Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 270 (2025) 2848–2857

Procedia Computer Science

www.elsevier.com/locate/procedia

29th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2025)

# Project Similarity Measures for Collaborative Filtering-based Effort Estimation: Review and Empirical Study

Ho Le Thi Kim Nhung[a,*], Radek Silhavy[a], Petr Silhavy[a]

[a]*Faculty of Applied Informatics, Tomas Bata University in Zlin, 760 01 Zlin, Czech Republic*

**Abstract**

As software project development becomes increasingly complex, accurate effort estimation is essential for successful delivery. This study investigates the impact of similarity measures on estimation accuracy within the Neighborhood-Based Collaborative Filtering for Effort Estimation (NCFEE) context. We analyzed the performance of 17 similarity measures using benchmark datasets, specifically fpa_china and fpa_isbsg. Effectiveness was assessed through Root Mean Squared Error (RMSE) to quantify prediction accuracy, supplemented by effect size analysis to gauge the practical significance of observed differences. The results demonstrate that Jaccard-based measures (JAC, DiceJAC, and TanimotoJAC) consistently achieved the lowest RMSE values, indicating their strong ability to capture effort-related similarities by focusing on overlapping project features. Effect size analysis confirmed that these performance advantages are highly practically significant. Furthermore, the optimal number of nearest neighbors varied between datasets, with effect sizes highlighting the substantial impact of dataset characteristics on model performance. These findings underscore the importance of selecting appropriate similarity measures, particularly Jaccard-based approaches, to enhance the effectiveness of NCFEE.



*Keywords:* Software effort estimation; Neighbor-based Collaborative Filtering; Similarity Measures

## 1. Introduction

The complexity of software project development has increased significantly, necessitating a high level of expertise among software professionals. Project managers endeavor to accurately estimate key parameters such as cost, effort, duration, and defect density to ensure timely and budget-conscious delivery of software products [1]. This underscores the critical importance of Software Development Effort Estimation (SDEE) methods for the success of software projects.

Accurate SDEE is critical for effective resource allocation and financial planning, directly influencing strategic decision-making and project outcomes. Recent advancements in machine learning, collaborative filtering, and con-

* Corresponding author. Tel.: (+420) 774886798
*E-mail address:* lho@utb.cz

1877-0509 
10.1016/j.procs.2025.09.407

textual data integration have considerably enhanced estimation accuracy in this field [2]. Collaborative Filtering (CF) has emerged as a promising alternative for estimation based on the principle that entities (such as projects) exhibiting similar historical patterns will likely continue to behave similarly [3]. CF leverages historical project data to predict the required effort for new projects when applied to software effort estimation.

Neighbor-based Collaborative Filtering Effort Estimation (NCFEE) specifically tailors this broader approach for effort estimation [4]. It identifies a set of $k$ neighboring past software projects resembling the target project. At the core of the NCFEE methodology lies the task of accurately quantifying similarity between projects. This is achieved through various similarity measures that mathematically assess resemblance based on distinct attributes. The choice of similarity measure is critical, directly impacting the reliability and accuracy of effort estimations. Each measure captures different facets of project-relatedness, leading to significant variations in predictive performance.

Distance-based similarity measures, including Euclidean, Manhattan, Chebyshev, and Minkowski distances, calculate geometric distances between project vectors in multidimensional feature space, where smaller distances indicate more significant similarity [5]. Despite the widespread use of these traditional distance measures in NCFEE, there is a growing recognition of the need to explore alternative similarity measures. Traditional measures may impose limitations, especially when project data exhibit non-normal distributions or complex project characteristics.

To address these challenges, this study aims to conduct an in-depth theoretical review of various similarity measures applicable to NCFEE, highlighting their theoretical foundations and practical relevance. Specifically, we will explore similarity measures traditionally used in neighborhood-based CF, which are widely applied in recommender systems. In this context, while recommendation systems compare users based on shared ratings, we will compare projects by identifying relevant attributes to determine similarity, specifically among projects with similar functional requirements, technical complexity, team composition, and development metrics. Additionally, we will empirically evaluate the performance of these measures using consistent datasets and a unified estimation methodology, emphasizing the critical relationship between estimation accuracy and the choice of similarity measure. To guide our investigation, we have formulated the following research questions:

- **RQ1:** Which similarity measures commonly used in neighbor-based collaborative filtering can be effectively applied in NCFEE?
- **RQ2:** How do different similarity measures impact effort estimation accuracy in NCFEE?

Below, we outline the specific contributions of this study:

- **Theoretical Review:** We comprehensively review similarity measures within neighbor-based collaborative filtering, emphasizing their relevance and utility in NCFEE.
- **Empirical Comparative Study:** An empirical study is conducted to identify the most effective similarity measures for NCFEE, using benchmark datasets like the International Software Benchmarking Standards Group (fpa_isbsg) [6] and the China dataset (fpa_china) [7].
- **Impact of Measure Selection:** Our exploration of the relationship between the choice of similarity measure and estimation accuracy highlights the critical importance of selecting appropriate measures in NCFEE, paving the way for improved estimation practices in software development.

The remaining sections are structured as follows: Section 2 reviews commonly used similarity measures for effort estimation. Section 3 outlines the NCFEE methodology, evaluation measures, and datasets. Section 4 presents the results and discusses their implications. Finally, Section 5 concludes the study and proposes future research directions.

