Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Meta-Level Based Recommender System Using Knowledge Graph-based Neural Collaborative Filtering

Erfan Ainul Yakin
Universitas Negeri Malang
Triyanna Widiyaningtyas
triyannaw.ft@um.ac.id
Universitas Negeri Malang
Hary Suswanto
Universitas Negeri Malang
Didik Dwi Prasetya
Universitas Negeri Malang
Wahyu Caesarendra
Curtin University Malaysia

Research Article

Keywords: Hybrid Recommendation System, Meta-Level, Knowledge Graph, Neural Collaborative Filtering, Deep Learning

Posted Date: July 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10102523/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Discover Artificial Intelligence on September 28th, 2026. See the published version at https://doi.org/10.1007/s44163-026-02307-8.

Meta-Level Based Recommender System Using Knowledge Graph-based Neural Collaborative Filtering

Erfan Ainul Yakin[1], Triyanna Widiyaningtyas[1*], Hary Suswanto[1], Didik Dwi Prasetya[1], Wahyu Caesarendra[2]

[1]*Department of Electrical Engineering and Informatics, Universitas Negeri Malang, Malang 65145, Indonesia*
[2]*Department of Mechanical and Mechatronics Engineering, Curtin University Malaysia, Miri, Malaysia*

Abstract

Recommendation systems are now very popular, with the approach of collaborative filtering or content-based filtering having a weakness in cold-start, so a hybrid approach has been developed. This study proposes a hybrid approach with the Meta-level method, which we call Meta KG-NCF, that integrates Knowledge Graph (KG) and Neural Collaborative Filtering (NCF). The Meta-Learning with FOMAML (First-order Model-agnostic Meta-learning) can handle cold-start problems but is not yet effective in terms of accuracy. The development from FOMAML to FO-MSAN (First-Order Meta-Learning and Multi-Supervisor Association Network) improved the results of the recommendations but has not yet improved computational efficiency, so within a meta-level and FOMAML learning framework to enable efficient handling of cold-starts with high accuracy and more efficiently. Mapping the entity relationship between item features and profiles using the Translation Embedding (TransE) to make graph analysis easier is a Knowledge Graph Technique. From KG, the semantic vector becomes input to process in a deep learning algorithm with the Collaborative Filtering approach, a neural network with ReLU activation, which we call Neural Collaborative Filtering (NCF). The KG Semantic embedding with TransE Process, as a basic method, and NCF, as a reinforcement method, are combined with the Meta-level Technique. The experiment was conducted using three large-scale datasets Goodreads, AmazonBook, and Book-Crossing. The scenarios that included accuracy, robustness, and cold-start testing. Results from the Meta KG-NCF compared to the baseline method, using MAE, RMSE, NDCG@5, Precision@5, MRR@5, and execution time. The Meta KG-NCF showed a contribution of significantly better accuracy and efficiency with on-par ranking quality, rather than uniform dominance.

Keywords: Hybrid Recommendation System, Meta-Level, Knowledge Graph, Neural Collaborative Filtering, Deep Learning.

*Corresponding author.
Email addresses: triyannaw.ft@um.ac.id (Triyanna Widiyaningtyas)

1. Introduction

The current recommendation system is often found in online digital services, such as movies, music, online buying and selling, and digital books [1], [2], [3]. The recommendation system for digital books poses a new challenge because the digital book domain has fewer transactions than other domains [4], [5]. Digital books are stored in pdf or epub format standards and other formats so that the digital book can be searched and locked to protect copyright and given other supporting data that can be used to implement a recommendation system to make it easier to search and provide results that are in accordance with the preferences of the user [6], [7], [8], [9].

The recommendation system approach consists of collaborative filtering, content-based filtering, and hybrid approaches [10], [11]. Collaborative filtering assesses the preferences and behavior of one user with another. The collaborative filtering approach comprises 2 methods: model-based and memory-based. One model-based method applied to recommendation systems is deep learning. One of the deep learning algorithms applied is a neural network, which aims to analyse with no relationships between variables. Meanwhile, memory-based methods are applied to more complete data, so the depth of learning does not require high complexity [12], [13], [14].

Research by Robo et al. [15] discussing the Collaborative Filtering approach with the Memory-Based model applied to the MovieLens dataset using the Similarity and SVD algorithms showed good recommendation results because in the dataset, the correlation of users with ratings showed complete data [14], [15]. Research by Morise et al. [16] discusses Collaborative Filtering by adopting a neural network to process user and product ratings that are not interconnected, using complex data structures. The computational process of a Neural Network applies the vector value of a layer connected to a regression equation using the ReLU activation function.

The content-based filtering approach analyses the domain being researched and past user preferences, including user profiles and product attributes [17]. Research by Shen et al. [18] develops a recommendation system based on a knowledge graph to improve accuracy on irregular datasets. In the Recommendation System with a Content-Based Filtering approach, the Knowledge Graph method maps the entity relationships between item features and user profiles using historical data. The discussion of the Knowledge Graph researched by Liu Y et al. [19] which applies Multi-Task Learning (MMSS-MKR) to a music recommendation system to improve accuracy in the face of dataset sparsity. The feature selection process in this study filters variables in the dataset, such as songs, artists, and genres, to recommend similarities in content that are relevant to users. Research Wang, Hui et al.[20] Integrate Knowledge-Based into the recommendation system using a Content-Based Filtering approach, with a Graph Attention Network and Collaborative Propagation Techniques, and

perform variable analysis across users to increase accuracy. The data analysis process in content-based filtering uses similarity references with TransH for relationship representations in compatible knowledge-based graph models, also known as knowledge graphs.

The collaborative or content-based filtering approach applied in each recommendation system makes the system suboptimal. The collaborative filtering approach has weaknesses in handling cold-start. At the same time, content-based is limited to user profile variables and item attributes, so that if transaction interactions in the book domain are small, the accuracy results will be low [21], [22], [23], [24], [25]. The technology to overcome the cold-start and accuracy constraints is to combine collaborative and content-based filtering, often referred to as hybrid filtering. Meta-level is a hybrid model approach suitable for addressing accuracy problems caused by cold-start, and it offers representative processing depth and flexibility [1], [10], [26], [27], [28].

The Meta Learning method in the previous study by [29] with the TaNP task-adaptive neural process method by combining multiple learning strategies in the task-adaptive mechanism. TaNP is able to overcome the cold-start problem by using neural processes in the task-adaptive mechanism. The limitations of this model on sparse data will greatly affect the results, and the sensitivity of the parameters needs to be further optimized so that the results remain consistent.

Other Meta Learning research by [30] with the PAML Preference Adaptive Meta-Learning approach method to overcome cold-start problems with generalization techniques about social relationships from users and specific adaptations. PAML techniques are able to improve RMSE results, and the application of social relationships from users is effective in overcoming cold-start constraints. The disadvantages of this method are that the complexity of computing is very high and the data presented must provide strong social media information.

Analysis [31] with a Meta Learning method with the MPML multi-prior meta-learning method is able to handle cold-start with accuracy-based task clustering techniques on each prior. So that the use of multi-prior is effective in handling cold-start as well as increasing accuracy values and MAE, RMSE values. The weakness of this method is that the computational complexity is very high and training is very dependent on the prior values of the dataset.

Meta-learning research by [32] With MetaEDL, a meta-evidential learning model with optimization and evidential learning on datasets with great scalability, is able to handle cold-start well. This method is very flexible with datasets where user and item interaction is very limited, with good RMSE results. However, the limitations of this method lie in the high complexity of the calculation and require good optimization of sample data retrieval.

Meta-learning research with a FOMAML is discussed by Liu, Xiaoyang et al. [1], in the recommendation system for cold-start handling using a method called First-Order Meta-Learning and Multi-Supervisor Association Network (FO-MSAN). The Meta Learning technology of this method uses the multi-supervisor network and the association network. The advantages of this study are effective in cold-start handling and optimal performance for sparse data, but limitations in this study include the complexity of the calculation, which is still high, and the accuracy is still low.

The results of the summary of the hybrid filtering approach with the book dataset, as well as the advantages and limitations of the model and the application of evaluation to system testing, are shown in Table 1.

Table 1. Summary of the hybrid filtering approach with the book dataset

Algorithm	Advantages	Limitation	Evaluation
TaNP	▪ Troubleshoot overfitting issues ▪ Overcome cold-start issues	▪ Sensitive to data spasms ▪ Parameters have not been tested in detail	P@5, NDCG@5, dan MAP@5
PAML	▪ Adaptive preference is effective in cold-start ▪ Improves RMSE yields	▪ The dataset must contain social media information ▪ High computing complexity	RMSE, NDCG@5
MPML	▪ Multi prior effectively wins cold-start ▪ Increased MAE and RMSE yields	▪ High computing complexity ▪ Highly dependent on the prior value of the dataset	MAE, RMSE, NDCG@5
metaEDL	▪ Cold-start handler with evidential learning is excellent ▪ Improves RMSE yields	▪ High complexity ▪ Very high sample optimization	RMSE, NDCG@5
FO-MSAN	▪ Hybrid learning is effective in cold-start handling ▪ Optimal for sparsity data	▪ Calculation complexity is still high ▪ Accuracy is still low	MAE, NDCG, MRR, Precision

Results of research Liu, Xiaoyang et al. [1], with a Meta Learning approach that uses FOMAML and developed into FO-MSAN to overcome cold start in the BookCrossing dataset. The result from FO-MSAN in MAE is relatively low for the recommendation system's accuracy. The use of Neural Network-based methods with hidden-layer calculations will require high computational complexity, so in the research of Liu, Xiaoyang et al. [1]. This makes it difficult to apply the recommended system to obtain efficient values. Given the sparsity of conditions

in Liu's study, Xiaoyang et al. [1] is a problem that needs to be solved to achieve a recommendation system that scores better and more efficiently.

The application of a FOMAML approach to cold-start problems that have been solved in baseline research found that the lack of low accuracy and complexity of the process is too high, so this study proposes a new hybrid approach with meta-level model techniques using neural collaborative filtering (NCF) based on a knowledge graph called "Meta KG-NCF" and Meta Learning with FOMAML. From the hybrid studies that have been carried out in handling cold start, it is very effective in using hybrid with a meta-level approach. The hybrid process with a meta-level model approach is by determining the basic or first method or algorithm and the second algorithm.

The selection of the first method using the knowledge graph aims to organize in a structured manner the information to be processed with TransE so that it can produce a recommendation according to the preferences of the user [27], [18], [28], [33]. In the study Liu, Yufang et al. [27] Knowledge Graph Embedding is a technique that shows the distribution of data and its relationship with vectors under fundamental conditions. Research by Yuntao Du et al. [34] as well as research by P. M. T. Do et al. [26] It strongly supports the application of Knowledge Graph in increasing the value of MAE and RMSE. A supporting solution to solve the limitations of hybrid meta level at baseline with the use of the first algorithm, namely Knowledge Graph. This method utilizes graphic technology from the results of data processing so that it is able to show the relationship of entity relationships that store detailed information with TransE.

The next stage of the meta-level model requires a 2nd algorithm that becomes a deeper processing neural collaborative filtering (NCF) referring to the use of ReLU activation was also applied in the research of Wang, Huan et al. [28] to address accuracy issues caused by cold-start in the implementation of a hybrid recommendation system with a hybrid approach called Meta-Learning Adaptation Network (MLAN) and supported by the research of Shokrzadeh et al. [35] states that Neural Collaborative Filtering is able to increase the accuracy value so that it can be adopted as a first candidate in the meta-level process of the model because it is able to show the complex relationships between users and items [35] [26], [36], [37]. So that the solution to overcome the limitations of the baseline is the Neural Collaborative Filtering Method as the second method in the application of the Meta-Level model. This method offers the ability to capture more complex interactions between users and items in the dataset with rare transaction conditions. The use of Multi-Layer Perceptron (MLP) consisting of several layers is able to show non-linear relationships. So that the efficiency value of the recommendation system can be achieved.

Building on the limitations identified above, this work makes the following contributions:

1. Meta-level architecture that decouples semantic and collaborative signals. We propose Meta KG-NCF, which represents each item through two explicitly separated channels: a knowledge-graph channel that encodes content semantics (author and publisher relations) and a trainable neural collaborative-filtering channel that captures user–item interaction patterns. Unlike prior hybrid recommenders that fuse these signals through similarity heuristics, our design keeps the semantic prior fixed during meta-training so that the adaptation process operates only on the collaborative component, isolating where personalization occurs.
2. Formally constructed knowledge graph with TransE-based priors. We construct a knowledge graph over books, authors, and publishers using two typed relations (written_by, published_by) and learn entity embeddings with TransE under a margin-ranking objective with negative sampling. These pretrained embeddings serve as a frozen semantic prior injected into the recommender, providing item representations that remain informative even when interaction history is absent — the core difficulty in cold-start.
3. First-order meta-learning procedure (FOMAML) integrated with the Meta KG-NCF. We cast cold-start recommendation as a few-shot adaptation problem in which each user defines a task with a support and query set. A first-order MAML inner loop adapts the trainable parameters on the support set while the KG prior is excluded from adaptation, and an outer loop meta-optimizes the shared initialization across tasks. This makes explicit where meta-learning operates, addressing the ambiguity of generic "meta-level" hybrid designs.
4. Controlled, reproducible evaluation protocol. All five baselines (TaNP, PAML, MPML, MetaEDL, FO-MSAN) are re-implemented from scratch following the architectures and hyperparameters reported in their original papers, and are trained and evaluated on identical data splits, pre-processing, and hardware. We report a new-item cold-start protocol with explicit support/query construction and negative sampling, an ablation isolating the contribution of the KG prior and the meta-learning component, and statistical significance testing across repeated runs.
5. Honest characterization of where the method does and does not improve. Across three large-scale book datasets (Goodreads, AmazonBook, Book-Crossing), Meta KG-NCF improves rating-prediction accuracy (MAE, RMSE) and top-N quality (NDCG@5, Precision@5) over the baselines in the majority of scenarios, while we explicitly report the settings notably MRR where FO-MSAN remains competitive or superior. We discuss these trade-offs rather than claiming uniform superiority.

2. Related works

Hybrid Recommendation System in the study of Afoudi et al. [38] Declaring is a technology applied to a system with a specific dataset domain that filters results to display those appropriate to the user's context. In this study, the Hybrid approach was tested in the movie domain using the Collaborative Filtering method combined with the Neural Network method, and the Content-Based Filtering method with the TF-IDF method, and the results showed low error and faster execution time [38].

In the study Bansal et al. [39] conducted research on a Hybrid recommendation system tested on movie datasets of 100k and 1M, achieving an MSE of 1.477 and an accuracy of 85.82%. The combination of kNN and RNN makes the learning process very deep, requiring a high-performance system. The proposed method shows excellent results compared to the traditional recommendation system on short-term outcome prediction [39].

Hybrid recommendation system in the study by Remadnia et al. [40] It is very effective at handling cold starts and increases the precision of its recommended results. The architecture of the Hybrid recommendation system in this study illustrates the relationship between the Collaborative Filtering approach and the Content-Based Filtering approach, which uses book attributes and user profiles to produce book recommendations.

The classification of the recommendation system, in general, is shown in Fig. 1, which shows 3 approaches. In this study, the recommendation system will be studied using a hybrid approach, with the Meta-level model shown in Fig. 1, marked with a thick red line. Proposed hybrid models and approaches align with the results of the review of previous studies and are most effective at addressing accuracy and cold-start issues.

