A CONFIDENCE RECOMMENDATION ALGORITHM FOR ENHANCING RECOMMENDATION QUALITY AND ATTACK RESISTANCE UNDER SPARSE DATA

Guiqin DOU[1], Yuyang GUO[1], Yansong ZHU[1]

To address the challenges of data sparsity and vulnerability to malicious attacks in traditional recommender systems, this paper proposes a novel confidence-based recommendation algorithm designed to enhance both recommendation quality and robustness under sparse data conditions. Central to our approach is the introduction of a trustworthiness function, which mitigates the impact of biased or deceptive ratings introduced by malicious users. By quantifying user trust, this function integrates trust metrics directly into the core decision-making processes of the recommender system. Additionally, we incorporate a user utility function that captures user feedback on recommendation quality, enabling a dynamic and rational update of trust values. This adaptive trust mechanism allows the system to respond effectively to evolving user behavior and feedback, improving long-term recommendation performance. The proposed algorithm is evaluated on the Movielens dataset, with experimental results demonstrating substantial improvements in both accuracy and coverage compared to conventional benchmark methods. Notably, our method maintains high recommendation quality even in adversarial scenarios involving malicious users.

Keywords: recommendation system; collaborative filtering; recommended quality; trustworthy service

1. Introduction

1.1 Background

Recommendation systems, based on users ’ historical behaviors and interest modeling, employ information filtering techniques to alleviate information overload. Common strategies include content-based, collaborative filtering, and hybrid approaches: the content-based method infers user preferences from historical data, while collaborative filtering identifies similar users through rating matrices to generate recommendations [1]. The hybrid approach integrates both methods, leveraging their strengths and compensating for their limitations to enhance recommendation accuracy and personalization.

With the development of service recommendation systems, various malicious behaviors against recommendation systems have also been derived, such

[1] Corresponding author’s e-mail: kuzj56976@163.com
Department of Computer Science, Zhongyuan University of Technology, Zhengzhou, China

as referral attacks， suppression attacks etc. In recommendation systems, some users act maliciously to manipulate service evaluations. They may inflate the ratings of services aligning with their interests, while deflating ratings of competing services. This behavior is often driven by the pursuit of financial gain. [2]. To enhance recommender system resilience against malicious attacks, researchers have explored trusted recommender systems. Foucart et al. [3] introduced a reputation-based trust algorithm for mobile agent e-commerce, weighting attributes and incorporating time-based trust dynamics. Shrivastava et al. [4] proposed using transaction history to build user reputation, integrated into a latent semantic collaborative filtering model. Barkan et al. [5] developed an evolutionary trust recommendation algorithm, filtering outliers with an X-bar control chart to mitigate malicious attacks. Iltaf et al. [6] employed Beta probabilistic models to calculate direct and indirect trust values, combining them for a comprehensive trust weight. Michael et al. [7] created an explicit trust-based recommendation model [8], assessing rating reliability and reconstructing the trust network based on positive and negative factors when reliability is low [9].

1.2 Our Research

General recommendation models often overlook how evaluation results relate to each other and how evaluation attributes connect. This can make it harder to distinguish good recommendations from bad ones and identify malicious or false recommendations, which ultimately hurts the accuracy of trust calculations. To address these issues, this paper introduces a social network trust recommendation algorithm that uses a trustworthiness metric. This aims to improve the system's robustness and trustworthiness, particularly in dealing with data sparsity and malicious attacks that challenge traditional recommendation systems.

Firstly, the concept of a credibility function is introduced. This function is designed to lessen the impact of malicious evaluations on recommendation outcomes. By integrating user interaction experience with the trustworthiness function, we can calculate trust values between users more precisely. This allows us to filter out unreliable users and improve the overall accuracy of recommendations.

Secondly, this paper establishes a feedback model of recommendation quality and defines a user utility function to realize the dynamic update of the trust relationship. This flexible mechanism enables the recommender system to better adapt to changes in user interests, thus providing more personalized and accurate recommendation services.

Thirdly, experiments conducted on the Movielens dataset demonstrate that the proposed algorithm outperforms traditional methods in accuracy and coverage. Notably, the system maintains recommendation quality even with malicious users, improving reliability.

This social network trust recommendation algorithm offers a novel approach to addressing data sparsity and malicious behavior, supported by experimental results that validate its superior performance and contribution to user satisfaction.