## 2. The function of similarity measures for effort estimation

Similarity measures are essential to NCFEE, as they identify past projects that serve as the most relevant *neighbors* for effort estimation. The selection of a specific similarity measure significantly influences the subsequent steps in NCFEE, determining which projects are considered similar and whose effort values are used to estimate the target project's effort. These measures compare project instances represented as vectors of features and use mathematical formulas to generate a similarity score. A higher or lower score indicates a more significant similarity, depending on

the measure. The following sections present the similarity functions used for effort estimation, with detailed formulas in Table 1.

### 2.1. Distance-based similarity measures

Distance-based measures calculate the geometric distance between project vectors in a multi-dimensional feature space. In these measures, a smaller distance indicates more remarkable similarity [8]. The distance is computed using specific mathematical formulas, with parameter $d$ variations resulting in different distance functions. The general formula for the distance between two projects, $u$ and $v$, in an n-dimensional space, is presented in Eq. (1).

$$sim(u, v) = \left( \sum_{i=1}^{n} |u_i - v_i|^d \right)^{\frac{1}{d}} \tag{1}$$

where $u_i$ and $v_i$ denote the attribute values for projects $u$ and $v$, $n$ is the number of attributes used to evaluate the relationship between the two projects, and $d$ indicates the type of distance metric utilized.

The Euclidean similarity measure (EUC), also known as the 2-norm, quantifies the straight-line distance between two points in Euclidean space, serving as a fundamental metric for assessing similarity based on geometric distance [9]. The Hamming similarity measure (HAM) assesses dissimilarity by counting the positions at which two corresponding elements differ, making it particularly useful for categorical data [10]. The Manhattan similarity measure (MHT), or 1-norm, often referred to as city block distance, sums the absolute differences between the Cartesian coordinates of two projects, representing the total distance required to traverse along axes at right angles [11]. In contrast, the Chebyshev similarity measure (CHE) evaluates similarity by identifying the maximum difference in coordinates between two points, focusing on the worst-case distance along any coordinate dimension [12]. Additionally, the Minkowski similarity measure (MKS) generalizes both EUC and MHT by incorporating parameter $d$ [13], which allows for flexibility in distance evaluation based on context. For instance, when $d = 1$, it corresponds to MHT, and when $d = 2$, it corresponds to EUC. The Akritean similarity measure (AKR) [14] is a hybrid metric combining the advantages of EUC and MHT similarity measures. It tends to yield better results in scenarios where data points are sparse or widely separated, while MHT performs better when points are closely packed together or densely distributed. Therefore, the AKR measure is particularly beneficial in environments where the distribution of points (or projects) varies significantly.

### 2.2. Pearson Correlation Coefficient-based similarity measure and its extensions

The Pearson Correlation Coefficient (PCC) [15] is one of recommender systems' most widely used traditional similarity measures. It quantifies the linear correlation between the attribute values of two entities, such as users or items, making it a powerful tool for assessing similarity based on recorded ratings. In the NCFEE context, we recognize that the PCC can be effectively adapted to evaluate project similarities by treating various project attributes as the variables to be compared [4], as shown in Eq. (10).

Several extensions have been proposed in the literature to enhance the traditional PCC, allowing for more accurate similarity assessments. We recognize that two specific extensions can be effectively adapted for NCFEE, particularly in scenarios where project attributes exhibit variability or their distributions differ significantly. The WeightedPCC [16] modifies the traditional PCC by applying weights that reflect the significance of each project attribute based on its prevalence and importance in the dataset. Specifically, the weight for an attribute can be calculated using the inverse frequency method, which has more influence on less common attributes across projects. This approach results in a more reliable similarity measure in diverse project evaluations, as it effectively emphasizes attributes that provide critical insights into project characteristics. Another relevant extension is the ModifiedPCC [17]. This adaptation centers the attribute values around their means and incorporates additional normalization techniques. This process improves the robustness of similarity assessments against data quality issues and scale discrepancies, making the correlation measure less sensitive to outliers and inconsistencies. ModifiedPCC offers a more precise and accurate evaluation of project similarities by effectively considering the context of project attributes.

### 2.3. Cosine-based similarity measure and its extensions

The Cosine similarity measure (COS) [18] can be effectively adapted for NCFEE by treating projects as vectors of their respective attributes. In this context, each project is represented as a vector where each dimension corresponds to a specific project attribute. This approach enables the calculation of similarity between projects based on their attributes, facilitating an assessment of how closely aligned the projects are concerning their characteristics, as shown in Eq. (11). We recognize that no extensions are directly applicable due to the inherent differences in data structures between user-based systems and project-based assessments. Thus, while Cosine similarity is a robust method for assessing project similarity in SDEE, the applicability of its extensions requires careful consideration to ensure they align with the specific characteristics and needs of project evaluations.

### 2.4. Jaccard-based similarity measure and its extensions

The Jaccard similarity measure (JAC) [19] is a widely recognized metric for quantifying the similarity between two entities based on their attributes. Originally developed for recommendation systems, the Jaccard measure evaluates the intersection and union of attribute sets, providing a straightforward method for assessing how closely two users are related. In the context of NCFEE, JAC calculates the ratio of common characteristics to the total number of unique attributes across both projects, as presented in Eq. (12). This measure quantifies the similarity between two projects by comparing the intersection of their attribute sets to their union.

