Mining Likes and Transactions per User for Cross-Domain Product Recommendation in Social Network and E-Commerce

Emmanuel Jojoe Ainoo[1], C. I. Ezeife*[1][0000−0002−9424−9223], and Abdulrauf Gidado[1][0000−0003−3013−1725]

University of Windsor, Onatrio N9B 3P4, Canada
https://cezeife.myweb.cs.uwindsor.ca {bandred,cezeife,gidado}@uwindsor.ca

Abstract. E-commerce recommendation accuracy can be improved by mining patterns from other domains such as e-commerce and social media (Facebook), to predict purchase behaviours. Existing systems such as the GaoLinRec23, GaoChenLin, WangZhaoRec, employ Collective Matrix Factorization to jointly factorize user-item interaction matrices from both domains have contributed deeply. However, the assumption that specific product details are shared between these domains does not align with the real-world scenario. Major e-commerce and social media platforms, such as Amazon and Facebook, typically do not exchange granular product information. This disconnect poses a critical obstacle for existing recommendation systems in providing accurate suggestions for users starting with no observable e-commerce activity. This paper proposes a system called Facebook Data Cross Recommendation ‘2023 (FD-CDR ’23), which uses the proposed MLTU (Mine Likes and Transactions per User) algorithm to extract Likes and purchase history of users from both domains, transforming them into itemsets. A modified association rule mining is then applied to uncover patterns of frequent co-occurrence between user Facebook post likes and e-commerce transactions as rules. It uses the proposed HARR (Hybrid Association Rule Recommendation) algorithm to match new user Facebook *likes* to, generate rules such as “Users who typically like cooking posts, buy cooking recipes” without needing to share products across platforms. Experimental results with precision and recall show that the proposed FD-CDR’23 system provides more accurate recommendations than the existing systems.

Keywords: Recommendation System · Cross-Domain Recommendation · Data Mining · Social E-commerce.

* This research was supported by the Natural Science and Engineering Research Council (NSERC) of Canada under an Operating grant (OGP0194134) and a University of Windsor grant.

1 Introduction

1.1 Background

In recent years, the lines separating e-commerce and social media networking have become less distinct. E-commerce platforms like Amazon[1] now incorporate several features commonly associated with social media networks, including real-time status updates and interactions between buyers and seller's media [13] Additionally, some e-commerce websites offer social login functionality, enabling new users to sign in with their credentials from popular social media platforms like Facebook [18], Instagram [8] or Twitter using tools like "Facebook Connect" [16]. E-commerce systems encompass a distinct component known as recommender systems (RS), aimed at predicting users' preferences and boosting sales [3]. These components employ advanced techniques like collaborative filtering and content filtering to predict users' potential purchases using user history [3]. Typically reliant on user history, these systems face a significant challenge when attempting predictions in the absence of any activity. Addressing this challenge, known as the "cross-site cold start," involves the transfer of activity or historical data from an alternate domain, such as social media [17]. This inter-domain transfer becomes instrumental in overcoming the hurdle of predicting user preferences when no observed activity exists in the e-commerce domain. To tackle the cross-site cold start problem, recommender systems make use of cross-domain techniques, including transfer learning or joint matrix factorization [4]. While online cross-domain product recommendation has been extensively studied before [3, 1, 10, 5, 7, 8, 14, 17], most of these studies focus on constructing solutions for specific e-commerce websites that have a database integration to allow shared user, products and transactional records or rely on transaction history confined to both domains. In most scenarios, the product data, which could include product details such as product names, IDs, categories, and other schema-related fields, are not shared across the two domains. The products themselves are not shared across the domains. This research aims to answer two questions: (1) How items be recommended to users with no historical activity by using historical activity from another domain? (2) How can user-item preferences be identified across domains without the platforms sharing any form of item data or schema structure?

1.2 Contributions

The main contributions of this paper are to: (i) Mine the frequent combinations of user purchases and likes to extract user-item preferences for recommendation without sharing item embeddings (products) across domains. (ii) Improve recommendation accuracy by predicting the purchase behaviours of users based on their Facebook likes activity. (iii) Provide extensive analysis of user purchasing patterns and Facebook activity logs of over 13,000 users and show a subset of Facebook features that correlate with purchase behaviours.

1.3 Problem Description

Given an e-commerce historical purchases, Facebook likes, and a set of users, the goal is to discover prevalent associations between new users' likes and purchases, sorting them based on confidence, and ultimately recommending items by prioritizing those associated with the highest-confidence rules. Let us define some terms: L is the set of user likes (from Facebook activity log). P is the set of e-commerce purchases. U is the set of users. R is the set of association rules, where each rule is $L_i \rightarrow P_j$, indicating a combination of likes leading to a purchase. Also, $Conf(L_i \rightarrow P_j)$ is the confidence level for the association rule $L_i \rightarrow P_j$. Now, the formula for the confidence level $Conf$ for an association rule $L_i \rightarrow P_j$ can be calculated as

$$Conf(L_i \rightarrow P_j) = \frac{Support(L_i \cup P_j)}{Support(L_i)} \tag{1}$$