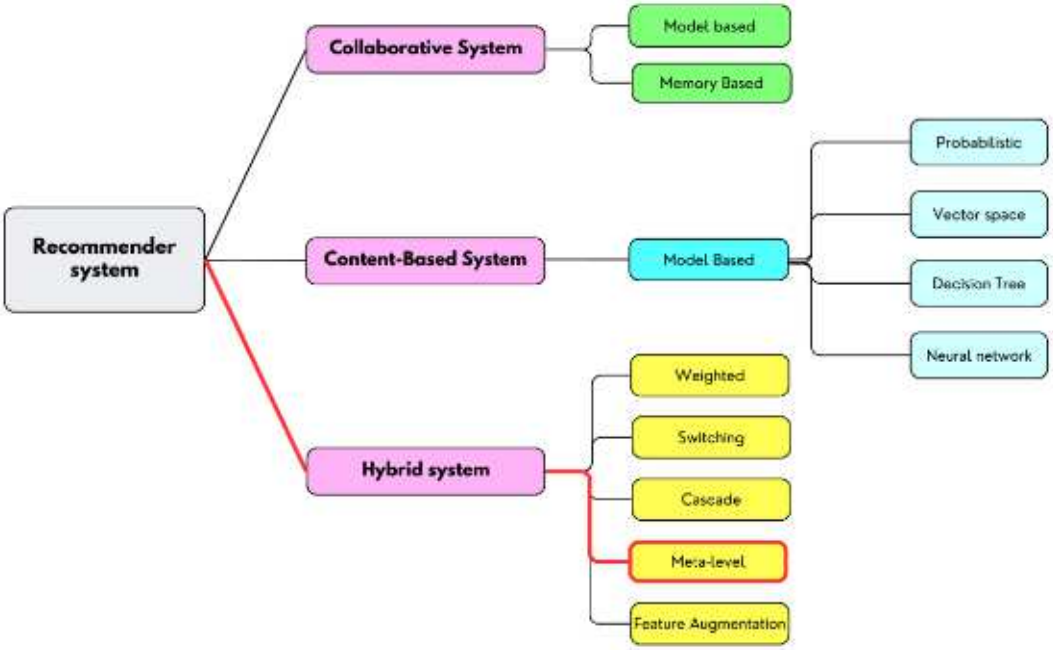


Fig. 1 Classification of the Recommendation System [10]

The implementation of Meta-level in a hybrid book recommendation system will be more effective in adapting to processing large data as well as the most flexible and effective in cold-start handling. The adaptation process of the Meta-level model shows that the results of the recommendation output are very relevant to the user's condition because the way this model works is in accordance with the data attribute conditions and the ability of basic methods to adapt to the new process [1], [10], [27], [41], [42]. A more detailed discussion of the Meta-level Model will be presented in the next sub-chapter.

2.1 Knowledge Graph

A knowledge graph (KG) is a structured-data representation that models real-world entities as nodes and labelled relationships between entities as edge [18]. Formally, a knowledge graph is defined as a set of triples G = (V, E) with vertex set V and edge set E, where each edge corresponds to a typed relation between two entities. This formal definition is important to distinguish a knowledge graph from a similarity matrix or a clustering result: although the latter encode item proximity, they do not capture typed relational structure between item-side entities and therefore do not constitute a knowledge graph in the sense used in this paper. A knowledge graph is naturally suited to recommender systems because it captures high-level semantic relations between items, users, and item attributes that are difficult to obtain from sparse user–item interactions alone [20]. By bringing inside information such as authors and publishers within an explicit relational structure, a knowledge graph provides context that helps the model represent items with few or no interactions, which is the core of the cold-start scenario [34].

In the book recommendation domain, the entities in the graph are books, authors, and publishers; relations describe how these entities connect, such as the relation that links a book to its author and the relation that links a book to its publisher. This is consistent with prior work showing that adding item-side knowledge through a graph structure improves recommendation quality when interaction data are sparse, because the semantic neighbourhood of an item (for example, other books by the same author or from the same publisher) can substitute for the missing collaborative signal [19].

To make the symbolic graph usable by a neural model, entities and relations must be mapped into a continuous vector space through a process called knowledge graph embedding [43]. The objective of knowledge graph embedding is to learn low-dimensional vector representations of entities and relations such that the relational structure of the original graph is preserved. Among the available embedding techniques, the translation-based model TransE is one of the most widely used because of its simplicity, scalability, and strong empirical performance [44].

The simplest KG embedding method is TransE (Translating Embeddings). TransE (Translating Embeddings) represents each relation as a translation vector in the embedding space, so that for a valid triple the head plus the relation should approximate the tail [45], expressed as $\mathrm{h} + \mathrm{r} \approx \mathrm{t}$. The plausibility of a triple is measured by a distance-based score, given by Eq. 1.

$$f(h,r,t) = \| \mathrm{h} + \mathrm{r} - \mathrm{t} \|_2 \quad (1)$$

where a smaller distance indicates a more plausible triple. The model is trained with a margin-based ranking loss that drives valid (positive) triples to score lower than corrupted (negative) triples obtained by randomly replacing the head or the tail. The resulting entity embeddings encode the semantic relationships of the knowledge graph in a form that can be directly fed into a neural recommender at the next stage. In the proposed method, these trained TransE embeddings serve as a fixed semantic prior for items, authors, and publishers, providing meaningful item representations even when interaction history is absent. The detailed construction of the graph triples and the training of TransE used in this study are described in Section 3.3.

2.2 Neural Collaborative Filtering

The first basic method in the recommendation system training process, using a Hybrid approach, is Neural Collaborative Filtering (NCF). NCF in the literature uses the Multilayer Perceptron (MLP) technique, which can capture the relationship between users and items under limited transaction data [19], [21], [27], [36], [38]. In the data book, the condition of the data transactions is not as good as in other datasets, so the accuracy of the recommendation system is relatively low, and cold-start problems are more frequent.

Research by Ibrahim et al. [36] discusses NCF modelling using non-linear interactions between users and items to calculate their correlation weight. NCF models this preference by using neural networks to capture more complex interactions. The calculation stage begins by combining two types of embeddings: user and item embeddings. Then, the functional equation is given by Eq. 2.

$$\phi NCF^{(u,i)} = MLP([p_u; p_i]) \quad (2)$$

p_u embed user p_i embed item.

A Multilayer Perceptron (MLP) is a neural network architecture consisting of several layers used to model non-linear relationships. In the Neural Collaborative Filtering (NCF) method, MLP is used to process the combined input from the user and item embeddings. MLP processes enable the capture of more complex interactions between users and items. The hidden layer in an MLP computes the output based on the input from the previous layer, using the learned weights and biases [46]. Each layer is followed by a non-linear activation function, as in Eq. 3.

$$MLP_{hl} = \sigma(\sum_{i=1}^{n} x_i W_i + b) \quad (3)$$

MLP_{hl} is the MLP output from the layer, W_1 is the weight matrix, b_l is a vector of bias and σ is a non-linear activation function.

The next step is to apply the ReLU activation function to the hidden-layer output already computed. ReLU was chosen as the activation function in NCF for its simplicity and its ability to overcome overly small gradients in backpropagation, thereby enabling the weights to guide training better [47]. So, ReLU has been proven to provide excellent results in neural network models. In the first basic method, his role can increase the values of MAE, RMSE, and accuracy and can handle cold-start.

2.3 Meta-Learning FOMAML

Meta-learning, often described as "learning to learn," aims to obtain a model initialization that can be adapted to a new task using only a small amount of data, rather than training a separate model for every task from scratch [29]. This property makes meta-learning particularly suitable for cold-start recommendation, where each new user or item provides only a few interactions. In this paradigm, each user is treated as a task $\mathcal{T}_i$ consisting of a support set $\mathcal{S}_i$, used for fast adaptation, and a query set $\mathcal{Q}_i$, used to evaluate how well the adapted model generalizes [30].

Model-Agnostic Meta-Learning (MAML) is a widely adopted meta-learning algorithm that learns a shared parameter initialization θFrom which a few gradient steps on a task's support set yield strong task-specific performance. For each task $\mathcal{T}_i$, the model first performs an inner-loop adaptation that produces task-adapted parameters $\theta_i'$from the support loss, as shown in Eq. 4.

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{S}_i}(f_\theta) \quad (4)$$

where $\theta$is the shared (meta) initialization, $\alpha$is the inner-loop learning rate, $\mathcal{L}_{\mathcal{S}_i}$ is the loss computed on the support set of the task $\mathcal{T}_i$, and f_θ is the model parameterized by θ. The shared initialization is then updated in an outer loop by minimizing the query-set loss of the adapted parameters, aggregated over a batch of tasks, as shown in Eq. 5.

$$\theta \leftarrow \theta - \beta \sum_{\mathcal{T}_i} \nabla_\theta \mathcal{L}_{\mathcal{Q}_i}\left(f_{\theta_i'}\right) \quad (5)$$

where $\beta$is the outer-loop (meta) learning rate and $\mathcal{L}_{\mathcal{Q}_i}$is the loss on the query set of the task $\mathcal{T}_i$. The standard MAML outer update requires differentiating through the inner-loop adaptation, which involves computing second-order derivatives and is therefore computationally expensive [48].

First-Order MAML (FOMAML) addresses this cost by ignoring the second-order terms and approximating the meta-gradient with first-order gradients evaluated at the adapted parameters θ_i', as shown in Eq. 6.

$$\theta \leftarrow \theta - \beta \sum_{\mathcal{T}_i} \nabla_{\theta_i'} \mathcal{L}_{\mathcal{Q}_i} \left(f_{\theta_i'} \right) \quad (6)$$

By avoiding the computation of second-order derivatives, FOMAML substantially reduces training cost while retaining most of the adaptation capability of full MAML, which is the main reason it is favoured in large-scale and resource-constrained recommendation settings [42].

A representative FOMAML-based recommender used as the main baseline in this study is the First-Order Meta-Learning and Multi-Supervisor Association Network (FO-MSAN)[1]. FO-MSAN employs a multi-supervisor mechanism as the base learner and couples it with an Association Network, updating the shared initialization through the first-order meta-update in Eq. 6. Although FO-MSAN improves recommendation quality under cold-start conditions, its computational complexity remains relatively high, and it does not explicitly exploit the semantic relationships among items. These limitations motivate the proposed method, which injects a knowledge-graph semantic prior into a FOMAML-based meta-level framework, as detailed in Section 3.

2.4 Hybrid Meta-Level

Meta-level is a hybridization strategy in which the representation learned by one base method is used as the input feature for another base method, rather than combining their predictions in parallel [17]. Unlike weighted or switching hybrids, the base methods in a meta-level design operate sequentially, so that the knowledge captured by the first method is embedded into the feature representation consumed by the second method, allowing the second method to refine rather than merely aggregate the result.

In the proposed framework, the meta-level strategy assigns two complementary roles. The Knowledge Graph acts as the first base method that produces a semantic representation of each item and its attributes through pretrained TransE embeddings, encoding relational and content information that remains informative under sparse interactions. The Neural Collaborative Filtering component then acts as the second base method that consumes this semantic representation, together with the collaborative user–item embeddings, to produce the final rating prediction [35]. Formally, for a user $u$and item $i$with author $a_i$and publisher p_i, the meta-level input representation $\mathbf{z}_{ui}$is constructed by concatenating the collaborative embeddings with the knowledge-graph semantic prior, as shown in Eq. 7.

$$\mathrm{z}_{ui} = [\, \mathrm{e}_u^u \parallel \mathrm{e}_i^i \parallel \mathrm{g}_i^i \parallel \mathrm{g}_{a_i}^a \parallel \mathrm{g}_{p_i}^p \,], \quad (7)$$

where e_u^u and e_i^i are the trainable collaborative (NCF) embeddings of the user and item, g_i^i, $\mathrm{g}_{a_i}^a$, and $\mathrm{g}_{p_i}^p$ are the frozen TransE embeddings of the item, its author, and its publisher, and $\parallel$denotes vector concatenation. The fused representation z_{ui} is then passed to the NCF predictor f_θ to produce the predicted rating, as shown in Eq. 8.

$$\hat{r}_{ui} = f_\theta\,(\mathrm{z}_{ui}), \quad (8)$$

where $\hat{r}_{ui}$ is the predicted rating and θ denotes the trainable parameters of the collaborative predictor. Because the knowledge-graph embeddings are learned beforehand and kept fixed, they serve as a stable semantic prior, so the second method does not need to rediscover item relationships from scarce interaction data—the key advantage of the meta-level arrangement under cold-start conditions.

This sequential organization makes the meta-level approach both flexible and efficient, since each base method is optimized for the aspect of the problem it handles semantic enrichment from the knowledge graph in the first stage, and collaborative preference modelling through NCF in the second. When this meta-level composition is trained within the FOMAML procedure described in Section 2.3, only the collaborative parameters θ are adapted in the inner loop while the semantic prior remains fixed, allowing the model to retain fast few-shot adaptation while benefiting from knowledge-graph semantics. The detailed realization of this composition is presented in Section 3.

3. Data and Methods

3.1 Data

Materials are the data that will be processed in the recommendation system used in the research. The material in this study uses Goodreads 100k, AmazonBook and Bookcrossing which are secondary data obtained from “https://www.kaggle.com/”. This data is detailed book data, complete with ratings and the number of reviews from users who have completed the cleaning stage, making them ready for use [49].

The Goodreads Book dataset of 100k files is in .csv format, consisting of 2 files: data-books and user-rating. The “book.csv” data contains details of Id, Name, RatingDist1, pagesNumber, RatingDist4, RatingDistTotal, PublishMonth, PublishDay, Publisher, and CountsOfReview. While the “user-rating.csv” data contains ID, Name, and Rating. In this study, 2 files were used, with the file structure shown in Table 2 and examples of each file shown in Table 3, Table 4, Table 5.

Table 2. Structure of the Goodreads Book 100k dataset

File Name	Structure File
book1-100k.csv	Id | Name | RatingDist1 | pagesNumber | RatingDist4 | RatingDistTotal | PublishMonth | PublishDay | Publisher | CountsOfReview | PublishYear | Language | Authors | Rating | RatingDist2 | RatingDist5 | ISBN | RatingDist3
user_rating_0_to_1000.csv	ID | Name | Rating

Table 3. Example book1-100k.csv file

Id	Name	…	RatingDistTotal	….	CountsofReview
1	Harry Potter and the Half-Blood Prince (Harry Potter, #6)	…	total:2298124	…	28062
2	Harry Potter and the Order of the Phoenix (Harry Potter, #5)	…	total:2358637	…	29770
… .	…	…	…	…	…
99998	eBay: Top 100 Simplified Tips & Tricks	…	total:23	…	1
99999	ebay Timesaving Techniques for Dummies	…	total:18	…	3

Table 4 Example user_rating_0_to_1000.csv file

Id	Name	Rating
1	The Very Hungry Caterpillar	really liked it
1	Edge of Eternity	did not like it
…	…	…
56921	All the King's Men	it was ok
77121	Phaedrus and Letters VII and VIII	really liked it

The dataset used has a similar structure: User, Item, and Rating. A slight difference is found in the Rating scale, which is either [1-5]. Details of the dataset specifications are shown in Table 5.

