

IMPACT-P: New Approach for Bundle Recommendation Using Implicit Constraints and Sentiment-Based Reranking

Andy Maulana Yusuf, Adiwijaya*, Agung Toto Wibowo, and Z. K. A. Baizal

Abstract

Constraint-based bundle recommendations are typically derived from explicit user requirements, often resulting in bundles with low or even nonexistent diversity. Consequently, recent studies have shifted their focus toward enhancing diversity in recommendations. However, this approach tends to compromise recommendation relevance. To address this issue, this study proposes a novel framework called IMPACT-P (implicit matching and constraint tuning for personalization), which leverages implicit user preferences and matrix sentiment-based reranking of bundles. The proposed framework significantly improves recommendation relevance, achieving up to a 300% increase in hit ratio, while diversity decreases by only 0.6% compared to the baseline framework (MF_ALL).

Keywords

Bundle Recommendation, Bundle Matching, Constraint Features, Implicit Preference, Reranking

1. Introduction

A bundle is a collection of interrelated items packaged to enhance user experience and boost sales [1]. However, service providers often offer a limited number of bundles, prompting the development of bundle recommendation systems to match available bundles with user preferences, a process known as bundle matching. A common approach involves using constraint features like price, quantity, distance, and time [2], which are typically explicitly defined based on user requirements [3]. This can lead to recommendations with low diversity, prompting research on increasing diversity [4]. However, excessive diversity can reduce relevance, especially for users with specific preferences [5]. Additionally, user's limited domain knowledge often results in generalized queries, such as searching for the lowest price, making constraint-based recommendations more challenging.

This study introduces a new bundle recommendation framework that derives constraint features from implicit user preferences. These preferences include user histories (interactions with individual items) and user bundle histories (interactions with bundles). The framework identifies similarities between these preferences and item characteristics to create implicit features, which are then processed by a binary classification model to generate recommendation constraints. Additionally, a reranking process based on a sentiment attribute matrix ensures that the recommended bundles remain highly relevant.

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received February 27, 2025; first revision May 12, 2026; accepted May 20, 2026.

*Corresponding Author: Adiwijaya (adiwijaya@telkomuniversity.ac.id)

School of Computing, Telkom University, Bandung, West Java, Indonesia (aitektele@student.telkomuniversity.ac.id, adiwijaya@telkomuniversity.ac.id, agungtoto@telkomuniversity.ac.id, baizal@telkomuniversity.ac.id)

Overall, our research contributions are summarized in two key points:

- We propose a novel framework that leverages the similarity between user's implicit preferences and item characteristics to generate implicit features. These features serve as the basis for providing more relevant recommendations without compromising diversity.
- We developed a reranking method that utilizes a sentiment attribute matrix of items within bundles. Sentiments are converted into a value range from 1 to -1, encompassing 18 types of sentiments obtained. The reranking process involves ordering bundles based on the total sentiment scores of the items within each bundle.

2. Related Work

Research on bundle matching has evolved by incorporating user preferences and constraint features to generate personalized recommendations. Early studies, such as [6], considered financial travel costs and time alongside tourist interests for realistic recommendations. Similarly, [7] proposed a multi-agent approach where users set initial preferences, and the system filters information from hotels, flights, and attractions to provide tailored tour recommendations. A graph-based method introduced by [8] demonstrated the potential of designing travel packages that account for users' spatio-temporal constraints, while [9] utilized check-in data from social networks to recommend travel routes based on user preferences and time limitations.

Despite its advantages, the constraint-based approach faces challenges in diversity and scalability. As noted by [10], bundle recommendations are NP-hard, especially when optimizing for profit. Additionally, [11] found that strict query criteria can limit options, leading to low diversity or no recommendations. To address this, [12] suggested relaxing constraints to capture more unstructured user preferences, though this often reduces recommendation relevance. Recent studies [13] introduced dynamic constraints, while [14] explored diverse user preferences and data variability. Optimization techniques like submodular optimization and the ant colony optimization algorithm [15] show promise but still suffer from data bias, favoring popular items over potentially relevant niche options. To overcome these issues, our research proposes IMPACT-P (implicit matching and constraint tuning for personalization), which leverages users' interaction histories to construct implicit user profiles. By matching these profiles with item characteristics in bundles, IMPACT-P enhances recommendation relevance while maintaining diversity, ensuring better alignment with user needs.

3. Research Methodology

Fig. 1 illustrates the proposed research method, which comprises five stages. In this section, we will describe each stage of the method in detail.

3.1 Steam Dataset

This study utilizes the Steam Video Game and Bundle Data, which is publicly available through this website (<https://cseweb.ucsd.edu/~jmcauley/datasets.html>) [16-19]. The dataset contains reviews from the Steam video game platform and information about games grouped into bundles. The key statistics of the dataset are as follows: users write reviews (25,799), users has games (88,310), Steam games (32,135), and Steam bundles (615).

This stage begins by preparing the Steam dataset, which includes four main entities:

- 1) Set of Bundles (B), a collection of bundles with attributes such as bundle name, bundle price, and items (a list of items).
- 2) Set of Items (I), a collection of items with attributes such as app name, sentiment, tags, price, and specs.
- 3) User interactions histories (UH), user interaction records consisting of items they have interacted with.
- 4) User bundle histories (UBH), Records of bundles that users have accessed. These four entities form the foundation for building a recommendation system using the bundle matching approach.