To generate personalized recommendations for a new user, the following steps are followed: (1) For the new user's like L_{new} , find all association rules $L_i \rightarrow P_j$ that contain L_{new}. (2) Calculate the confidence level *Conf* for each of these rules. (3) Rank the rules by confidence level in descending order. (4) Return the purchased items P_j from the top-ranked rule as the recommended items for the new user. The formula to generate the top personalized recommendation for a new user is given as :

$Rnew = Top_{conf}(P_j for L_i \rightarrow P_j$ where $L_{new} \epsilon L_i)$

R_{new} is the top rule based on the confidence level for the new user's likes and for extracting the associated purchased items. $Top_{conf}(P_j for L_i \rightarrow P_j)$ selects the top-ranked items based on confidence from the association rules that involve the new user's like $L_{new}\{P_j\}$. This denotes the set of purchased items $P_j.L_i \rightarrow P_j$ represents association rules where L_i includes new user's like L_{new}.

2 Related Work

2.1 Data Mining

Data mining is a methodology designed to derive valuable insights from extensive datasets aligned with specific business objectives [9]. Given the notion of being "rich in data but lacking in information," data mining has garnered significant attention due to its pivotal role in transforming large datasets into meaningful information and knowledge [1]. The conventional data mining methodologies that aid in predictive analysis include Association Rule Mining which identifies patterns where the occurrence of one set of items implies the occurrence of another set of items [3], Classification which involves the process of categorizing data into predefined classes or groups based on identified patterns and features [15]. Lastly, Clustering involves grouping similar observation points into segments based on their inherent similarities [11]. This paper uses association rule mining to discover associations between users' likes and Facebook posts.

2.2 E-commerce Recommender Systems

In the realm of recommender systems, various approaches have been explored to enhance recommendation accuracy and address specific challenges. In [6] association rule algorithm for online e-commerce recommendation, incorporating item profitability into the recommendation process was introduced. Despite its potential to refine recommendations based on sales volumes, challenges such as defining optimal profit thresholds and continuous adjustment to market conditions were identified. The paper [11] tackled product recommendation through a hybrid of sequential rules and collaborative filtering, integrating RFM segmentation and association rule mining. Challenges in this approach include potential noise in RFM clustering and sensitivity to parameter choices. Additionally, [15] focused on custom data mining association rules for preventing Pottery product recommendations, highlighting challenges related to biases in user registration data and adapting to evolving user preferences. The paper [3] delved into mining sequential patterns for collaborative filtering, enhancing user-item rating matrices with sequential purchase history. However, inefficiencies in pattern extraction may impact matrix enhancement.

2.3 Cross Domain Recommender Systems

Cross-domain recommender systems [1, 2, 10, 4, 5, 14] aim to extend recommendation capabilities across different domains, presenting unique challenges and opportunities. The paper [12] addressed data sparsity issues in collaborative filtering through transfer learning, leveraging auxiliary data and applying sparse Singular Value Decomposition for coordinate system adaptation. Challenges here lie in the quality and relevance of auxiliary data. Authors in [1] proposed CD-SPM, a cross-domain recommendation algorithm integrating sequential pattern mining and rule mining from movies and books to e-commerce. Despite its potential, challenges include accurately recommending to new users with minimal historical data. In [7], cross-platform item recommendation for online social e-commerce, utilizing Collective Matrix Factorization and Social Matrix Factorization, was explored. Challenges include the need for shared product item data across domains and the sparsity of user-item interactions. Then, [8] introduced neural collaborative filtering with graph regularization for cross-domain social recommendation, facing similar challenges related to shared product item data and sparsity in user-item interactions.

3 PROPOSED FACEBOOK-DATA CROSS DOMAIN RECOMMENDER

This paper proposes a way to extract and transform user product interests from their Facebook likes to understand what products they may like to buy based on their likes. It means something when a lot of users who liked a specific post A, also buy a specific product B. Once the activities on both platforms can be transformed into itemsets and discover frequent occurrences of activities, the need to have the same product on both platforms is eliminated, essentially solving this paper's main challenge. For example, if it is discovered that likes on post A always lead to product B, product B can be recommended to users who eventually like post A without needing to directly have product B on the social media platform to model a representation of user's interests in product B.

Algorithm 1 Algorithm1: FD-CDR (Facebook Data Cross-Domain Recommendation)

Input : minimum support (s), minimum confidence (c), historical Facebook activity log (F), historical purchase database (PDB), Facebook to e-commerce user mapping (FP), new user like (l)

Output : Top recommended items ranked by confidence level.

Intermediates : Itemsets per User (IPDB), Association Rules (AR), Confidence level Rule Set (CRS), number of recommendations (n)

```
Generate Itemsets per User, u(IPDB) ← MLTU(F,PDB,FP) as in Algorithm 2 in section 3
Generate Association rules (AR) ← ModifiedApriori(s, c, IPDB) as in Algorithm 3 in section 3
for each rule r in Association Rules (AR) do
    if r antecedent contains new user like l then
        Confidence Rule Set (CRS) = calculate conf (r)
    end if
end for
sort Confidence Rule Set (CRS) by descending order
return Confidence Rule Set for n recommendations (CRS)[:n] consequent
```

[1]

The steps below highlights how FD-CDR works.

