Power System Protection and Control
ISSN:1674-3415

RECOMMENDER SYSTEMS IN SOCIAL NETWORKS USING CROWDSOURCING

Masoumeh Ansarifar[1], Erfaneh Noroozi*[2], Mehdi Hosseinzadeh[3]

[1]Department Of Computer Engineering, Qe.C., Islamic Azad University, Qeshm, Iran
Masoumeh.ansarifar@iau.ac.ir 0000-0001-9590-7550
[2]Department Of Computer Engineering, Sep.C., Islamic Azad University, Sepidan, Iran
Erfaneh.Noroozi@iau.ac.ir 0000-0002-1397-4870
[3]International Center for Materials Sciences and Technology, Western Caspian University, Baku, Azerbaijan.
mehdihosseinzadeh@duytan.edu.vn 0000-0003-1088-4551

Received: 08/05/2026 **Revised**: 10/06/2026 **Accepted**: 05/07/2026

***ABSTRACT*:**

The rapid growth of content in social networks has made it hard for users to find relevant information and has raised the need for effective recommender systems. Collaborative filtering is one of the most widely used recommendation approaches, yet it faces the challenges of data sparsity and cold start. The aim of this study was to design and evaluate a hybrid recommender system that strengthens collaborative filtering with crowdsourced data. The study followed an applied and developmental design, and the data were gathered from an active social-network group in the health-news domain. The implicit data were drawn from the users' behavioral traces, and the explicit data were obtained from a researcher-made questionnaire across four dimensions, namely satisfaction, relevance, transparency, and trust. The model performance was measured with precision, recall, the F1 score, coverage, diversity, and prediction error. The findings showed that crowdsourcing increases data richness and density and raises the density of the interaction matrix from 6.3 percent to 28.9 percent. The structured integration of explicit and implicit data also lifted the precision of the optimized version to 0.45 at the top ten recommendations and its recall to 0.44, while lowering the prediction error. These improvements enhanced recommendation quality for low-activity users and eased the cold-start problem. Overall, integrating crowdsourcing with collaborative filtering provides a human-centered and effective framework for improving recommendation in social networks.

***Keywords**: Collaborative Filtering, Crowdsourcing, Explicit Data, Recommender System, Social Network.*

INTRODUCTION

Online social networks have become a pervasive part of daily life, and nearly half of the world population holds active membership in them [1]. This rapid expansion has produced enormous volumes of user-generated content and personal information across digital platforms [2]. At a structural level, a social network is modeled as a graph of nodes representing users and edges representing their relations [3]. Within these connected structures, content and user preferences propagate quickly among neighboring nodes [4]. This growing flood of information makes relevant content selection difficult and gives rise to information overload [5]. Recommender systems have emerged as intelligent tools that filter and personalize content within social networks [6]. Large-scale platforms rely on these systems to manage billions of daily interactions efficiently [7]. Such systems now determine which content users encounter and thereby shape their decision-making processes [8]. Their application has further extended to mobile and social services, including mechanisms based on trust evaluation [9]. The effectiveness of these systems depends heavily on the algorithmic approach, and collaborative filtering based on user similarity remains the most widely used method [10]. Collaborative filtering predicts user interests from similar users, yet its accuracy declines under cold-start and data-sparsity conditions [11]. To improve performance, approaches based on deep learning and reinforcement learning have been developed [12]. Knowledge-graph-based methods enrich recommendations through structured relations among entities [13]. Conversational recommender systems elicit user preferences through multi-turn interaction [14]. Session-based recommenders model the short-term and dynamic patterns of user behavior [15]. Context-aware mobile recommenders incorporate situational information into the recommendation process [16]. Graph-convolution methods attempt to balance the accuracy and diversity of recommendations [17]. Deep-learning multi-criteria algorithms capture nonlinear and multidimensional relations among user preferences [18]. Nevertheless, deep-learning-based systems remain vulnerable to threats

DOI: 10.46121/pspc.54.3.23

such as data-poisoning attacks [19]. The performance of collaborative filtering relies on the volume and quality of interaction data, and richer contextual data improves rating accuracy [20]. Accordingly, graph-based hybrid systems have been proposed to mitigate the cold-start problem through demographic and structural information [21]. Enriching the model with auxiliary information, such as user experience and emotion, improves graph-based recommendation accuracy [22]. Hybrid models based on resource allocation specifically target the cold-start problem for newly introduced items [23]. Combining classical and deep-learning methods in rating-prediction models has increased predictive accuracy [24]. The use of auxiliary and visual features also partially compensates for the data-sparsity problem [25]. Multimodal trust-based approaches simultaneously address cold start, sparsity, and malicious data [26]. Beyond accuracy, the transparency and explainability of recommendations have become key factors in strengthening user trust [27]. Among emerging solutions, crowdsourcing leverages collective intelligence to enable large-scale data generation and evaluation by users [28]. Integrated with social platforms, this approach enables dynamic and continuous knowledge collection [29]. In recent years, several studies have examined the use of crowdsourcing within recommender systems [30], [31]. Mirbahar et al. (2025) improved mobile-application recommendations by combining crowdsourced data with collaborative filtering and machine- and deep-learning techniques. Their results showed that deep-learning methods substantially enhance the accuracy and relevance of recommendations by capturing complex and context-aware patterns [32]. Yin et al. (2025) designed a multi-task deep-learning model for task recommendation in knowledge-intensive crowdsourcing to balance the mutual benefits of users and platforms. Their findings indicated that jointly modeling solver preference and ability reduces sample-selection bias and improves recommendation accuracy and resource allocation [33]. Despite these advances, most prior studies have focused on algorithmic data centrality and have paid limited attention to collective intelligence for the structured enrichment of input data and for active user participation in refining recommendations. Moreover, a framework that dynamically integrates explicit crowdsourced data with implicit behavioral data in a real social-network environment has rarely been presented in a systematic manner. The objective of this study is to propose and evaluate a hybrid recommender model in social networks. This model strengthens the collaborative-filtering algorithm with crowdsourced data to improve recommendation accuracy and reduce the cold-start problem. Three hypotheses are pursued in this study:

1. Crowdsourcing increases the quality and diversity of users' behavioral data in the recommender system.
2. The structured integration of crowdsourced data with the collaborative-filtering algorithm improves the algorithm's accuracy and the effectiveness of recommendations.
3. The use of crowdsourced data reduces the cold-start problem in the recommender system.