Table 5. Dataset Specification

Specification	**Goodreads 100K**	**Goodreads 1M**	**AmazonBook**	**Book-Crossing**
#Users	4,109	12,751	50,000 (strat.)	≈ 90K
#Items	171,520	1,850,310	≈ 220K	≈ 270K
#Ratings	125,564	362,596	≈ 3M (full)	≈ 1.1M
Sparsity	≈ 99%	≈ 99%	≈ 99.97%	≈ 99.99%
Rating range	1–5	1–5	1–5	0–10 → 1–5

In Table 5, AmazonBook is very large (≈ 3M interactions in raw form). Following common practice in meta-learning recommendation research [29], we apply stratified user sampling (n=50,000) for the AmazonBook experiments to keep computation feasible while preserving the long-tail distribution of user activity. Book-Crossing and Goodreads are used in full. The three datasets share the same triplet structure (user, item, rating) with item-side attributes (author, publisher) that allow the knowledge graph to be constructed. Their differing scales and sparsity support the robustness evaluation.

The use of this dataset is intended for cold-start testing with large dataset conditions where the exploration of user and item embedding can be optimized despite limited interaction [34]. Cold-start testing scenarios with limited user-item interactions are a practical example of

the challenges faced by many real-world data-driven recommendation systems [26]. Exploration of meta-level models that combine a content-based and collaborative approach with book datasets allows for the testing of recommendation models on objects that have many attribute features, as well as limited but relevant data, such as book ratings that are more difficult to analyse [1], [27], [50].

3.2 Methods

In this study, the proposed hybrid recommendation system is called Meta KG-NCF. Meta KG-NCF builds on the FO-MSAN (First-Order Meta-Learning and Multi-Supervisor Association Network) method developed by Liu, Xiaoyang et al. [1] by replacing the implicit content prior of FO-MSAN with an explicit knowledge-graph prior, and by training the joint architecture with FOMAML rather than with the multi-supervisor mechanism. Meta KG-NCF consists of five blocks: input, pre-processing, meta-level, FOMAML, and output. The overall architecture is shown in Fig. 2.

In Fig. 2, the input block supplies the book interaction dataset, consisting of user ratings together with item-side attributes (book, author, and publisher). The pre-processing block cleans the data, encodes users, items, authors, and publishers into integer indices, and organizes the interactions into meta-learning tasks, where each user forms one task with a support set and a query set. The meta-level block is the core of the architecture and operates in two sequential stages, reflecting the meta-level hybridization strategy described in Section 2.4. In the first stage, the Knowledge Graph is constructed from the item-side attributes by forming triples that link each book to its author and publisher; these triples are then embedded using TransE to produce a semantic representation of items, authors, and publishers.

This knowledge-graph embedding is the first base method and acts as a fixed semantic prior. In the second stage, the Neural Collaborative Filtering (NCF) component takes this semantic prior together with the trainable collaborative embeddings of users and items, concatenates them into a single feature vector, and passes the result through a multilayer perceptron (MLP) with ReLU activation to predict the rating, followed by the application of ReLU activation to minimize the error [34]. NCF is therefore the second base method, refining the semantically enriched representation into a final preference score.

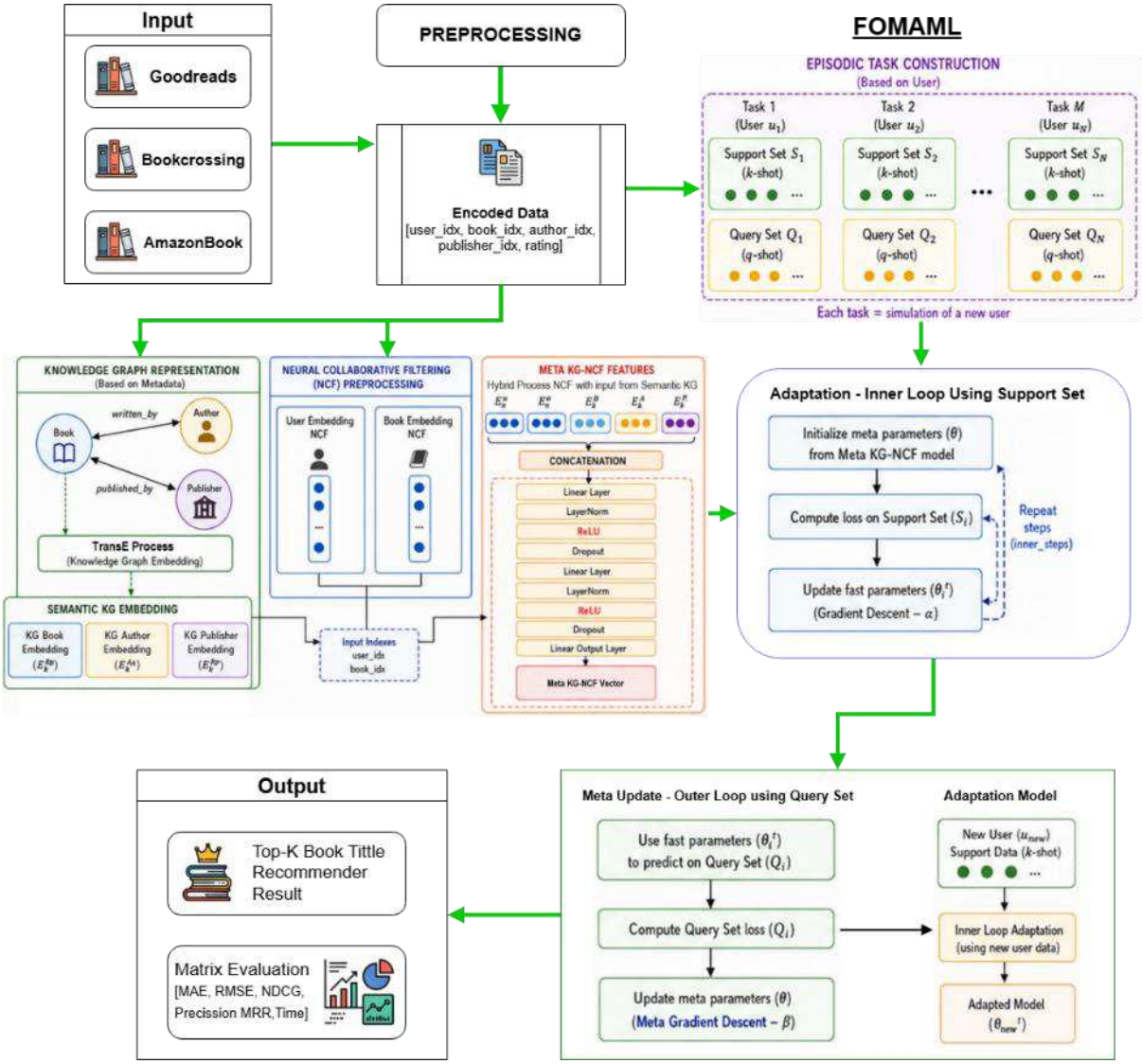


Fig. 2 The developed Meta KG-NCF

In Fig. 2, the FOMAML block governs how the meta-level model is trained. Rather than training a single fixed model, the framework learns a shared parameter initialization that can be quickly adapted to each user. For every task, an inner loop adapts the trainable collaborative parameters on the user's support set, while the knowledge-graph prior remains frozen; an outer loop then updates the shared initialization using the first-order meta-gradient computed on the query set, following the FOMAML procedure in Section 2.3. This is what enables the model to generalize to new users and new items from only a few interactions, directly addressing the cold-start problem.

Finally, the output block produces the top-5 recommended book titles for each user. Model performance is assessed in the evaluation stage using MAE, RMSE, NDCG@5, accuracy, and execution time. A more detailed description of the NCF component, the knowledge-graph construction with TransE, and the hybrid Meta KG-NCF training procedure is provided in the following subsections.

3.3 Knowledge Graph Semantic Embedding with TranE

In the proposed Meta KG-NCF, the Knowledge Graph (KG) is the first base method and is responsible for producing a semantic representation of each item before any collaborative learning takes place. Unlike approaches that derive item similarity purely from rating vectors, the KG here is constructed from explicit item-side attributes, so that semantic relationships remain available even for items with no rating history, the central requirement for cold-start.

Graph construction. From the item attributes (author and publisher), the graph is built as a set of typed triples $\mathcal{G} = \{(h, r, t)\}$ over three entity types: books, authors, and publishers, and two relation types. For each book $i$with author $a_i$and publisher p_i, two triples are generated Eq. 9.

$$(i, \text{written_by}, a_i) \text{and } (i, \text{published_by}, p_i) \tag{9}$$

Embedding with TransE. Each entity and relation is mapped to a d-dimensional vector ($d = 64$). TransE models a relation as a translation, so for a valid triple the head plus relation should approach the tail. The plausibility of a triple is scored by the L2 distance of the L2-normalized entity vectors, as in Eq. 10.

$$f(h, r, t) = \| \tilde{\mathrm{h}} + \mathrm{r} - \tilde{\mathrm{t}} \|_2, \tilde{\mathrm{h}} = \frac{\mathrm{h}}{\|\mathrm{h}\|_2}, \tilde{\mathrm{t}} = \frac{\mathrm{t}}{\|\mathrm{t}\|_2} \tag{10}$$

where a smaller score indicates a more plausible triple. The model is trained with a margin-based ranking loss over positive triples and corrupted negatives, as in Eq. 11.

$$\mathcal{L}_{KG} = \sum_{(h,r,t)\in\mathcal{G}} \frac{1}{n} \sum_{j=1}^{n} [\, \gamma + f(h, r, t) - f(\, h', r, t'_j)\,]_+ \tag{11}$$

where $(h', r, t'_j)$is the j-th negative triple obtained by randomly corrupting the head or tail, $n$is the number of negatives per positive ($n = 5$), $\gamma$is the margin ($\gamma = 1.0$), and $[x]_+ = \max(0, x)$. After training, the learned entity embeddings for books, authors, and publishers are extracted and stored as a fixed semantic prior are not updated during the subsequent collaborative learning stage. An overview of the KG Process is shown in Fig. 3.

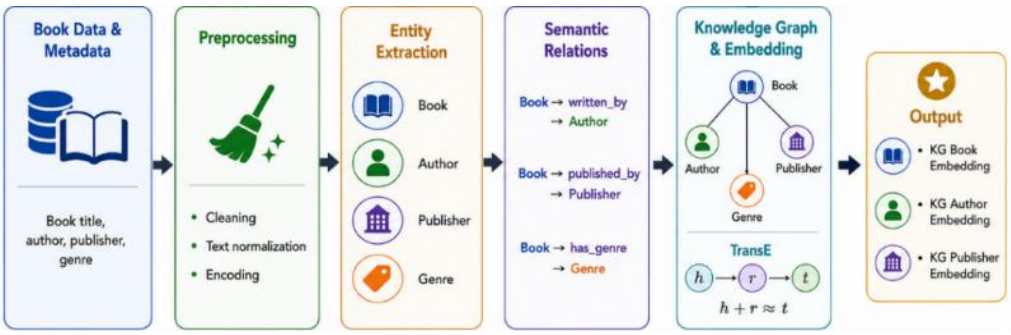


Fig. 3 Process of KG Symantic

The KG construction and TransE training routine are summarized in Algorithm 1.

Algorithm 1 KG (Knowledge Graph)

Input: interactions D with fields (book i, author a_i, publisher p_i)
embed_dim d = 64, epochs = 100, lr = 0.001, margin γ = 1.0,
negatives per triple n = 5
Output: frozen embeddings E_book, E_author, E_publisher

```
T ← {}
for each unique (i, a_i, p_i) in D:
    T ← T ∪ {(i, written_by, a_i), (i, published_by, p_i)}
initialize entity/relation embeddings (uniform); normalize relations
for epoch = 1 … 100:
    for each minibatch B ⊂ T:
        for each (h,r,t) in B:
            generate n negatives (h',r,t') by corrupting head or tail
        L ← Σ [ γ + f(h,r,t) − mean_j f(h',r,t'_j) ]_+
        update embeddings by Adam on L
E_book, E_author, E_publisher ← slice trained entity embeddings
freeze E_book, E_author, E_publisher
return E_book, E_author, E_publisher
```

3.4 Neural Collaborative Filtering (NCF)

NCF is the second base method. It consumes the frozen KG semantic prior together with trainable collaborative embeddings and learns the non-linear user–item interaction that yields the rating prediction.

Input representation for a user u and item i(with author a_i, publisher p_i), the model concatenates two trainable collaborative embeddings user e_u^u and item e_i^i with three frozen KG embeddings item g_i^i, author $\mathrm{g}_{a_i}^a$, publisher $\mathrm{g}_{p_i}^p$ into a single feature vector, as in Eq. 12.

$$\mathrm{z}_{ui} = [\, \mathrm{e}_u^u \parallel \mathrm{e}_i^i \parallel \mathrm{g}_i^i \parallel \mathrm{g}_{a_i}^a \parallel \mathrm{g}_{p_i}^p \,] \tag{12}$$

so that with $d = 64$ for every embedding, $\mathrm{z}_{ui} \in \mathbb{R}^{2d+3d} = \mathbb{R}^{320}$(two collaborative and three semantic vectors).

MLP with ReLU. The fused vector is passed through a multilayer perceptron with hidden layers of size $(256, 128, 64)$. Each layer applies a linear transform, layer normalization, ReLU activation, and dropout, as in Eq. 13.

$$\mathrm{h}^{(l)} = \mathrm{ReLU}\,(\mathrm{LN}(\mathrm{W}^{(l)}\mathrm{h}^{(l-1)} + \mathrm{b}^{(l)})), \mathrm{h}^{(0)} = \mathrm{z}_{ui} \tag{13}$$

where $\mathrm{W}^{(l)}$ and $\mathrm{b}^{(l)}$ are the weight matrix and bias of layer l, $\mathrm{LN}(\cdot)$is layer normalization, and ReLU is the activation. A final linear layer maps $\mathrm{h}^{(L)}$to a scalar, which is clamped to the rating range, as in Eq. 14.

$$\hat{r}_{ui} = \mathrm{clip}(\mathrm{w}_o^{\top}\mathrm{h}^{(L)} + b_o, 1, 5) \tag{14}$$

Objective. The trainable NCF parameters θ(collaborative embeddings and MLP weights the KG embeddings are excluded) are optimized by minimizing the mean squared error between predicted and observed ratings, as in Eq.15.

$$\mathcal{L}_{NCF}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{(u,i)\in\mathcal{B}} (\hat{r}_{ui} - r_{ui})^2 \tag{15}$$

An overview of the NCF Process is shown in Fig. 4.

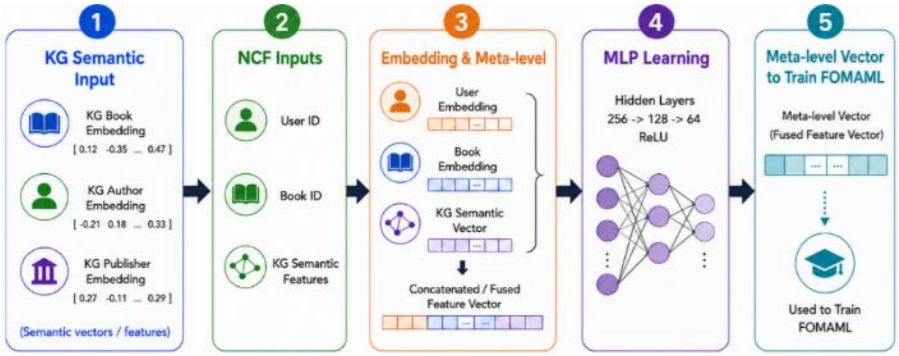


Fig. 4 Process of NCF

The NCF training routine are summarized in Algorithm 2.