2. Construction of Recommendation Models

According to the theory of social cognition, the behavior of the individual, the cognition of the subject and the social environment are dynamically influenced by each other. In traditional transactions, individuals seek to minimize risk by evaluating the transaction object and the trustworthiness of the other party. This often involves assessing the other party's reputation or seeking validation from others within a social network, forming a trust network.

2.1 Basic Concepts of the Web of Trust

1) Node: A node is defined as an individual entity or object within the trust network, which can take on various roles such as evaluator, recommender, or evaluation object, among others.

2) The trust degree: The trust degree, a quantifiable measure on a scale of 0 to 1, reflects the level of trust between nodes in this network. A value of 0 represents complete distrust, while 1 indicates complete trust.

3) The trust relationship: In recommendation systems, trust is crucial for effective evaluation. A trust relationship links the evaluator and the item being evaluated, fostering mutual recognition. This link can be direct, based on past interactions, or indirect. Without direct experience, evaluators rely on recommendations from trusted sources. These recommenders, deemed highly trustworthy, allow the evaluator to infer a trust value for the item – this constitutes the recommendation trust relationship.

4) The trust path: The trust path signifies tracing trust from one entity to another through a network, identifying a chain of trustworthy connections.

5) The continuous trusted service subsequence: It describes a series of reliable interactions between two entities over time.

2.2 Trust relationship

In the trust network, the trust relationship often exhibits the characteristics of time sensitivity, asymmetry and transitivity, and it is dynamic. Time sensitivity means that the degree of trust will decrease with time, and the trust decay can be used. Function to simulate the change rule of trust relationship between long-term non-interacting nodes. Asymmetry means that the trust relationship between individuals is asymmetric. Suppose that the individual's trust degree to the individual is expressed as T_{uv}, The individual-to-individual trust scale is T_{vu},not necessarily satisfied. Transitivity means that trust relationships can be transmitted through individuals, Its transmission follows a certain communication strategy.

According to the previous definition of trust relationship, Direct trust comes from the accumulation of experience in the interaction between two individuals, which can be measured by the similarity of individuals. For individual u to individual v, If the two are more similar, Indicates that they can be grouped into the same category, they can trust each other. Individual trust derivation using method JMSD [10], Gain direct trust between individuals, so:

$$T_{uv} = (1 - \frac{\sum_{i \in I_{uv}} (R_{ui} - R_{vi})^2}{I_{uv}}) \times \frac{|I_u \cap I_v|}{I_u \cup I_v} \tag{1}$$

Here I_u , I_v respectively represent the user u and v set of ratings. $I_u \cap I_v$ for users u, v A collection of common rating items, R_{ui} and R_{vi} is the normalized parameter after processing.

2.3 Definition of credibility function

By definition2 of the trust degree, in recommendation systems, direct trust between users is established based on shared item ratings. If two users have rated the same item, a non-zero trust value is assigned; otherwise, the trust is zero. Sociological research suggests that interaction influences trust. Therefore, in recommender systems, trust values should dynamically adjust based on user interactions, specifically, their ratings. To account for both positive and negative feedback, user ratings are categorized relative to their average rating. Ratings equal to or exceeding the user's average are considered positive, while those below are deemed negative.

An interaction between two users is successful only when both users rate an item as either positive or negative, otherwise it fails. then:

$$Int_{uv} = \begin{cases} 1, \ (R_{ui} - \tilde{R}_u)(R_{vi} - \tilde{R}_v) \geq 0 \\ 0, \ (R_{ui} - \tilde{R}_u)(R_{vi} - \tilde{R}_v) < 0 \end{cases} \tag{2}$$

An expression value of 1 indicates that the interaction is successful, 0 means the interaction failed. Use the counters sus and fal to record the consecutive number of successes and failures of two user interactions.

In real life, the degree of trusted interaction is closely related to the persistence of behavior, That is, individuals are more able to trust objects with whom they have continuous and credible interactions, on the contrary, Trust in unstable objects gradually weakens. In order to characterize the above characteristics in a measure of direct trust, this paper introduces the credibility function, its definition is as follows:

$$\begin{cases} F(u,v) = \dfrac{\sum_{k=1}^{m} g(k) \times length_k(u,v)}{M} \\ g(k) = \gamma^{(sus_k - fal_k)/(sus_k + fal_k)} \end{cases} \tag{3}$$

$length_k(u,v)$ Indicates the length of the k persistent trusted service subsequence, $g(k)$ is the time and sequence based decay factor of the k persistent credible subsequence, M is the total number of services based on time and duration sequence, parameter $\gamma \in (0,1)$.