The main novelty of this study lies in a human-centered framework in which the user simultaneously acts as a data source and as a validator of recommendations. Explicit data obtained through voluntary participation are integrated with implicit behavioral data within a dynamic feedback mechanism. This design transforms the system from a static structure into a self-evolving model. The proposed method, the data collection and integration process, and the quantitative and qualitative evaluation of the model are described in the following sections.

METHODOLOGY

Research Design

This study follows an applied and developmental design, and it centers on building a hybrid social recommender system. The system couples collaborative filtering with crowdsourced user data. Its input comes from two complementary sources, namely implicit behavioral data and explicit data contributed through voluntary participation. The proposed model merges these two sources within a single structure to raise the accuracy and quality of its recommendations. The primary aim is to improve recommendation performance and to ease the cold-start problem in a social-network setting. The overall architecture of the model and the sequence of its execution phases are shown in Figure 1.

DOI: 10.46121/pspc.54.3.23

Power System Protection and Control
ISSN:1674-3415

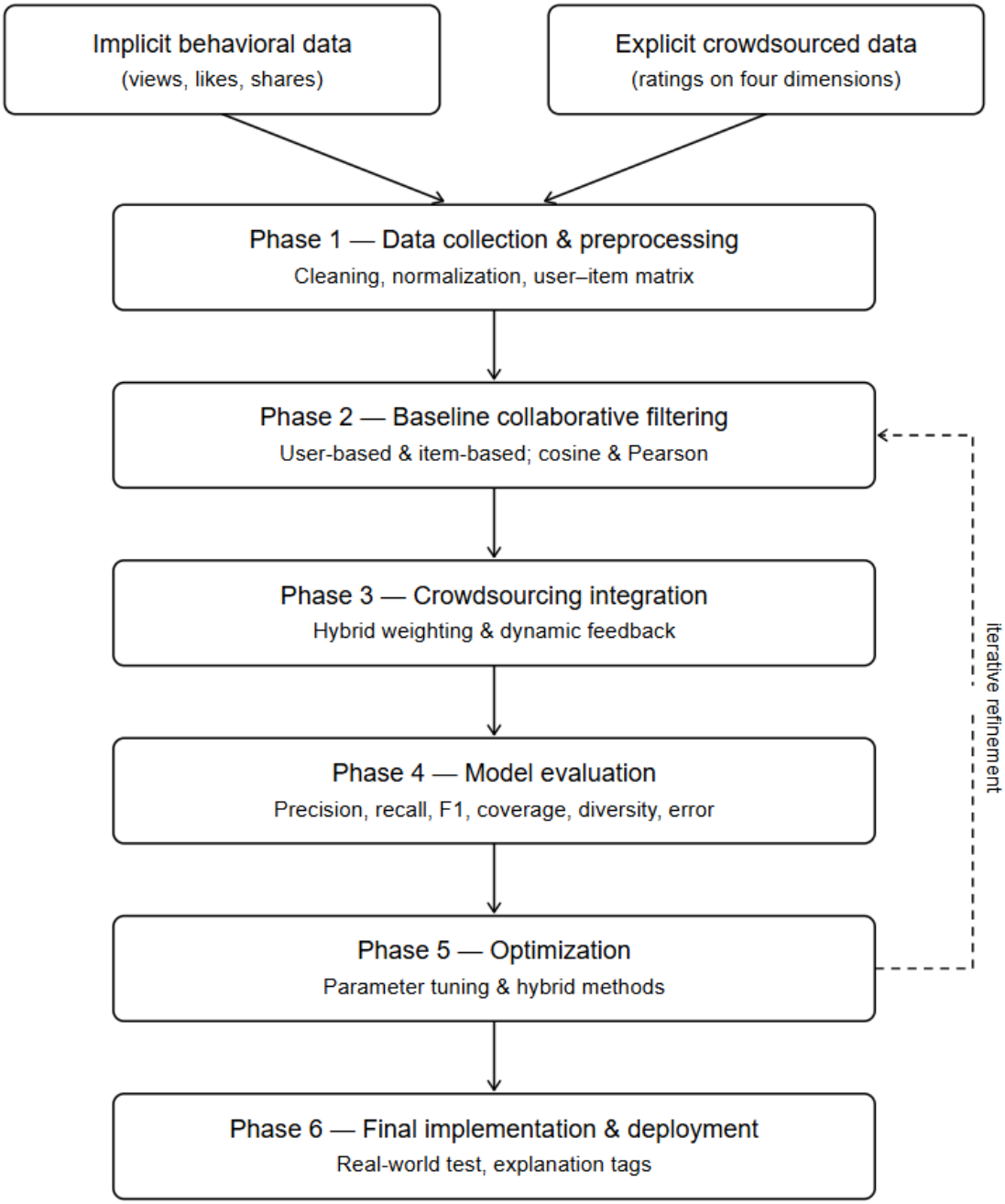

Figure 1. Architecture of the proposed hybrid recommender model.

Statistical Population and Sampling

The statistical population of this study consisted of the active members of a social-network group who were exposed daily to health-related news and social content during December 2020. This population comprised 110 volunteer users who took part in the crowdsourced data collection process. Sampling was carried out through a convenience and voluntary procedure. Among the members present in the group, those who were willing to participate were selected as the units of analysis. Because all 110 active members joined the crowdsourcing process, the entire population was treated as the study sample. Each user expressed an opinion on the published content using a score from one to five, based on a set of predefined criteria.

DOI: 10.46121/pspc.54.3.23

Power System Protection and Control
ISSN:1674-3415

Data Collection Method and Instrument

The study data were gathered through a social crowdsourcing process from the users of an active social-network group. The explicit data were collected using a researcher-made questionnaire. Within this questionnaire, users rated the published content along four dimensions, namely overall satisfaction, relevance of the content to their interests, transparency, and trust in the system. The rating scale ranged from one to five, and it reflected the users' genuine perception of recommendation quality. Alongside the explicit data, implicit data were extracted from the users' behavioral traces. These data included the number of views, likes, shares, repeated interactions, and participation in discussions. Data collection began in December 2020 and continued online in real time. The general specifications of the study dataset are presented in Table 1.