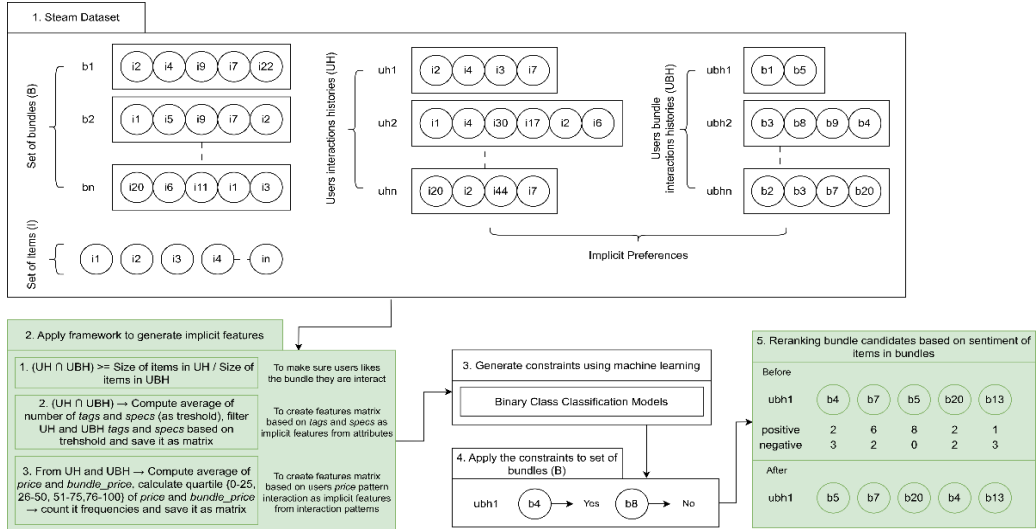


Fig. 1. The proposed framework is based on implicit preference and item characteristics. Stage which is green is our contribution.

3.2 Apply Framework to Generate Implicit Features

To construct the proposed implicit features, we calculate the similarity between UH and UBH. These similarities involve computing the number of similar items, similarities in types and attributes such as tags and specs, purchasing patterns, and item price groups. Overall, the creation of these implicit features consists of three main processes as follows.

3.2.1 Finding the bundle users actually likes

This process involves calculating the number of overlapping items (intersection) between UH and UBH. The calculation aims to identify the correlation between user's interaction histories and the bundles they have interacted with.

Based on our dataset and statistical analysis, it is observed that bundles with fewer than 5 items typically share only one similar item between UH and UBH. Conversely, for bundles containing 5 or more items, at least 3/4 of the items exhibit similarity between UH and UBH, as calculated using Eq. (1):

$$ULB = \frac{\text{Size of items in UH}}{\text{Size of items in UBH}} \quad (1)$$

where ULB represents the number of similar items between UH and UBH.

Based on this phenomenon, we established a statistical rule to identify users as threshold candidates, which is then used to construct the feature matrix in the subsequent stage. This process ensures that the recommended bundles are truly relevant to the user's preferences.

3.2.2 Creating implicit features matrix based on attributes

After identifying the threshold candidates, the next step is to determine additional reasons that might drive users to interact with a bundle. At this stage, we calculate the frequency of the tags and specs attributes of items within UH and UBH. These attributes are selected because they are categorical and easily recognizable by users. For example, users who prefer games tagged as “Adventure” and specified as “Mobile” are more likely to choose bundles with similar attributes, either due to their preferences or device limitations.

After calculating the frequency for each “tags” and “specs,” the average frequency is used as the second threshold to separate attributes that are favored by users from those that are less favored. The favored attributes are included in a checklist matrix, which is then utilized as features. This process results in 204 attributes for tags and 21 attributes for specs. The checklist matrix is applied to all bundles B to generate a uniform input for the binary classification model, which determines the constraints. The outcome represents the first variation of the framework based on implicit attributes.

3.2.3 Creating implicit features matrix based on interaction patterns

This process focuses on user interaction patterns by analyzing the costs incurred in UH and UBH by the threshold candidates. We calculate the average cost incurred by users during each interaction with UH, then match it with the prices of the related bundles. Additionally, we create price ranges in the form of quartiles (0–25, 26–50, 51–75, 76–100) and compute the frequency of user interactions within each price range. This process aims to identify the price tendencies where users most frequently interact with items or bundles.

The result is an implicit feature matrix representing user interaction patterns and relevant price groups. This matrix is also applied to all bundles B to generate a uniform input for the binary classification model to define constraints. Consequently, the second variation of the framework is obtained, based on user interaction patterns and price groups.

3.2.4 Feature selection and leakage mitigation

It is pertinent to emphasize that the model architecture is specifically designed to prevent label leakage. While the UBH dataset labels are generated using a heuristic threshold covering over 75% of items within a bundle, the model inputs are strictly limited to content-based features, such as game tags, hardware specifications, and pricing patterns. By deliberately excluding interaction-count features from the training process, the model is compelled to learn the underlying semantic and behavioral associations between item attributes and user interest, rather than relying on the mechanics of the label generation rule itself as we show Step 2 in Fig. 1.

This architecture ensures that the model cannot “calculate” or “memorize” the label generation threshold, as the heuristic rule is not provided as an input signal. Instead, the model learns to predict user preference based on the intrinsic content of the bundles, effectively decoupling the synthetic label generation process from the predictive inference mechanism. Consequently, this approach enforces generalization, ensuring that the recommendations are driven by the inherent relevance of game characteristics to user preferences, rather than a replication of the deterministic rules used to construct the ground truth.

3.3 Generate Constraints using Machine Learning

Constraints for recommending bundles are generated using binary classification algorithms. At least five algorithms have been tested to demonstrate the flexibility of the framework proposed in this study. The binary classification algorithms used include the following:

- Artificial neural network (ANN): Utilizes layers of neurons to process input data and, through forward propagation, maps complex patterns to separate binary classes based on nonlinear relationships.
- k-Nearest neighbors (k-NN): Classifies new data into one of two classes based on the majority class of its k-nearest neighbors in the feature space.
- Decision tree (DT): Constructs a decision tree by splitting data on attributes that maximize information gain, leading to leaves that represent binary classes.
- Logistic regression (LR): Models the linear relationship between input variables and the probability of belonging to one of the binary classes using the sigmoid function.
- Naive Bayes (NB): Classifies data by calculating the posterior probability for each binary class based on Bayes theorem, assuming independence among features.