Credibility functions in recommendation systems serve a dual purpose. First, they mitigate the impact of biased or malicious ratings by weighting interactions based on trustworthiness. Second, they incentivize users to consistently offer honest and dependable service by rewarding persistent, reliable behavior.

2.4 Calculation of direct and indirect trust

Introducing the credibility function to the measure of direct trust value, the direct trust calculation formula combining user interaction experience and credibility function evaluation is as follows:

$$DT_{uv} = T_{uv} \times F(u,v) \tag{4}$$

By definition of the trust relationship, in a social trust network, when there is no direct trust relationship between the two users, Indirect trust relationships that can be obtained from a network of trust through trust transfer. E.g, User x exists trusting user y, and user y trusts user z, Then by establishing the trust transfer matrix, you can get the trust level of user x for user z, this process is called trust aggregation. The main trust aggregation methods are max-based, Min-based and mean-based, etc. This paper adopts the method of weighted mean aggregation, In order to ensure that the trust value of the individual closer to the source individual on the trust path can obtain a larger weight value, ensuring the reliability of trust delivery. Then, the indirect trust calculation method is:

$$\begin{cases} IT_{uv} = \sum\limits_{v \in Neighbor(u)} DT_{uv} \times (DT_{uv} \times \delta_d) / \sum\limits_{v \in Neighbor(u)} DT_{uv} \\ \delta_d = (\max_0 - d_{uv} + 1) / \max_0 \end{cases} \tag{5}$$

The $Neighbor(u)$ represents the set of neighbor individuals of user u, and need to satisfy $DT_{uv} \geq \lambda$ (λ is a trust threshold) to ensure that individuals with low credibility are not involved in the trust transfer process. Taking into account that indirect trust decays with distance, δ_d as an adjustment factor, it is used to set different weights for individuals with different distances. $\max_0$ The maximum number of hops allowed for reliable links for the recommender system, d_{uv} distance between individuals.

2.5 Dynamic update of trust

To model feedback on recommendation quality, it's need to set up a utility function for the user first. Define the utility function for each user $U(t)$ is the utility obtained after experiencing the recommendation of each neighbor node, The formal description is as follows:

$$U_{uv}(t) = 1 - |R_u - R_v| \tag{6}$$

$U_{uv}(t)$ Represents the utility value of the individual at time t after the recommendation of the trusted neighbor node v of u. If user u does not directly score the item, the expected score value on the item can be estimated from the score values of all trusted neighbor nodes, then:

$$R_u = \sum_{v\in Neighbor(u)} R_v \times \frac{IT(u,v)}{\sum_{w\in Neighbor(u)} IT(u,w)} \tag{7}$$

A recommendation system's utility function assesses the value of a neighbor's recommendation to an individual. The utility function increases when a neighbor's rating of an item aligns with the individual's own rating. This suggests the neighbor provides valuable recommendations, increasing the individual's trust. Conversely, a low utility value indicates a poor recommendation, which decreases the individual's trust in that neighbor [11].

Therefore, the update of trust can be described formally using the utility function:

$$T_{uv}(t+1) = \begin{cases} T_{uv}(t) + U_{uv}(t) \times (1+\varepsilon),\ if\ U_{uv}(t) \ge U^* \\ [T_{uv}(t) - U_{uv}(t)]/(1+\varepsilon),\ else \end{cases} \tag{8}$$

U^* is a threshold used to distinguish between good and bad utility function values.ε is the scaling factor, Used to reflect the rate at which trust accumulates and is depleted, Generally it should be expressed as slow accumulation and fast consumption. Based on the above description, the specific recommendation algorithm process is summarized as follows.