Table 1. General specifications of the study dataset

Attribute	Value
Platform type	Social-network group (messaging app)
Content domain	Health news and social events
Number of users (behavioral data)	300
Number of content items	200
Crowdsourcing volunteers	110
Explicit rating scale	1 to 5
Crowdsourcing dimensions	Satisfaction, relevance, transparency, trust
Collection period	December 2020

Proposed Model (Integration of Collaborative Filtering and Crowdsourcing)

The model inputs are organized as a user–item matrix. In this matrix, each row represents a user, and each column represents a content item. The explicit rating matrix R is defined in Equation 1, where every element denotes the score assigned by a user to an item.

$$R = \{r_{ui} | u = 1. \ldots . U; i = 1. \ldots . I\} \quad (1)$$

In this equation, R is the explicit rating matrix, and r_{ui} is the score that user u assigns to item i. The symbol U denotes the total number of users, and I denotes the total number of items.

Raw data usually contain noise, missing entries, and inconsistent scales. For this reason, the values are normalized through a min–max transformation before modeling, so that they fall within the range of zero to one. This normalization is given in Equation 2.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2)$$

In this equation, x is the original value, and x′ is its normalized counterpart. The terms x_{min} and x_{max} represent the minimum and maximum values of the corresponding variable.

After the matrix is prepared, the similarity between users or items is computed. This study employs two similarity measures, namely cosine similarity and Pearson correlation. Cosine similarity measures the angle between two vectors, and it is well suited to sparse data. This measure is defined in Equation 3.

$$sim_{cos}(A.B) = \frac{\sum_{k=1}^{n} A_k \times B_k}{\sqrt{\sum_{k=1}^{n} A_k^2} \times \sqrt{\sum_{k=1}^{n} B_k^2}} \quad (3)$$

In this equation, A and B are the two compared vectors, and A_k and B_k are their kt_h components. The term n denotes the number of features, that is, the scores given to items or users.

Pearson correlation measures the linear association between two vectors, and it reduces the effect of scale differences by subtracting the mean. This measure is given in Equation 4.

$$sim_{Pearson}(A.B) = \frac{\sum_{k=1}^{n} (A_k - \bar{A})(B_k - \bar{B})}{\sqrt{\sum_{k=1}^{n} (A_k - \bar{A})^2} \sqrt{\sum_{k=1}^{n} (B_k - \bar{B})^2}} \quad (4)$$

In this equation, A and B are the two compared vectors, while $\bar{A}$ and $\bar{B}$ are the mean values of these vectors. The terms A_k and B_k again denote their k_{th} components.

Based on the computed similarities, the score of a user for a given item is predicted. In the user-based approach, the prediction is obtained from the weighted average of the scores of similar users. This prediction is shown in Equation 5.

DOI: 10.46121/pspc.54.3.23

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N(u)} sim(u.v) \times (r_{vi} - \bar{r}_v)}{\sum_{v \in N(u)} |sim(u.v)|} \quad (5)$$

In this equation, $\hat{r}_{ui}$ is the predicted score for user u and item i. The set N(u) contains the users similar to u, and r_{vi} is the score that user v gives to item i. The terms $\bar{r}_u$ and $\bar{r}_v$ are the mean scores of users u and v, while sim(u,v) is the similarity between users u and v.
In the item-based approach, the prediction relies on the similarity among items. This prediction is given in Equation 6.

$$\hat{r}_{ui} = \frac{\sum_{j \in S(i)} sim(i.j) \times r_{uj}}{\sum_{j \in S(i)} |sim(i.j)|} \quad (6)$$

In this equation, $\hat{r}_{ui}$ is the predicted score for user u and item i. The set S(i) contains the items similar to item i, and r_{uj} is the score that user u gives to item j. The term sim(i,j) denotes the similarity between items i and j.

In the integration step, the output of the implicit-data model is combined with the output of the explicit crowdsourced model. This combination relies on hybrid weighting, where the weight of the implicit data was set to 0.7 and that of the explicit data to 0.3. These weights were later tuned to 0.62 and 0.38 during the optimization phase. The explicit data act as a filler in the cells where behavioral data are scarce, and they raise the density of the matrix. Furthermore, direct user feedback enters the model dynamically, and the algorithm is updated in an iterative manner. This mechanism turns the system from a static design into a learning and adaptive model.

Model Validation and Data Splitting
To assess the model performance fairly, the dataset was divided into training and test subsets. Eighty percent of the data were assigned to training, and twenty percent were held out for testing. This split was performed through stratified random selection, so that the overall distribution of interactions was preserved in both subsets. This approach prevents the model from overfitting the training data. To strengthen the external validity of the model, a secondary dataset of user ratings in the accommodation-services domain, namely hotel rankings, was also used. This complementary data tested the proposed algorithm in a setting with a different behavioral and social nature, and it reinforced the generalizability of the model.

Data Analysis Method and Evaluation Metrics
Data analysis was carried out with the collaborative-filtering algorithm in both the user-based and item-based approaches. The algorithm was implemented in the Java programming language, which supports the processing of large data volumes and the execution of iterative analyses. The model performance was measured through a set of standard quantitative metrics. These metrics included precision, recall, and the F1 score for assessing the correctness of the recommendations. Two further metrics, namely coverage and diversity, were used to gauge the breadth and variety of the suggestions. The prediction error was assessed through the mean absolute error and the root mean square error. Beyond the quantitative assessment, a qualitative analysis was conducted based on user feedback regarding satisfaction, usability, and the transparency of the recommendations.

RESULTS

This section presents the empirical results obtained from implementing the proposed model. It first describes the statistical properties of the behavioral and crowdsourced data. It then reports the results of the baseline collaborative-filtering model and the effect of integrating crowdsourced data. Next, the performance of four model versions is compared across the quantitative metrics. Finally, the optimized parameters, the qualitative assessment, and the real-world performance of the model are analyzed.