Step 1: *Generate all historic purchases per User*

Extract user e-commerce IDS and corresponding product IDS from sample Historical Purchase Database (PDB) present in Table 1 using step 1 procedure in Algorithm 2.

Table 1. Sample Historical Purchase Database (PDB)

userID	transactionID	productID	timestamp
u1	1001	p1	2023-01-15T08:30:00
u2	1002	p2	2023-01-16T10:45:00
u3	1004	p4	2023-02-10T14:00:00
u2	1005	p1	2023-02-18T09:10:00
u4	1006	p5	2023-03-05T11:45:00
...	...	...	...
u1	1011	p6	2023-05-20T08:45:00

Following the procedure in step 1 of algorithm 2, the following steps are followed to retrieve purchases of each user. (1) Create an empty dictionary data structure of Product Database User (PDBU), to store user e-commerce IDs (userID) from Table 1 and associated product IDs for each user transaction in Table 1. (2) For each entry in the original data structure PDB in Table 1, retrieve the userID and productID. (3) Check if the userID is not in the set of known users. If the user is new, initialize an empty list for that user in PDBU. (4) Append the productId to the list associated with the userId in PDBU. Return Result: After processing all entries in PDB, return the resulting PDBU dictionary containing user e-commerce IDs and associated product IDs. PDBU =extracted_ecommerce_user_products(PDB)

Output: = PDBU = { 'u1': ['p1', 'p3', 'p2', 'p4', 'p6'], 'u2': ['p2', 'p1', 'p5', 'p3'], 'u3': ['p4', 'p3', 'p5'], 'u4': ['p5', 'p2'] }

Step 2: *Generate E-commerce to Facebook userID mapping* For each e-commerce user, retrieve their Facebook UserID from Facebook to e-commerce mapping (FP) in Table 2 using step 2 procedure in Algorithm 2

Table 2. Facebook and E-commerce UserID Mapping

userEcommerceID	userFacebookID
u1	1041851b-87db-4054-b56a-d1e4c280f39b
u2	0844682d-6e20-4f92-86e9-a0c60ee81eb2
u3	3b404f36-7ed0-44d2-9a57-e3980e433825
u4	0a175d58-bcac-40ec-b1e8-a89ae079f945
u5	1b2c3d4e-5f6g-7h8i-9j0k-a1b2c3d4e5f6

Following the procedure in step 2 of Algorithm 3, the following steps are followed (1) Create an empty dictionary of Facebook Mapped User(FMU) to store the mapping of Facebook IDs to ecommerce user IDs. (1) For each entry in the original data structure FP in Table 2, extract the userEcommerceID and userFacebookID. (2) Assign the userEcommerceID to the key userFacebookID in the FMU dictionary. (3) After processing all entries in FP, return the resulting FMU dictionary containing the mapping of Facebook IDs to e-commerce user IDs, as shown below. # Output: FMU = { '1041851b-87db-4054-b56a-d1e4c280f39b': 'u1',
'0844682d-6e20-4f92-86e9-a0c60ee81eb2': 'u2',
'3b404f36-7ed0-44d2-9a57-e3980e433825': 'u3',
'0a175d58-bcac-40ec-b1e8-a89ae079f945': 'u4',
'1b2c3d4e-5f6g-7h8i-9j0k-a1b2c3d4e5f6': 'u5'
}

Step 3: *Generate liked posts per User*

Use the Facebook User IDs to extract liked posts from the Facebook activity log (F) using the procedure in step 3 of Algorithm 3.

Following the procedure in step 3 of Algorithm 2, the following steps are defined: (1) Create an empty dictionary for Facebook Likes Data (FDL) to store liked posts organized by e-commerce user IDs. For each post in the data structure, extract the userFacebookID from the likes data of the post. (2) Check if the userFacebookID is present in the mapping FMU. (3) If userFacebookID is found in FMU, retrieve the corresponding userEcommerceID. (4) Check if the userEcommerceID is not in FDL. If userEcommerceID is not in FDL, initialize an empty list for that user in FDL and append the post to the list associated with userEcommerceID in FDL. (5) Return Result: After processing all posts in F, return the resulting FDL dictionary containing liked posts organized by e-commerce user IDs.

Step 4: C*reate Itemsets of activities (Likes and Purchases) per User*: Create a transactional table by combining outputs from step 3 (FDL) and step 1 (PDBU) to produce a set of items per user.