These algorithms are employed to create constraints that determine whether a bundle is suitable for recommendation based on user preferences. With the two previous framework variations (attribute-based and interaction-pattern-based) and the five tested algorithms, this research produces a total of 10 constraint feature models for the bundle recommendation system. This demonstrates the flexibility and adaptability of the proposed framework

3.4 Apply Constraints to Set of Bundles B

At this stage, the designed constraints are applied to bundle set B to select candidates suitable for recommendation to users. Each bundle is evaluated based on criteria that reflect the user's specific preferences and needs. These constraints allow the selection process to consider not only the individual attributes of each bundle but also the relationships between attributes that are relevant to user preferences. This approach enables the system to eliminate bundles that are less aligned with user preferences. The outcome of this selection process is a set of candidate bundles, which are then further processed in the ranking stage to generate the top-N bundle recommendations that are most relevant and aligned with the user's needs.

3.5 Reranking Bundle Candidates based on Sentiment Matrix of Items in Bundles

The bundles evaluated in the previous stage are reranked based on the frequency of positive and negative sentiments associated with the items within each bundle. This process ensures that the final recommendations reflect user's preferences for items with the most relevant sentiment.

In this study, the reranking process employs a sentiment-based approach that integrates qualitative user satisfaction with statistical confidence. Sentiment polarity values for items, denoted as S_i , are defined on a normalized intensity scale ranging from -1.00 (Overwhelmingly Negative) to +1.00 (Overwhelmingly Positive). The results of this conversion are presented in Table 1.

Table 1. Value setting based on sentiment

Sentiment	Value
Overwhelmingly positive	1.00
Very positive	0.80
Positive	0.60
Mostly positive	0.40
Mixed	0.0
Mostly negative	-0.40
Negative	-0.60
Very negative	-0.85
Overwhelmingly negative	-1.00

To calculate the final reranking score for a bundle B , the framework aggregates the sentiment across its constituent items (I_B), incorporating the review volume (N_i) as a statistical confidence weight. Review data typically follows a long-tail distribution, where a few items accumulate a disproportionately large share of interactions. Consequently, utilizing raw review counts as direct weights would inadvertently amplify popularity bias, causing high-volume items to dominate the reranking process regardless of the actual sentiment. To address this, we define a confidence weight that exhibits "diminishing returns," recognizing that the statistical confidence gained from the first few reviews is significantly more valuable than the incremental confidence gained from hundreds of subsequent reviews. To achieve this, we apply logarithmic scaling to the review counts, ensuring that the confidence weight grows rapidly for low-volume items but stabilizes for high-volume items. The scoring function is defined as follows:

$$V_B = \frac{1}{|I_B|} \sum_{i \in I_B} (S_i \times \log_{10}(N_i + 1)) \quad (2)$$

where V_B is the final reranking score for bundle B ; $|I_B|$ is the total number of items contained within the bundle; S_i is the qualitative sentiment polarity score of item i , and N_i is the number of reviews associated with item i .

The term $\log_{10}(N_i + 1)$ serves as the confidence weight. This approach prevents popular items from disproportionately dominating the bundle ranking while still ensuring that items with more substantial evidence (higher review counts) carry more statistical significance than items with a single, potentially noisy review. The addition of 1 ensures the function is well-defined for items with zero reviews ($N_i = 0$), preventing undefined values while maintaining a baseline weight. This mathematical formulation effectively mitigates popularity bias, ensuring that recommendations are prioritized based on a balanced combination of high-quality sentiment and statistical reliability.

4. Scenario and Experiment

In this session, we evaluate the performance of each binary classification model to demonstrate the flexibility of the two types of implicit matrices developed earlier (see Section 3.2).

4.1 Research Scenario

This study was conducted using offline experiments due to time and budget constraints. The first stage involved constructing the UBH dataset using statistical methods. This was done by calculating the average number of items in UH relative to the Set of Bundles (B) and including bundles that met this condition in UBH as user interaction data. Since a complete dataset covering user interactions with bundles and their attributes was unavailable, this method helped create a more representative dataset for evaluating the recommendation system. The bundle matching process then assessed the suitability of bundles to user preferences, functioning like a binary classification task where users' implicit matrices served as input, and training labels were derived from UBH, allowing the system to predict relevant bundles.

The final stage, reranking, refined the recommendation results by selecting the top- N most relevant bundles. Since binary classification only determines whether a bundle is eligible for recommendation, it may lead to an excessive number of recommendations. To address this, a sentiment-based reranking approach was applied, leveraging user reviews and item attributes like ratings to filter and rank bundles. This process ensured that the most relevant bundles aligned with users' needs and preferences, improving recommendation quality.

4.2 Evaluation of Binary Classification Model Performance

In building the binary classification model, we implemented various tuning efforts to optimize model performance. The tuning process was conducted using hyperparameter tuning via grid search and k-fold cross-validation, with k-fold set to 5 folds.

To ensure the reproducibility of our research results, all computational experiments were conducted on a standardized hardware configuration. The system operates on Linux Mint 22.1 (Kernel 6.8.0-88-generic, 64-bit architecture). The primary computing platform is powered by an Intel Core i7-13700F (13th Gen, 16 cores, up to 5.2 GHz) processor, supported by 16 GB of RAM. To accelerate the training process of our machine learning and deep learning models, we utilized GPU acceleration via an NVIDIA GeForce RTX 4070 SUPER (12 GB VRAM). The software environment was configured with NVIDIA driver version 580.95.05 and CUDA version 13.0, ensuring the stability and consistency of all algorithms implemented in this study.