Trust Network Recommendation Model Algorithm Flow

```
  Input：user set $U$，item set I，User-item rating data D，Maximum training rounds T，
Time decay factor λ, Trust threshold θ ,Update rate parameter β , τ
  Output：Top-K recommendation list
1. Initialize nodes and trust relationships:
        For each user u ∈ U:
          u.ratings ← D[u]  // User ratings for items
          u.avg_rating ← mean(u.ratings.values())  // Average rating
          u.trust_neighbors ← []  // Initialize trust neighbor list
          u.direct_trust ← {}   // Initialize the direct trust dictionary
          u.indirect_trust ← {}  // Initialize the indirect trust dictionary
        For each item i ∈ I:
          i.rating_history ← get_ratings(i)  // Get historical rating records
2. Build the initial trust network:
3. Training/update cycle:
  For iteration t = 1 to T do:
4.   For each user u ∈ U, execute in parallel:
5.     Build local trust subgraph:
         neighbors = [v for v in u.trust_neighbors if u.direct_trust[v] ⩾ θ ]
6.     Make recommendation predictions for items not directly rated:
```

```
            recommendations = {}
            for item in I:
               if item.id not in u.ratings:
                  predicted_score = estimate_item_score(u, item)
                  recommendations[item.id] = predicted_score
7. Sort and generate Top-K recommendations:
            ranked_items = sort_by_value(recommendations, descending=True)
            top_k_recommendations = take_top_k(ranked_items, K)
8. Simulate user feedback:
            actual_ratings = get_user_actual_ratings(u, top_k_recommendations)
            utility = compute_utility(u, top_k_recommendations)
9. Dynamic update of trust:
10. End for
11. End loop
12. Return final recommendation results:
```

3. Results and Analysis

This study investigates the performance of a specific recommendation model using two common datasets: Book-Crossings and MovieLens. The model's algorithmic process was applied to both datasets, and similar results were observed across both under the same parameter configurations. To conserve space, this paper will focus on the analysis and presentation of the validation results obtained from the MovieLens dataset. The typical MovieLens in recommender system research (The dataset contains 943 users, 1682 movies, and 100,000 ratings) [12]. Considering the manifestation of data sparsity, The proportion of unrated items in the entire dataset is 75.3%, The scoring attribute ranges from 1 to 5.The parameters used in the implementation of this algorithm are as follows:

Setting the trust threshold to 0.7 maximizes the average user influence. This aligns with real-world observations where users tend to accept roughly half of recommendations. Furthermore, a positive correlation exists between the frequency of information exchange and trust among users [13]. Consequently, a trust threshold of 0.7 represents an optimal balance, as illustrated in Fig. 1.

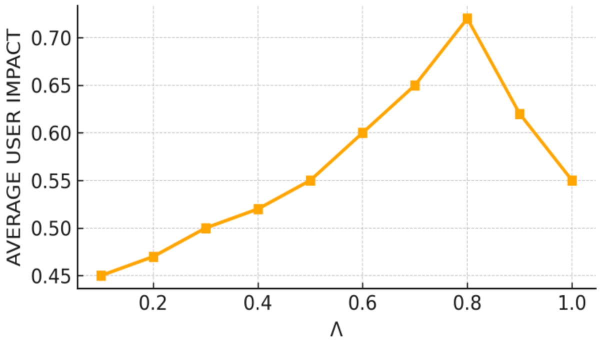

Fig. 1. The effect of different parameter λ value on the average influence of users

The maximum distance $\max_0$ of a reliable link allowed is 4 mean absolute error (Mean Absolute Error, MAE) and coverage (Coverage). The former is by comparing the deviation between the predicted score and the actual score, Accuracy as an Algorithm Prediction. The latter is a measure of the percentage of all items that an algorithm can predict. The definitions of the two-algorithm metrics are shown in equations (9) and (10).

$$MAE = \sum_{i=1}^{n} \left| P_i - R_i \right| / n \tag{9}$$

P_i is the user's actual rating of the target, R_i is the predicted score, n is the number of predicted scores.

$$COV = \sum_{u \in U} \left| P(u) \cap R(u) \right| / \sum_{u \in U} \left| P(u) \right| \tag{10}$$

$P(u)$ is the score set of user u on the test set, $R(u)$ is the set of items recommended by user u.

This study investigates the influence of algorithm parameters on recommendation accuracy, specifically focusing on the scaling factor and utility function. Initially, users with limited recommendation ability, determined by a queue-based threshold, were excluded from the nearest neighbor set. The distribution of recommendation abilities among core users was then analyzed (Fig. 2).