Data Description
The descriptive statistics of the two data sources are presented in Table 2. The behavioral data include the likes, comments, shares, and views of each user, drawn from their interaction traces. The crowdsourced data capture the users' ratings across four dimensions, namely satisfaction, content relevance, transparency, and trust. The table reports the mean, standard deviation, and range of every variable, so that the distribution and heterogeneity of the data become clear.

DOI: 10.46121/pspc.54.3.23

Power System Protection and Control
ISSN:1674-3415

Table 2. Descriptive statistics of the behavioral and crowdsourced data

Statistic	Likes	Comments	Shares	Views	Satisfaction	Relevance	Transparency	Trust
Count	300	300	300	300	120	120	120	120
Mean	5.2	1.8	0.9	125.0	3.9	4.1	3.7	3.8
Std. dev.	3.6	1.5	1.1	32.0	0.8	0.7	0.9	0.8
Minimum	0	0	0	30	2	2	1	1
Median	5	2	1	120	4	4	4	4
Maximum	20	10	8	250	5	5	5	5

The results in Table 2 showed that the behavioral data follow a heterogeneous distribution. The mean number of likes per user is about 5.2, and its median is close to this value, which points to a fairly balanced distribution of likes. By contrast, the mean number of comments and shares is below two per user, and the first quartile of shares equals zero. This pattern reflects the strong presence of low-activity users who act as silent consumers. Among the crowdsourced data, content relevance holds the highest value with a mean of 4.1, which confirms the alignment between the recommended content and the users' interests. Transparency shows the lowest level with a mean of 3.7 and a higher standard deviation, which points to the uneven experience of users in following the logic of the system.

Data Preprocessing

A summary of the cleaning and reduction process is presented in Table 3. At this stage, incomplete records, duplicates, and values outside the natural range were removed. Inactive users who logged no interaction were also excluded from the training set, so that the noise in the similarity computation would decrease. The table reports the volume of raw, removed, and final records for each data source.

Table 3. Summary of data cleaning and reduction

Data type	Raw records	Removed records	Removal rate (%)	Final records
Behavioral interactions	18000	2500	13.9	15500
Crowdsourcing responses	140	20	14.3	120
Inactive users	80	80	100	0

The results in Table 3 showed that nearly fourteen percent of the eighteen thousand raw behavioral records were removed during cleaning. These records were mostly incomplete, duplicated, or outside the natural range. In the crowdsourced data, about fourteen percent of the responses were discarded due to incompleteness or inconsistency. The full removal of inactive users meant that the final set consisted mainly of users with a minimum level of interaction. The removal rate in both sources lies within a reasonable range, which shows that the cleaning was strict yet balanced.

Baseline Collaborative-Filtering Results

The descriptive statistics of user similarity and item similarity in the baseline model are presented in Table 4. These similarities were computed from the user–item interaction matrix using the cosine and Pearson measures. A comparison of the two columns reveals the structural difference between user similarity and item similarity. This difference has a direct effect on the choice of the more stable approach in the baseline model.

Table 4. Descriptive statistics of user and item similarity in the baseline model

Statistic	User similarity	Item similarity
Analyzed pairs	44850	19900
Mean	0.31	0.41
Std. dev.	0.14	0.18
Minimum	0.02	0.01
First quartile	0.21	0.28
Median	0.29	0.39
Third quartile	0.39	0.52
Maximum	0.87	0.96

DOI: 10.46121/pspc.54.3.23

Power System Protection and Control
ISSN:1674-3415

The results in Table 4 showed that the mean item similarity equals 0.41 and exceeds the mean user similarity of 0.31. The closeness of the mean and median in both columns confirms the absence of severe skewness in the similarity distributions. The wider gap between the quartiles in item similarity indicates the formation of more distinct content clusters. This structure makes the item-based approach more stable under sparse interactions. On this basis, the stable relations among neighboring items compensate for part of the weakness caused by sparse user data.

The frequency distribution of user and item similarity values is shown in Figure 2. In panel (a), the distribution of user similarity concentrates in the range of 0.2 to 0.4 and forms around the mean of 0.31. The frequency of similarities below 0.1 and above 0.7 in this panel is very low, which shows that most user pairs share moderate similarity and that highly similar pairs are rare. In panel (b), the distribution of item similarity shifts toward higher values, and its peak falls within the range of 0.4 to 0.5. This panel has a long tail reaching 0.96, which confirms the presence of content groups with an almost identical consumption pattern. The difference between the means of the two panels, equal to 0.10, clearly reflects the higher concentration of similarity at the item level. This pattern reveals the advantage of the item-based approach in locating relevant and stable neighbors.

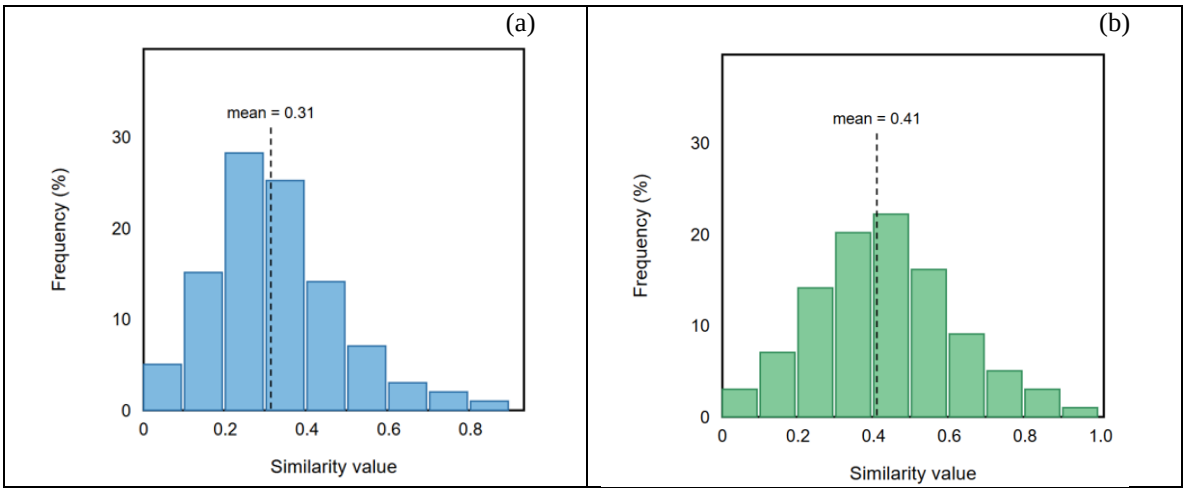

Figure 2. Similarity distribution in the baseline model. (a) User similarity, (b) item similarity.