1. Create an empty list IPDB to store transactional tables as itemsets per user.
2. For each userId and associated products in the dictionary PDBU items, retrieve the liked posts (likes) for the current userId from FDL, or an empty list if the userId is not present in FDL.
3. Combine the user's products and liked posts to create combinedItems.
4. Append a dictionary with keys 'UserID' and 'Items' to IPDB, where 'UserID' is the current userId and 'Items' is the combined list of products and likes.
5. Return Result: After processing all user-product combinations, return the resulting IPDB list containing transactional tables as itemsets per user, as shown below:

The resulting output (IPDB), which is the set of items per each user is shown below:

Output: IPDB = transactional_data = [
$[‘p1', ‘p3', ‘p2', ‘p4', ‘p6', ‘post1', ‘post2'],$
$[‘p2', ‘p1', ‘p5', ‘p3', ‘post1', ‘post3'],$
$[‘p4', ‘p3', ‘p5', ‘post2'],$
$[‘p5', ‘p2', ‘post1']$
]

In the resulting outputs, the datatypes are attached to each item in IPDB, which denotes if they are posts or products. To find frequent combinations of which liked posts lead to product purchases to tackle the cold start on e-commerce. It looks more like the output below in implementation:

transactional_data = [{‘UserID’: ‘u1’, ‘Items’: [{‘p1Id’: ‘p1’, ‘Type’: ‘Product’}, {‘post1Id’: ‘post1’, ‘Type’: ‘Post’}, {‘post2Id’: ‘post2’, ‘Type’: ‘Post’}]}, {‘UserID’: ‘u2’, ‘Items’: [{‘p2Id’: ‘p2’, ‘Type’: ‘Product’}, {‘post3Id’: ‘post3’, ‘Type’: ‘Post’}]}, {‘UserID’: ‘u3’, ‘Items’: [{‘p3Id’: ‘p3’, ‘Type’: ‘Product’}, {‘post1Id’: ‘post1’, ‘Type’: ‘Post’}, {‘post2Id’: ‘post2’, ‘Type’: ‘Post’}, {‘post4Id’: ‘post4’, ‘Type’: ‘Post’}]}, {‘UserID’: ‘u4’, ‘Items’: [{‘p4Id’: ‘p4’, ‘Type’: ‘Product’}, {‘post2Id’: ‘post2’, ‘Type’: ‘Post’}]}]

Algorithm 2 (i.e., MTLU: Mine Likes and Transactions Per User) combines all these steps to achieve the itemsets per user.

Step 5: *Generate frequent itemsets and Association Rules* After retrieving itemsets per User (IPDB), frequent itemsets and association rules are generated using the Modified Apriori Algorithm presented in Algorithm 3

Step 5a: *Initialize:* Generate 1-itemsets (C1) from unique items in the transactional data (IPDB) shown here:

Algorithm 2 Algorithm 2: MLTU (Mine Likes and Transactions Per User)

Input: historical Facebook activity log (F), historical purchase database (PDB), Facebook to e-commerce user mapping (FP)

Output: Itemsets per User (CRS)

Intermediates: Historical Purchase per User (PDBU), Facebook to E-commerce unique Identifier Mapping per User (FMU), Facebook likes per E-commerce user (FDL), Itemsets per User (IPDB)

```
Extract user ecommerce IDS and product IDs from PDB, PDBU = {}
for each in PDB: do
    userId = each(userId), productId = each(productId)
    if userId not in FP(users) then
        PDBU [userId] = [ ] # initialize empty list
    end if
    PDBU[userId] append productId.
end for
return PDBU.
for each e-commerce user, retrieve Facebook ID from FP, FMU = {}  do
    for each in FP do
        userEcommerceID = each(userEcommerceID)
        userFacebookID = each(userFacebookID)
        FMU[userFacebookID] = userEcommerceID
    end for
end for
return FMU
Use Facebook ID to extract liked posts from F, FDL = {}
for each post in F[data] do
    userFacebookID post['likes']['data']['user']
    if userFacebookID is in FMU then
        userEcommerceID = FMU[userFacebookID]
        if userEcommerceID not in FDL then
            FDL[userEcommerceID] = [ ]
        end if
        append post to FDL[userEcommerceID]
    end if
end for
return FDL
Create transactional table as itemsets per user, IPDB=[ ]
for userId, products in PDBU items do
    likes = FDL.get(userId, [ ])
    combinedItems = products + likes
    append ('UserID': userId, 'Items':combinedItems) to IPDB.
end for
return IPDB.
```

[1]

Output: IPDB = transactional_data = [
['p1', 'p3', 'p2', 'p4', 'p6', 'post1', 'post2'],
['p2', 'p1', 'p5', 'p3', 'post1', 'post3'],
['p4', 'p3', 'p5', 'post2'],
['p5', 'p2', 'post1']
]
Calculate support for each 1-itemset. C1 = [['p1'], ['p2'], ['p3'], ['p4'], ['p5'], ['p6'], ['post1'], ['post2'], ['post3'], . . .]
Calculate support for each 1-itemset L1 = [['p1'], ['p2'], ['p3'], ['post1'], ['post2'], . . .]
Iterate ($k = 2, 3, \ldots$): Join frequent (k-1)-itemsets to generate candidate k-itemsets (Ck). Calculate support for each candidate k-itemset. Prune candidate k-itemsets that don't meet the minimum support.
k = 2 C2 = [['p1', 'p2'], ['p1', 'p3'], ['p1', 'post1'], ['p1', 'post2'], ['p2', 'p3'], . . .]
Calculate support for each candidate 2-itemset. L2 = [['p1', 'p2'], ['p1', 'post1'], ['p1', 'post2'], ['p2', 'post1'], ['p2', 'post3'], . . .]
k = 3 C3 = [['p1', 'p2', 'post1'], ['p1', 'p2', 'post2'], . . .]
Calculate support for each candidate 3-itemset. L3 = [['p1', 'p2', 'post1'], . . .]

