Improving Ranking Quality via Consequent Set Quartile Partitioning in Rule-Based Recommendation

Tubagus Mohammad Akhriza[1*], Khoerul Anwar[2], and Weda Adistianaya Dewa[3]
[1,3]Department of Information System, STMIK PPKIA Pradnya Paramita
Jawa Timur, Indonesia 65126
[2]Department of Information Technology, STMIK PPKIA Pradnya Paramita
Jawa Timur, Indonesia 65126
Email: [1]akhriza@stimata.ac.id, [2]alqhoir@stimata.ac.id, [3]weda@stimata.ac.id

***Abstract*—Accurate ranking remains challenging in session-based recommender systems, particularly in dynamic domains such as e-commerce and digital news. Neural models can achieve high accuracy but require substantial computational resources, while Association Rule (AR)-based methods are computationally efficient yet may rank relevant items poorly when candidate weights are similar. The research aims to improve the ranking quality of AR-based session recommendations while retaining computational efficiency through Association Rule with Top-k Quartile Filtering (ART-Q). The proposed framework constructs an association-rule dictionary, partitions each consequent set into four quartiles according to rule weights, and independently selects top-k candidates from each quartile. ART-Q is evaluated using two real-world datasets: YooChoose (YOO) and Malang Posco Media (MPM). Performance is evaluated using HR@20, MRR@20, and NDCG@20 against AR, k-Nearest Neighbor (KNN), and neural baselines. On YOO, ART-Q-V6 achieves HR@20 of 0.6899, MRR@20 of 0.7500, and NDCG@20 of 0.5436, representing gains of 11.9% in HR and 31.6% in Normalized Discounted Cumulative Gain (NDCG) over traditional AR, with more than twofold improvement in Mean Reciprocal Rank (MRR). On MPM, ART-Q-V1 achieves HR@20 of 0.7498, MRR@20 of 0.2717, and NDCG@20 of 0.3260, outperforming all evaluated baselines. ART-Q also completes training and inference in less than two minutes on a laptop CPU, demonstrating its computational efficiency. These results indicate that quartile-based filtering effectively reduces ranking ambiguity while maintaining a lightweight recommendation framework.**

***Index Terms*—Association rules, Ranking Quality, Recommendation System, Session-Based Recommendation, Top-k Recommendation**

Received: June 29, 2025; received in revised form: Oct. 30, 2025; accepted: Oct. 30, 2025; available online: Aug. 13, 2026.
*Corresponding Author

I. Introduction

RECOMMENDER Systems (RS) play a pivotal role in enhancing digital platforms, such as e-commerce and online news, by delivering tailored content based on user preferences [1, 2]. However, a distinct challenge emerges within anonymous platforms due to the absence of explicit user preference data. To address this limitation, session-based RS leverages short-term, intra-session user interactions to predict subsequent items in real time [3–5]. Optimizing this system is critical because ranking quality directly influences user engagement and satisfaction. When relevant items are positioned too low within the recommendation list, the probability of user engagement diminishes, leading to suboptimal experiences and reduced platform effectiveness [6, 7].

Session-based RS research has historically produced two main methodologies. Neural approaches (e.g., Gated Recurrent Unit for Recommendation (GRU4Rec) [8], Next Item recommendation Network (NextItNet) [9] and Session-based Recommendation with Graph Neural Networks (SR-GNN) [10]) deliver high performance by modeling sequential or graph-structured dependencies. However, a heavy reliance on extensive datasets and substantial computational resources restrict practical deployment in real-time settings [4]. Non-neural approaches (i.e., Association Rule (AR) mining and k-Nearest Neighbor (KNN) algorithms) remain advantageous due to high computational efficiency and clear interpretability. Prior studies have demonstrated that properly configured simpler techniques can match or exceed the performance of neural models under specific conditions [4].