This method aimed to comprehensively evaluate model performance across various hyperparameter combinations. For ANN, epochs [10, 20] and batch sizes [32, 64] were tested. k-NN used n_neighbors [3, 5] with weights ['distance']. DT explored max_depth [None, 10, 20, 30], min_samples_split [2, 5, 10], and min_samples_leaf [1, 2, 4]. LR optimized C [0.001, 0.01, 0.1], penalty ['l2'], and solver ['lbfgs', 'liblinear', 'newton - cg']. Finally, NB was tested with alpha [0.1, 0.5, 1.0, 2.0, 5.0] and binarize [0.0, 0.5, 1.0].

After the tuning process, each model was evaluated using the best hyperparameter combinations for the two primary features, "Tags and Specs" and "Price Patterns." The results are presented in Table 2.

Table 2. Results of binary classification model based on optimized hyperparameters

Model	With tags and specs			With price patterns		
	Precision	Recall	F1-score	Precision	Recall	F1-score
ANN	0.8766	0.8762	0.8763	0.9980	0.9979	0.9980
k-NN	0.8549	0.8550	0.8550	0.9771	0.9789	0.9779
DT	0.8759	0.8742	0.8744	0.9974	0.9976	0.9975
LR	0.8684	0.8663	0.8665	0.9006	0.9002	0.9003
NB	0.8257	0.8242	0.8243	0.6793	0.6750	0.6735

The evaluation results, summarized in Table 2, compare binary classification models using two implicit matrix variations: "Tags and Specs" (attributes) and "Price Patterns" (interaction patterns). Metrics such as Precision, Recall, and F1-score were used to assess performance. "Price Patterns" consistently outperformed "Tags and Specs" indicating that interaction patterns provide stronger classification signals. DT and k-NN models were the most stable across both features, while ANN excelled with "Price Patterns" but struggled with "Tags and Specs." NB had the lowest performance overall. These findings emphasize the importance of feature selection and model choice in optimizing bundle matching classification.

4.3 Baselines

To evaluate the proposed framework, we compared it with a previous approach [20] that used Matrix Factorization for bundle recommendations. That study determined bundle ratings using two methods: the minimum item rating and the harmonic mean of item predictions within the bundle. These variations served as benchmarks for assessing our framework's performance. Additionally, the prior research introduced two Matrix Factorization variations, one based on specific categories (e.g., "tops" and "bottoms") and another using all items for training.

Since our dataset consists solely of “games” but includes over 200 tags, implementing the category-based variation was challenging due to the limited number of bundles per tag. Consequently, we utilize the all-item variation baseline, referred to as MF_ALL, as the primary baseline. MF_ALL is selected as a robust representative of the MF approach because it leverages the complete user-item interaction space, ensuring a fair comparison by avoiding the selection bias associated with category-based filtering.

To ensure the optimality and robustness of the Matrix Factorization baseline models (MF_ALL), we conducted rigorous hyperparameter tuning using GridSearchCV with a 5-fold cross-validation strategy, allowing us to explore the full user-item interaction spectrum effectively. For the SVD model, the search space for hyperparameters was defined as $n_factors$ [50, 100, 150], n_epochs [20, 30], lr_all [0.005, 0.01], and reg_all [0.02, 0.1]. Similarly, for the SVD++ model, we utilized a search space comprising $n_factors$ [20, 50], n_epochs [20, 30], lr_all [0.005, 0.01], and reg_all [0.02, 0.1]. These configurations ensured that the baseline models operated at their maximum capacity, thereby providing a fair, consistent, and reliable benchmark for comparison with our proposed framework.

4.4 Analysis of Recommendation Quality and Diversity Levels

To assess the impact of the proposed framework on the quality and diversity of recommendations, this session analyzes the performance distribution variations across models and the correlation between relevance and diversity.

To measure recommendation relevance, we used the evaluation metric hit ratio (HR). HR calculates the proportion of users who have at least one item from their interaction history included in the generated bundles, as defined by the following formula:

$$HR@K = \frac{1}{|U|} \sum_{u \in U} \mathbb{I}(|B_{r,u} \cap B_{k,u}| > 0) \quad (3)$$

where $HR@K$ is the hit ratio metric calculated at the top-K recommendation rank; $|U|$ is the total number of unique users in the test set; $\mathbb{I}()$ is an indicator function that outputs 1 if the condition within the parentheses is true (i.e., at least one relevant bundle is found), and 0 otherwise; $B_{r,u}$ is the set of ground truth bundles for user u that were rated positively ≥ 3 ; $B_{k,u}$ is the set of bundles appearing in the top-K recommendations for user u ; and $|B_{r,u} \cap B_{k,u}|$ is the cardinality of the intersection between the ground truth bundles and the recommended bundles.

To measure diversity, we analyzed the variety of bundles received by users from the perspective of Inter Diversity, which evaluates the diversity of bundles recommended across different users.

$$Div@K = \frac{1}{|U|} \sum_{u \in U} \left(1 - \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^K Sim(b_{u,i}, b_{u,j}) \right) \quad (4)$$

where $Div@K$ is the average diversity of the top-K recommendation lists across all users; $|U|$ is the total number of unique users in the test set (consistent with HR definition); K is the number of bundles in the top-K recommendation list (consistent with HR definition); $b_{u,i}$ is the i -th bundle in the top-K recommendation list for user u ; and $b_{u,j}$ is another bundle in the top-K list at a position other than i , specifically starting from the $(i+1)$ -th position, denoted here as j ; $Sim(b_{u,i}, b_{u,j})$ is the similarity score between two recommended bundles $b_{u,i}$ and $b_{u,j}$ (e.g., Jaccard similarity of their attributes).