Step 5c: *Stop when no frequent k-itemsets can be generated.* Frequent itemsets (L_k) for $k = 1, 2, 3, \ldots$

frequent_itemsets = [
[{'p1Id': 'p1', 'Type': 'Product'}, {'p2Id': 'p2', 'Type': 'Product'}],
[{'p1Id': 'p1', 'Type': 'Product', 'post1Id': 'post1', 'Type': 'Post'}],
[{'p1Id': 'p1', 'Type': 'Product', 'p2Id': 'p2', 'Type': 'Product', 'post1Id': 'post1', 'Type': 'Post'}]]

Step 5d: *Association Rule Generation* Input: Frequent itemsets (Lk) from Procedure 3 in step 2

Minimum confidence (min_confidence): 0.6 Generate association rules from frequent itemsets and filter out rules where the potential antecedents are posts to obtain Posts => Products. Association Rules: (L1 is not used for rules)

Rules from L2:
- [$`p1'$] => [$`p2'$], Confidence:0.6
- [$`p2'$] => [$`p1'$], Confidence: 0.7
- . . .

Rules from L3:
- [$`p1', `p2'$] => [$`post1'$], Confidence: 0.9
-[$post1$] => [$p2$], confidence 0.9
- . . .

Output: Association rules with confidence for each rule.

association_rules (AR) = [
{'Antecedent': [{'p1Id': 'Post1', 'Type': 'Post'}], 'Consequent': [{'p2Id': 'p2', 'Type': 'Product'}], 'Confidence': 0.9},]

What the rule generated above $post1 => p2$ implies is that most of the time, users who liked post1 purchased p2, and thus, p2 can be recommended to a new user who likes post1, provided the confidence level is high. Here is the modified Apriori algorithm to achieve step 5 above.

Algorithm 3 :Modified Apriori (*Posts* => *Products*)

Input: minimum support (s), minimum confidence (c), Itemsets per User (IPDB)
Output: Association Rules (AR)

Initialize frequent_itemsets with frequent itemsets of size 1.
Set k to 2.
while frequent_itemsets[k-2] is not empty **do**
    Generate candidate_itemsets by joining frequent_itemsets[k-2].
    Prune candidate_itemsets that do not meet min_support.
    Add remaining candidate_itemsets to frequent_itemsets.
    Increment k
end while
Generate association_rules:
for each frequent_itemset in frequent_itemsets **do**
    Generate all possible non-empty subsets of the itemset.
    for each subset **do**
        if the subset contains only items of type “Post" **then**
            consider it as a potential antecedent.
        end if
        if the remaining items in the itemset (excluding the subset) are of type "Product," **then**
            consider them as potential consequent.
        end if
        Calculate confidence for the rule (antecedent implies consequent).
        if confidence exceeds the min_confidence threshold **then**
            Add the rule to the list of association_rules
        end if
    end for
end for
Return association_rules (AR)

The goal here is to solve the cross-site cold start problem where users may have low activity on e-commerce. Thus, the aim is to generate rules where actions on social media lead to e-commerce purchases, such that when new users come in with no e-commerce history, their social media history can be leveraged. Algorithm 3 considers the data types of the items in the Itemsets: Posts or Products, so the rules for where posts imply products can be filtered.

Step 6: *Generate Recommendations*

Using HARR (Hybrid Association Rule Recommendation) Algorithm 3 to generate personalized recommendations based on association rules (AR) derived from step 5. The new user like (l) is considered to be post1. Using the steps in Algorithm 3 (1) Going through the association rules (AR) in step 5d to retrieve all rules that contain post1 $[post1] => [p2]$ (2) Calculate the confidence level of this rule, to be 0.9. (3) There is only one rule with post1; sorting by confidence level, resulting in $[post1] => [p2]$, on top of the list in the ranking (4) Hence, p2 is extracted as the item to be recommended to the user.

Algorithm 4 Algorithm 4: HARR (Hybrid Association Rule Recommendation)

Input: Association rules (AR), new user like (l)
Output: Top recommended items ranked by confidence level
Intermediates: Rule Set (CRS), number of recommendations (n)

```
for each rule r in (AR) do
    if r antecedent contains l then
        (CRS) ← calculate conf(r)
    end if
end for
sort (CRS) by descending order
return (CRS)[:n] consequent
```

4 Experiment, Result and Analysis

To evaluate the system, the Facebook-eBay dataset of real Facebook activity and e-commerce purchases of anonymous users on a study conducted by eBay [16] is used. This comprised of 13,619 randomized eBay users who anonymously signed up using their Facebook accounts between June and August 2012. The distribution of this dataset is presented in Table 3.