Traditional recommendation methods typically select the top-k items from a single candidate pool on the highest probabilities observed within a session [11–

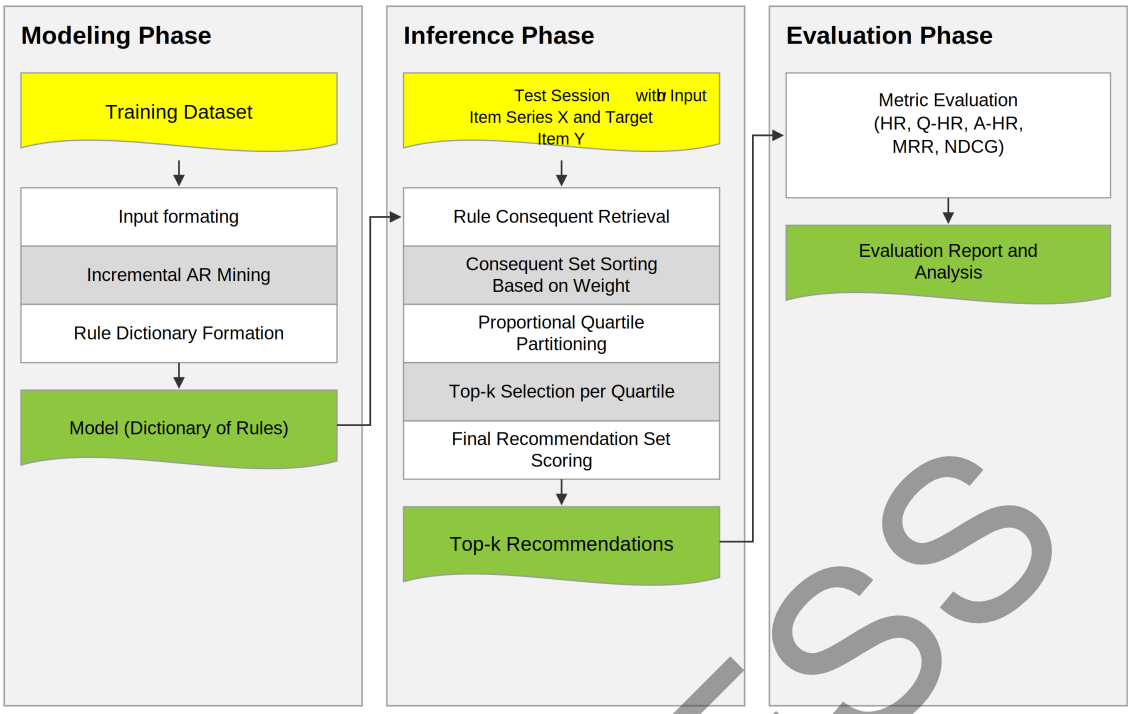

Fig. 1. General workflow of Association Rule with Top-k Quartile Filtering (ART-Q) method. Note: Hit Rate (HR), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG), Quartile-Level HR Evaluation (Q-HR), and Antecedent-Level HR Evaluation (A-HR).

13]. Similarly, AR methods select items from consequent sets by maximizing support, which measures how frequently an itemset appears in the dataset, and confidence, which measures the conditional probability of a consequent given the antecedent [14]. However, when multiple candidate items share similar support and confidence values, ranking ambiguity often occurs. This ambiguity can misplace relevant items lower in the list, which ultimately reduces evaluation metrics, such as Hit Rate (HR), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG). Furthermore, traditional AR methods in session-based recommendation typically rely solely on the final item in a session as the antecedent reference to identify consequent sets [4, 15]. Consequently, this limitation narrows item coverage and potentially fails to produce relevant recommendations.

The research introduces Association Rule with Top-k Quartile Filtering (ART-Q) to effectively address these challenges. The ART-Q framework reorganizes consequent sets into four distinct quartiles (Q1–Q4) based on support-confidence weights, allowing it to select multiple top-k items across each tier. Beyond drawing candidates from individual quartiles, the system leverages every item in the active session as an antecedent reference, generating multiple top-k recommendations for each consequent set. This approach broadens candidate coverage to reduce ranking ambiguity and boost key metrics, e.g., HR, MRR, and NDCG, without substantially increasing computational complexity.

The objective of this research is to improve the ranking quality of AR-based session recommendation in dynamic domains while retaining computational efficiency. To achieve this goal, the research introduces quartile-based filtering as a refinement strategy. The effectiveness of this approach is evaluated using two real-world datasets from the YooChoose (YOO) e-commerce dataset and the Malang Posco Media (MPM) digital news dataset. Furthermore, the performance is analyzed at both quartile and antecedent levels to provide deeper insights into how rule-based recommenders can be enhanced for session recommendation tasks.