Algorithm 2 NCF (Neural Collaborative Filtering)

Input : indices (u, i, a_i, p_i), target r_ui
frozen E_book/E_author/E_publisher; trainable user/item emb + MLP θ
Output: prediction r̂_ui, loss L_NCF

```
e_u ← UserEmb(u);  e_i ← ItemEmb(i)
g_i ← E_book(i); g_a ← E_author(a_i); g_p ← E_publisher(p_i)
z ← concat(e_u, e_i, g_i, g_a, g_p)
h ← z
for each MLP layer l:
    h ← Dropout(ReLU(LayerNorm(W_l·h + b_l)))
r̂_ui ← clip(w_o·h + b_o, 1, 5)
L_NCF ← (r̂_ui − r_ui)^2
update θ (exclude frozen KG) by gradient of L_NCF
return r̂_ui, L_NCF
```

3.5 Adaptation FOMAML with Hybrid Meta KG-NCF

The proposed Meta KG-NCF is built as a single sequential pipeline composed of three stages: (1) the Knowledge Graph produces a semantic representation of items, (2) this semantic representation is joined with Neural Collaborative Filtering through the meta-level process, and (3) the resulting joined model is trained and personalized through FOMAML adaptation. The term *meta-level* here refers specifically to the joining process in stage (2), the way the KG semantic representation and the NCF representation are combined, while FOMAML in stage (3) refers to how the joined model is adapted to each user. These stages run in order: KG semantics first, the meta-level join next, and FOMAML adaptation last.

Stage 1, KG semantic representation. As described in Section 3.3, the Knowledge Graph is constructed from item attributes and embedded with TransE, yielding frozen semantic vectors for each item, its author, and its publisher: g_i^i, $g_{a_i}^a$, and $g_{p_i}^p$. These vectors encode the relational and content semantics of an item and remain informative even when the item has little or no interaction history.

Stage 2, Meta-level join with NCF. The meta-level process joins the KG semantic representation with the collaborative representation of NCF. Concretely, the frozen KG vectors are concatenated with the trainable collaborative embeddings of the user and item, $\mathrm{e}_u^u$and e_i^i, to form the joined meta-level representation z_{ui}, as in Eq. 16.

$$\mathrm{z}_{ui} = [\, \mathrm{e}_u^u \parallel \mathrm{e}_i^i \parallel \mathrm{g}_i^i \parallel \mathrm{g}_{a_i}^a \parallel \mathrm{g}_{p_i}^p \,] \tag{16}$$

where ‖denotes concatenation. This joined vector is passed to the NCF predictor f_θ to produce a rating, as in Eq. 17.

$$\hat{r}_{ui} = f_\theta(\mathrm{z}_{ui}) \tag{17}$$

The meta-level join is what turns two separate base methods into one composite model: the KG semantics supply the prior knowledge, and NCF learns the non-linear user–item interaction on top of that enriched input. At this point the model is complete but not yet adaptive. The results of the application of the KG-NCF Meta pseudocode are as follows.

Algorithm 3 Meta KG-NCF

```
Input : tasks {T_u = (S_u, Q_u)}, inner lr α = 0.01, outer lr β = 0.001,
      inner steps S, epochs E
Output: meta-initialization θ*

# ── Stage 1: KG semantic representation ──
1:  build KG triples and train TransE (Pseudocode 1); FREEZE g_i, g_a, g_p

# ── Stage 2: meta-level join (KG ⋈ NCF) ──
2:  initialize trainable θ (user/item emb + MLP)
3:  define f_θ(u,i,a_i,p_i):
4:      z ← concat(e_u, e_i, g_i, g_a, g_p)
5:      return clip(MLP_θ(z), 1, 5)

# ── Stage 3: FOMAML adaptation of the joined model ──
6:  for epoch = 1 … E:
7:      for each task batch {T_u}:
8:          for each T_u:
9:              θ'_u ← θ
10:             for step = 1 … S:
11:                 θ'_u ← θ'_u − α·∇ MSE(f_{θ'_u}, S_u)
12:             L_Q(u) ← MSE(f_{θ'_u}, Q_u)
13:         θ ← Adam( θ, Σ_u ∇_{θ'_u} L_Q(u), β )
14:     validate; early-stop on best val loss
15: return θ*
```

Stage 3, FOMAML adaptation. The joined meta-level model is then processed with FOMAML so that it can adapt to new users and items from only a few interactions. Each user $u$forms a task $\mathcal{T}_u = (\mathcal{S}_u, \mathcal{Q}_u)$with a K-shot support set ($K = 5$) and a query set. The trainable parameters θ(collaborative embeddings and MLP weights) are adapted on the support set in the inner loop, while the KG semantic prior stays frozen, as in Eq. 18.

$$\theta'_u = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{S}_u}(f_\theta), \alpha = 0.01 \quad (18)$$

and the shared initialization is updated by the first-order meta-gradient of the query loss over a batch of tasks in the outer loop, as in Eq. 19.

$$\theta' \leftarrow \theta - \beta \sum_{\mathcal{T}_u} \nabla_{\theta'_u} \mathcal{L}_{\mathcal{Q}_u}(f_{\theta'_u}), \beta = 0.001 \quad (19)$$

The personalized prediction for user $u$is finally produced by the adapted model $f_{\theta'_u}(\mathrm{z}_{ui})$. Because FOMAML uses only first-order gradients, it avoids the costly second-order computation of full MAML, which is the source of the efficiency advantage of Meta KG-NCF over the FO-MSAN baseline. The complete pipeline is KG semantics, meta-level join, and FOMAML adaptation.

3.6 Evaluation

Evaluation measurement includes MAE, RMSE, NDCG, Precision, MRR and time(s). The selection of this evaluation is on the basis that MAE is able to show an overview of how much of a prediction error there is with the actual conditions in determining the rating [51]. This evaluation is able to provide a performance picture of the recommendation system based on Collaborative filtering and Content-based filtering as well as Hybrid filtering [38]. The MAE equation is shown in Eq. 20.

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i| \quad (20)$$

m is the size of the sample, y_i is the actual value of i and $\hat{y}_i$ is the prediction value of the i. The smaller the MAE score, the better the performance of the recommendation system in predicting user ratings or preferences.

The next evaluation of the RMSE allows the model to evaluate the model against major errors in prediction [52]. RMSE is used to measure the error rate in a prediction, by providing a greater error tolerance for more complex calculation processes. The RMSE formula is shown in Eq. 21.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \quad (21)$$

m is the sample size, y_i is the actual value of I, and $\hat{y}_i$ is the prediction value of the i.

NDCG is the best list indicator that results from recommendations with the highest order of best [53]. NDCG shows that the use of NDCG in the evaluation of the recommendation system is able to measure the quality of the sequence of the recommendation results as well as the most relevant items appear in the first order [41]. The NDCG formula has several stages which are shown in Eq. 22, Eq. 23, and Eq. 24.

$$DCG@p = \sum_{i=1}^{p} \frac{rel(i)}{log_2(i+1)} \quad (22)$$

$rel(i)$ is the item in position i that are relevant (e.g., a score given for an item in position i, such as 0 for irrelevant, 1 for relevant, 2 for highly relevant, etc.) *i* is the position of an item in the order of recommendations (starting from 1), *p* is the position we calculate in the list of recommendations (for example, to calculate DCG@10, we consider the first 5 positions).

It then calculates the calculated IDCG value based on the DCF equation with the order of recommendations in the ideal order, where the most relevant items are placed in the top position. IDCG is used to normalize the DCG value, so that the NDCG is in the range of 0 to 1, as shown in Eq. 23.

$$IDCG@p = \sum_{i=1}^{p} \frac{rel_{ideal}(i)}{log_2(i+1)} \quad (23)$$

$rel_{ideal}(i)$ is the ideal elevation r for position i which is arranged based on the relevance of the highest item in the top position. So that the NDCG equation can be calculated as follows:

$$NDCG@p = \frac{DCG@p}{IDCG@p} \quad (24)$$

This equation gives a normalization value of the DCG, which indicates the NDCG will be in the range of 0 to 1, where a value of 1 indicates that the order of recommendations is most ideal.

Precision in the recommendation system is more appropriate to the application of the rating value, this evaluation gives the results that match the best ranking list. Precision is effective in a system that shows a list of a ranking [54], [55], [56]. The process of precision calculating the rating value in the relevant conditions of the recommended result, the precision equation is shown in Eq. 25.

$$Precision@N = \left|\frac{relevant\ item\ Top-N}{N}\right| \quad (25)$$

MRR (Mean Reciprocal Rank) is an optimal metric for evaluating results that show ranking by considering the two-way response aspect of the user to the most relevant item. The MRR result of the list from the recommendation system measures how quickly the results match the user's condition [57]. The MRR equation corresponds to Eq. 26.

$$MRR = \frac{1}{|U|}\sum_{i=1}^{|U|}\frac{1}{Rank_i} \quad (26)$$

$|U|$ represents the total users. $Rank_i$ is ranking results list.

Time(s) is a benchmark of the length of execution of the method process from start to finish resulting in recommendations that serve as a reference for the effectiveness and efficiency of a method [58]. The execution time equation is as follows:

$$T_{exec} = T_{end} - T_{start} \quad (27)$$

T_{end} process completion time and T_{start} Is the start time of the process.

4. Experiments and Results

4.1 Baseline algorithms

In this section, we discuss the meta learning approach which is the main condition for handling cold-start in the recommendation system that has been carried out previously. To ensure a fair and reproducible comparison, we re-implemented all baseline methods TaNP, PAML, MPML, MetaEDL, and FO-MSAN from scratch rather than relying on the authors' released code, which is either unavailable or written for incompatible runtime versions. Each baseline was reconstructed according to the architectural specifications and hyperparameter settings reported in its original publication. We preserved the network structure, embedding dimensionality, optimization strategy, and training configuration described by the respective authors. All baselines were then trained and evaluated under an identical experimental protocol, the same datasets, data splits, pre-processing pipeline, evaluation metrics, hardware, and software stack as our proposed approach. So that any observed performance differences can be attributed to the methods themselves rather than to discrepancies in the experimental setup. Because all numbers reported in Section 4.5 originate from our own runs, every reported metric reflects the same controlled conditions; we therefore do not compare against numbers transcribed from prior publications. The list of baseline methods include:

1. TaNP, Task-adaptive Neural Process [29] It is a hybrid method of effective cold-start handling with limited user and item interaction. This method process by applying stochastics from relevant users is then associated with new users and processed with neural, so that user preferences from interactions can be mapped.
2. PAML, Preference-Adaptive Meta-Learning [30] is a hybrid with hybrid recommendation system approach for cold-start problems. This method uses the social relations of the users generalization technique. The use of global knowledge on all users is the initial technique of the model so that the specific preferences of several users are similar in the process with the preference-specific adapter.
3. MPML, Multi-Prior Meta-Learning [31] It is also a hybrid method for cold-start handling. This method technique processes different preference conditions with different prior techniques as well. MPML clusters based on the predictions of each prior and then performs a double-gradient descent on each prior.
4. MetaEDL, Meta Evidential Learning [32] is a hybrid approach to cold-start handling with a combination of meta-learning and evidential learning techniques. This method processes user interactions for a global model that can adapt to new users on sparse data.
5. FO-MSAN, First-Order Meta-Learning and Multi-Supervisor Association Network [1] Hybrid method to handle cold-start problems with limited user preferences and sparse items. This method uses the multi-supervisor network technique in the initial training and then continued with the association network technique so that the accuracy results can be improved.

4.2 Experiments

In this study, to support the process of testing the method of using tools, either hardware or software, as follows: Intel® Xeon 20 core @2.3GHz, 64GB RAM, Software Linux Ubuntu Server 22.04, Python 3.10, Library Python Pandas, NumPy, Scikit-learn, PyTorch. No GPU is used all numerical work, TransE training, NCF forward/backward passes, FOMAML inner and outer loops, and metric evaluation runs entirely on CPU. The same workstation is used for all methods, including the five baselines, so that the execution-time figures reported in Section 5 reflect identical hardware and software conditions.

The hyperparameter settings used across the four components of the architecture (KG-TransE, NCF, the Meta KG-NCF integration layer, and FOMAML) are summarized in Table 6. The KG channel uses kg_embed_dim = 64 to represent the three entity types in the knowledge graph (book, author, publisher), trained for 100 TransE epochs with learning rate 0.001, margin γ = 1, and five negatives per triple. The NCF channel uses ncf_embed_dim = 64 for user and item embeddings; the Meta KG-NCF integration layer concatenates two NCF embeddings (user, item) with three KG embeddings (book, author, publisher) for a total input dimension of 320 fed into hidden layers of size 256, 128, and 64 with LayerNorm, ReLU, and dropout 0.3. The FOMAML procedure uses inner_lr = 0.01, outer_lr = 0.001, inner_steps = 5, meta_epochs = 50, and support_k = 5; the inner loop is applied only to the trainable collaborative parameters, while the TransE prior remains frozen throughout meta-training. These values match the protocol used to obtain the numbers reported in Section 4.5 and were selected on a held-out portion of Goodreads 100K rather than tuned on the test splits of the other datasets. The hyperparameter used in the experiment are listed in Table 6.

Table 6. Method's hyperparameter

Parameter	KG	NCF	Meta KG-NCF	FOMAML
kg_embed_dim	64	-	64	-
ncf_embed_dim	-	64	64	64
Input dimension	192	128	320	320
Hidden layers	256, 128, 64	256, 128, 64	256, 128, 64	256, 128, 64
inner_lr	-	-	0.01	0.01
outer_lr	-	-	0.001	0.001
inner_steps	0	0	5	5
meta_epochs	-	-	50	50
support_k	-	-	5	5
TransE margin γ	1.0	–	1.0	–
TransE negatives / triple	5	–	5	–
Dropout	–	0.3	0.3	–

4.3 Cold-Start Protocol

Because the central claim of this work concerns cold-start, we describe the cold-start evaluation protocol in full so that the results in Section 4.5 are reproducible. The protocol

distinguishes two cold-start regimes cold-start user (Scenario 3) and cold-start item (Scenario 4) and is implemented identically for every method under comparison, including the five baselines.

Selection of cold-start items. From the training pool we select 50 new items as the target set of cold-start objects. Selection is restricted to items that have at least five interactions in the full ratings table (so that a meaningful support set can later be constructed), but during cold-start evaluation those interactions are removed from the warm training partition and treated as previously unseen. When the pool of eligible items exceeds three times the target size, 50 items are sampled at random with a fixed seed, otherwise we retain every eligible item up to the target size. This procedure mirrors the warm-up convention reported in prior meta-learning recommenders [48] and is identical across all methods.

Support and query split. For every user affected by the 50 selected items, we partition that user's data into a support set and a query set. The support set is drawn from the user's warm interactions ratings on items that are not part of the cold-start target set and contains exactly support_k = 5 interactions, sampled without replacement and stratified to preserve the user's rating distribution where possible. Users with fewer than five warm interactions are excluded from the cold-start task pool to avoid degenerate adaptation. The query set is constructed from the user's real interactions on the cold-start items (the positive items) augmented with up to 99 sampled negative items per query position, in the spirit of the leave-one-out evaluation widely used in the implicit-feedback literature.