Experimental results demonstrate that increasing the threshold initially improves recommendation accuracy. However, exceeding an optimal threshold value leads to a decline in performance. This is attributed to the stricter consistency requirements between a user's item ratings and those of their trusted neighbors, negatively impacting items with sparse data. The value of the adjustment parameter has an impact on the update of trust, as you can see, when the value of ε is 0.4 optimized the speed of trust accumulation and consumption, Expressed as slow accumulation and fast consumption. At the same time, when the utility function value is 0.6, the MAE is optimal. Fig. 3 shows the effect of scaling factor and utility function threshold setting on MAE. When the value is set to 0.4, a balanced trade-off between the accumulation and decay rates of trust is achieved. A slower trust accumulation rate prevents early reliance on unreliable trust relationships, while a faster decay rate effectively removes invalid or inaccurate trust connections, thereby maintaining the stability and accuracy of the recommendation system.

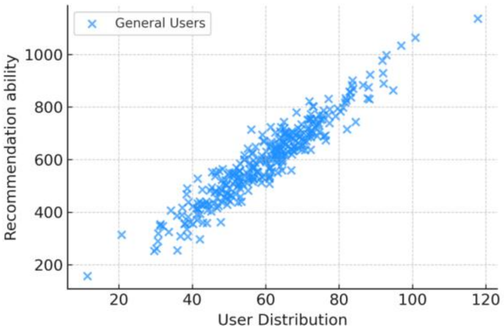

Fig. 2. Distribution of users' recommendation ability

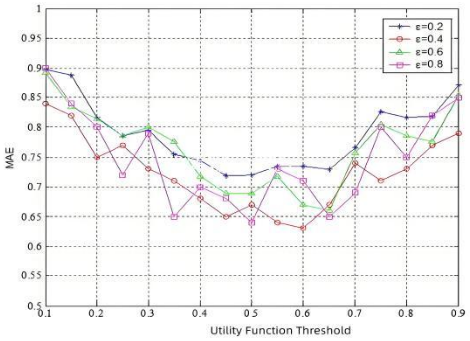

Fig. 3. Influence of algorithm parameters on MAE

To verify the effectiveness of the algorithm, experiments were compared with traditional algorithms such as CF, [14]Trust-aware and Merge .The value range of the number of neighbors is [10, 90]， The utility function threshold U^* of this paper is set to 0.6,The value of the adjustment parameter ε is 0.4.Fig. 4 is a comparison of the accuracy, It can be seen that as the number of neighbors increases, the mean absolute error decreases significantly, In contrast, the accuracy of the algorithm in this paper shows a good performance when there are few neighbors, And overall, the accuracy of the algorithm proposed in this paper has always been better than other algorithms. This is because the credibility measure can better improve the selection of trusted neighbor nodes and reduce the impact of negative scores.

Fig. 5 illustrates how the coverage rate changes with varying numbers of neighbors. The results indicate that increasing the number of neighbors generally improves the coverage rate for all tested recommendation algorithms. Our proposed algorithm demonstrates a significant performance improvement compared to traditional collaborative filtering (CF) and trust-aware approaches. These traditional methods, which rely on shared user ratings for similarity calculation, exhibit lower coverage. The Merge algorithm, by incorporating common user scores, achieves a moderate performance gain. While data sparsity in the Movielens dataset initially limits our algorithm's performance with few neighbors, the trust value update mechanism effectively compensates for the scarcity of shared ratings.

In order to reflect the performance of each algorithm under the trust attack. We randomly generate a certain proportion of malicious individuals in the data set to carry out the trust attack, Adopt the random attack model of literature [15].The neighbors of the target user are all set to 45, and the value of M is 1000. Fig. 6 shows the changes of MAE under different proportions of malicious users. It can be clearly seen that the increase in the proportion of malicious users makes the prediction accuracy of the recommendation algorithm decline, and brings about an increase in Mae, which shows that the trust attack behavior of malicious users does interfere with the recommendation results The anti-interference ability of the method in this paper is stronger, because the continuity of successful interaction is taken into account in scoring prediction. It is difficult for malicious users to improve their weight in scoring prediction of target individuals only by copying the evaluation attributes of target individuals.