II. Research Method

The general workflow of the ART-Q method is illustrated in Fig. 1. The framework is divided into three main phases: modeling, inference, and evaluation. The detailed mechanisms of each phase are provided in the following subsections. Prior to those descriptions, the ART-Q model is formalized.

A. Model Formalization

In accordance with prior next-item recommendation frameworks [9, 16, 17], a session $u = \{x_1, x_2, \ldots, x_i\}$ composed of unique clicked items, where $Y = \{Y_1, Y_2, \ldots\}$ denotes the set of consequent items Y_i corresponding to each $x_i \in u$. ART-Q extends this formulation by aggregating rule-based predictions across all antecedents. It ensures that the probability distribution accounts for the combined influence of every antecedent x_i and its corresponding consequents $y \in Y_i$, which represent the next-item candidates for session u. Therefore, each pair $x_i \rightarrow Y_i$ constitutes a distinct association rule. Combining these rules derives the overall probability distribution of recommendations, as formulated in Eq. (1).

$$P\left(y|u\right) = \sum_{x_i \in u} \sum_{y \ \in Y_i} P\left(y|x_i\right) \times w\left(x_i, y\right). \quad (1)$$

Where $w\left(x_i, y\right) = conf\left(x_i, y\right) \times \sup(y)$ represents the weight of y relative to x_i. It has u as the user session, x_i as the i-th item in session u, y_i as the target item to be recommended, $P(y|u)$ as the conditional probability of target item y in given session u, $P(y|x_i)$ as the conditional probability of item y in given item x_i, and w as the weight of the association rule between x_i and y_i. Prior literature of general AR-based methods often employs the Lift metric to indicate the correlation strength between x and y [7, 18–21], which normalizes the rule confidence by the support of y, or $Lift(xy) = \frac{conf(x,y)}{sup(y)}$, where $conf(x, y) = \frac{sup(xy)}{sup(x)}$. However, ART-Q deliberately avoids using Lift because such normalization may unintentionally penalize the relevance of less frequent items. Additionally, this weighting approach mitigates uniformity in rule scoring, preventing excessive reliance on confidence alone. After sorting the items in Y_i by descending weight, the items are divided into four quartiles Q_1, Q_2, Q_3, Q_4, where each quartile covers a proportion p_i, ensuring that $\sum_{i=1\ldots4} p_i = 1$. Recommendations are generated by independently selecting top-k items from each quartile after ordering items within Q_i by weight.

Let $Topk\text{-}Q_i$ represent the set of $y_j \in Q_i$, with $j \leq k$, with the highest weights in Q_i. The probability of selecting item y within an individual quartile Q_i is given in Eq. (2). Then, the total probability across all quartiles is then computed in Eq. (3).

$$P\left(y|u,\ Q_i\right) = \sum_{x_i \in u} P\left(y|x_i, Topk\text{-}Q_i\right) \times w\left(x_i, y\right), \quad (2)$$

$$P\left(y|u, Q\right) = \sum_{j=1\ldots4} P\left(y|u, Q_j\right). \quad (3)$$

Given that the recommendation space is constrained within $Topk\text{-}Q_i$, the probability for an individual antecedent x_i across all quartiles with respect to $Topk\text{-}Q_i$ is defined in Eq. (4). Since multiple antecedents contribute to session-based recommendations, ART-Q computes the total session-level probability by aggregating over all antecedents. Thus, the final probability is derived in Eq. (5).

$$P\left(y|x_i\right) = \sum_{j=1\ldots4} P\left(y|x_i, Topk\text{-}Q_i\right) \times w\left(x_i, y\right), \quad (4)$$

$$P\left(y|u\right) = \sum_{x_i \in u} P\left(y|x_i\right). \quad (5)$$

B. Modeling Phase

The modeling phase of ART-Q constructs a rule dictionary from the training dataset T to capture relationships between antecedents and corresponding consequent sets. Within this framework, antecedents serve as keys and consequent sets function as values to guide the recommendation process. The system executes rule construction and dictionary formation in a sequential manner.

The first stage involves input formatting. Preprocessing prepares the session data in dataset T to ensure that each row corresponds to a unique SessionId containing a sequence of ItemId entries ordered by timestamp. Following this formatting, incremental AR mining begins.