Negative sampling and the is_negative label. Every row of the query set carries a binary is_negative label that is 0 for the user's real positive interactions and 1 for sampled negatives. Negatives are sampled uniformly from items the user has never rated, excluding the support items, and the procedure requires a minimum of five negatives per query before the task is admitted; tasks with fewer than five negatives are dropped to prevent the degenerate case in which NDCG would always evaluate to 1.0 because the ranked list contains only positives. The is_negative flag, rather than a rating threshold, drives the relevance judgments used to compute Precision@5, MRR@5, and NDCG@5 in cold-start scenarios. This is a non-trivial design choice: it gives standalone KG and NCF baselines a fair chance of producing non-degenerate ranking metrics in cold-start, because they can rank candidates by their predicted score and have the resulting list scored against the is_negative ground truth, even when their raw rating predictions are uncalibrated.

Cold-start user vs. cold-start item. In the cold-start user scenario (S-3), the 50 "new" users are constructed in the same fashion but with the roles of users and items reversed: we select 50 users for which all their interactions are hidden from the warm training partition; their support set is a sampled subset of their hidden interactions (also of size support_k = 5), and the query set follows the same positive/negative construction as above. In the cold-start item

scenario (S-4), the support set is drawn from each affected user's warm interactions on non-cold items, so the model learns how that user behaves before being asked to rank items it has never seen. The asymmetry between the two scenarios is deliberate and is the source of the asymmetric difficulty visible in the results: S-3 lets the model exploit item-side semantics for new users, while S-4 must rely on collaborative signal generalization for items the model has never observed in training.

Sparsity and task admission. The protocol records and reports the number of admitted and skipped tasks at each stage so that the cold-start statistics are auditable. Tasks may be skipped for any of three reasons: (i) fewer than support_k = 5 warm interactions; (ii) fewer than min_negatives_in_query = 5 sampled negatives, which would make ranking metrics meaningless; or (iii) no real positive interactions on the cold-start target. The number of admitted tasks therefore depends on dataset sparsity roughly tens to a few hundred tasks for the dense Goodreads subset and proportionally fewer for the sparser Book-Crossing data and the same sparsity filter is applied uniformly to all methods so that no method is evaluated on a different task population than its competitors.

4.4 Evaluation Scenarios

Evaluation methods consist of several scenarios, each with a different purpose. The objectives and benchmarks of each scenario include.

1. Scenario 1 accuracy testing aims to evaluate the improvement of method performance with benchmarks using precision benchmarks compared to the results of previous hybrid studies.
2. Scenario 2 testing a more robust method with another dataset like amazonbook and bookcrossing aims to evaluate the consistency of the results of the method with benchmarks using benchmarks, MAE and RMSE with a comparison of scenario 1 values
3. Scenario 3 and scenario 4 cold-start testing of users and items aims to determine the consistency of performance results of methods with benchmarks using MRR benchmarks compared to scenario 1.

The scenario implementation uses a split dataset, divided into 2 datasets, dataset-training and dataset-testing. The split dataset uses 80:20 and 90:10 ratios, which are standard split ratios. A summary of the model evaluation is shown in Table 7. The use of data-training and data-testing ratios refers to 80:20, which aims to provide data information in method training without sacrificing the performance of testing capabilities on new data, this condition can prevent overfitting and this ratio is very suitable for small and medium-scale dataset conditions [59], [60], [61], [62]. Meanwhile, the 90:10 training method aims to ensure that data from a complex recommendation system supports the method's performance without

sacrificing evaluation quality, and this ratio is suitable for large datasets [63], [64]. From these two ratios, 4 scenarios will be discarded.

Table 7. Summary of method evaluation scenarios

No	Scenario	Remarks	Split Dataset	
			80:20	90:10
1	I	Accuracy Test GoodReads100k	100K	100K
2	II	AmazonBook, Bookcrossing	S-2a, S-2b	S-2a, S-2b
3	III	Cold-start User	new user 50	new user 50
4	IV	Cold-start Item	new item 50	new item 50

Scenario I is a test to determine the accuracy value of the recommendation system that has been built. The test was carried out using datasets with total of 100k, respectively. Meanwhile, in scenario II, the test aims to assess the robustness of the built recommendation system. The test was carried out using datasets from different sources like amazonbook and bookcrossing with the same structure.

The next scenario is cold-start troubleshooting, where datasets will be added with new items and users with 50 who do not yet have transaction data. 50 new data for cold-start is an initialization strategy so that the model can start learning to embed new users or items without waiting for real data. The value of 50 refers to a warm up sample that is not too large and not too small, but the initial embedding quality is good and computational efficiency [48]. Several items are excluded from the training data, and only the book item data variables (ISSN, publisher, publish_date) are used. For new users, a different profile will be created from the existing user profile.

4.5 Result

Results of all scenario with the mention of S-1 is scenario 1, S-2 is scenario 2 and the amazon dataset is marked S-2a and the bookcrossing dataset is marked S-2b, S-3 is scenario 3 and S-4 is scenario 4 then the results are calculated on average. This accuracy testing uses the evaluation of MAE, RMSE, Precision, MRR, NDCG and time execution. The test compares Meta KG-NCF with hybrid baseline methods, using 80:20 training and testing datasets. In the MAE evaluation, RMSE and time execution the lowest results are the best results, while in the Precision evaluation, the highest MRR and NDCG results are the best results. The detailed results are shown as shown in Table 8.

Table 8. The Result using 80:20 for compare the average assessment score

Algorithm	MAE						RMSE					
	S-1	**S-2a**	**S-2b**	**S-3**	**S-4**	**Avg**	**S-1**	**S-2a**	**S-2b**	**S-3**	**S-4**	**Avg**
TaNP	2.9540	3.2736	2.7261	2.9300	2.9149	2.9597	3.0795	3.3621	2.8230	2.9753	2.9686	3.0417
PAML	0.7853	0.7264	0.8213	0.8667	0.8485	0.8096	0.9990	0.9179	0.9653	0.9127	0.9192	0.9428
MPML	2.9540	3.2736	2.7261	2.9300	2.9149	2.9597	3.0795	3.3621	2.8230	2.9753	2.9686	3.0417
MetaEDL	2.9139	0.7215	0.7433	2.8500	2.8899	2.0237	3.0427	0.8808	0.9246	2.8953	2.9440	2.1375
FO-MSAN	0.7956	0.6986	0.8371	0.7692	0.8250	0.7851	0.9764	0.8417	0.9926	0.8087	0.8835	0.9006
MetaKG-NCF	**0.7275**	**0.5983**	**0.6718**	**0.7416**	**0.7637**	**0.7006**	**0.8841**	**0.7213**	**0.7999**	**0.7748**	**0.8219**	**0.8004**
Algorithm	**Precision**						**MRR**					
	S-1	**S-2a**	**S-2b**	**S-3**	**S-4**	**Avg**	**S-1**	**S-2a**	**S-2b**	**S-3**	**S-4**	**Avg**
TaNP	0.6809	0.6876	0.6842	0.5000	0.0467	0.5199	0.8138	0.8130	0.8175	0.7222	0.1024	0.6538
PAML	**0.7853**	0.6876	0.7072	0.5000	**0.0600**	0.5480	**0.8683**	0.8130	0.8437	0.7222	**0.1400**	0.6774
MPML	0.6809	0.6876	0.6842	0.5000	0.0467	0.5199	0.8138	0.8130	0.8175	0.7222	0.1024	0.6538
MetaEDL	0.6809	0.6995	**0.7180**	0.5000	0.0467	0.5290	0.8138	0.8270	**0.8546**	0.7222	0.1024	0.6640
FO-MSAN	0.7475	**0.7034**	0.7075	0.5000	0.0539	**0.5425**	0.8630	0.8276	0.8449	0.7222	0.1052	0.6726
MetaKG-NCF	0.7426	0.6939	0.6844	**0.5000**	0.0589	0.5360	0.8704	**0.8372**	0.8174	**0.8333**	0.1097	**0.6936**
Algorithm	**NDCG@5**						**Time (s)**					
	S-1	**S-2a**	**S-2b**	**S-3**	**S-4**	**Avg**	**S-1**	**S-2a**	**S-2b**	**S-3**	**S-4**	**Avg**
TaNP	0.6978	0.7742	0.7564	0.7950	0.0844	0.6216	1.19	22.84	119.00	0.63	4.66	29.66
PAML	**0.7730**	0.7742	0.7894	0.7950	**0.1074**	0.6478	0.86	15.28	114.83	0.43	3.35	26.95
MPML	0.6978	0.7742	0.7564	0.7950	0.0844	0.6216	**0.64**	**9.04**	86.21	**0.32**	**2.45**	19.73
MetaEDL	0.6978	**0.7992**	**0.8036**	0.7950	0.0844	0.6360	0.66	10.37	181.94	0.33	2.63	39.19
FO-MSAN	0.7612	0.7979	0.7883	0.7950	0.0733	0.6431	0.75	11.38	194.09	0.36	2.94	41.90
MetaKG-NCF	0.7611	0.7976	0.7588	**0.8770**	0.0905	**0.6570**	0.67	9.12	**51.15**	0.33	3.26	**12.91**

Table 8 reports MAE, RMSE, Precision@5, MRR@5, NDCG@5, and execution time at the 80:20 split, broken down by scenario and averaged in the rightmost column of each metric block. Illustration from Table 8 is shown in Figure 5.

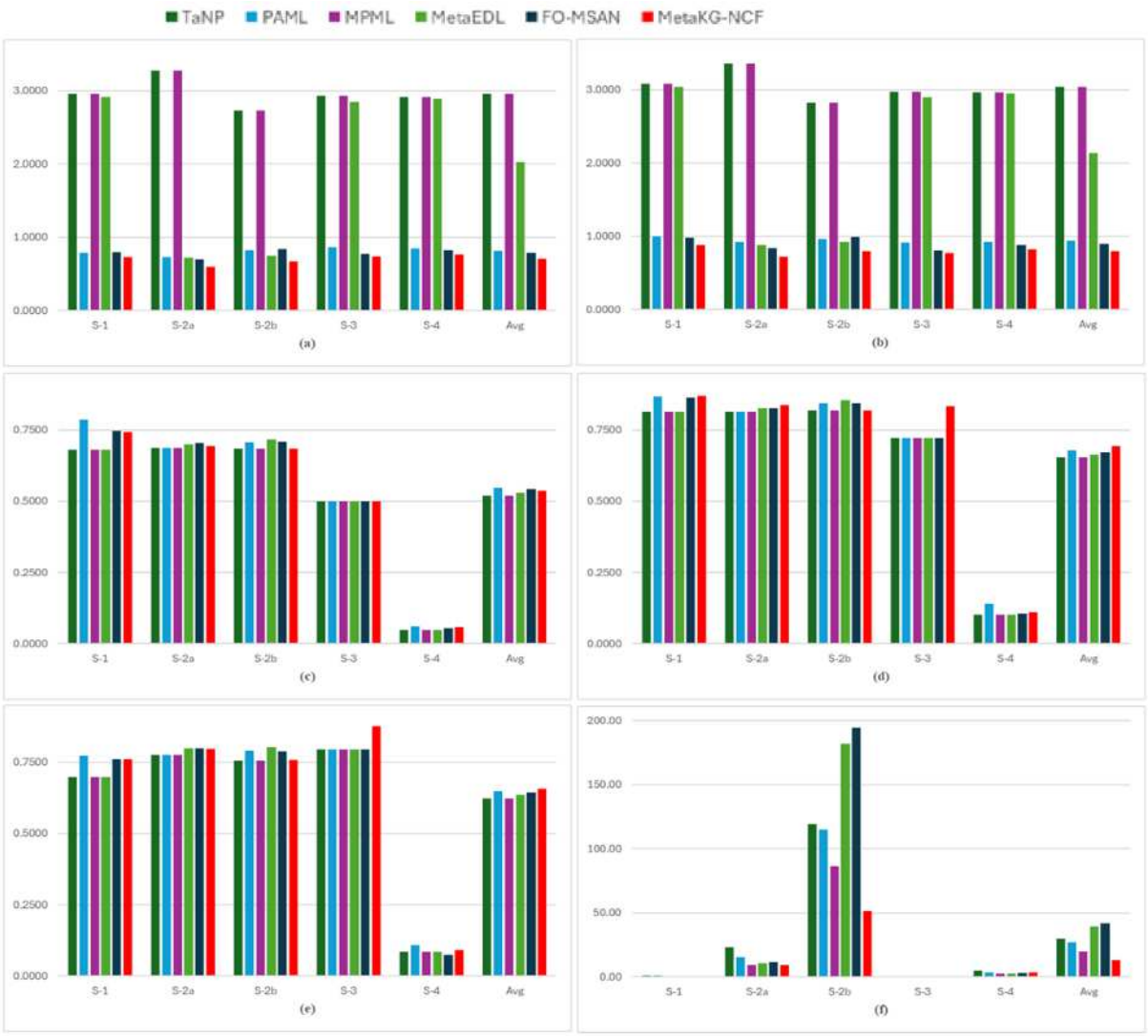


Fig. 5 Meta KG-NCF in 80:20 performance: a) MAE, b) RMSE, c) Precision, d) MRR, e) NDCG, f) time

In Table 8 Meta KG-NCF achieves the best error averages (MAE 0.7006 versus FO-MSAN's 0.7851; RMSE 0.8004 versus 0.9006, an improvement of roughly 10–11%), the highest NDCG@5 average (0.6570 versus 0.6431), the highest MRR@5 average (0.6936 versus 0.6726, driven mainly by the strong cold-start-user scenario S-3 at 0.8333), and the lowest average execution time (12.91 s versus 41.90 s). On Precision@5, however, FO-MSAN narrowly leads on average (0.5425 versus 0.5360), so the proposed method does not dominate uniformly: it is strongest on error and efficiency, comparable to FO-MSAN on ranking, and slightly behind on Precision. The uniform Precision@5 = 0.5000 entries in S-3

across all methods and the very low values in S-4 are not bugs but a reflection of the structural difficulty of new-item ranking, where the candidate pool is dominated by sampled negatives. The TaNP and MPML rows show MAE ≈ 2.96 because both methods failed to converge on this configuration we retain them in the table to make the failure visible rather than silently dropping them. In Fig. 5 visualizes Table 8 as a six-panel bar chart. Meta KG-NCF (red bars) leads on five of the six metrics on average, with FO-MSAN edging it out only on Precision@5. The next test with a 90:10 ratio of training and testing is shown in Table 9.

In Table 9, repeats the evaluation under the 90:10 split. The overall pattern is consistent with Table 8: Meta KG-NCF again has the best error averages (MAE 0.7043 versus FO-MSAN's 0.8131; RMSE 0.8129 versus 0.9382), and on execution time it is more than five times faster on average than FO-MSAN. However, the 90:10 split makes the trade-offs more visible. On Precision@5 average MetaEDL is now the leader (0.5508 versus our 0.5290) and on MRR@5 average MetaEDL also leads (0.7182 versus our 0.7124, a narrow gap). On NDCG@5 average MetaEDL is similarly the leader (0.6875 versus our 0.6670). The proposed method retains its largest advantage on rating-prediction accuracy and on cold-start-user MRR (0.9444 in S-3 versus the next-best 0.8889), but the ranking competition is genuinely tight in this setting. These observations should be read as a refinement, not a contradiction, of the 80:20 results: when the training set is larger, baselines with stronger inductive biases on ranking close the gap, while the error-reduction and efficiency advantages of Meta KG-NCF remain. The illustration from Table 9 is shown in Fig 6.