The performance of the two baseline approaches on the test set is presented in Table 5. The table reports the precision values at two levels and the prediction error for the item-based and user-based approaches. The comparison of these two approaches forms the basis for selecting the reference configuration for the later phases. The figures in the table were computed from the same metrics defined in the methodology.

Table 5. Performance of the baseline model in the item-based and user-based approaches

Model type	Precision@5	Precision@10	RMSE	MAE
Item-based collaborative filtering	0.41	0.37	0.182	0.139
User-based collaborative filtering	0.38	0.34	0.195	0.148

The results in Table 5 showed that the item-based approach performs better at both precision levels. In the top five recommendations, the precision of the item-based approach is about three percentage points higher than that of the user-based approach. This advantage also appears in the error metrics, where the RMSE and MAE values are lower in the item-based approach. This finding aligns with the sparse nature of the study data, since the relations among items are more stable than the relations among users. On this basis, the item-based approach was selected as the reference configuration for comparison with the integrated version.

Effect of Crowdsourcing Integration

The effect of adding the explicit data on model performance is presented in Table 6. The table compares the baseline and integrated versions on the same test set. The aim of this comparison is to measure the real role of the crowdsourced data in improving accuracy and reducing error. The reported metrics include precision, recall, and prediction error.

DOI: 10.46121/pspc.54.3.23

Power System Protection and Control
ISSN:1674-3415

Table 6. Model performance with and without crowdsourced data

Metric	Baseline model	Integrated model	Improvement (%)
Precision@5	0.41	0.48	+17
Precision@10	0.37	0.41	+12
Recall@10	0.32	0.39	+22
RMSE	0.182	0.164	−9.8
MAE	0.139	0.124	−10.8

The results in Table 6 showed that the integrated version outperforms the baseline version across all key metrics. The precision at the top ten recommendations rose by about twelve percent, and the recall rose by about twenty-two percent. The RMSE and MAE values also fell by close to ten percent each. This simultaneous reduction in error and rise in accuracy shows that the explicit data added meaningful and complementary information to the model. The performance gain is more pronounced for low-activity users, since the explicit data played a compensating role in these cases.

The change in the density of the interaction matrix before and after integrating the explicit data is shown in Figure 3. The matrix density in the baseline version was only 6.3 percent, and a large portion of the matrix cells remained empty. After adding the explicit data, this value rose to 28.9 percent, which represents a near four-and-a-half-fold increase in density. This jump means that a wide portion of the empty cells was filled with real information on user preferences. The reduction of empty space in the matrix made the estimation of hidden relations among users more stable and lowered the error caused by data sparsity. This rise in density is the main driver of the improved similarity computation and the reduced cold-start effect in the integrated version.

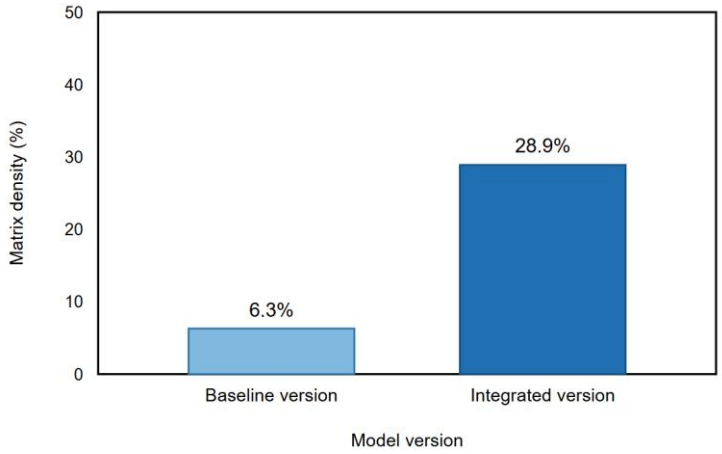

Figure 3. Density of the user–item interaction matrix before and after crowdsourcing integration.

Final Quantitative Evaluation

A full comparison of the four model versions across all quantitative metrics is presented in Table 7. The table places the baseline, integrated, hybrid, and optimized versions side by side. The reported metrics cover precision, recall, the F1 score, prediction error, diversity, and coverage. The final column reports the improvement of the final version relative to the baseline version.

Table 7. Performance comparison of the four model versions across all metrics

Metric	Baseline	Integrated	Hybrid	Optimized	Improvement over baseline (%)
Precision@10	0.37	0.41	0.44	0.45	+22
Recall@10	0.32	0.39	0.43	0.44	+37
F1@10	0.34	0.39	0.42	0.43	+26
RMSE	0.182	0.164	0.158	0.151	−17
Diversity	0.43	0.51	0.57	0.58	+35

DOI: 10.46121/pspc.54.3.23

Coverage	0.21	0.29	0.32	0.34	+62

The results in Table 7 showed that the optimized version holds the highest performance across all metrics. The precision at the top ten recommendations rose from 0.37 in the baseline version to 0.45 in the optimized version. The largest improvement belongs to coverage with sixty-two percent and recall with thirty-seven percent, which shows the increased ability of the model to retrieve desirable items. The RMSE also fell by seventeen percent, which confirms the higher prediction accuracy. The simultaneous rise in accuracy, diversity, and coverage is an important achievement, since these three metrics usually trade off against one another. This upward trend from the baseline version to the optimized version reveals the cumulative effect of crowdsourcing integration and optimization.

The improvement trend of the three main accuracy metrics across the four model versions is shown in Figure 4. The precision at the top ten recommendations begins at 0.37 in the baseline version and rises to 0.41, 0.44, and 0.45 in the subsequent versions. The recall follows a steeper trend and rises from 0.32 to 0.44, which marks the largest improvement slope. The F1 score also rises from 0.34 to 0.43 and confirms the steady growth across all three metrics. The gap between the baseline and integrated versions reveals the effect of adding crowdsourced data. The gap between the hybrid and optimized versions reflects the effect of parameter tuning and the added content layer. This upward pattern shows that the performance gain results from an effective structural change rather than mere parameter adjustment.