Using the formula above, we conducted evaluations for all users by varying the number of top-N recommendations generated, where $N = [5, 10, 15, 20]$. Table 3 presents the performance measurements of recommendation quality using HR and recommendation diversity using Diversity for each model variation, comparing the MF_ALL framework with our proposed IMPACT-P framework.

Overall, the IMPACT-P framework significantly outperforms the MF_ALL framework in recommendation relevance, as shown by a 300% increase in HR. This confirms that leveraging implicit interactions captures user intent more accurately than explicit ratings, which are often sparse and lead to poor preference predictions. Interestingly, while MF_ALL generates highly diverse recommendations, nearly all bundles differ for each user, aligning with studies [11] that suggest excessive diversity can reduce relevance. To balance both factors, IMPACT-P effectively improves relevance while maintaining diversity, with only a 0.6% reduction, ensuring users receive both meaningful and varied recommendations.

Table 3. Comparison of average hit ratio and diversity across frameworks and models

Framework	Model	Avg. HR	Avg. Diversity
MF ALL	SVD	<u>0.0427</u>	0.9042
	SVD++	0.0389	<u>0.9092</u>
MF ALL MIN	SVD min	0.0339	<u>0.9614</u>
	SVD++ min	<u>0.0344</u>	0.9421
MF ALL MUL	SVD hmean	0.0441	<u>0.9189</u>
	SVD++ hmean	<u>0.0473</u>	0.9143
IMPACT-P ATTR (proposed)	ANN	<u>0.1821</u>	0.7984
	DT	0.1002	0.8038
	k-NN	0.0875	0.8179
	LR	0.1812	<u>0.8231</u>
	NB	0.0965	0.7812
IMPACT-P PATRN (proposed)	ANN	0.1719	0.8236
	DT	0.1720	<u>0.8238</u>
	k-NN	0.0637	0.6047
	LR	<u>0.1781</u>	0.8232
	NB	0.1766	0.8075
Improvement		327.79%	-0.60%

Bold text indicates the best result for each test among the proposed method and all baselines, while underlined text indicates the best result for each model.

4.5 Statistical Test

After observing in initial experiments that the IMPACT-P framework excels in improving recommendation relevance (HR) without sacrificing diversity, as shown in Table 3, we conducted a statistical analysis to ensure the consistent performance of our proposed framework. The statistical test involved 50 batch scenarios. The test data was randomly sampled using a cross-validation setup.

The results of this statistical analysis are presented in Fig. 2 and Table 4. These results provide deeper insight into the robustness and consistency of our framework across multiple test scenarios, validating its ability to outperform baseline methods statistically.

Fig. 2 illustrates that, based on statistical testing, the IMPACT-P framework consistently maintains recommendation relevance even across 50 batches. The best variation of IMPACT-P is IMPACT-P_PATRN, indicating that interaction patterns consistently outperform attribute-based or explicit interaction methods in capturing user preferences for bundles.

For the MF ALL framework, the harmonic mean variation (MF_ALL_MUL) performs best, demonstrating that averaging predicted item ratings to represent bundle ratings can effectively capture the overall value of the bundle. These findings highlight the advantages of leveraging implicit interaction patterns and the potential of harmonic mean calculations for bundle rating representation.

The Wilcoxon signed-rank test was used to compare the IMPACT-P framework against MF_ALL and its variations, with results shown in Table 4. This test assesses whether the performance differences between two groups are statistically significant. A p -value below 0.05 indicates a meaningful difference, while a higher value suggests no significant distinction. In this study, all p -values were exceptionally small (1.776×10^{-15}), confirming that IMPACT-P significantly outperforms MF_ALL, MF_ALL_MUL, and MF_ALL_MIN across all evaluations. These results highlight IMPACT-P's consistent superiority, demonstrating that its attribute and pattern-based approach effectively models key influencing factors better than MF_ALL and its variations. Consequently, IMPACT-P stands as a robust choice for future applications in similar domains.

Table 4. Comparison of average hit ratio and diversity across frameworks and models

Comparison	p -value
IMPACT-P ATTR vs. MF_ALL	1.776×10^{-15}
IMPACT-P ATTR vs. MF_ALL MUL	1.776×10^{-15}
IMPACT-P ATTR vs. MF_ALL MIN	1.776×10^{-15}
IMPACT-P PATRN vs. MF_ALL	1.776×10^{-15}
IMPACT-P PATRN vs. MF_ALL MIN	1.776×10^{-15}
IMPACT-P PATRN vs. MF_ALL MUL	1.776×10^{-15}

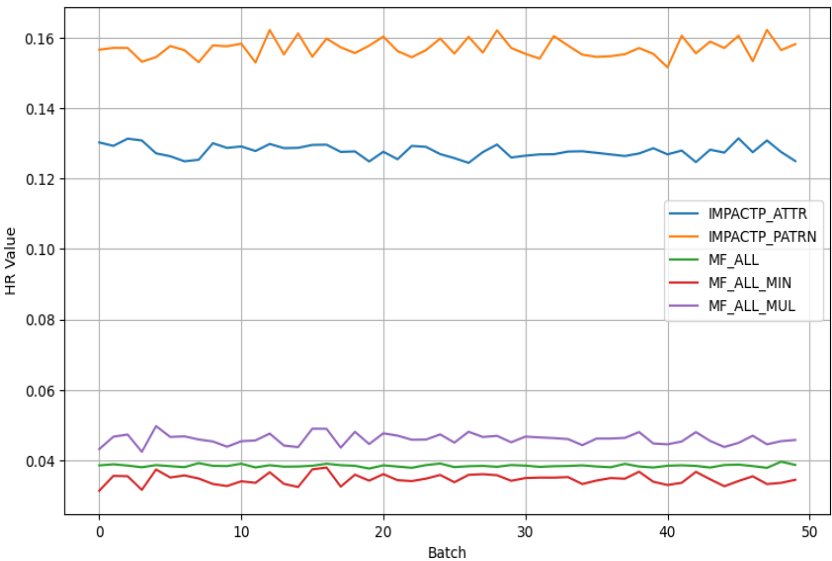


Fig. 2. Statistical test by running models to 50 batch.