Table 3. Example User Information of Facebook-eBay Dataset

Name	Anonymous
Gender	Male
Age Group	35-44
Facebook Likes	Beatles (Musician/band) iPhone 5 (Electronics) Starbucks (Food/beverage) Walt Disney Studios (Movie)
eBay Purchases	iPhone 4S (Electronics) Beatles T-shirt (Clothing) Beatles Mug (Collectibles)

From Table 3, it can be observed that the user's interests are similar on both platforms, where they like Beatles music on Facebook and buy their shirts on eBay. This denotes that a lot of rich information can be used from user activity on social media to predict purchase behaviour and interests. Table 4 shows the statistics of the dataset used for experiments.

4.1 Evaluation Metrics

The evaluation of this work focuses on three key metrics: precision, recall, and F-score, which are used in existing benchmark systems.

Table 4. Facebook-eBay Dataset Summary Statistics

Users	13,619
Facebook Likes	4,165,690
Facebook pages	1,373,984
eBay purchases	628,753

4.2 Experimental Setup

(1) The dataset used for testing FD-CDR '23 is compatible with comparable systems. Many existing e-commerce recommendation systems would not be able to test on such data because of the integration of the two domains to share product details. (2) The experiments were run a series of over 10 times to get a range of values for precision and recall. The highest recorded within the range of the 10-time run was used to evaluate the system. (3) The systems compared with FD-CDR'23 include GaoLinRec23[7], WangZhoaHeLi15 [14], and Zhang-Penacchiotti15[17].

4.3 Results and Discussion

The first step is to run MLTU on the dataset to retrieve a usable data format for the recommender system. The e-commerce historical datacontains UserIDs and purchased product IDs as extracted using MLTU of this paper. In this data, we are interested in what purchases users have had.

The proposed MLTU converts the dataset into itemsets of per user as explained in section 3, which contains all the IDS of purchased products and liked pages per user.

The data was then split into 80% for training and 20% for testing, which were used for evaluating the systems. The proposed FD-CDR'23 system is then experimented with different numbers of users ranging from 10 to 1000 and then compared to existing systems on recall, precision and F1 as shown in Table5 below:

Table 5. Evaluation Results from Experiments on Tested Systems

Recommender System	**Number of Users**	**Precision**	**Recall**	**F1Score**
GaoLinRec23	100	0.222	0.21	0.225
	1000	0.216	0.19	0.204
WangZhoaHeLi15	100	0.104	0.07	0.067797
	1000	0.120	0.03	0.075000
ZhangPenacchiotti15	100	0.6	0.7	0.647059
	1000	0.4	0.63	0.480000
FD-CDR'23	100	0.61	0.54	0.697248
	1000	0.57	0.44	0.682927

The precision run times for FD-CDR exhibit some variability, with values ranging between 0.52 and 0.61. This indicates that the precision scores are not static and can vary across different runs. The system might be sensitive to certain conditions or user interactions, leading to fluctuations in the recommendation accuracy. While there is variability, there is a pattern of relatively stable precision around the range of 0.58 to 0.61 in several consecutive runs. This stability suggests that, under certain conditions or datasets, FD-CDR consistently performs well in maintaining high precision. This is a positive sign as it indicates the system's robustness and reliability in delivering accurate recommendations for 100 users. The run times present opportunities for optimization. If the goal is to achieve even higher precision or reduce variability, further fine-tuning of the recommender algorithm, parameter adjustments, or incorporating additional features might be considered.

Figure 1, 2, and 3 show the tested systems' precision, recall and F1 score run graphs.

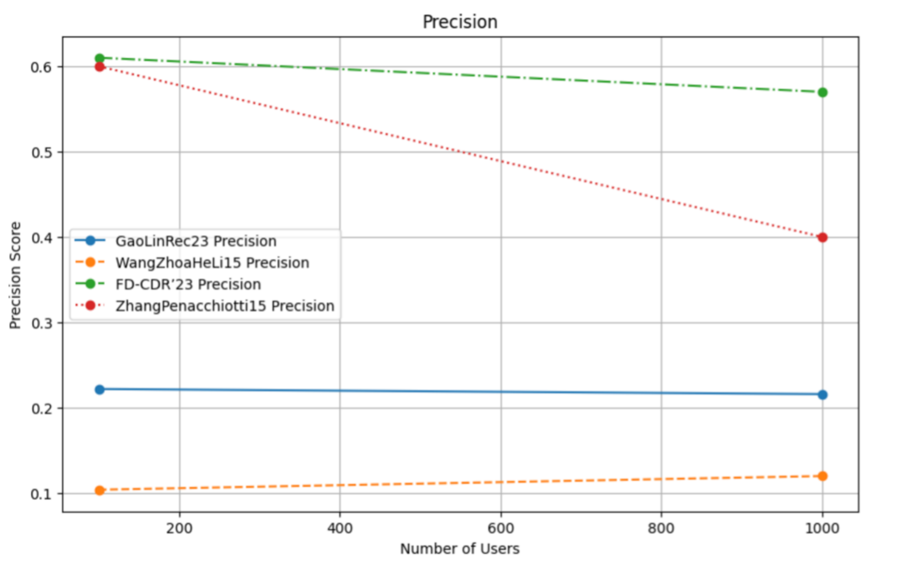

Fig. 1. Precision Results for Tested Systems Compared to FD-CDR'23

From the results analysis as shown in Figures 1, 2 and 3, it is observed that FD-CDR '23 performs better in accurately recommending items for 100 users and more [7] exhibits a moderate precision, hovering around 22.2% for 100 users and slightly decreasing to 21.6% as the user base expands to 1000. This suggests that, on average, about 22% of the recommended items are relevant to the user's interests. While the system provides reasonably accurate recommendations, there is room for improvement. [14] on the other hand, starts with a precision of 10.4% for 100 users, but interestingly, precision increases to 12% with a larger user base of 1000. This improvement might indicate that the system benefits from a larger dataset, leading to more accurate recommendations