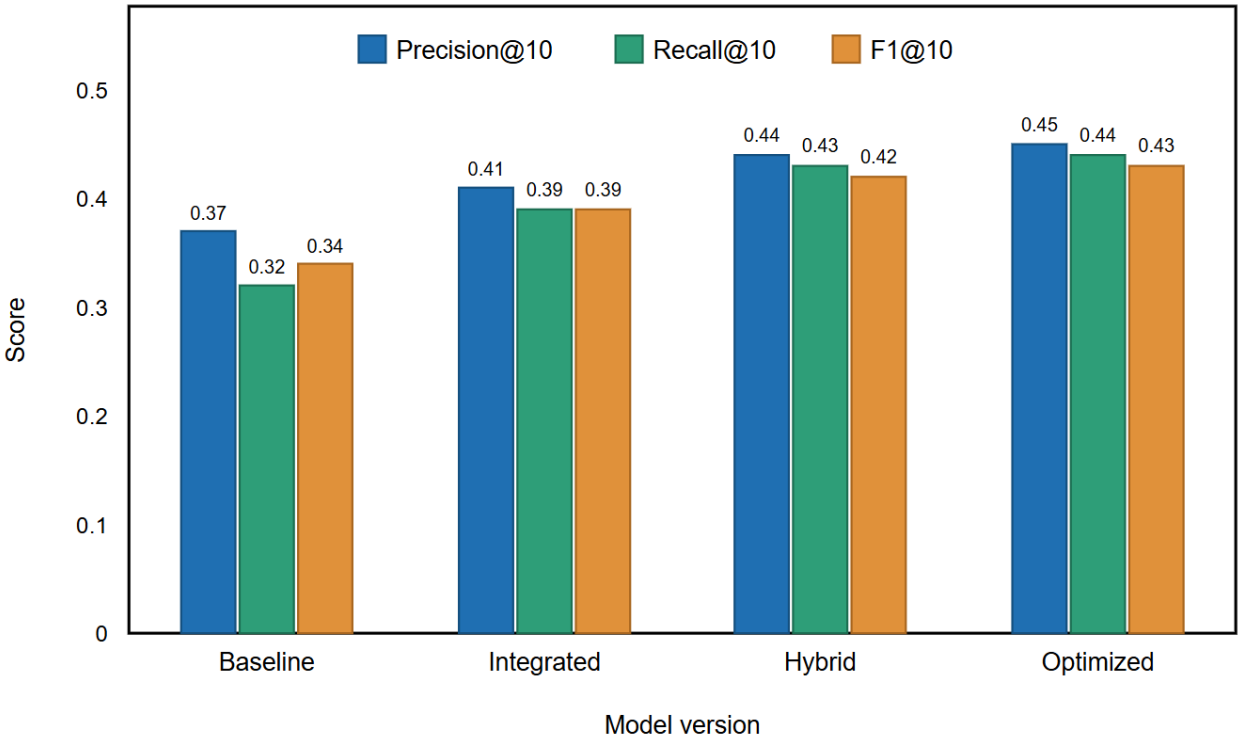

Figure 4. Comparison of Precision@10, Recall@10, and F1@10 across the four model versions.

Optimized Parameters and Model Efficiency

The key adjustments to the parameters and the model structure in the optimization phase are presented in Table 8. The table reports the value of each parameter before and after optimization, together with the reason for each change. These parameters were tuned based on the evaluation results of the earlier versions. The aim of these adjustments was to raise the stability of the similarities and to reduce the prediction error.

DOI: 10.46121/pspc.54.3.23

Power System Protection and Control
ISSN:1674-3415

Table 8. Optimized parameters of the final model

Component	Before optimization	After optimization	Reason for change
Number of neighbors K	20	28	Higher similarity stability
Explicit-data weight	0.30	0.38	Positive effect in the integrated version
Similarity measure	Cosine	Cosine and Pearson	Lower sensitivity to noise
Learning rate	0.010	0.005	Reduced prediction fluctuation
Regularization coefficient	—	0.12	Prevention of overfitting

The results in Table 8 showed that raising the number of neighbors from twenty to twenty-eight increased the stability of the similarity computation. Raising the explicit-data weight to 0.38 also reduced the fluctuation of the similarities and made the ranking more stable. Changing the similarity measure to a cosine–Pearson combination lowered the model's sensitivity to noisy scores. Reducing the learning rate and adding the regularization coefficient also prevented overfitting. Alongside these adjustments, the rearrangement of the compressed matrix reduced the processing time of the final version by close to nineteen percent.

Qualitative and Real-World Evaluation

The qualitative user assessment and the real-world performance of the model are presented in Table 9. The first part of the table reports the users' ratings of five dimensions of the user experience on a five-point scale. The second part reports the performance metrics of the final version in a twelve-day field test with two hundred and forty users. Together, these two parts offer a picture of the perceived quality and the practical efficiency of the model.

Table 9. Qualitative user assessment and real-world performance of the final version

Part	Indicator	Value
Qualitative assessment	Relevance of recommendations	4.2
Qualitative assessment	User trust	4.1
Qualitative assessment	Response speed	4.0
Qualitative assessment	Diversity of suggestions	3.8
Qualitative assessment	Transparency of recommendations	3.4
Real-world performance	Precision@10	0.44
Real-world performance	Recall@10	0.42
Real-world performance	Overall user satisfaction	4.1
Real-world performance	Mean response time (seconds)	0.17

The results in Table 9 showed that the highest qualitative score belongs to the relevance of recommendations with a value of 4.2. Trust and response speed, with scores close to four, also confirm a suitable level of user satisfaction. The lowest score belongs to transparency with a value of 3.4, which represents a structural weakness of the current version. In the real-world part, the precision and recall values are very close to the laboratory results, which shows the stability of the model under real conditions. The mean response time of 0.17 seconds also confirms the model's capacity for deployment at a medium scale.