Several extensions of the Jaccard similarity measure (JAC) have been proposed, enhancing its applicability and effectiveness for evaluating similarities in Software Development Effort Estimation (SDEE). Notably, the Dice Coefficient (DiccJAC) [20], Tanimoto Index (TanimotoJAC) [21], and Triangle Multiplying Jaccard (TMJ) [22] can all be adapted for use in NCFEE. The DiccJAC evaluates the similarity of attribute distributions between two projects, providing insight into how closely aligned their characteristics are. This measure emphasizes the significance of each project's attributes, offering a more nuanced assessment than the traditional Jaccard measure. The TanimotoJAC modifies the Jaccard measure by calculating the ratio of the intersection of attribute values to the total number of distinct attributes across both projects. This offers a precise method for gauging similarity based on the attributes of each project. The TMJ further extends the JAC by incorporating additional factors that account for the magnitude of attribute values. This adjustment enhances the accuracy of similarity assessments in project evaluations by providing a refined similarity score.

### 2.5. Mean Squared Difference-based similarity measure and its extensions

The Mean Squared Difference (MSD) function assesses dissimilarity by calculating the squared differences between the attribute values of two projects [23]. This measure effectively captures the degree of dissimilarity, allowing for a quantitative evaluation of the differences between the projects. The formula for MSD in the NCFEE context is defined in Eq. (16).

Two extensions of the MSD are the Jaccard Mean Squared Difference (JMSD) [24], which combines the principles of MSD with the Jaccard similarity measure, and the Coverage - Jaccard - MSD (CJMSD) [25], which refines the approach further by considering the average attribute values across projects. This adaptation provides a clearer understanding of project dissimilarities. While traditional MSD is adequate for assessing project dissimilarities, extensions such as JMSD and CJMSD enhance its applicability by addressing data sparsity and computational efficiency challenges.

## 3. Methodology

### 3.1. Research design

This study investigates the relationship between the choice of similarity measures and estimation accuracy, emphasizing the importance of selecting appropriate measures in NCFEE. The research employs a comprehensive framework to structure the effort estimation process by identifying similar historical projects. This framework encompasses selecting and applying various similarity measures and follows a structured two-phase estimation process. To thoroughly

Table 1: Similarity Measures and Their Corresponding Formulas

| No. | Category | Function | Similarity Formula | |
|---|---|---|---|---|
| 1 | Distance-Based | EUC [9] | $\text{sim}(u,v)^{\text{EUC}} = \frac{1}{1+\sqrt{\sum_{i=1}^{n}(u_i-v_i)^2}}$ | (2) |
| 2 | | HAM [10] | $\text{sim}(u,v)^{\text{HAM}} = 1 - \frac{\sum_{i=1}^{n}\delta(u_i,v_i)}{n}$ | (3) |
| 3 | | MHT [11] | $\text{sim}(u,v)^{\text{MHT}} = \frac{1}{1+\sum_{i=1}^{n}\lvert u_i-v_i\rvert}$ | (4) |
| 4 | | CHE [12] | $\text{sim}(u,v)^{\text{CHE}} = \frac{1}{1+\max_i\lvert u_i-v_i\rvert}$ | (5) |
| 5 | | MKS [13] | $\text{sim}(u,v)^{\text{MKS}} = \frac{1}{1+(\sum_{i=1}^{n}\lvert u_i-v_i\rvert^d)^{\frac{1}{d}}}$ | (6) |
| 6 | | AKR [14] | $\text{sim}(u,v)^{\text{AKR}} = \alpha\cdot\left(\frac{1}{1+\sqrt{\sum_{i=1}^{n}(u_i-v_i)^2}}\right)+(1-\alpha)\cdot\left(\frac{1}{1+\sum_{i=1}^{n}\lvert u_i-v_i\rvert}\right)$ | (7) |
| 7 | Pearson Correlation Coefficient | PCC [15] | $\text{sim}(u,v)^{\text{PCC}} = \frac{\sum_{i=1}^{n}(u_i-\bar{u})(v_i-\bar{v})}{\sqrt{\sum_{i=1}^{n}(u_i-\bar{u})^2}\cdot\sqrt{\sum_{i=1}^{n}(v_i-\bar{v})^2}}$ | (8) |
| 8 | | WeightedPCC [16] | $\text{sim}(u,v)^{\text{WPCC}} = \frac{\sum_{i=1}^{n}w_i(u_i-\bar{u})(v_i-\bar{v})}{\sqrt{\sum_{i=1}^{n}w_i(u_i-\bar{u})^2}\cdot\sqrt{\sum_{i=1}^{n}w_i(v_i-\bar{v})^2}}$ | (9) |
| 9 | | ModifiedPCC [17] | $\text{sim}(u,v)^{\text{ModifiedCCC}} = \frac{\sum_{i=1}^{n}F_i(u_i-\widetilde{u})(v_i-\widetilde{v})}{\sqrt{\sum_{i=1}^{n}F_i(u_i-\widetilde{u})^2}\cdot\sqrt{\sum_{i=1}^{n}F_i(v_i-\widetilde{v})^2}}$ | (10) |
| 10 | Cosine | COS [18] | $\text{sim}(u,v)^{\text{COS}} = \frac{\sum_i(u_i\cdot v_i)}{\sqrt{\sum_i u_i^2}\cdot\sqrt{\sum_i v_i^2}}$ | (11) |
| 11 | Jaccard | JAC [19] | $\text{sim}(u,v)^{\text{JAC}} = \frac{\lvert I_u\cap I_v\rvert}{\lvert I_u\cup I_v\rvert}$ | (12) |
| 12 | | DiceJAC [20] | $\text{sim}(u,v)^{\text{DiceJAC}} = \frac{2\lvert I_u\cap I_v\rvert}{\lvert I_u\rvert+\lvert I_v\rvert}$ | (13) |
| 13 | | TanimotoJAC [21] | $\text{sim}(u,v)^{\text{TanimotoJAC}} = \frac{\lvert I_u\cap I_v\rvert}{\lvert I_u\rvert+\lvert I_v\rvert-\lvert I_u\cap I_v\rvert}$ | (14) |
| 14 | | TMJ [22] | $\text{sim}(u,v)^{\text{TMJ}} = \text{sim}(u,v)^{\text{JAC}}\times\left(1-\frac{\sqrt{\sum_{i\in I_u}(u_i-v_i)^2}}{\sqrt{\sum_{i\in I_u}(u_i)^2}+\sqrt{\sum_{i\in I_v}(v_i)^2}}\right)$ | (15) |
| 15 | Mean Squared Difference | MSD [23] | $\text{sim}(u,v)^{\text{MSD}} = \frac{1}{n}\sum_{i=1}^{n}(u_i-v_i)^2$ | (16) |
| 16 | | JMSD [24] | $\text{sim}(u,v)^{\text{JMSD}} = \frac{\sum_{i\in I_u\cap I_v}(u_i-v_i)^2}{\lvert I_u\cup I_v\rvert}$ | (17) |
| 17 | | CJMSD [25] | $\text{sim}(u,v)^{\text{CJMSD}} = \frac{\sum_{i\in I_u\cap I_v}((u_i-\bar{u})(v_i-\bar{v}))^2}{\lvert I_u\cup I_v\rvert}$ | (18) |