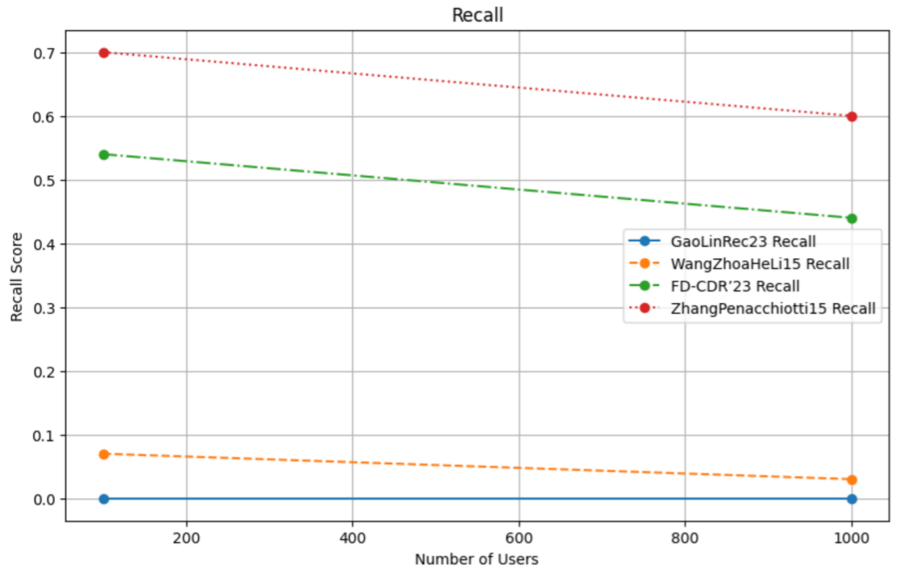

Fig. 2. Recall Results for Tested Systems Compared to FD-CDR‘23

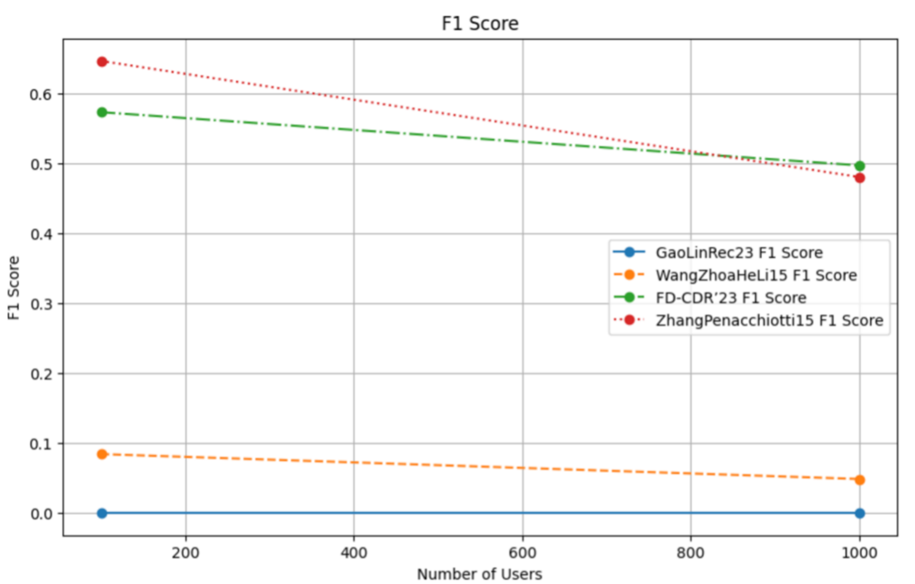

Fig. 3. F1 Score Results for Tested Systems Compared to FD-CDR‘23

for a broader user population. FD-CDR‘23 stands out with a high precision of 61% for 100 users, showcasing its effectiveness in providing accurate recommendations. Impressively, the precision remains robust even as the user base expands to 1000 users, maintaining a precision of 57%. This suggests that FD-CDR'23 excels in accurately identifying relevant items, making it a commendable choice for recommendation tasks.