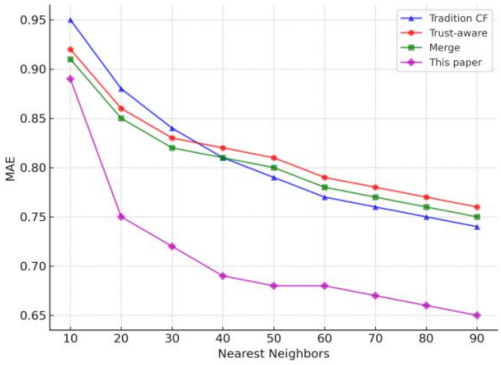

Fig. 4. The effect of the number of neighbors on MAE

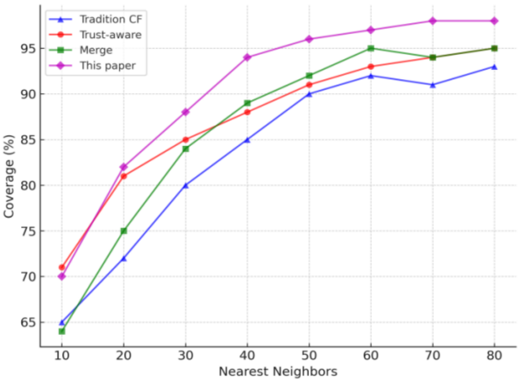

Fig. 5. The effect of the number of neighbors on the coverage

To simulate potential malicious attacks in real-world recommendation systems, we introduce three typical attack models.

Random Attack: Attackers randomly assign ratings to multiple items, increasing the uncertainty of system predictions and reducing recommendation accuracy.

Push Attack: Attackers give extremely high ratings to target items to artificially boost their recommendation weight, thereby disrupting the ranking.

Nuke Attack: Attackers assign extremely low ratings to specific items to decrease their recommendation probability, compromising the system’ s fairness.

In the experiment, we control the proportion of malicious users and set five different attack intensities at 5%, 10%, 15%, 20%, and 25% to observe the system’s performance under varying attack strengths. As shown in Fig. 7 below.

The trend graph shows the impact of the three attack types (random attack, push-up attack, and destroy attack) on the MAE of the recommender system for different percentages of malicious users. It can be seen that all attack types lead to an increase in MAE as the percentage of malicious users increases. Among them, the destruction attack (green dashed line) has the greatest impact on the system, while the random attack (blue solid line) has relatively less impact.

To ensure the stability of the recommendation system, this algorithm employs a dynamic trust update mechanism that adjusts trust levels based on users' historical behavior [16]. This approach mitigates the impact of malicious users' ratings on the system, thereby maintaining its stability [17].

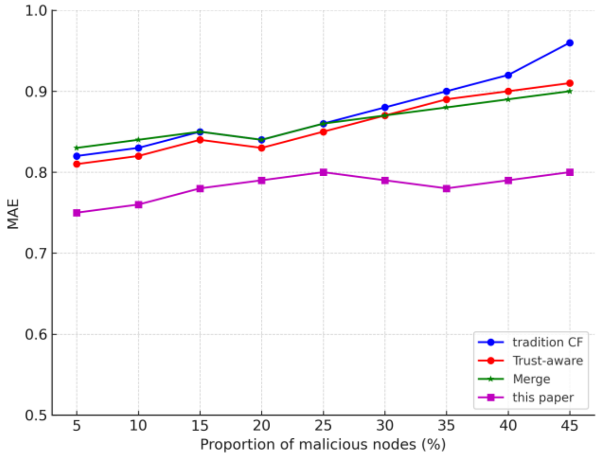

Fig. 6. Change of MAE under different proportions of malicious users

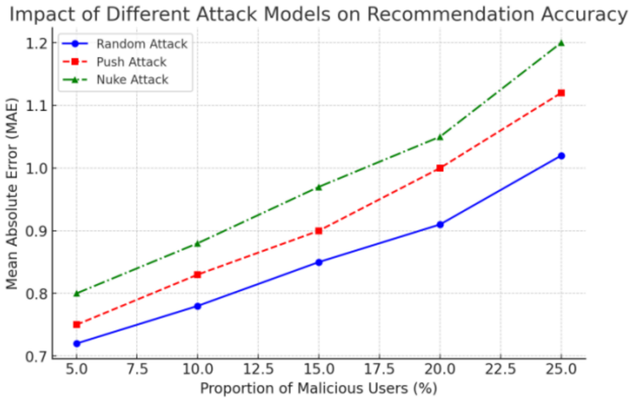

Fig. 7. Impact of attack types on MAE with different percentages of malicious users