explore project similarity, we implemented 17 distinct similarity functions, including distance-based, correlation-based, and set-based measures. This diverse set of functions allows for an in-depth examination of how different interpretations of project similarity impact the accuracy of software effort estimations. Detailed mathematical formulations of these functions are provided in Table 1, and Fig. 1 illustrates the architecture of the NCFEE framework, which operates through two key phases.

- **Phase 1(Project similarity measurement):** In this phase, the similarity between a target project and all historical projects in the dataset is calculated using 17 implemented similarity measures, producing a similarity score for each project pair. The main objective is to identify a subset of historical projects most similar to the target project, thereby establishing a relevant neighborhood. This aligns with the collaborative filtering process, where project proximity defines the neighborhood for predictions. The top $k$ most similar projects are selected as neighbors based on their similarity scores, with the parameter $k$ being a critical factor explored at various values throughout the study.
- **Phase 2 (Model fitting using stepwise regression):** The second phase estimates the development effort for the target project by analyzing the effort values of its neighboring projects identified in the first phase. The study

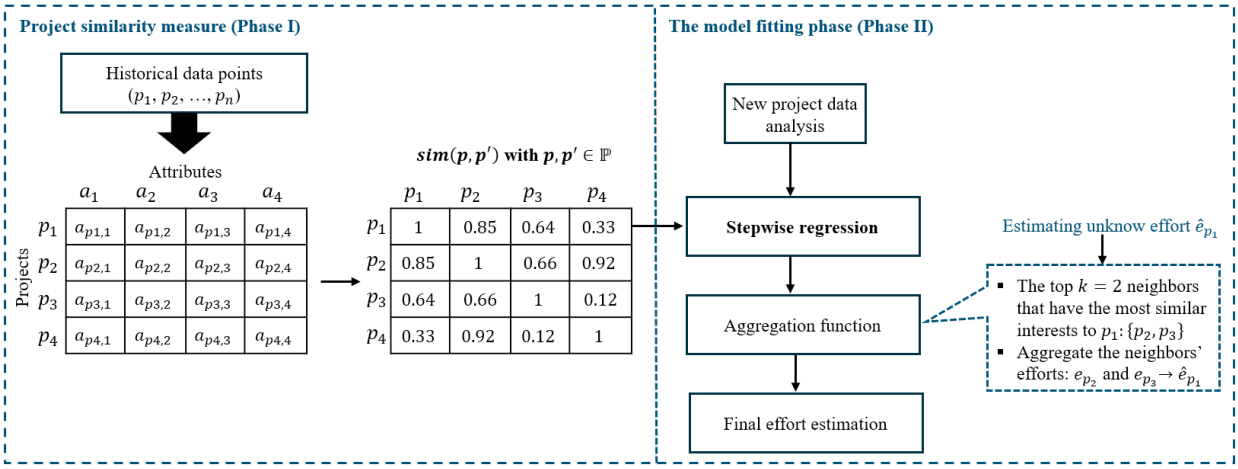


Fig. 1: The processing of the NCFEE framework

employs the Stepwise Regression (StepR) method [26], a statistical technique for developing predictive models that automatically selects the most relevant predictor variables from a larger pool of potential candidates. In this context, the effort values of the neighboring projects serve as the predictor variables for the target project's effort. StepR iteratively adds or removes these variables based on a pre-defined statistical criterion, refining the model by including only those neighboring projects whose effort values significantly influence the target project's effort, ultimately enhancing estimation accuracy.

