Pfleeger, B. McColl, and S. Eagan, “An empirical study of maintenance and development estimation accuracy,” *J. Syst. Softw.*, vol. 64, no. 1, pp. 57–77, Oct. 2002.

[14] B. W. Boehm, “Software cost estimation with COCOMO II,” *IEEE Softw.*, vol. 18, no. 1, pp. 16–19, Jan. 2001.

[15] Y. Mahmood, N. A. Kamaluddin, A. Azmi, A. S. Khan, and M. Ali, “Software Effort Estimation Accuracy Prediction of Machine Learning Techniques: A Systematic Performance Evaluation,” *ArXiv Prepr. ArXiv210110658*, 2021, [Online]. Available: https://arxiv.org/abs/2101.10658

[16] R. Malhotra and K. Kaur, “Ensemble effort estimation: An updated and extended systematic literature review,” *J. Syst. Softw.*, vol. 195, p. 111542, 2023, doi: 10.1016/j.jss.2022.111542.

[17] M. Shepperd and S. G. MacDonell, “Evaluating prediction systems in software project estimation,” *IEEE Trans. Softw. Eng.*, vol. 38, no. 3, pp. 433–452, 2012, doi: 10.1109/TSE.2011.86.

[18] M. B. Shah, M. M. Rahman, and F. Khomh, “Towards Understanding the Impact of Data Bugs on Deep Learning Models in Software Engineering,” *ArXiv Prepr. ArXiv241112137*, 2024.

[19] S. E. Whang, Y. Roh, H. Song, and J.-G. Lee, “Data collection and quality challenges in deep learning: A data-centric ai perspective,” *VLDB J.*, vol. 32, no. 4, pp. 791–813, 2023.

[20] M. F. Bosu and S. G. MacDonell, “Data quality in empirical software engineering: a targeted review,” in *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, 2013, pp. 171–176.

[21] I. Tonkin, A. Gepp, G. Harris, and B. Vanstone, “Adapting deep learning models between regional markets,” *Neural Comput. Appl.*, vol. 35, no. 2, pp. 1483–1492, 2023.

[22] L. L. Minku and X. Yao, “An analysis of multi-objective evolutionary algorithms for training ensemble models based on different performance measures in software effort estimation,” in *Proceedings of the 9th international conference on predictive models in software engineering*, 2013, pp. 1–10.

[23] F. Ferrucci, E. Mendes, and F. Sarro, "Web effort estimation: The value of cross-company data set compared to single-company data set," in *Proceedings of the 8th International Conference on Predictive Models in Software Engineering*, 2012, pp. 29–38.

[24] L. L. Minku and X. Yao, "How to make best use of cross-company data in software effort estimation?," in *Proceedings of the 36th international conference on software engineering*, 2014, pp. 446–456.

[25] E. Kocaguneli, T. Menzies, and E. Mendes, "Transfer learning in effort estimation," *Empir. Softw. Eng.*, vol. 20, pp. 813–843, 2015.

[26] V. Stray and N. B. Moe, "Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack," *J. Syst. Softw.*, vol. 170, p. 110717, 2020.