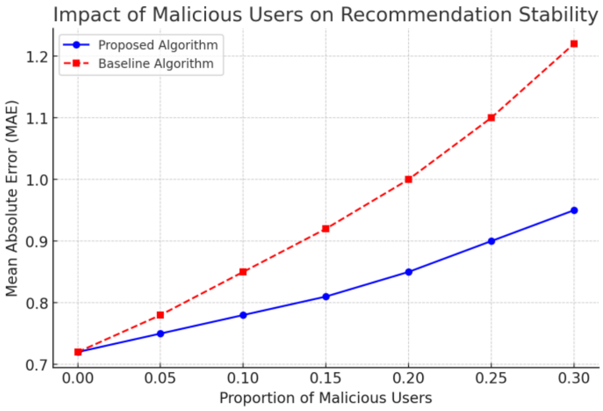

Fig. 8. Changes in Recommendation Stability under different proportions of malicious users

A test set comprising both normal and malicious users is selected, and a trust decay factor is used to evaluate the extent to which the influence of malicious users on the system is weakened over time. The disturbance caused by malicious users' ratings on the overall trust network is measured and compared with a baseline model (a recommendation system without trust updates), as shown in Fig. 8 above.

The trend graph illustrates the impact of different proportions of malicious users on the stability of the recommendation system. It can be observed that the proposed algorithm (solid blue line) exhibits minimal MAE variation as the proportion of malicious users increases, indicating strong robustness. In contrast, the traditional recommendation algorithm (dashed red line) is significantly affected by malicious users, with MAE rising rapidly, demonstrating its poor stability.

In recent years, neural network – based and graph model – based approaches have followed distinct trajectories in the measurement of trustworthiness for social recommendation. Neural network methods typically employ multi-layer nonlinear architectures to model user – item interactions and social relations, where attention mechanisms or weighting strategies are applied to dynamically assign trust scores across different relationships. These methods excel in feature fusion and nonlinear relation modeling, making them particularly effective in multimodal recommendation scenarios involving text, images, or behavioral sequences. However, they often suffer from high training costs, increased model complexity, and limited scalability in large-scale social networks.

By contrast, graph-based approaches (e.g., GCN, GAT, and their variants) operate directly on user – item or social trust graphs, embedding trustworthiness into edge weights and capturing higher-order trust relations through multi-hop propagation. These methods demonstrate stronger capabilities in structured relation modeling and robustness, effectively mitigating the problem of data sparsity. Nevertheless, they face considerable computational and storage overhead when applied to ultra-large-scale graph networks.

To assess the competitiveness of our proposed algorithm, we conducted experiments on the Book-Crossings dataset. The results indicate that our method consistently outperforms both neural network – based and graph-based approaches in terms of trustworthiness (Accuracy and Coverage), robustness, and scalability, thereby demonstrating superior practical potential. As shown in the Fig. 9 below, in the experiments conducted on the entire Book-Crossings dataset, the proposed algorithm demonstrates superior performance in both the precision, as reflected by the proportion of recommended books that align with users' actual preferences, and the coverage, indicated by the proportion of recommended books within the entire dataset.

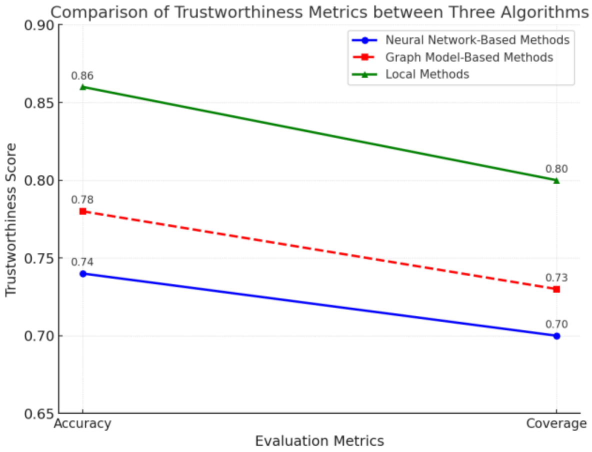

Fig. 9. Our Methods and Neural Network Methods and Graph Model-Based Methods

This highlights the algorithm's effectiveness in enhancing both recommendation quality and system scalability.

4. Concluding remarks

Traditional recommendation systems face challenges including data sparsity, cold start problems, and vulnerability to malicious attacks. To address these issues, a trust-based recommendation algorithm leveraging reliability measurement within social networks is proposed. First, a credibility function is defined to mitigate bias from malicious evaluations. A trust metric is then calculated, incorporating user interaction history and credibility assessments. Second, user utility function is defined to model feedback of recommendation quality for dynamically updating trust in a rational way.