5 CONCLUSION

This research presents a novel approach to enhancing cross-domain recommendation systems through the integration of social media and e-commerce data. The proposed system demonstrates the potential of utilizing patterns to capture hidden associations between user activities in different domains, leading to more accurate and personalized recommendations. The experimental evaluation showcases the effectiveness of the developed system in terms of precision and recall. Experimental results on DF-CDR'23 show that users' interests are similar in these two domains and answered the research question of recommending items to users on the e-commerce platform with no activity by mining activity from the social media domain. In conclusion, the FD-CDR'23 recommender system stands out as a robust and efficient solution for making accurate recommendations. Its superior precision and commendable recall, coupled with scalability, position FD-CDR'23 as a promising choice for practical implementation in various recommendation scenarios. The findings of this study provide valuable guidance for researchers, developers, and businesses seeking effective recommender systems, emphasizing the noteworthy performance of FD-CDR'23 in delivering accurate and relevant recommendations to users. However, there are several avenues for future exploration in this domain. (1) One major setback of the proposed algorithm is answering what happens when a new user likes a post that has not been liked before. The system would not be able to generate recommendations for unknown posts. A good direction in the future is to explore ways to handle scenarios of users liking unknown posts and how else such users' preferences can be discovered for product recommendation. (2) One significant direction is to explore the reverse scenario, where social media and e-commerce domains are exchanged, allowing recommendations to flow from the e-commerce domain back to social media or even between both domains simultaneously. This two-way recommendation system could potentially uncover even more nuanced user preferences and behaviours across platforms, leading to more comprehensive and context-aware recommendations. (3) Additionally, further research could focus on adapting the proposed approach to different types of cross-domain recommendation scenarios, such as music and movie recommendations or news and entertainment recommendations, thereby broadening its applicability and impact.

References

1. Anwar, T., Uma, V.: Cd-spm: Cross-domain book recommendation using sequential pattern mining and rule mining. Journal of King Saud University - Computer and Information Sciences **34**(3), 793–800 (2022). https://doi.org/10.1016/j.jksuci.2019.01.012

2. Bahaweres, R.B., Almujaddidi, A.R.: Improving recommender systems performance with cross-domain scenario: Anime and manga domain studies. In: 2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI). pp. 361–365 (2022). https://doi.org/10.23919/EECSI56542.2022.9946560
3. Bhatta, R., Ezeife, C.I., Butt, M.: Mining sequential patterns of historical purchases for e-commerce recommendation. In: International Conference on Big Data Analytics and Knowledge Discovery. pp. 57–72 (2019)
4. Cantador, I., Fernández-Tobías, I., Berkovsky, S., Cremonesi, P.: Cross-domain recommender systems. In: Recommender Systems Handbook, pp. 919–959. Springer US (2015). https://doi.org/10.1007/978-1-4899-7637-6_27
5. Chen, J., WU, C.: The design of cross-border e-commerce recommendation system based on big data technology. In: 2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP). pp. 381–384 (2021). https://doi.org/10.1109/ICSP51882.2021.9409014
6. Dai, J., Zeng, B.: An association rule algorithm for online e-commerce recommendation service. Journal of Economics, Business and Management **4**(10), 573–576 (2016). https://doi.org/10.18178/joebm.2016.4.10.454x
7. Gao, C., Chen, X., Feng, F., Zhao, K., He, X., Li, Y., Jin, D.: Cross-domain recommendation without sharing user-relevant data. In: Proceedings of the World Wide Web Conference (WWW). pp. 491–502 (2019)
8. Gao, C., Lin, T.H., Li, N., Jin, D., Li, Y.: Cross-platform item recommendation for online social e-commerce. IEEE Transactions on Knowledge and Data Engineering **35**(2), 1351–1364 (2023). https://doi.org/10.1109/TKDE.2021.3098702
9. Han, J., Pei, J., Tong, H.: Data Mining: Concepts and Techniques. Elsevier Science (2022), https://books.google.ca/books?id=NR1oEAAAQBAJ
10. Li, B.: Cross-domain collaborative filtering: A brief survey. In: International Conference on Tools with Artificial Intelligence. pp. 1085–1086 (2011)
11. Liu, D.R., Lai, C.H., Lee, W.J.: A hybrid of sequential rules and collaborative filtering for product recommendation. Information Sciences **179**(20), 3505–3519 (2009)
12. Pan, W., Xiang, E.W., Liu, N.N., Yang, Q.: Transfer learning in collaborative filtering for sparsity reduction. In: Proceedings of the 10th IEEE International Conference on Data Mining (ICDM). pp. 956–961 (2010)
13. Safko, L., Brake, D.K.: The social media bible: Tactics, tools, and strategies for business success. John Wiley & Sons (2009)
14. Wang, J., Zhao, W., He, Y., Li, X.: Leveraging product adopter information from online reviews for product recommendation. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 9, p. Article 1 (2015). https://doi.org/10.1609/icwsm.v9i1.14585
15. Yang, J., Chen, H., Xiong, S., Yang, Z., Jiang, Y.: Custom data mining association rules of nixing pottery product recommendation system. In: Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering. pp. 89–93 (2021). https://doi.org/10.1145/3443467.3443734
16. Zhang, Y., Pennacchiotti, M.: Recommending branded products from social media. In: Proceedings of the 7th ACM Conference on Recommender Systems. pp. 77–84 (2013). https://doi.org/10.1145/2507157.2507170
17. Zhao, W.X., Li, S., He, Y., Chang, E.Y., Wen, J.R., Li, X.: Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. IEEE Transactions on Knowledge and Data Engineering **28**(5), 1147–1159 (2016). https://doi.org/10.1109/TKDE.2015.2508816