DISCUSSION

This study was conducted to design and evaluate a hybrid recommender system that strengthens collaborative filtering with crowdsourced data. The findings showed that adding explicit user data to the model steadily improves the accuracy and quality of the recommendations. The optimized version outperformed the baseline version across all metrics and noticeably reduced the cold-start problem. This achievement confirms that the

DOI: 10.46121/pspc.54.3.23

informed participation of users can play a complementary role for implicit behavioral data. In what follows, each key finding is analyzed in light of prior research.

The results showed that crowdsourcing increases the richness and density of the behavioral data, and on this basis the first hypothesis was confirmed. Each interaction shifted from a simple event to a set of qualitative indicators, and the mean amount of useful information per user grew several times. This change deepened the preference profile of the users. Rahi et al. (2025) [28] reached a similar result and showed that combining crowdsourcing with artificial intelligence raises data quality in public-health systems. Juárez-Varón and Monreal (2025) [29] also reported that crowdsourcing through social media strengthens organizational learning and knowledge richness. The alignment of these two studies with the present finding reinforces the role of explicit data in filling the gap left by implicit data.

The results also showed that the structured integration of explicit and implicit data improves model accuracy, and on this basis the second hypothesis was confirmed. Merging the two data sources raised precision and recall at all evaluation levels and lowered the prediction error. This improvement showed that explicit data add meaningful and complementary information to the model rather than mere volume. Mirbahar et al. (2025) [32] reported a similar finding and showed that crowdsourced educational data improve the accuracy of mobile-app recommendations. Darban and Valipour (2022) [21] also proved, in their hybrid model, that combining diverse data sources raises recommendation performance over single-source methods. This consistency confirms the effectiveness of the proposed integrated structure for improving accuracy.

In addition, the results showed that crowdsourced data reduce the cold-start problem, and accordingly the third hypothesis was confirmed. Adding the explicit data raised the density of the interaction matrix from 6.3 percent to 28.9 percent and improved recommendation quality for low-activity users. This rise in density made the estimation of users' hidden relations more stable. Chen et al. (2018) [10] pointed to a consistent conclusion in their review and identified data sparsity and cold start as the main challenges of collaborative filtering, which complementary data sources can ease. Ahsan et al. (2025) [31] also showed that crowdsourced data cover the data-deficient points in monitoring systems. This agreement reinforces the role of crowdsourcing in easing the cold-start problem.

Beyond the main hypotheses, the results showed that the item-based approach performs more stably under sparse data. This complementary finding emerged outside the primary hypotheses. Item similarity proved more stable than user similarity, and the item-based model recorded higher accuracy in the baseline version. This stability traces back to the clustered structure of the content. Anwar and Uma (2021) [11] reached a similar result and found the item-based approach more stable on sparse data. Sanwal and Çaliskan (2021) [24] also highlighted, in their hybrid model, the role of stable item relations in improving prediction. This alignment justifies the choice of the item-based approach as the reference configuration.

Furthermore, the results showed that the optimized version raises accuracy, diversity, and coverage at the same time. The model lifted all three metrics together, although these metrics usually trade off against one another. This outcome showed that the hybrid structure prevents the loss of diversity when accuracy rises. Isufi et al. (2021) [17] reached a similar finding and showed that careful model design can moderate the trade-off between accuracy and diversity. Khalaji et al. (2021) [23] also reported a simultaneous improvement in accuracy and coverage in their resource-allocation-based hybrid model. This consistency confirms the favorable balance that the proposed model strikes among these metrics.

Finally, the results showed that transparency is the weakest dimension of the system from the users' viewpoint. Transparency received the lowest qualitative score, yet adding explanation tags in the final version raised it from 3.4 to 4.0. This improvement showed that offering a short reason for each recommendation enhances the user experience. Samih et al. (2021) [27] reached a consistent result and showed that explainable mechanisms increase user trust and satisfaction. Jannach et al. (2021) [14] also identified explainability as a key factor in the adoption of recommender systems. This agreement highlights the development of explainability as a path for future improvement.

This study faced several limitations that define the scope for generalizing its findings. The data were gathered from a single social-network platform and a single content domain, and the crowdsourcing sample was limited in size. These conditions allowed the model to be tested in a defined environment but make its generalization to all

DOI: 10.46121/pspc.54.3.23

platforms conditional. For future research, testing the model across several platforms and diverse content domains is suggested. Adding advanced explainability mechanisms and using deep-learning methods for dynamic weighting can also further strengthen the performance and adoption of the system.

CONCLUSION

Recommender systems have become a key tool for managing information overload in social networks, yet their performance is limited by challenges such as data sparsity and cold start. On this basis, the present study was conducted to design and evaluate a hybrid recommender system that strengthens collaborative filtering with crowdsourced data. The study followed an applied and developmental design, and its data were gathered from an active social-network group in the health-news domain. The implicit data were drawn from the users' behavioral traces, and the explicit data were obtained from a researcher-made questionnaire across four dimensions, namely satisfaction, relevance, transparency, and trust. The model performance was measured with precision, recall, the F1 score, coverage, diversity, and prediction error. The findings showed that adding crowdsourced data noticeably increases the richness and density of the behavioral data and turns each interaction from a simple event into a set of qualitative indicators. The structured integration of explicit and implicit data also raised precision and recall at all levels and lowered the prediction error, so that the precision of the optimized version reached 0.45 at the top ten recommendations and its recall reached 0.44. In addition, the explicit data raised the density of the interaction matrix from 6.3 percent to 28.9 percent and improved recommendation quality for low-activity users, which indicates an effective reduction of the cold-start problem. Overall, the optimized version outperformed the baseline version across all metrics, and the largest improvement was achieved in coverage with sixty-two percent. The main novelty of this study lies in its human-centered framework, in which the user acts at once as a data source and a validator of the recommendations. This approach merges explicit crowdsourced data with implicit behavioral data in a dynamic structure and offers a practical path for improving recommendation quality in social networks. The proposed model is deployable at a medium scale and can be applied to social-content platforms. For further strengthening, developing explainability mechanisms and using deep-learning methods for dynamic weighting are suggested as directions for future research.