### 3.2. Aggregation function and number of neighbors

The aggregation function is vital for combining the effort estimates of selected neighboring projects into a single effort estimation for the target project. This study employs a weighted average aggregation function, as represented by Eq. (19):

$$\hat{e}_u = \frac{\sum_{j \in N_u^k} \text{sim}(u, v_j) \cdot e_j}{\sum_{j \in N_u^k} \text{sim}(u, v_j)} \tag{19}$$

where $\hat{e}_u$ denotes the estimated effort for the target project $u$, $N_u^k$ is the set of $k$ neighboring projects for $u$, sim(u, $v_j$) represents the similarity score between $u$ and neighboring project $v_j$, and $e_j$ is the actual effort of $v_j$. The weighted average approach is intuitive and commonly used in collaborative filtering, where more similar neighbors have a more significant impact on the prediction, thus reflecting the principle that closely related projects are likely to have identical effort requirements.

The choice of $k$ significantly affects the model's accuracy. A small $k$ may lead to unstable predictions based on a limited number of similar projects. In contrast, a large $k$ can dilute the influence of the most relevant neighbors by including less representative projects. In this study, $k$ varies between 5 and 60, reflecting previous research and initial experimentation recommendations to identify an appropriate range for the datasets and similarity measures utilized. Exploring various $k$ values demonstrates a thorough approach to optimizing this critical parameter and assessing its impact on estimation accuracy across different similarity measures.

### 3.3. Evaluation measure

Evaluating effort estimation methods requires understanding their estimate accuracy and the practical relevance of performance differences. For accuracy, we employ the Root Mean Square Error (RMSE) [27], which measures the

typical error magnitude between predicted $\hat{y}_i$ and actual $y_i$ efforts across $m$ observations:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{m}(\hat{y}_i - y_i)^2}{m}} \tag{20}$$

However, comparing RMSE values may not reveal the practical importance of observed differences, especially when assessing variations of similarity measures. Therefore, we use Cohen's d [28] to incorporate effect size analysis. This standardized measure is applied in pairwise comparisons to quantify the magnitude of the difference between the mean RMSEs of two groups, scaled by their pooled standard deviation. By doing so, Cohen's d provides crucial insights into whether the observed differences in estimation accuracy are substantial enough to matter in practice, complementing the accuracy assessment provided by RMSE.

### *3.4. Dataset*

The study employs two distinct datasets, fpa_isbsg and fpa_china, both recognized benchmarks in SDEE. The fpa_isbsg dataset [6], sourced from the ISBSG repository, was refined from over 8,000 projects to 1,700 according to IFPUG standards, while fpa_china [7] comprises 499 cases. Both datasets focus on functional effort characteristics, with Summary Work Effort (SWE) measured in person-hours as the dependent variable. Independent variables representing project complexity include External Inputs (EI), External Outputs (EO), External Queries (EQ), Internal Logical Files (ILF), and External Interface Files (EIF), which are key to Function Point Analysis. Descriptive statistics for both datasets are provided in Table 3.

The datasets were preprocessed to address missing values and normalize the independent variables to ensure a balanced influence on similarity calculations. Subsequently, each dataset was split into a training set 80% and a testing set 20%. This process was repeated five times to enhance evaluation robustness, using 5-fold cross-validation to minimize the impact of specific data splits and generate reliable predictive estimates.

Table 2: Descriptive statistics for the dataset

| Dataset | Sizing Method | Instances | Features | Mean | Median | Min | Max |
|---|---|---|---|---|---|---|---|
| fpa_isbsg | Functional Point | 1,641 | 58 | 5,122 | 2,419 | 8 | 186,203 |
| fpa_china | Functional Point | 499 | 14 | 487 | 215 | 9 | 17,518 |