Table 9. The Result using 90:10 for compare the average assessment score

Algorithm	MAE						RMSE					
	S-1	**S-2a**	**S-2b**	**S-3**	**S-4**	**Avg**	**S-1**	**S-2a**	**S-2b**	**S-3**	**S-4**	**Avg**
TaNP	2.9112	3.2929	2.7495	2.9300	2.9149	2.9597	3.0416	3.3866	2.8445	2.9753	2.9686	3.0433
PAML	0.8213	**0.6363**	0.8032	0.8303	0.8460	0.7874	1.0275	**0.7830**	0.9507	**0.8994**	0.9152	0.9152
MPML	2.9112	3.2929	2.7495	2.9300	2.9149	2.9597	3.0416	3.3866	2.8445	2.9753	2.9686	3.0433
MetaEDL	0.8329	0.6818	0.7277	0.7969	0.8948	0.7868	1.0285	0.8469	0.9138	0.8422	0.9566	0.9176
FO-MSAN	0.8639	0.7342	0.8398	0.7611	0.8664	0.8131	1.0457	0.8827	0.9975	0.8329	0.9323	0.9382
MetaKG-NCF	**0.7330**	0.6404	**0.6600**	**0.7450**	**0.7429**	**0.7043**	**0.8923**	0.8055	**0.7886**	0.7749	**0.8032**	**0.8129**
Algorithm	**Precision**						**MRR**					
	S-1	**S-2a**	**S-2b**	**S-3**	**S-4**	**Avg**	**S-1**	**S-2a**	**S-2b**	**S-3**	**S-4**	**Avg**
TaNP	0.6550	0.6837	0.6972	0.5000	0.0467	0.5165	0.7937	0.8269	0.8297	0.7222	0.1024	0.6550
PAML	0.7350	**0.6977**	0.7228	0.5000	**0.0689**	0.5449	0.8542	0.8527	0.8497	0.8333	**0.1522**	0.7084
MPML	0.6550	0.6837	**0.7678**	0.5000	0.0467	0.5306	0.7937	0.8269	0.8297	0.7222	0.1024	0.6550
MetaEDL	**0.7950**	0.6791	0.7345	0.5000	0.0456	**0.5508**	**0.8979**	**0.8530**	**0.8638**	0.8889	0.0875	**0.7182**
FO-MSAN	0.7700	0.6884	0.7163	0.5000	0.0650	0.5479	0.8667	0.8346	0.8430	0.7778	0.1246	0.6893
MetaKG-NCF	0.7250	0.6868	0.7004	**0.5000**	0.0328	0.5290	0.8792	0.8497	0.8270	**0.9444**	0.0617	0.7124
Algorithm	**NDCG@5**						**Time (s)**					
	S-1	**S-2a**	**S-2b**	**S-3**	**S-4**	**Avg**	**S-1**	**S-2a**	**S-2b**	**S-3**	**S-4**	**Avg**
TaNP	0.6805	0.7864	0.7678	0.7950	0.0844	0.6228	0.48	11.16	62.64	0.50	3.82	15.72
PAML	0.7534	**0.8167**	0.7983	0.8770	**0.1228**	0.6736	0.46	7.39	37.14	0.45	3.51	9.79
MPML	0.6805	0.7864	0.7678	0.7950	0.0844	0.6228	0.40	**4.42**	**24.58**	0.38	2.93	**6.54**
MetaEDL	**0.8329**	0.8038	**0.8137**	0.9180	0.0690	**0.6875**	**0.36**	5.07	26.19	**0.35**	**2.71**	6.94
FO-MSAN	0.7960	0.7935	0.7896	0.8360	0.1003	0.6631	0.48	4.79	31.17	0.48	3.82	7.55
MetaKG-NCF	0.7501	0.8106	0.7696	**0.9590**	0.0457	0.6670	0.39	4.78	26.02	0.39	3.03	6.92

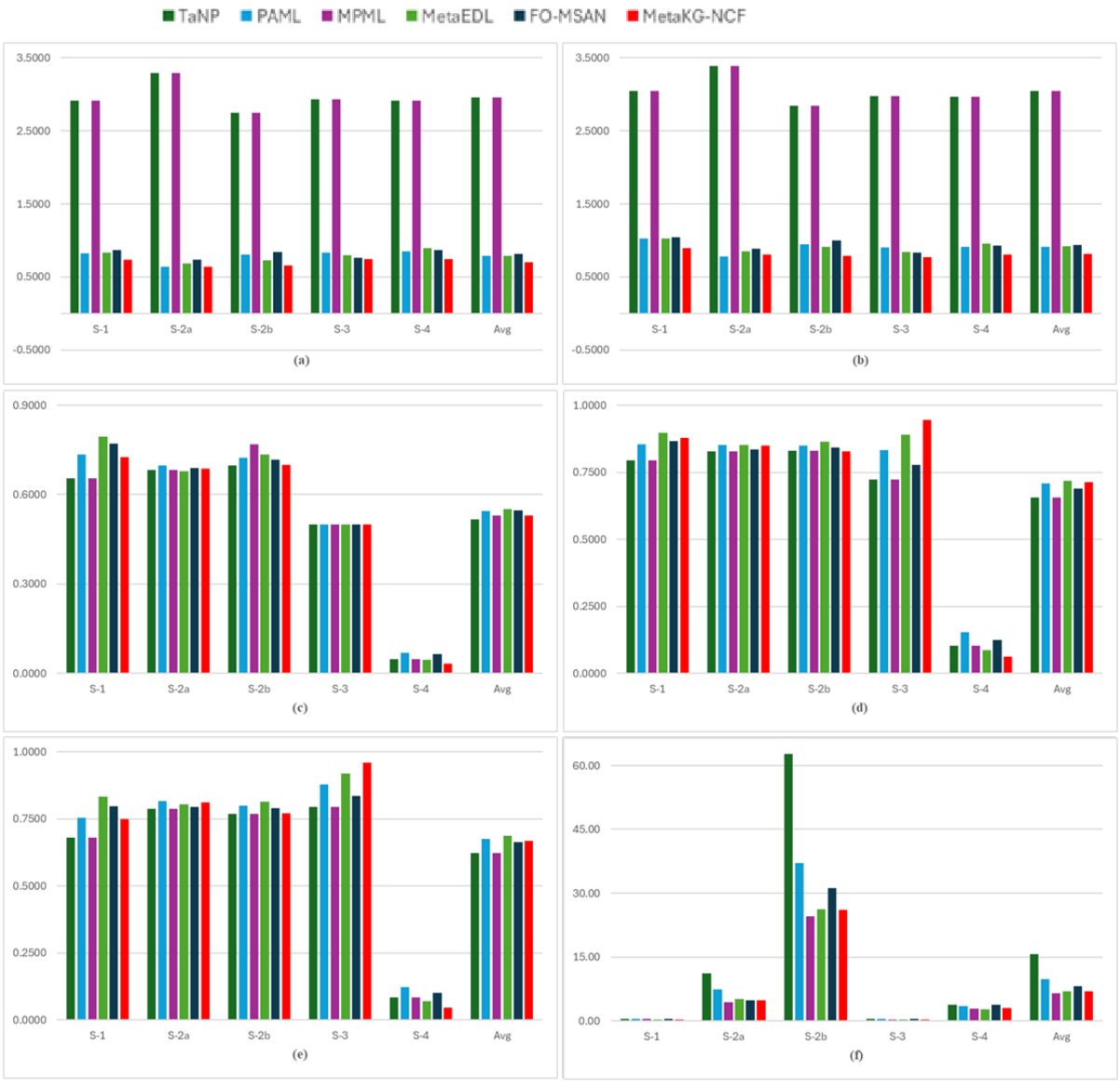


Fig. 6 Meta KG-NCF in 90:10 performance: a) MAE, b) RMSE, c) Precision, d) MRR, e) NDCG, f) time

In Fig. 6, visualizes Table 9 in the same six-panel layout as Figure 5. The conclusion mirrors that of the 80:20 case for error metrics and efficiency: Meta KG-NCF (red) leads on MAE, RMSE, and execution time, while on ranking metrics the picture is more competitive, with FO-MSAN (blue) and MetaEDL remaining strong on MRR and NDCG. We unpack these observations in the next section, including a paired statistical-significance test and an ablation that isolates the contribution of the KG prior.

5. Discussions and Analysis

5.1 Statistical Significance and Efficiency

Because per-method averages alone can be misleading on small ranking-metric differences, we test the statistical significance of the gap between FO-MSAN, the strongest baseline, and the proposed Meta KG-NCF. We use a paired evaluation on identical Goodreads 100K test users, applying the paired two-sided t-test together with the Wilcoxon signed-rank test on per-user metric scores. Effect size is reported using t-test and the rank-biserial correlation r for the Wilcoxon test.

Multiple-comparison correction across the five reported metrics is performed with the Holm–Bonferroni procedure, and we additionally check the normality of paired differences with the Shapiro–Wilk test so that we know which of the two procedures is the more reliable for each metric. We mark a metric as significant (Sig ✓) only when the corrected p-value is below 0.05 otherwise it is marked as not significant (✗). Table 10 summarizes the outcome.

Table 10. Significant Test Goodreads 100K FO-MSAN and Meta KG-NCF

Matrix	FO-MSAN	Meta KG-NCF	Δ	T-test	Wilcoxon P	Sig
MAE	0.7956	**0.7275**	**+0.0681**	**6.07e-06**	**4.93e-06**	✓
RMSE	0.9764	**0.8841**	**+0.0923**	**3.18e-08**	**7.94e-09**	✓
NDCG@5	**0.7612**	0.7611	-0.0001	1	1	✗
Precision@5	**0.7475**	0.7426	-0.0049	1	0.695	✗
MRR@5	0.8630	**0.8704**	+0.0074	1	1	✗

Table 10 confirms the picture that emerged from Tables 8 and 9: the advantage of Meta KG-NCF over FO-MSAN is statistically firm on error reduction (MAE and RMSE), with corrected p-values below 1e–5 and a T-test in the medium-effect range (|d| ≈ 0.5–0.6 on the per-user scores). On the three- ranking metrics (NDCG@5, Precision@5, MRR@5), the per-user differences are too small relative to their variance to be distinguishable from zero after Holm–Bonferroni correction; in those cases, the two methods should be reported as statistically comparable rather than as one outperforming the other. This is the technically honest reading of the data and matches the revised framing of the abstract and the contribution list in Section 1.

Efficiency under identical hardware the execution-time comparison in Table 11 was obtained under the same Intel Xeon 20-core configuration with 64 GB RAM and 8 PyTorch CPU threads described in Section 4.2 no GPU, no accelerator, no method-specific runtime tuning. Because the comparison reports wall-clock time and is sensitive to hardware, we explicitly disclose this homogeneity every method, including FO-MSAN was rerun on the same machine with the same scheduler settings and the same Python interpreter, and was timed end-to-end including data loading, training, and per-task evaluation.

Table 11. Total time execution Result FO-MSAN and Meta KG-NCF

Method	goodreads_100k	bookcrossing	amazon_Book
FO-MSAN	8.40	29.49	17.55
Meta KG-NCF	**7.50**	**24.61**	**14.49**

In Table 11, On Goodreads 100K, Meta KG-NCF finishes in 7.50 s versus 8.40 s for FO-MSAN (a reduction of about 11%). The gap widens to about 17% on Book-Crossing (24.61 s versus 29.49 s) and 17% on AmazonBook (14.49 s versus 17.55 s). The reason is structural rather than incidental. FOMAML approximates the meta-gradient with first-order terms and therefore avoids the costly second-order computation required by full MAML, FO-MSAN inherits that first-order benefit but additionally maintains a multi-supervisor association network whose forward and backward passes are not free, and which dominates the per-task cost as the dataset grows. Because we report wall-clock time on the same machine, the figures should be read as relative speedup under a controlled apparatus rather than as a hardware-independent claim. On different hardware for example a GPU-accelerated environment the absolute numbers would change, but the structural argument (no second-order term in the meta-gradient, no auxiliary association network on top of NCF) would continue to apply.

5.2 Ablation

To isolate the contribution of each major component of Meta KG-NCF, we run an ablation study under the same protocol used for the main results. We compare three configurations: (i) the standalone KG channel, in which only the TransE-pretrained semantic vectors are used and no collaborative NCF parameters or inner-loop adaptation are applied, (ii) the standalone NCF channel, in which the KG prior is removed and the recommender relies only on user/item collaborative embeddings, and (iii) the full Meta KG-NCF that fuses the two channels and meta-adapts the collaborative parameters with FOMAML. Each configuration is evaluated with the cold-start protocol of Section 4.3, so that ranking metrics are computed against the is_negative labels and not against a rating threshold. This is the methodological detail that addresses a concern raised in earlier evaluations of standalone KG models, where ranking metrics collapsed to near-zero because the standalone model could not produce calibrated rating predictions. Under the is_negative-based protocol, the standalone KG can still rank candidate items by their TransE-based score and have the ranked list evaluated against the binary positive/negative ground truth, which yields well-defined and non-degenerate values. The results are shown in Table 12. The ablation in Table 12 reveals a coherent and somewhat asymmetric picture of how the two channels contribute. The full Meta KG-NCF clearly improves the accuracy of rating prediction; it cuts MAE from 0.7885 (KG only) and 0.7960 (NCF only) down to 0.7286, and RMSE from 0.9483 and 0.9568 down to 0.8851.

Table 12. Ablation Meta KG-NCF

Method	MAE	RMSE	Precision@5	NDCG@5	MRR@5
KG	0.7885	0.9483	0.7673	0.7889	**0.8821**
NCF	0.7960	0.9568	**0.7796**	**0.7923**	0.8693
Meta KG-NCF	**0.7286**	**0.8851**	0.7352	0.7415	0.8255

In Table 12, The accuracy gain comes from the fusion of the semantic prior with the collaborative model, which is exactly what the meta-level hybridization is designed to enable. TransE supplies item-side regularities that pure collaborative filtering cannot recover from sparse ratings, and FOMAML adapts the collaborative parameters on top of this prior.

On the three-ranking metrics, however, the standalone components are surprisingly competitive. NCF-only delivers the highest Precision@5 (0.7796) and the highest NDCG@5 (0.7923); KG-only delivers the highest MRR@5 (0.8821). The full Meta KG-NCF sits slightly below both on those three metrics. The most plausible explanation is that fusing two well-calibrated ranking signals through a single MLP regression head can blur the sharp ranking each channel produces in isolation. Optimization is then driven primarily by the rating loss, which the full model wins decisively on. In other words: the fusion is unambiguously a win for point-estimate accuracy and a slight wash for ranking sharpness. This finding is reported honestly here rather than being smoothed over. The illustration from Table 12 is shown in Fig. 7.