REFERENCES

1. Frauenstein, E. D., & Flowerday, S. (2020). Susceptibility to phishing on social network sites: A personality information processing model. Computers & Security, 94, 101862.
2. Beigi, G., & Liu, H. (2018). Privacy in social media: Identification, mitigation and applications. arXiv preprint arXiv:1808.02191.
3. Singh, S. S., Singh, K., Kumar, A., & Biswas, B. (2020). ACO-IM: Maximizing influence in social networks using ant colony optimization. Soft Computing, 24(13), 10181-10203.
4. Deng, X., Zhong, Y., Lü, L., Xiong, N., & Yeung, C. (2017). A general and effective diffusion-based recommendation scheme on coupled social networks. Information Sciences, 417, 420-434.
5. Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. Journal of Big Data, 9(1), 59.
6. Campana, M. G., & Delmastro, F. (2017). Recommender systems for online and mobile social networks: A survey. Online Social Networks and Media, 3, 75-97.
7. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018, July). Graph convolutional neural networks for web-scale recommender systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 974-983).
8. Jesse, M., & Jannach, D. (2021). Digital nudging with recommender systems: Survey and future directions. Computers in Human Behavior Reports, 3, 100052.
9. Xu, K., Zhang, W., & Yan, Z. (2018). A privacy-preserving mobile application recommender system based on trust evaluation. Journal of Computational Science, 26, 87-107.

DOI: 10.46121/pspc.54.3.23

10. Chen, R., Hua, Q., Chang, Y. S., Wang, B., Zhang, L., & Kong, X. (2018). A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. IEEE Access, 6, 64301-64320.
11. Anwar, T., & Uma, V. (2021). Comparative study of recommender system approaches and movie recommendation using collaborative filtering. International Journal of System Assurance Engineering and Management, 12(3), 426-436.
12. Chen, X., Yao, L., McAuley, J., Zhou, G., & Wang, X. (2021). A survey of deep reinforcement learning in recommender systems: A systematic review and future directions. arXiv preprint arXiv:2109.03540.
13. Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., & He, Q. (2020). A survey on knowledge graph-based recommender systems. IEEE Transactions on Knowledge and Data Engineering, 34(8), 3549-3568.
14. Jannach, D., Manzoor, A., Cai, W., & Chen, L. (2021). A survey on conversational recommender systems. ACM Computing Surveys (CSUR), 54(5), 1-36.
15. Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M. A., & Lian, D. (2021). A survey on session-based recommender systems. ACM Computing Surveys (CSUR), 54(7), 1-38.
16. del Carmen Rodríguez-Hernández, M., & Ilarri, S. (2021). AI-based mobile context-aware recommender systems from an information management perspective: Progress and directions. Knowledge-Based Systems, 215, 106740.
17. Isufi, E., Pocchiari, M., & Hanjalic, A. (2021). Accuracy-diversity trade-off in recommender systems via graph convolutions. Information Processing & Management, 58(2), 102459.
18. Shambour, Q. (2021). A deep learning based algorithm for multi-criteria recommender systems. Knowledge-Based Systems, 211, 106545.
19. Huang, H., Mu, J., Gong, N. Z., Li, Q., Liu, B., & Xu, M. (2021). Data poisoning attacks to deep learning based recommender systems. arXiv preprint arXiv:2101.02644.
20. Krishna, C. V. M., Rao, G. A., & Anuradha, S. (2023). Analysing the impact of contextual segments on the overall rating in multi-criteria recommender systems. Journal of Big Data, 10(1), 16.
21. Darban, Z. Z., & Valipour, M. H. (2022). GHRS: Graph-based hybrid recommendation system with application to movie recommendation. Expert Systems with Applications, 200, 116850.
22. Lee, C., Han, D., Han, K., & Yi, M. (2022). Improving graph-based movie recommender system using cinematic experience. Applied Sciences, 12(3), 1493.
23. Khalaji, M., Dadkhah, C., & Gharibshah, J. (2021). Hybrid movie recommender system based on resource allocation. arXiv preprint arXiv:2105.11678.
24. Sanwal, M., & Çaliskan, C. (2021). A hybrid movie recommender system and rating prediction model. International Journal of Information Technology and Applied Sciences, 3(3), 161-168.
25. Wang, Z., Chen, H., Li, Z., Lin, K., Jiang, N., & Xia, F. (2021). VRConvMF: Visual recurrent convolutional matrix factorization for movie recommendation. IEEE Transactions on Emerging Topics in Computational Intelligence, 6(3), 519-529.
26. Choudhury, S. S., Mohanty, S. N., & Jagadev, A. K. (2021). Multimodal trust based recommender system with machine learning approaches for movie recommendation. International Journal of Information Technology, 13(2), 475-482.

DOI: 10.46121/pspc.54.3.23

27. Samih, A., Ghadi, A., & Fennan, A. (2021). ExMrec2vec: Explainable movie recommender system based on Word2vec. International Journal of Advanced Computer Science and Applications, 12(8).
28. Rahi, P., Dandotiya, M., Basa, S., Sen, S., Jakhete, M. D., & Vijayakumar, P. (2025). Crowdsourcing and artificial intelligence based modeling framework for effective Public Healthcare Informatics and Smart eHealth System. In Artificial Intelligence in Biomedical and Modern Healthcare Informatics (pp. 597-608). Academic Press.
29. Juárez-Varón, D., & Monreal, E. G. (2025). Market-oriented entrepreneurship and the impact of social media and crowdsourcing on continuous organizational learning. An empirical study. Sustainable Technology and Entrepreneurship, 4(1), 100088.
30. Munir, Y., Umer, Q., Faheem, M., Akram, S., & Jaffar, A. (2025). Developer recommendation and team formation in collaborative crowdsourcing platforms. IEEE Access.
31. Ahsan, R., Panja, A. K., Roy, M., & Chowdhury, C. (2025). Design of an end-to-end recommendation system for crowdsourced road monitoring applications based on machine learning. Multimedia Tools and Applications, 84(30), 37191-37213.
32. Mirbahar, N., Kumar, K., & Laghari, A. A. (2025). Enhancing mobile app recommendations with crowdsourced educational data using machine learning and deep learning. IEEE Access, 13, 38778-38792.
33. Yin, X., Li, J., Zhu, K., Wang, W., & Wang, H. (2025). Willing and able: Task recommendation with a trade-off of the bilateral benefits for knowledge-intensive crowdsourcing. Decision Support Systems, 190, 114400.

DOI: 10.46121/pspc.54.3.23