## 4. Results and discussion

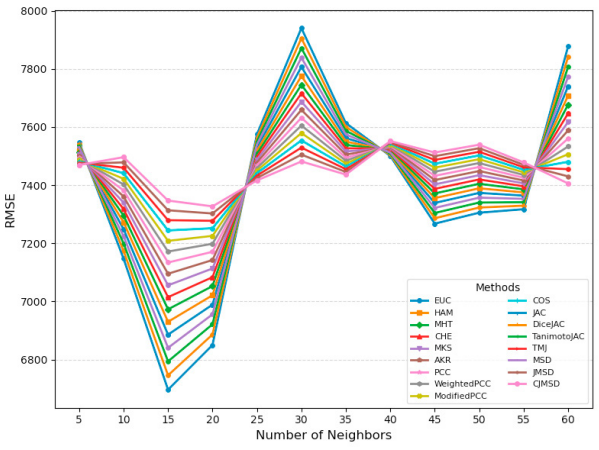


Fig. 2: RMSE results of the methods in the fpa_china dataset

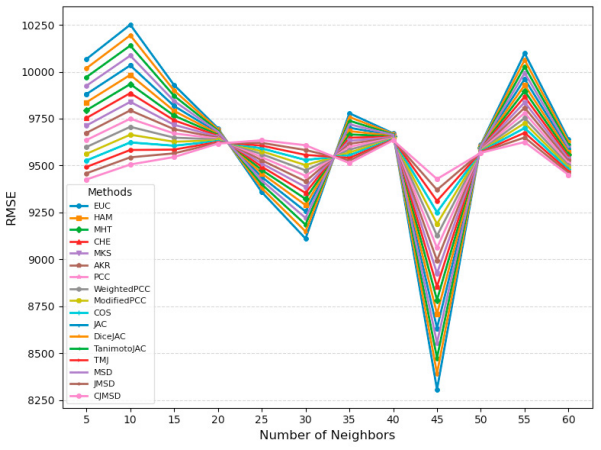


Fig. 3: RMSE results of the methods in the fpa_isbsg dataset

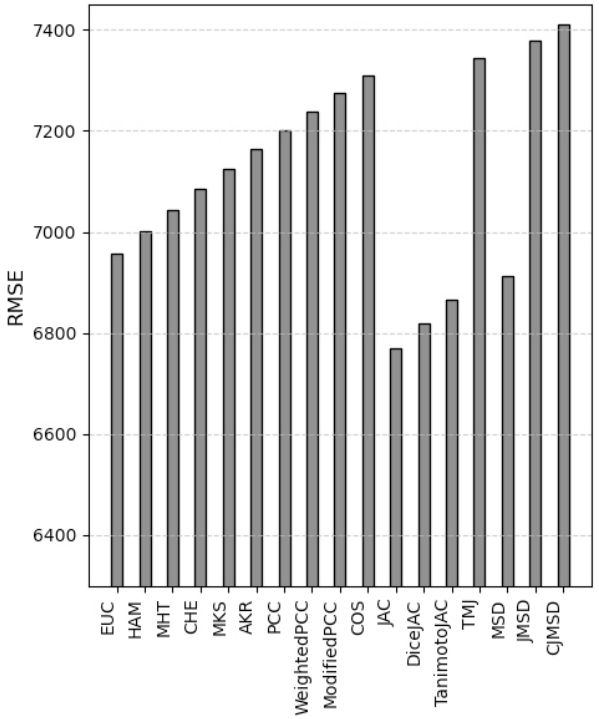


Fig. 4: RMSE results at the optimal number of neighbors $k = 15$ in the fpa_china dataset

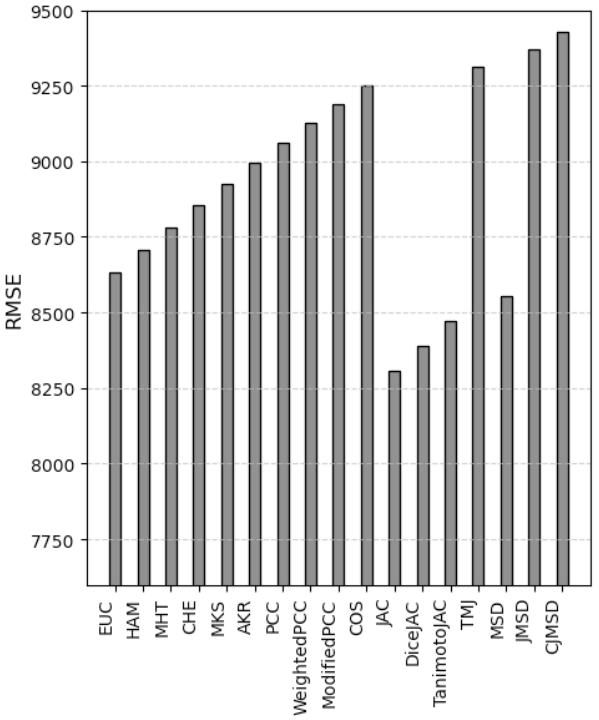


Fig. 5: RMSE results at the optimal number of neighbors $k = 45$ in the fpa_isbsg dataset

This study evaluated the influence of 17 similarity measures on NCFEE accuracy using the fpa_china and fpa_isbsg datasets, with performance measured by RMSE values, where lower values signify better accuracy. We initially examined the impact of the similarity measure and neighbor size $k$. Analysis of RMSE across neighbor sizes from 5 to 60 (shown in Fig. 2 and Fig. 3) reveals that estimation accuracy is sensitive to the choice of both similarity measure and neighbor size. Optimal performance typically occurs at intermediate $k$ values, avoiding overly small neighborhoods sensitive to noise and overly large neighborhoods diluted by dissimilar neighbors. Specifically, Jaccard-based measures (JAC, DiceJAC, and TanimotoJAC) demonstrated strong performance, generally showing decreasing RMSE with increasing $k$ within optimal ranges. However, these ranges varied by dataset, with the optimal $k$ values ranging from 5 to 15 for fpa_china and 10 to 45 for fpa_isbsg. Further analysis pinpointed optimal values at $k = 15$ for fpa_china and $k = 45$ for fpa_isbsg (Fig. 4 and Fig. 5). This difference suggests that the denser fpa_china dataset benefits from a more focused neighborhood, while the larger, more heterogeneous fpa_isbsg dataset requires a broader context. These findings underscore the necessity of dataset-specific tuning of $k$ for each similarity measure.

Next, we assessed the practical significance of performance differences using Cohen's d effect sizes (Table 3), where negative values indicate superior performance (lower mean RMSE) for the first group in comparison. The results confirm the strong performance of Jaccard-based measures, showing very large, practically significant advantages over other measure types on both datasets. Specifically, key comparisons revealed Jaccard-based vs. Distance-based with $d = -3.427$, Jaccard-based vs. PCC-based with $d = -9.827$ (an extremely large effect), Jaccard vs. MSD-based with $d = -2.406$, and Jaccard-based vs. all others combined (Distance-based, PCC-based, COS, and MSD-based) with $d = -2.367$. These substantial effect sizes reflect the inherent strengths of Jaccard-based measures. Specifically, they evaluate similarity based on the overlap of project attributes, assessing the proportion of shared attributes relative to the total unique attributes. This focus makes them particularly effective when project attributes differ in scale. Furthermore, these measures highlight relevant project features that contribute to similarity and can capture non-linear relationships among attributes, unlike PCC-based measures, which focus solely on linear correlations. Moreover, Jaccard-based measures are less susceptible to the curse of dimensionality compared to distance-based measures,