4.6 Computational Complexity Analysis on Binary Classification Models

To assess the practicality and scalability of the IMPACT-P framework, we provide a *Big-O* complexity analysis for both the training and inference phases. Let n denote the number of training samples, d the number of features, k the number of latent factors (for SVD) or neighbors (for k-NN), and e the number of training epochs. Additionally, we define h as the average number of hidden units per layer in the ANN architecture, and $depth$ as the maximum height (number of levels) of the DT model.

The detailed computational complexity analysis, including the *Big-O* notation for both training and inference phases across all models, is provided in Table 5.

Table 5. Computational complexity analysis

Model	Training time complexity	Inference time complexity	Space complexity
MF_ALL (SVD/SVD++)	$O(kne)$	$O(K)$	$O((Users + Items) \cdot K)$
IMPACT-P (ANN)	$O(endh)$	$O(dh)$	$O(dh)$
IMPACT-P (k-NN)	$O(n)$	$O(nd)$	$O(nd)$
IMPACT-P (DT)	$O(n \log n d)$	$O(depth)$	$O(depth)$
IMPACT-P (LR)	$O(nd)$	$O(d)$	$O(d)$
IMPACT-P (NB)	$O(nd)$	$O(d)$	$O(d)$

4.7 Convergence Analysis and Stability Evaluation on Binary Classification Models

To ensure the models achieved optimal convergence and are free from training instability, we employed a 5-fold cross-validation protocol. Table 6 summarizes the performance stability across all iterations.

Regarding the high accuracy (>0.99) achieved by the ANN and DT models on the Price Pattern dataset, we conducted a rigorous diagnostic analysis to address potential concerns regarding feature leakage. Our investigation confirms that the results are valid and reflect the genuine discriminative power of the engineered features. This conclusion is supported by three key observations:

Model disparity, if the high accuracy were caused by feature leakage, such as the presence of a target identifier within the features for all models, including the probabilistic NB, would have achieved near-perfect accuracy. However, the NB model achieved an accuracy of only 0.67 on the same dataset. This performance gap indicates that the Price Pattern dataset contains complex, higher-order non-linear interactions that only models capable of capturing nonlinear boundaries (ANN, DT, and kNN) can effectively exploit.

Table 6. Model performance stability analysis F1-score (5-fold cross-validation) for IMPACT-P

Model	Variation	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
ANN	ATTR	0.8763	0.8744	0.8722	0.8753	0.8698	0.8736	0.0023
	PATRN	0.9927	0.9966	0.9926	0.9986	0.9961	0.9953	0.0023
DT	ATTR	0.8722	0.8711	0.8755	0.8717	0.8697	0.8720	0.0019
	PATRN	0.9976	0.9970	0.9972	0.9979	0.9966	0.9973	0.0005
k-NN	ATTR	0.8200	0.8569	0.8531	0.8638	0.8512	0.8490	0.0151
	PATRN	0.9784	0.9782	0.9772	0.9739	0.9742	0.9764	0.0020
LR	ATTR	0.8648	0.8622	0.8660	0.8635	0.8630	0.8639	0.0013
	PATRN	0.8993	0.8993	0.8992	0.8994	0.9000	0.8994	0.0003
NB	ATTR	0.8201	0.8194	0.8250	0.8192	0.8214	0.8210	0.0021
	PATRN	0.6753	0.6752	0.6751	0.6760	0.6748	0.6753	0.0004

Statistical consistency, the consistency observed across the k-fold cross-validation folds (low standard deviation) serves as a robust defense against overfitting. An overfitted model or one suffering from leakage typically exhibits erratic performance when exposed to different data splits. Our models demonstrated high, uniform stability across all iterations.

In other hand, to evaluate the predictive stability and generalization capability of the proposed models, we conducted an analysis of the F1-Score convergence across training iterations. Figs. 3 and 4 present the convergence curves for the Tags and Specs and Price Patterns variant of IMPACT-P, respectively, illustrating the performance trajectory for both training (solid lines) and validation (dashed lines) sets. The analysis of these convergence profiles reveals three critical findings:

Convergence stability: All models exhibit a positive convergence trend, reaching a state of predictive

saturation, or a plateau, as training iterations increase. Nonlinear architectures, specifically the ANN (red) and DT (green), demonstrate a more efficient convergence rate compared to the statistical baselines. This suggests that the higher representational capacity of these models allows them to effectively capture and stabilize on the underlying data features within the initial stages of the training process.

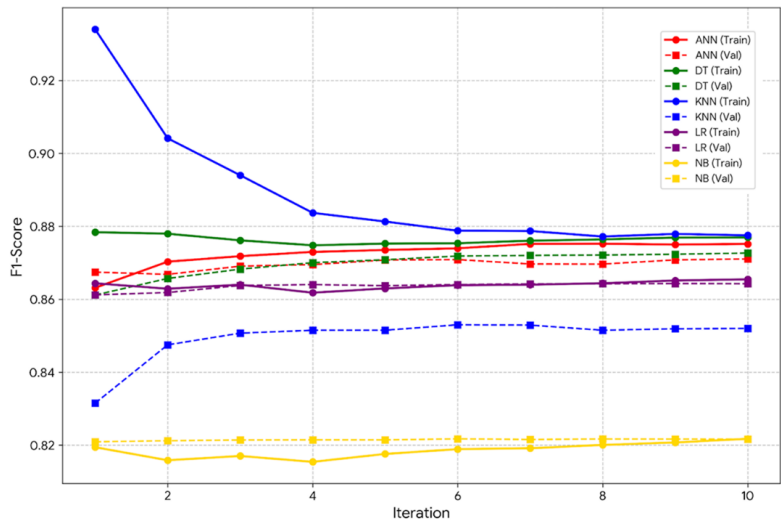


Fig. 3. Learning curve for binary classification model training of IMPACT-P ATTR.