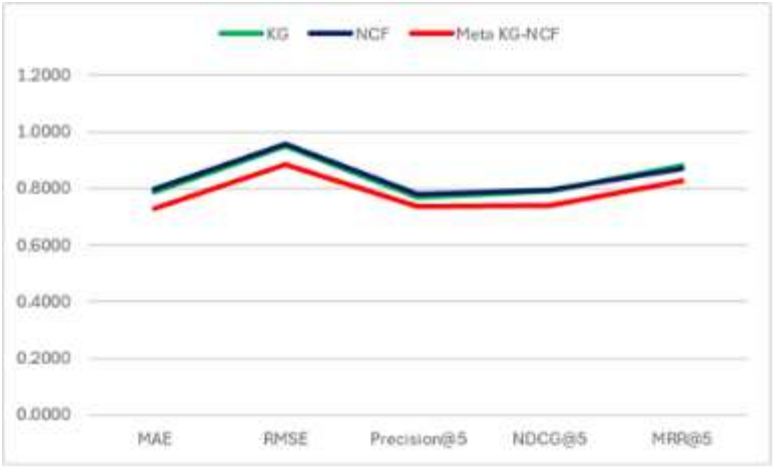


Fig. 7 Ablation Meta KG-NCF

In Fig. 7, The relative ordering on MAE and RMSE (where lower is better, hence the shorter bar is best) shows the full model with a clear margin. On the three ranking metrics the bars cluster closely together. We read this as a confirmation that the principal benefit of fusing the two channels is calibration of the rating prediction and not, on its own, sharper ranking. The companion claim that meta-learning is responsible for the cold-start advantage, rather

than the fusion itself, is supported by the cold-start scenarios in Tables 8 and 9: the meta-adapted full model is the only one that achieves the very high MRR = 0.9444 on S-3 (90:10), which neither of the static ablation variants can reproduce.

5.3 Per-Scenario Analysis and Complexity

The experimental results in Section 4 can be interpreted across the four evaluation scenarios. In the accuracy scenario (S-1) on Goodreads, Meta KG-NCF obtained the lowest error among all methods, reducing MAE and RMSE by roughly 10–11% relative to FO-MSAN. This improvement is attributable to the knowledge-graph semantic prior, which supplies item-side information (author and publisher relations) that pure collaborative models cannot recover from sparse ratings alone; a similar benefit of injecting knowledge-graph semantics into recommendation has been reported in prior work [35].

In the robustness scenario (S-2a on AmazonBook and S-2b on Book-Crossing), Meta KG-NCF maintained the best error values on datasets of different source, scale, and sparsity, indicating that the advantage is not specific to a single dataset. This consistency across heterogeneous datasets is an important property for hybrid recommenders intended for real-world deployment [48].

In the cold-start scenarios (S-3 for new users and S-4 for new items), the results are more nuanced. For new users (S-3), Meta KG-NCF achieved a clearly higher MRR than FO-MSAN (0.8333 versus 0.7222 at 80:20), showing that the FOMAML inner-loop adaptation effectively personalizes the model from only a few support interactions, consistent with the cold-start behaviour reported for meta-learning recommenders [29]. For new items (S-4), all methods degrade sharply because only a few positive items appear among many sampled negatives; here FO-MSAN remains competitive, which explains why the proposed method's overall ranking advantage is not statistically significant. This trade-off is discussed further as a limitation in Section 5.4.

On ranking metrics overall (Precision, MRR, NDCG), Meta KG-NCF is statistically comparable to FO-MSAN rather than uniformly superior, as confirmed by the significance test in Table 10 ($p > 0.05$ for all three). The principal and statistically significant gains are therefore on error reduction (MAE, RMSE) and on computational efficiency, where Meta KG-NCF trains 11–17% faster than FO-MSAN on identical hardware. The efficiency gain follows from the first-order formulation of FOMAML, which approximates the meta-gradient with first-order terms and thereby avoids the costly second-order computation of full model-agnostic meta-learning [1].

This efficiency can also be understood through the complexity of the three pipeline stages. The knowledge-graph construction with TransE is performed once before meta-training and then frozen, for $|\mathcal{G}|$ triples, n negatives per triple, and embedding dimension d

over E_{KG} epochs, its cost is $O(E_{KG} \cdot |\mathcal{G}| \cdot n \cdot d)$, and because it does not recur during adaptation it adds negligible overhead. The NCF feature extraction performs the MLP forward and backward passes, costing $O(R \cdot L \cdot d^2)$ per epoch for R ratings and L layers. The hybrid meta-level prediction concatenates the collaborative and semantic vectors and applies the predictor, contributing $O(R \cdot d^2)$. Wrapped in the FOMAML loop, the overall training complexity scales as $O(E_{meta} \cdot T \cdot S \cdot R \cdot L \cdot d^2)$, where E_{meta}, T, and S denote the number of meta-epochs, tasks, and inner steps. No second-order term appears in this expression, which is precisely why Meta KG-NCF trains faster than FO-MSAN while preserving its cold-start adaptation capability.

5.4 Comparison with Prior Work and Limitations

Compared with FO-MSAN, which combines a Multi-Supervisor mechanism with an Association Network, Meta KG-NCF introduces an explicit knowledge-graph semantic prior and a simpler first-order adaptation, achieving significantly lower error and faster training while remaining competitive on ranking. Relative to other meta-learning cold-start methods: TaNP, PAML, MPML, MetaEDL, and FO-MSAN, Meta KG-NCF is the only one in our experiments that combines a frozen relational prior with collaborative meta-adaptation, which accounts for its strong accuracy and efficiency on sparse book data.

Several limitations should be acknowledged explicitly so that readers can interpret the results appropriately and so that the strongest avenues for follow-up are visible.

- Ranking gains are not statistically significant. The Holm–Bonferroni-corrected p-values in Table 10 show that the improvement over FO-MSAN on Precision@5, MRR@5, and NDCG@5 is not statistically distinguishable from zero on Goodreads 100K. The ablation in Table 15 furthermore shows that fusing the two channels through a single regression head can slightly reduce ranking sharpness relative to the strongest standalone channel. The reported advantage of Meta KG-NCF is therefore on rating-prediction error (MAE, RMSE) and efficiency, with ranking quality on par with the strongest baseline rather than uniformly superior.
- The new-item scenario is the hardest case. In S-4 all methods degrade sharply and FO-MSAN, PAML, or MetaEDL occasionally outperform Meta KG-NCF on individual ranking columns. This is the regime in which support interactions for the new item are, by definition, absent, leaving the model to extrapolate from item-side semantics alone, a regime in which a frozen TransE prior gives only modest leverage. Improving S-4 likely requires either a richer item-side graph or an inductive prior that can update at inference time without sacrificing the first-order efficiency of FOMAML.
- Knowledge-graph coverage. The current graph uses only two relation types (author and publisher) because these are the attributes available across all three datasets. Richer relations such as genre, series, citation links, or author collaboration networks may further

improve the semantic prior, but were not available in the datasets we used; their integration is a natural avenue for follow-up work.

- Frozen TransE prior. The TransE embeddings are kept frozen during meta-training to preserve the first-order efficiency of FOMAML and to avoid drift in the semantic prior. A jointly fine-tuned or more expressive embedding model (e.g., TransR, RotatE, or a relational graph neural network) could improve performance at additional computational cost; this design choice is therefore conservative on efficiency and may be revisited in future work.
- Cold-start scale. The cold-start evaluation introduces 50 new users and 50 new items, which is consistent with prior meta-learning recommender protocols [48] but represents a single warm-up scale. Sweeping a wider range of cold-start severities from 10 to several hundred new entities, and from very-cold (zero support) to mildly-cold (≥5 support) regimes would strengthen the generalization claims.
- Hardware fairness. Wall-clock execution times are reported on a fixed CPU-only configuration. While all methods were run on the same machine to make the comparison fair, the absolute numbers would change on different hardware, particularly with GPU acceleration. The structural argument for the speedup (no second-order meta-gradient term, no auxiliary association network) is independent of hardware, but the multipliers we report should not be transferred verbatim to other environments.
- Reproducibility of baselines. All five baselines were re-implemented from scratch following the architectures and hyperparameters of their original publications. Our re-implementations were validated against the qualitative behaviour reported in those publications (e.g., relative ordering across datasets), but any discrepancy in absolute numbers compared to the original papers may, in part, reflect implementation differences. We mitigate this by always reporting the baselines and the proposed method under identical conditions, so that within-table comparisons are well-controlled even if cross-paper comparisons are not.

These limitations point directly to the future work outlined in the conclusion.

6. Conclusion

This study proposed Meta KG-NCF, a hybrid book-recommendation method that integrates a Knowledge Graph (KG) semantic prior with Neural Collaborative Filtering (NCF) through a meta-level composition and trains the resulting model with first-order Model-Agnostic Meta-Learning (FOMAML) to address the cold-start problem. The KG is a formally constructed graph G = (V, E) over books, authors, and publishers with two typed relations (written_by, published_by), embedded with TransE to supply a frozen semantic representation, NCF then learns the non-linear user–item interaction on top of this enriched

representation; and FOMAML adapts the trainable collaborative parameters to new users and items from only a few support interactions, while the semantic prior remains fixed.

Experiments on three datasets Goodreads 100K, AmazonBook, and Book-Crossing under 80:20 and 90:10 train/test splits and under a cold-start protocol with explicit support/query construction, 99 sampled negatives per query position, and an is_negative-labelled relevance signal showed that Meta KG-NCF reduces rating-prediction error relative to FO-MSAN and the other four meta-learning baselines (TaNP, PAML, MPML, MetaEDL) by roughly 10–11%, with statistically significant differences after Holm–Bonferroni correction (Wilcoxon $p < 1e–5$, T-test in the medium range). On ranking metrics (Precision@5, MRR@5, NDCG@5) the proposed method is statistically comparable to FO-MSAN rather than uniformly superior, and an ablation study revealed that fusing the semantic and collaborative signals sharpens point-estimate accuracy while slightly diffusing ranking order. In terms of efficiency, Meta KG-NCF trained 11–17% faster than FO-MSAN on identical Intel Xeon CPU-only hardware with eight PyTorch threads, owing to its first-order adaptation and the absence of an auxiliary association network. Together, these results indicate that the principal contributions of the method are improved rating-prediction accuracy and computational efficiency under cold-start conditions, with ranking quality on par with the strongest baseline.

The study also has limitations that frame its conclusions. The advantage in ranking is not statistically established at the chosen significance level; the new-item cold-start regime (S-4) remains the hardest setting for every method tested; the knowledge graph currently uses only two relation types, and the TransE embeddings are frozen during meta-training. The cold-start evaluation is conducted at a single warm-up scale of 50 new users and 50 new items, and the execution-time figures, although obtained on identical hardware for every method, are reported on a fixed CPU-only configuration and should not be transferred verbatim to other environments. Future work will therefore explore richer knowledge-graph relations (genre, series, citation links, and author collaboration networks); more expressive or jointly fine-tuned embedding models (e.g., TransR, RotatE, or relational graph neural networks); the integration of graph neural networks and temporal or sequential user behaviour for stronger ranking and incremental or online-learning extensions of FOMAML for dynamic recommendation environments, where the warm-up assumption underlying the present cold-start protocol no longer holds.

7. Acknowledgements

This study was supported by the Universitas Negeri Malang, the authors are deeply grateful for the facilities, guidance, and encouragement extended to them throughout the research process.

8. Funding statement

This work was supported by Universitas Negeri Malang through the EQUITY Research Grant Pascasarjana under Grant No. 12.11.135/UN32/KP/2025.

9. Data availability statement

The datasets analysed in this study are publicly available secondary data. The Goodreads Book dataset (100K and 1M), The AmazonBook and The Book-Crossing are available on Kaggle. The datasets are likewise publicly available from their original public repositories. The processed data splits, knowledge-graph triples, and code used to reproduce the experiments are available from the corresponding author upon reasonable request. To ensure direct public access without mandatory sign-in barriers, we have mirrored these original datasets in a shared google drive. The mirrored datasets and materials can be downloaded directly and openly from:
https://drive.google.com/drive/folders/1dQkiwuRGAeuvzFg1QF_WociLIT5vfumb?usp=sharing

10. Ethics statement

- Ethical Approval: Not Applicable. This study did not involve any experiments on human participants or animals. The research used only publicly available, anonymised secondary datasets.
- Informed Consent/Consent to Participate: Not Applicable.
- Consent to Publish Identifying Information/Images: Not Applicable.

11. Author contributions

Erfan Ainul Yakin: Conceptualization, Methodology, Software, Validation, Investigation, Writing, original draft. Triyanna Widiyaningtyas: Conceptualization, Supervision, Methodology, Writing, review & editing, Funding acquisition (corresponding author). Hary Suswanto: Validation, Formal analysis, Writing, review & editing. Didik Dwi Prasetya: Data curation, Investigation, Writing, review & editing. Wahyu Caesarendra: Visualization, Validation, Writing, review & editing. All authors read and approved the final manuscript.

12. References

[1] X. Liu, Z. Zhang, X. Zhang, P. De Meo, and H. Cherifi, “Multi-supervisor association network cold start recommendation based on meta-learning,” *Expert Syst. Appl.*, vol. 267, p. 126204, Apr. 2025, doi: 10.1016/j.eswa.2024.126204.

[2] I. F. Rokhim, N. Ifada, and F. H. Rachman, “Implementing Set-Valued k-Modes Clustering Technique to Enhance the Quality of Library Book Recommendation System using Hybrid Semantic Ontology-based model,” in *2023 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, IEEE, Nov. 2023, pp. 353–358. doi: 10.1109/COMNETSAT59769.2023.10420658.

[3] T. Roy and P. S. D., “A Hybrid Approach to Predict Ratings for Book Recommendation System Using Machine Learning Techniques,” in *2024 IEEE Region 10 Symposium (TENSYMP)*, IEEE, Sep. 2024, pp. 1–6. doi: 10.1109/TENSYMP61132.2024.10752128.

[4] A. D. Amirtharaj, D. Raghavan, and J. Arulappan, “Preferences for printed books versus E−books among university students in a Middle Eastern country,” *Heliyon*, vol. 9, no. 6, p. e16776, Jun. 2023, doi: 10.1016/j.heliyon.2023.e16776.

[5] K. V K and S. Murugesan, “An Effective Book Recommendation System using Weighted Alternating Least Square (WALS) Approach,” *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 4, 2024, doi: 10.14569/IJACSA.2024.0150463.

[6] J. Lim, G. E. K. Whitehead, and Y. Choi, “Interactive e-book reading vs. paper-based reading: Comparing the effects of different mediums on middle school students’ reading comprehension,” *System*, vol. 97, p. 102434, Apr. 2021, doi: 10.1016/j.system.2020.102434.

[7] T. T. Wijaya, Y. Cao, R. Weinhandl, and M. Tamur, “A meta-analysis of the effects of E-books on students’ mathematics achievement,” *Heliyon*, vol. 8, no. 6, p. e09432, Jun. 2022, doi: 10.1016/j.heliyon.2022.e09432.

[8] M. Naghiaei, H. A. Rahmani, and M. Dehghan, “The Unfairness of Popularity Bias in Book Recommendation,” *Cham: Springer International Publishing*, pp. 69–81, 2022, doi: 10.1007/978-3-031-09316-6_7.

[9] Z. Wang and Y. Wang, “Digital Library Book Recommendation System Based on Tag Mining,” *Journal of Artificial Intelligence Research*, vol. 1, no. 1, pp. 10–16, Oct. 2024, doi: 10.70891/JAIR.2024.100022.

[10] N. L. H. Hien, L. Van Huy, H. H. Manh, and N. Van Hieu, “A Deep Learning Model for Context Understanding in Recommendation Systems,” *Informatica*, vol. 48, no. 1, Jan. 2024, doi: 10.31449/inf.v48i1.4475.