This approach currently focuses solely on the structural aspects of user trust networks. Future research will explore integrating additional contextual information, such as user location, comments, and demographics, to enhance recommendation accuracy and performance.

Acknowledgments

This research was supported by Zhongyuan University of Technology, Horizontal Research Project, Energy Efficiency Simulation System Platform for Hydrogen-rich Clean Fuel Production Process, Project No. 22002349.

REFERENCES

[1] *Abdi, M., Okeyo, G. and Mwangi, R.*, 2025. Improved Collaborative Filtering Recommender System Based on Hybrid Similarity Measures. International Arab Journal of Information Technology (IAJIT), 22(1).

[2] *Kurniawan, R. and Wijaya, Y.A.*, 2025. Association Analysis of Printing and Photocopying Sales Data in Adzmi Art Shop Cirebon Uses the FP-Growth Algorithm. Journal of Artificial Intelligence and Engineering Applications (JAIEA), 4(2), pp.1119-1124.

[3] *Foucart, A., Elskens, A. and Decaestecker12, C.*, 2025, April. Ranking the scores of algorithms with confidence. In ESANN 2025.

[4] *Shrivastava, V., Kumar, A. and Liang, P.*, 2025. Language Models Prefer What They Know: Relative Confidence Estimation via Confidence Preferences. arXiv preprint arXiv:2502.01126.

[5] *Oren Barkan, Veronika Bogina, Liya Gurevitch, Yuval Asher, Noam Koenigstein*, A Counterfactual Framework for Learning and Evaluating Explanations for Recommender Systems,2024(5), WWW '24: Proceedings of the ACM Web Conference 2024, Pages 3723 - 3733.

[6] *Chen, C., Zhang, M., Liu, Y., & Ma, S.* (2022). Robust recommendation with adversarial training for combating malicious attacks. Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, 1234-1244.

[7] *Michael D. Ekstrand, Ben Carterette, Fernando Diaz*. Distributionally-Informed Recommender System Evaluation, 2024(3), ACM Transactions on Recommender Systems, Volume 2, Issue 1 Article No.: 6, Pages 1 - 27.

[8] *Pratama, Y. C. A., & Dewi, C.* (2025). Analysis of Consumer Purchasing Patterns Using the Apriori Algorithm on Sales Transaction Data from Anak Panah Kopi Salatiga. International Journal Software Engineering and Computer Science (IJSECS), 5(1), 1–10.

[9] *Shigan Yu, Fujun Ren*. (2023). IUCFAMR: an improved movie recommendation algorithm. Series C, Vol. 85, Iss. 4, 2023.

[10] *Shijie Zhang; Wei Yuan; Hongzhi Yin*, Comprehensive Privacy Analysis on Federated Recommender System against Attribute Inference Attacks, IEEE,2023,6:1-13

[11] *Rand Jawad Kadhim Almahmood, Adem Tekerek*,Issues and Solutions in Deep Learning-Enabled Recommendation Systems within the E-Commerce Field, Applied Sciences,2022,12(21)

[12] *Gao, M., Wu, J., & Li, X.* (2022). A dynamic trust-aware recommendation model for sparse data in social networks. Information Processing & Management, 59(1), 102768.

[13] *Li, X., Chen, H., & Zhang, W.* (2023). A hybrid trust-aware recommendation framework for sparse data and malicious attack scenarios. Expert Systems with Applications, 213, 119234.

[14] *Kumar, V.V., Raj, T.S.R., Raj, J.R.F., Sornavalli, M., Krishnan, R.S. and Soundiraraj, N.,* 2025, January. A Hybrid Machine Learning Approach for Top-N Bike Recommendations. In 2025 International Conference on Multi-Agent Systems for Collaborative Intelligence (ICMSCI) (pp. 1380-1387). IEEE

[15] *Baheri, A.*, 2025. Multilevel Constrained Bandits: A Hierarchical Upper Confidence Bound Approach with Safety Guarantees. Mathematics, 13(1), p.149.

[16] *Veronica Opranescu, Anca Daniela Ionita*. Towards a recommendation system for an educational profile in systems engineering. Series C, Vol. 86, Iss. 1, 2024.

[17] *Wenhao Wu, Rong Liu* et al. Research on personalized recommendation algorithm based on trust relationship. Series C, Vol. 83, Iss. 4, 2021