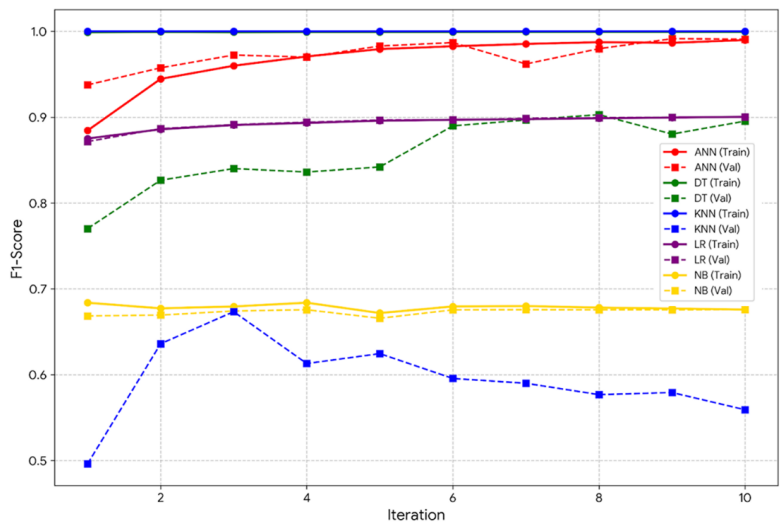


Fig. 4. Learning curve for binary classification model training of IMPACT-P PATRN.

Analysis of generalization: The generalization gap, defined by the divergence between the training and validation F1-Scores, remains within acceptable parameters for most of the models. The ANN exhibits the most consistent performance, with the narrowest gap between training and validation scores, signifying excellent generalization and a robust resistance to overfitting. Conversely, models showing a wider divergence, such as ANN, indicate a tendency toward memorizing the training data, necessitating more rigorous regularization in future iterations.

Comparison of linear vs. nonlinear architectures: Linear models, including LR (purple) and NB (yellow)

(gold), demonstrate stable convergence profiles but consistently stabilize at lower F1-Score levels compared to their non-linear counterparts. This empirical evidence supports our hypothesis that user preferences and bundle relevance, within both Tags and Specs and Price Patterns feature sets, are governed by complex, non-linear dependencies that simple linear estimators cannot fully characterize.

5. Conclusion

This study shows that the IMPACT-P framework significantly improves recommendation relevance, achieving a 300% increase in HR while maintaining high diversity, with only a 0.6% decrease compared to the MF_ALL framework. The effectiveness of implicit interactions in capturing user intent surpasses explicit interactions like ratings, which are often sparse. Among the models tested, LR performed best, while k-NN had the lowest accuracy due to suboptimal bundle predictions. Currently, this research focuses on a collaborative scope with specific datasets, and the initial development of IMPACT-P. Future studies will explore temporal aspects, test on larger datasets, and enhance recommendation interpretability. All the source code of our work is available at <https://huggingface.co/recommender-system/impact-p>.

Conflict of Interest

The authors declare that they have no competing interests.

Funding

This research was funded by the Directorate of Research, Technology, and Community Service (DRTPM) under the Fundamental Research Scheme, with contract number 106/E5/PG.02.00.PL/2024. We would like to express our gratitude for their support, which made this research possible.

References

- [1] Y. Fang, Q. An, X. Jin, and Y. Liu, "Bundle recommendation methods considering rating data differences for online retailers," *PLoS One*, vol. 20, no. 9, article no. e0328245, 2025. <https://doi.org/10.1371/journal.pone.0328245>
- [2] Y. Qin, P. Wang, and C. Li, "the world is binary: contrastive learning for denoising next basket recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Virtual Event, Canada, 2021, pp. 859-868. <https://doi.org/10.1145/3404835.3462836>
- [3] Z. He, H. Zhao, T. Yu, S. Kim, F. Du, and J. McAuley, "Bundle MCR: towards conversational bundle recommendation," in *Proceedings of the 16th ACM Conference on Recommender Systems*, Seattle, WA, USA, 2022, pp. 288-298. <https://doi.org/10.1145/3523227.3546755>
- [4] N. Ebrahimi, Z. Zhang, and K. Stefanidis, "A diversity-aware approach to bundle recommendations," in *Proceedings of the 27th International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP)*, Barcelona, Spain, 2025, pp. 49-53.