maintaining their effectiveness in high-dimensional datasets by concentrating on attribute overlap. Furthermore, a substantial dataset effect was observed. All measure types performed significantly better on fpa_china than fpa_isbsg, with extremely large effect sizes ranging from $d = -5.873$ (MSD-based) to $d = -37.187$ (PCC-based). This highlights the profound impact of dataset characteristics on estimation accuracy across all tested similarity approaches. The findings demonstrate that Jaccard-based measures offer a consistent and practically significant performance advantage over Distance-based, PCC-based, and MSD-based measures.

## 5. Conclusion and future work

This study investigated the critical role of similarity measures in the accuracy of NCFEE, utilizing the fpa_china and fpa_isbsg datasets. The findings offer valuable insights for enhancing software effort estimation reliability. Our investigation confirmed that the choice of similarity measure significantly impacts NCFEE performance. Addressing **RQ1: Which similarity measures commonly used in neighbor-based collaborative filtering can be effectively applied in NCFEE?** We found that Jaccard-based measures (JAC, DiceJAC, TanimotoJAC) consistently outperformed other evaluated measures (Distance-based, PCC-based, MSD-based) by achieving lower RMSE values. Additionally, effect size analysis revealed these performance differences to be highly significant. In response to **RQ2: How do different similarity measures impact effort estimation accuracy in NCFEE?** Our empirical analysis demonstrated that estimation accuracy is highly sensitive to the similarity measure chosen and the neighborhood size. This underscores the necessity for careful, dataset-specific tuning of both the similarity measure and the parameter k to maximize estimation accuracy.

Table 3: Effect Sizes of Jaccard-based Measures Compared to Other Similarity Measures in Software Effort Estimation

| Comparison description | Dataset | Cohen's d |
|---|---|---|
| Jaccard-based measures vs. Distance-based measures | fpa_china | -3.427 |
| Jaccard-based measures vs. Distance-based measures | fpa_isbsg | -3.427 |
| Jaccard-based measures vs. PCC-based measures | fpa_china | -9.827 |
| Jaccard-based measures vs. PCC-based measures | fpa_isbsg | -9.827 |
| Jaccard-based measures vs. MSD-based measures | fpa_china | -2.406 |
| Jaccard-based measures vs. MSD-based measures | fpa_isbsg | -2.406 |
| Jaccard-based measures vs. All others combined | fpa_china | -2.367 |
| Jaccard-based measures vs. All others combined | fpa_isbsg | -2.367 |
| JAC-based on fpa_china vs. JAC-based on fpa_isbsg | both | -30.374 |
| Distance-based on fpa_china vs. Distance-based on fpa_isbsg | both | -16.513 |
| PCC-based on fpa_china vs. PCC-based on fpa_isbsg | both | -37.187 |
| MSD-based on fpa_china vs. MSD-based on fpa_isbsg | both | -5.873 |

While this study provides valuable insights, it has limitations. The findings are based on only two specific datasets, and their generalizability to other software project datasets requires further investigation. Additionally, while the impact of $k$ was explored, a deeper analysis of which specific dataset characteristics drive the optimal k was beyond the scope of this work. Future work will focus on validating these findings across a broader range of diverse software project datasets. Building on the strengths of Jaccard measures while addressing their contextual limitations, we will develop and evaluate a novel similarity measure, which we will refer to as Multi-Factor Project Similarity (MFPS). Envisioned as an enhanced form of Jaccard similarity, MFPS will integrate domain-specific knowledge by explicitly considering factors such as industry sector, project size, and programming language, alongside functional characteristics. This approach is intended to enable a more comprehensive and context-aware assessment of project similarity. A key objective will be formally defining MFPS and rigorously evaluating its performance against benchmark measures across diverse datasets.

## Acknowledgements

This work was supported by Tomas Bata University in Zlin, Faculty of Applied Informatics under Grant No. RVO/FAI/2021/002, IGA/ CebiaTech/2022/001, and RO30246061025/2102.

## References


[1] A. Saeed, W. H. Butt, F. Kazmi, M. Arif, Survey of software development effort estimation techniques, in: Proceedings of the 2018 7th International Conference on Software and Computer Applications, ICSCA '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 82–86. doi:10.1145/3185089.3185140.

[2] H. L. T. K. Nhung, V. Van Hai, R. Silhavy, Z. Prokopova, P. Silhavy, Parametric software effort estimation based on optimizing correction factors and multiple linear regression, IEEE Access 10 (2022) 2963–2986. doi:10.1109/ACCESS.2021.3139183.

[3] P. Phannachitta, Robust comparison of similarity measures in analogy-based software effort estimation, in: 2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 2017, pp. 1–7. doi:10.1109/SKIMA.2017.8294126.

[4] H. L. T. Kim Nhung, P. Silhavy, R. Silhavy, Enhancing software effort estimation through influencers-based project similarity measurement, Procedia Computer Science 246 (2024) 3256–3264, 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024). doi:https://doi.org/10.1016/j.procs.2024.09.314.