The items within each session form 1-itemsets and 2-itemsets through permutation. A frequency counter increments the occurrence of individual items and item pairs (x, y). Following the counting step, association rules of length two are constructed from the mined 2-itemsets, representing simple item-to-item transitions within sessions. An advantage of this incremental mining approach is the ability to enable fast, continuous rule construction in streaming environments where transaction data arrives session by session [18, 22]. This processing eliminates the requirement to wait for data accumulation windows, such as one day or week, before updating the model. As a trade-off, the computed support and confidence values are approximations, though the data remains effective in capturing the relative strength and relevance of item associations.

In the final stage, a rule dictionary forms. After processing all sessions in T, a dictionary of rules is constructed. Each key corresponds to an antecedent item x, and the value is a nested dictionary of consequent items y paired with corresponding rule weights $w(x, y)$. Since the supports of x, y, and the co-occurrence (x, y) are accumulated during step two, computing this weight is straightforward. The final dictionary format is structured as $x_i : \{y_1 : w_1, y_2 : w_2, \ldots\}$. Once all

records are processed, the resulting rule dictionary fully encapsulates the learned model from the session data, serving as the foundation for the subsequent inference and recommendation phases.

C. Inference Phase

In practical applications, the inference phase processes a session $u = x_1, \ldots, x_{i+1}$ to generate multiple next-item predictions $x_{i+1}, x_{i+2}, \ldots$. However, during the experimental evaluation, each session u in the testing set is split into two parts, i.e., $u_x = x_1, \ldots, x_i$ representing the observed antecedents, and $y = x_{i+1}$ serving as the ground-truth next item. The model aims to predict y within the top-k recommendations by the subsequent procedure.

The process begins with retrieving rule consequents. For each antecedent $x \in u_x$, the corresponding set of consequents Y is retrieved from the rule dictionary, where each consequent $y \in Y$ is already associated with a rule weight $w(x, y)$. Next, this consequent set is sorted. All consequent items $y \in Y$ are sorted in descending order based on the respective rule weights $w(x, y)$, yielding a ranked list of candidate consequents for each antecedent.

Following the sorting stage, the framework performs proportional quartile partitioning. The ranked list of consequent items is partitioned into four quartiles, denoted as Q_1, Q_2, Q_3, and Q_4, representing segments from the highest to the lowest rule weights, respectively. Each quartile is assigned a proportional weight $p_1 : p_2 : p_3 : p_4$ that it is $\sum_{i=1..4} p_i = 1$. This quartile-based partitioning serves two primary purposes. First, it enables low-scoring but contextually relevant consequents, which frequently appear found in Q_3 and Q_4. Second, it allows an item near the tail end of a higher quartile, such as Q_2, to be re-ranked into the head of a lower quartile Q_3 under top-k selection, thereby increasing the probability of inclusion in the final recommendation list.

Subsequently, top-k selection is applied to each quartile. From each quartile Q_i, a subset of k top-ranked items is selected, defined as $Topk\text{-}Q_i = \{y_j \in Q_i \mid j \leq k\}$, where y_j represents a candidate next-item selected within quartile Q_i. Through this approach, multiple top-k recommendations are generated across quartiles, ensuring that not only the most frequent items but also lower-weighted yet contextually relevant candidates are considered. Finally, the total recommendation probability is computed. The total probability across all quartiles is then computed following Eq. (3), while the total probability across all antecedents in session u is computed following Eq. (5).

D. Evaluation Phase

Standard ranking metrics, i.e., HR, MRR, and NDCG, are utilized to assess the performance of baseline models [12, 23, 24]. As formulated in Eq. (6), Hit Rate at top-k (HR@k) measures the proportion of sessions in which the ground-truth item y appears within the top-k recommendations [25, 26].

$$HR@k = \frac{1}{|U|} \sum_{u \in U} 1 \ \left(I_u \in R_u^k\right) , \tag{6}$$

where U is the set of test sessions, and $|U|$ is the total number of test sessions. Then, it has u as an individual session in U, R_u^k as the set of top-k recommended items generated for session u, I_u as the ground-truth item I associated with session u, and k the number of top recommendations considered.