- [5] H. S. Nguyen, Y. Liu, M. Mansoury, M. Aliannejadi, A. Hanjalic, and M. de Rijke, "A reproducibility study of product-side fairness in bundle recommendation," in *Proceedings of the 19th ACM Conference on Recommender Systems*, Prague, Czech Republic, 2025, pp. 696-705. <https://doi.org/10.1145/3705328.3748169>
- [6] M. Zanker, M. Aschinger, and M. Jessenitschnig, "Constraint-based personalised configuring of product and service bundles," *International Journal of Mass Customisation*, vol. 3, no. 4, pp. 407-425, 2010. <https://doi.org/10.1504/IJMASSC.2010.037653>
- [7] F. Lorenzi, S. Loh, and M. Abel, "PersonalTour: a recommender system for travel packages," in *Proceedings of 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Lyon, France, 2011, pp. 333-336. <https://doi.org/10.1109/WI-IAT.2011.69>
- [8] R. Hti and M. S. Desarkar, "Personalized tourist package recommendation using graph based approach," in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP)*, Singapore, 2018, pp. 257-262. <https://doi.org/10.1145/3213586.3225233>
- [9] A. Gionis, T. Lappas, K. Pelechrinis, and E. Terzi, "Customized tour recommendations in urban areas," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, New York, NY, USA, 2014, pp. 313-322. <https://doi.org/10.1145/2556195.2559893>
- [10] T. Zhu, P. Harrington, J. Li, and L. Tang, "Bundle recommendation in ecommerce," in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, Gold Coast, Queensland, Australia, 2014, pp. 657-666. <https://doi.org/10.1145/2600428.2609603>
- [11] M. Brucato, A. Abouzied, and A. Meliou, "Improving package recommendations through query relaxation," in *Proceedings of the 1st International Workshop on Bringing the Value of "Big Data" Users (Data4U 2014)*, Hangzhou, China, 2014, pp. 13-18. <https://doi.org/10.1145/2658840.2658843>
- [12] T. Deng, W. Fan, and F. Geerts, "On the complexity of package recommendation problems," in *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, Scottsdale, AZ, USA, 2012, pp. 261-272. <https://doi.org/10.1145/2213556.2213592>
- [13] C. Zhang, H. Liang, K. Wang, and J. Sun, "Personalized trip recommendation with POI availability and uncertain traveling time," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, Melbourne, Australia, 2015, pp. 911-920. <https://doi.org/10.1145/2806416.2806558>
- [14] C. Tan, Q. Liu, E. Chen, H. Xiong, and X. Wu, "Object-oriented travel package recommendation," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 3, pp. 1-26, 2014. <https://doi.org/10.1145/2542665>
- [15] H. Liang and K. Wang, "Top-k route search through submodularity modeling of recurrent POI features," in *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, Ann Arbor, MI, USA, 2018, pp. 545-554. <https://doi.org/10.1145/3209978.3210038>
- [16] H. Chen, Y. Lin, M. Pan, L. Wang, C. C. M. Yeh, X. Li, Y. Zheng, F. Wang, and H. Yang, "Denoising self-attentive sequential recommendation," in *Proceedings of the 16th ACM Conference on Recommender Systems*, Seattle, WA, USA, 2022, pp. 92-101. <https://doi.org/10.1145/3523227.3546788>
- [17] Z. Liu, Y. Hou, and J. McAuley, "Multi-behavior generative recommendation," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, Boise, ID, USA, 2024, pp. 1575-1585. <https://doi.org/10.1145/3627673.3679730>
- [18] M. Wan and J. McAuley, "Item recommendation on monotonic behavior chains," in *Proceedings of the 12th ACM Conference on Recommender Systems*, Vancouver, Canada, 2018, pp. 86-94. <https://doi.org/10.1145/3240323.3240369>
- [19] A. Pathak, K. Gupta, and J. McAuley, "Generating and Personalizing bundle recommendations on steam," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in*

Information Retrieval, Shinjuku, Tokyo, Japan, 2017, pp. 1073-1076. <https://doi.org/10.1145/3077136.3080724>

- [20] A. T. Wibowo, A. Siddharthan, J. Masthoff, and C. Lin, "Incorporating constraints into matrix factorization for clothes package recommendation," in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, Singapore, 2018, pp. 111-119. <https://doi.org/10.1145/3209219.3209228>



Andy Maulana Yusuf <https://orcid.org/0000-0001-7712-5221>

He received the B.S. and M.S. degrees in Informatics Department from Telkom University, Indonesia, in 2023 and currently continuing to the Ph.D. program in Informatics Department from Telkom University, Indonesia. From 2017 to 2018, he was a computer laboratory assistant in Applied Science Department, Telkom University. Since 2019, he has been a lecturer assistant with the Electronics Engineering Department, Telkom University.



Adiwijaya <https://orcid.org/0000-0002-3518-7587>

He was born in Majalengka, 21 September 1974. He graduated with a bachelor of Mathematics from the Bandung Institute of Technology (ITB) in 1993–1998, graduated from master of Mathematics at Bandung Institute of Technology (ITB) in 2001–2004, and graduated with a doctoral degree in Mathematics at the Bandung Technology Institute (ITB) in 2007–2012. Currently he serves as rector of Telkom University. He is now the Chair & TPC Chair of the International Conference on Data and Information Science (ICoDIS). He is also Chair of the Editor-in-Chief of the *Journal of Data Science and Its Applications*.



Agung Toto Wibowo <https://orcid.org/0000-0003-2434-2518>

He holds a Ph.D. in Computing Science from the University of Aberdeen (2016–2018) and a master's degree in digital media and game technology from Institut Teknologi Bandung. His research focuses on recommender systems, matrix and tensor factorization, machine learning, swarm intelligence, and plagiarism detection. Since 2006, he has been a senior lecturer at Telkom University, teaching artificial intelligence, machine learning, and theory of computation. His research applies machine learning and swarm intelligence to areas like collaborative filtering and time series prediction.



Z. K. Abdurahman Baizal <https://orcid.org/0000-0003-0795-9559>

He was a distinguished academic with a rich background in Computer Science. He earned his Doctorate in Computer Science from Institut Teknologi Bandung (ITB) and holds both a master's and bachelor's degree from Universitas Gadjah Mada (UGM). His research focuses on artificial intelligence, recommender systems, ontology, and the semantic web. He served as the Dean of the Faculty of Informatics at Telkom University from December 2019 to June 2022, where he played a key role in advancing the faculty's academic and research.