[5] I. Abnane, M. Hosni, A. Idri, A. Abran, Analogy software effort estimation using ensemble knn imputation, in: 2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2019, pp. 228–235. doi:10.1109/SEAA.2019.00044.

[6] ISBSG, Isbsg development enhancement repository release 13., [Online]. Available: http://isbsg.org (Feb. 2, 2015).

[7] J. S. Shirabad, T. J. Menzies, et al., The promise repository of software engineering databases, School of information technology and engineering, University of Ottawa, Canada 24 (3) (2005).

[8] S. Bagchi, Performance and quality assessment of similarity measures in collaborative filtering using mahout, Procedia Computer Science 50 (2015) 229–234, big Data, Cloud and Computing Challenges. doi:https://doi.org/10.1016/j.procs.2015.04.055.

[9] M. V. Kosti, N. Mittas, L. Angelis, Alternative methods using similarities in software effort estimation, in: Proceedings of the 8th International Conference on Predictive Models in Software Engineering, PROMISE '12, Association for Computing Machinery, New York, NY, USA, 2012, p. 59–68. doi:10.1145/2365324.2365333.

[10] P. Phannachitta, J. Keung, A. Monden, K.-i. Matsumoto, Improving analogy-based software cost estimation through probabilistic-based similarity measures, in: 20th Asia-Pacific Software Engineering Conference, Vol. 1, 2013, pp. 541–546. doi:10.1109/APSEC.2013.78.

[11] A. Idri, I. Abnane, A. Abran, Missing data techniques in analogy-based software development effort estimation, Journal of Systems and Software 117 (2016) 595–611. doi:https://doi.org/10.1016/j.jss.2016.04.058.

[12] Q. Liu, X. Chu, J. Xiao, H. Zhu, Optimizing non-orthogonal space distance using pso in software cost estimation, in: 2014 IEEE 38th Annual Computer Software and Applications Conference, 2014, pp. 21–26. doi:10.1109/COMPSAC.2014.9.

[13] Z. Shahpar, V. Khatibi, A. Khatibi Bardsiri, Hybrid pso-sa approach for feature weighting in analogy-based software project effort estimation, Journal of AI and Data Mining 9 (3) (2021) 329–340. doi:10.22044/jadm.2021.10119.2152.

[14] Z. Shahpar, V. K. Bardsiri, A. K. Bardsiri, An evolutionary ensemble analogy-based software effort estimation, Software: Practice and Experience 52 (4) (2022) 929–946. doi:https://doi.org/10.1002/spe.3040.

[15] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94, Association for Computing Machinery, New York, NY, USA, 1994, p. 175–186. doi:10.1145/192844.192905.

[16] J. L. Herlocker, J. A. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, in: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999, pp. 230–237.

[17] N. F. AL-Bakri, S. H. Hashim, A modified similarity measure for improving accuracy of user-based collaborative filtering., Iraqi Journal of Science 59 (2018).

[18] R. A. . K. H. Abdulgabber, M. A., An improved memory-based collaborative filtering method based on the topsis technique, PLOS ONE 13 (10) (2018). doi:10.1371/journal.pone.0204434.

[19] K. Saranya, G. Sudha Sadasivam, Modified heuristic similarity measure for personalization using collaborative filtering technique, Applied Mathematics and Information Sciences 11 (1) (2017) 307 – 315. doi:10.18576/amis/110137.

[20] M. Y. H. Al-Shamri, Power coefficient as a similarity measure for memory-based collaborative recommender systems, Expert Systems with Applications 41 (13) (2014) 5680–5688.

[21] A. Kumar, S. Gupta, S. K. Singh, K. K. Shukla, Comparison of various metrics used in collaborative filtering for recommendation system, in: 2015 Eighth International Conference on Contemporary Computing (IC3), IEEE, 2015, pp. 150–154.

[22] S.-B. Sun, Z.-H. Zhang, X.-L. Dong, H.-R. Zhang, T.-J. Li, L. Zhang, F. Min, Integrating triangle and jaccard similarities for recommendation, PloS one 12 (8) (2017) e0183570.

[23] N. Sivaramakrishnan, V. Subramaniyaswamy, S. Arunkumar, A. Renugadevi, et al., Neighborhood-based approach of collaborative filtering techniques for book recommendation system, International Journal of Pure and Applied Mathematics (2018).

[24] J. Bobadilla, F. Ortega, A. Hernando, J. Bernal, A collaborative filtering approach to mitigate the new user cold start problem, Knowledge-based systems 26 (2012) 225–238.

[25] J. Bobadilla, F. Ortega, A. Hernando, Á. Arroyo, A balanced memory-based collaborative filtering similarity measure, International journal of intelligent systems 27 (10) (2012) 939–946.

[26] P. Silhavy, R. Silhavy, Z. Prokopova, Evaluation of data clustering for stepwise linear regression on use case points estimation, in: Advances in Intelligent Systems and Computing, Vol. 575, 2017, pp. 491–496. doi:10.1007/978-3-319-57141-6_52.

[27] M. Rahman, H. Sarwar, M. A. Kader, T. Gonçalves, T. T. Tin, Review and empirical analysis of machine learning-based software effort estimation, IEEE Access 12 (2024) 85661–85680. doi:10.1109/ACCESS.2024.3404879.

[28] A. J. Larner, Effect size (cohen's d) of cognitive screening instruments examined in pragmatic diagnostic accuracy studies, Dementia and Geriatric Cognitive Disorders Extra 4 (2) (2014) 236–241.