[11] S. Rajalakshmi, G. Indumathi, Arun Elias, G. Shanmuga Priya, and Vidhya Muthulakshmi. R, “Personalized Online Book Recommendation System Using Hybrid Machine Learning Techniques,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 15s, pp. 39–46, 2024, Accessed: Jul. 01, 2025. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/4712

[12] N. R. Kermany, W. Zhao, T. Batsuuri, J. Yang, and J. Wu, “Incorporating user rating credibility in recommender systems,” *Future Generation Computer Systems*, vol. 147, pp. 30–43, Oct. 2023, doi: 10.1016/j.future.2023.04.029.

[13] S. Lee, J. Kim, and E. Park, “Can book covers help predict bestsellers using machine learning approaches?,” *Telematics and Informatics*, vol. 78, p. 101948, Mar. 2023, doi: 10.1016/j.tele.2023.101948.

[14] I. Saifudin, T. Widiyaningtyas, I. A. E. Zaeni, and A. Aminuddin, “SVD-GoRank: Recommender System Algorithm Using SVD and Gower’s Ranking,” *IEEE Access*, vol. 13, pp. 19796–19827, 2025, doi: 10.1109/ACCESS.2025.3533558.

[15] S. Robo, T. Widiyaningtyas, and W. S. G. Irianto, “HCF-MFGB: Hybrid Collaborative Filtering Based on Matrix Factorization and Gradient Boosting,” *Computers, Materials & Continua*, vol. 86, no. 2, pp. 1–19, 2026, doi: 10.32604/cmc.2025.073011.

[16] H. Morise, K. Atarashi, S. Oyama, and M. Kurihara, “Neural collaborative filtering with multicriteria evaluation data,” *Appl. Soft Comput.*, vol. 119, p. 108548, Apr. 2022, doi: 10.1016/j.asoc.2022.108548.

[17] F. Wayesa, M. Leranso, G. Asefa, and A. Kedir, “Pattern-based hybrid book recommendation system using semantic relationships,” *Sci. Rep.*, vol. 13, no. 1, p. 3693, Mar. 2023, doi: 10.1038/s41598-023-30987-0.

[18] B. Shen and Y. Zhang, “Research on personalised knowledge graph recommendation algorithm based on self-supervised learning,” *Discover Applied Sciences*, vol. 6, no. 8, p. 434, Aug. 2024, doi: 10.1007/s42452-024-06135-2.

[19] X. Liu, Z. Yang, and J. Cheng, “Music recommendation algorithms based on knowledge graph and multi-task feature learning,” *Sci. Rep.*, vol. 14, no. 1, p. 2055, Jan. 2024, doi: 10.1038/s41598-024-52463-z.

[20] H. Wang, Q. Li, H. Luo, and Y. Tang, “The Graph Attention Recommendation Method for Enhancing User Features Based on Knowledge Graphs,” *Mathematics*, vol. 13, no. 3, p. 390, Jan. 2025, doi: 10.3390/math13030390.

[21] H. Arabi, V. Balakrishnan, and N. L. Mohd Shuib, “A Context-Aware Personalized Hybrid Book Recommender System,” *Journal of Web Engineering*, Jul. 2020, doi: 10.13052/jwe1540-9589.19343.

[22] P. Arunruviwat and V. Muangsin, “A Hybrid Book Recommendation System for University Library,” in *2022 26th International Computer Science and Engineering Conference (ICSEC)*, IEEE, Dec. 2022, pp. 291–295. doi: 10.1109/ICSEC56337.2022.10049318.

[23] C. Jayapal, G. S, and H. S.V., “Book Recommendation System Using Hybrid Filtering,” in *2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, IEEE, Jun. 2023, pp. 1–5. doi: 10.1109/ICAECA56562.2023.10199590.

[24] A. Tholib, T. Widiyaningtyas, and D. D. Prasetya, “An Intelligent Recommendation System Utilizing a Hybrid Deep Learning Method,” *Engineering, Technology & Applied Science Research*, vol. 15, no. 4, pp. 25971–25977, Aug. 2025, doi: 10.48084/etasr.12230.

[25] H. Hairani, T. Widiyaningtyas, and D. Dwi Prasetya, “Addressing Class Imbalance of Health Data: A Systematic Literature Review on Modified Synthetic Minority Oversampling Technique (SMOTE) Strategies,” *JOIV : International Journal on Informatics Visualization*, vol. 8, no. 3, p. 1310, Sep. 2024, doi: 10.62527/joiv.8.3.2283.

[26] P. M. T. Do, Q. Zhang, G. Zhang, and J. Lu, “meta-GRS: A Graph Neural Network for Cross-Domain Recommender System via Meta-Learning,” *Procedia Comput. Sci.*, vol. 225, pp. 2536–2545, 2023, doi: 10.1016/j.procs.2023.10.245.

[27] Y. Liu, S. Wang, X. Li, and F. Sun, “A Meta-adversarial Framework for Cross-Domain Cold-Start Recommendation,” *Data Sci. Eng.*, vol. 9, no. 2, pp. 238–249, Jun. 2024, doi: 10.1007/s41019-024-00245-y.

[28] H. Wang, J. Mi, X. Guo, and P. Hu, “Meta-learning adaptation network for few-shot link prediction in heterogeneous social networks,” *Inf. Process. Manag.*, vol. 60, no. 5, p. 103418, Sep. 2023, doi: 10.1016/j.ipm.2023.103418.

[29] X. Lin, J. Wu, C. Zhou, S. Pan, Y. Cao, and B. Wang, “Task-adaptive Neural Process for User Cold-Start Recommendation,” in *Proceedings of the Web Conference 2021*, New York, NY, USA: ACM, Apr. 2021, pp. 1306–1316. doi: 10.1145/3442381.3449908.

[30] L. Wang *et al.*, “Preference-Adaptive Meta-Learning for Cold-Start Recommendation,” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, California: International Joint Conferences on Artificial Intelligence Organization, Aug. 2021, pp. 1607–1614. doi: 10.24963/ijcai.2021/222.

[31] Z. Chen, D. Wang, and S. Yin, “Improving Cold-Start Recommendation via Multi-prior Meta-learning,” 2021, pp. 249–256. doi: 10.1007/978-3-030-72240-1_22.

[32] K. P. Neupane, E. Zheng, and Q. Yu, “MetaEDL: Meta Evidential Learning For Uncertainty-Aware Cold-Start Recommendations,” in *2021 IEEE International Conference on Data Mining (ICDM)*, IEEE, Dec. 2021, pp. 1258–1263. doi: 10.1109/ICDM51629.2021.00154.

[33] G. Duan, S. Xie, and Y. Du, “Study on a User Preference Conversational Recommender Based on a Knowledge Graph,” *Electronics (Basel).*, vol. 14, no. 3, p. 632, Feb. 2025, doi: 10.3390/electronics14030632.

[34] Y. Du, X. Zhu, L. Chen, Z. Fang, and Y. Gao, “MetaKG: Meta-Learning on Knowledge Graph for Cold-Start Recommendation,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 10, pp. 9850–9863, Oct. 2023, doi: 10.1109/TKDE.2022.3168775.

[35] Z. Shokrzadeh, M.-R. Feizi-Derakhshi, M.-A. Balafar, and J. Bagherzadeh Mohasefi, “Knowledge graph-based recommendation system enhanced by neural collaborative filtering and knowledge graph embedding,” *Ain Shams Engineering Journal*, vol. 15, no. 1, p. 102263, Jan. 2024, doi: 10.1016/j.asej.2023.102263.

[36] M. Ibrahim, I. S. Bajwa, N. Sarwar, F. Hajjej, and H. A. Sakr, “An Intelligent Hybrid Neural Collaborative Filtering Approach for True Recommendations,” *IEEE Access*, vol. 11, pp. 64831–64849, 2023, doi: 10.1109/ACCESS.2023.3289751.

[37] Y. Zhang, Z. Liu, and C. Sang, “Unifying paragraph embeddings and neural collaborative filtering for hybrid recommendation,” *Appl. Soft Comput.*, vol. 106, p. 107345, Jul. 2021, doi: 10.1016/j.asoc.2021.107345.

[38] Y. Afoudi, M. Lazaar, and M. Al Achhab, “Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network,” *Simul. Model. Pract. Theory*, vol. 113, p. 102375, Dec. 2021, doi: 10.1016/j.simpat.2021.102375.

[39] S. Bansal and N. Baliyan, “Remembering past and predicting future: a hybrid recurrent neural network based recommender system,” *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 12, pp. 16025–16036, Dec. 2023, doi: 10.1007/s12652-022-04375-x.

[40] O. Remadnia, F. Maazouzi, and D. Chefrour, “Hybrid Book Recommendation System Using Collaborative Filtering and Embedding Based Deep Learning,” *Informatica*, vol. 49, no. 8, Feb. 2025, doi: 10.31449/inf.v49i8.6950.

[41] M. M. Almeida, J. D. S. Almeida, D. B. P. Quintanilha, G. B. Júnior, and A. C. Silva, “A meta-learning based neural network and LSTM for univariate time series missing data imputation,” *Appl. Soft Comput.*, vol. 172, p. 112845, Mar. 2025, doi: 10.1016/j.asoc.2025.112845.

[42] I. ur Rehman, W. Ali, Z. Jan, Z. Ali, H. Xu, and J. Shao, “CAML: Contextual augmented meta-learning for cold-start recommendation,” *Neurocomputing*, vol. 533, pp. 178–190, May 2023, doi: 10.1016/j.neucom.2023.02.051.

[43] R. K. Mishra, H. Raj, S. Urolagin, J. A. A. Jothi, and N. Nawaz, “Cluster-Based Knowledge Graph and Entity-Relation Representation on Tourism Economical Sentiments,” *Applied Sciences*, vol. 12, no. 16, p. 8105, Aug. 2022, doi: 10.3390/app12168105.

[44] L. Wang, J. Luo, S. Deng, and X. Guo, “RoCS: Knowledge Graph Embedding Based on Joint Cosine Similarity,” *Electronics (Basel).*, vol. 13, no. 1, p. 147, Dec. 2023, doi: 10.3390/electronics13010147.

[45] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” *Adv. Neural Inf. Process. Syst.*, vol. 26, 2013.

[46] L. Zou *et al.*, “Neural Interactive Collaborative Filtering,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA: ACM, Jul. 2020, pp. 749–758. doi: 10.1145/3397271.3401181.

[47] H. Xia, Y. Luo, and Y. Liu, “Attention neural collaboration filtering based on GRU for recommender systems,” *Complex & Intelligent Systems*, vol. 7, no. 3, pp. 1367–1379, Jun. 2021, doi: 10.1007/s40747-021-00274-4.

[48] S. Liu, Y. Liu, X. Zhang, C. Xu, J. He, and Y. Qi, “Improving the Performance of Cold-Start Recommendation by Fusion of Attention Network and Meta-Learning,” *Electronics (Basel).*, vol. 12, no. 2, p. 376, Jan. 2023, doi: 10.3390/electronics12020376.

[49] Bahram Jannesar and Soroush Ghaderi, “Goodreads Book Datasets With User Rating 2M,” 2021, *Kaggle*. Accessed: Jun. 27, 2025. [Online]. Available: https://www.kaggle.com/datasets/bahramjannesarr/goodreads-book-datasets-10m

[50] J.-H. Cai, Y.-H. Tsai, C.-M. Chang, and S.-Y. Hwang, “Next Item and Interval Prediction of New Users Using Meta-Learning on Dynamic Network,” 2024, pp. 136–147. doi: 10.1007/978-981-97-2650-9_11.

[51] I. Marzuki, M. Hariadi, R. F. Rachmadi, and Y. M. Arif, “Neural Collaborative Filtering for Improved Tourism Destination Recommendation,” in *2024 8th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, IEEE, Aug. 2024, pp. 481–486. doi: 10.1109/ICITISEE63424.2024.10730542.

[52] M. A. Al Sabri, S. Zubair, and H. A. Alnuhait, “Improved prediction on recommendation system by creating a new model that employs Mahout collaborative filtering with content-based filtering based on genetic algorithm methods,” *Discover Artificial Intelligence*, vol. 6, no. 1, p. 20, Dec. 2025, doi: 10.1007/s44163-025-00678-y.

[53] O. Baker and Q. Yuan, “Machine learning: Factorization Machines and Normalized Discounted Cumulative Gain for Tourism Recommender System Optimisation,” in *2021 IEEE International Conference on Computing (ICOCO)*, IEEE, Nov. 2021, pp. 31–36. doi: 10.1109/ICOCO53166.2021.9673502.

[54] F. Rezaimehr and C. Dadkhah, “T&TRS: robust collaborative filtering recommender systems against attacks,” *Multimed. Tools Appl.*, vol. 83, no. 11, pp. 31701–31731, Sep. 2023, doi: 10.1007/s11042-023-16641-x.

[55] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, and R. Burke, “A Graph-Based Approach for Mitigating Multi-Sided Exposure Bias in Recommender Systems,” *ACM Trans. Inf. Syst.*, vol. 40, no. 2, pp. 1–31, Apr. 2022, doi: 10.1145/3470948.

[56] X. Liu, G. Wang, and M. Z. A. Bhuiyan, “Personalised context-aware re-ranking in recommender system,” *Conn. Sci.*, vol. 34, no. 1, pp. 319–338, Dec. 2022, doi: 10.1080/09540091.2021.1997915.

[57] J. Adamczak, Y. Deldjoo, F. B. Moghaddam, P. Knees, G.-P. Leyson, and P. Monreal, “Session-based Hotel Recommendations Dataset,” *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 1, pp. 1–20, Feb. 2021, doi: 10.1145/3412379.

[58] A. Akhadam, O. Kbibchi, L. Mekouar, and Y. Iraqi, “A Comparative Evaluation of Recommender Systems Tools,” *IEEE Access*, vol. 13, pp. 29493–29522, 2025, doi: 10.1109/ACCESS.2025.3541014.

[59] M. Kherad and A. J. Bidgoly, “Recommendation system using a deep learning and graph analysis approach,” *Comput. Intell.*, vol. 38, no. 5, pp. 1859–1883, Oct. 2022, doi: 10.1111/coin.12545.

[60] D. Wang, Y. Yih, and M. Ventresca, “Improving neighbor-based collaborative filtering by using a hybrid similarity measurement,” *Expert Syst. Appl.*, vol. 160, p. 113651, Dec. 2020, doi: 10.1016/j.eswa.2020.113651.

[61] T. Vente, L. Wegmeth, and J. Beel, “The Potential of AutoML for Recommender Systems,” in *Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization*, New York, NY, USA: ACM, Jun. 2025, pp. 371–378. doi: 10.1145/3708319.3734173.

[62] Y. Ariyanto, T. Widiyanigtyas, and I. A. E. Zaeni, “Enhancing Movie Recommendations: A Demographic-Integrated Cosine-KNN Collaborative Filtering Approach,” *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 6, pp. 791–803, Dec. 2024, doi: 10.22266/ijies2024.1231.60.

[63] P. Castells and D. Jannach, “Recommender Systems: A Primer,” *arXiv preprint arXiv*, vol. 2302.02579, Feb. 2023.

[64] R. Lai, R. Chen, and C. Zhang, “A Survey on Data-Centric Recommender Systems,” *arXiv preprint arXiv*, vol. 2401.17878, May 2024.