MRR measures the average reciprocal rank of the first correctly predicted ground-truth item, ensuring that highly relevant items appear earlier in the recommendation list [4, 26, 27]. It is defined in Eq. (7) where $rank_u$ represents the position of the first correctly recommended item in session u. A higher MRR score indicates improved ranking quality, with relevant items appearing closer to the top.

$$MRR@k = \frac{1}{|U|} \sum_{u \in U} \frac{1}{rank_u}. \tag{7}$$

On the other hand, NDCG evaluates ranking quality and relevance weighting. The model prioritizes highly relevant items earlier in the list [26–28]. Equation (8) demonstrates this computation:

$$NDCG@k = \frac{DCG@k}{IDCG@k}. \tag{8}$$

where $DCG@k$ is the Discounted Cumulative Gain of the top-k recommendations, while $IDCG@k$ is the Ideal Discounted Cumulative Gain, representing the maximum possible DCG@k under an ideal ranking.

The variable $DCG@k$ is calculated using a logarithmic discount factor, ensuring that top-ranked items contribute more to the final score, as defined in Eq. (9). The term rel_j represents the relevance score of the item at position j in the ranked list. Thus, j denotes the rank position of an item, where j = 1 is the highest-ranked position. It is $rel_j = 1$ if the recommended item matches the ground-truth item, and 0 otherwise.

$$DCG@k = \sum_{j=1\ldots k} \frac{2^{rel_j} - 1}{\log_2(j+1)}. \tag{9}$$

These standard metrics are also applied within ART-Q to assess recommendation quality at the session, quartile, and antecedent levels, as detailed in the following explanations. The evaluation includes session-

level HR, which applies the common HR@k metric as defined in Eq. (6). It measures whether the ground-truth next-item y is successfully identified within any $Topk\text{-}Q_i$. It is defined in Eq. (10):

$$HR@k = \begin{cases} 1, & \text{if } \exists\, Q_i \text{ and } y \in Topk\text{-}Q_i \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

HR@k is counted as 1 if at least one quartile contains y within its top-k items. This metric evaluates the overall success of the model per session. It ensures that at least one ranked quartile includes the actual next-item prediction [24–26].

In this research, Quartile-Level HR Evaluation (Q-HR@k) is newly proposed to measure whether the ground-truth next-item appears within the top-k items in each quartile independently.This formulation is provided in Eq. (11):

$$Q\text{-}HR@k = \begin{cases} 1, & \text{if } y \in Topk\text{-}Q_i \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The metric Q-HR@k is computed separately for each quartile Q_i. Higher Q-HR@k scores indicate broader quartile relevance, ensuring that lower-ranked quartiles also contribute meaningfully to the recommendation process.

In addition, Antecedent-Level HR Evaluation (A-HR@k) is newly proposed. It measures how many antecedents in a session contribute to identifying the ground-truth next-item. Equation (12) outlines this approach.

$$A\text{-}HR@k\,(x_i) = \begin{cases} 1, & \text{if } \exists\, Q_i,\ y \in Topk\text{-}Q_i \text{ for } x_i \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The variable $x_i \in u$ represents an individual antecedent observed. A-HR@k counts as 1 for each antecedent that successfully ranks y within its top-k predictions across quartiles. This metric evaluates the distribution of predictive contributions across antecedents, ensuring that recommendations are not overly dependent on a single antecedent. It is worth noting that Q-HR and A-HR accommodate the model's ability to match relevant items across different quartiles through distinct antecedents. As a result, a single test session may have more than one Q-HR and A-HR. Consequently, the average Q-HR and A-HR across the entire test dataset can also exceed 1 if the total hits surpass the number of test sessions.

To assess ranking precision, ART-Q applies MRR@k on quartile-specific predictions. It applies MRR@k separately to each $Topk\text{-}Q_i$ that contains the ground-truth next-item y, denoted as $MRR@k\,(Topk\text{-}Q_i)$. The formula, adapted from Eq. (7), is defined in Eq. (13).

$$MRR@k\,(Topk\text{-}Q_i) = \frac{1}{|U|} \sum_{u \in U} \frac{1}{rank_u\,(Topk\text{-}Q_i)}. \tag{13}$$

Here, $rank_u\,(Topk\text{-}Q_i)$ is the position of the first correctly recommended next-item y within $Topk\text{-}Q_i$. The final session-level MRR@k score is determined by selecting the maximum $MRR@k\,(Topk\text{-}Q_i)$ value from across four quartiles within that session. The overall session MRR@k is then computed as the average of all session-level MRR@k scores.

Similarly, $NDCG@k\,(Topk\text{-}Q_i)$ measures ranking quality and relevance weighting within each quartile Q_i, ensuring that highly relevant items appearing earlier receive higher scores. This improves the efficiency of recommendation rankings. The formula, adapted from Eq. (8), is defined in Eq. (14):

$$NDCG@k\,(Topk\text{-}Q_i) = \frac{DCG@k\,(Topk\text{-}Q_i)}{IDCG@k\,(Topk\text{-}Q_i)}. \tag{14}$$

The principles of $DCG@k\,(Topk\text{-}Q_i)$ and $IDCG@k\,(Topk\text{-}Q_i)$ remain the same as the original version presented in Eq. (9). However, rel_j represents the relevance of item y, assigned a value of 1 if $y \in Topk\text{-}Q_i$, and 0 otherwise. The final session-level $NDCG@k$ score is determined by selecting the maximum $NDCG@k\,(Topk\text{-}Q_i)$ value obtained across four quartiles within that session. The overall session $NDCG@k$ is then computed as the average of all session-level scores.

E. Pseudocode of Association Rule with Top-k Quartile Filtering (ART-Q)

The integration of quartile partitioning within the recommendation process is detailed in the pseudocode in Appendix. The framework operates in two primary stages: an incremental rule-generation phase followed by a session-based inference and evaluation phase. By analyzing the training data, the algorithm constructs a weighted rule dictionary that enables the subsequent partitioning of items into quartiles, extraction of top-k recommendations, and computation of normalized performance metrics.

F. Experimental Works Design

The research evaluates the performance of ART-Q with baseline models using two datasets from different domains. The first is YOO, a public e-commerce transaction dataset containing user session data from the RecSys Challenge 2015. The second is MPM, a

TABLE I
SUMMARY OF DATASET CHARACTERISTICS.

Dataset	Session Number	Item Number	Min-Max Session Length
YOO-train	1,410,210	28,450	2–200
YOO-test	1,606	2,603	2–84
MPM-train	17,244	31,226	2–421 (orig.) and 2–10
MPM-test	3,999	6,494	2–10

Note: YooChoose (YOO) and Malang Posco Media (MPM).

private dataset collected from a digital news platform through cookie-based logging of user reading activities over two weeks in August 2024. The cookies capture sequences of news article IDs accessed by readers. Session lengths vary from very short, spanning one to three clicks, to very long, depending on user browser settings rather than program constraints. The private dataset is available upon request to the authors for research purposes. However, potential biases exist within this data owing to variations in cookie retention policies and the frequent appearance of popular articles within user sessions.

These datasets present unique challenges for session-based RS. YOO features goal-driven interactions where users make rapid purchasing decisions influenced by product descriptions and promotions [29]. In contrast, MPM represents digital news consumption characterized by exploratory behavior, in which users engage with multiple articles in a single session, driven by evolving interests [1, 30]. Furthermore, item characteristics differ substantially between domains. Product availability in e-commerce fluctuates based on inventory levels and market trends, while news article relevance evolves dynamically in response to editorial curation, breaking news, and audience engagement [31, 32].

The experimental design partitions each dataset into training and testing sets, with core characteristics summarized in Table I. The train-test split for YOO replicates the version from the official GitHub repository maintained by the original authors [33] to ensure consistency with prior comparative studies. Meanwhile, the MPM dataset is split into training and testing sets at an approximate ratio of 85:15. Across both datasets, the minimum session length is two items. The training phase for MPM involves two distinct treatments: the first using the original session data with a maximum length of 421, and the second partitioning the data into sub-sessions with a maximum length of 10. The training results represent the average of these two treatments.

The baseline models for benchmarking include Simple AR, Session-based KNN (SKNN), Vector Multiplication Session-based KNN (VSKNN), GRU4Rec, NextItNet, and SR-GNN. The Python implementations for these models are available on GitHub [34], maintained by the authors of a comparative study on session-based recommendation models [4, 35]. For brevity, this benchmarking suite is referred to as Recommendation System (SREC). Notably, NextItNet is also run using the original source code provided by the developers on GitHub [33], as it employs a different data-loading mechanism than the SREC suite. The proposed ART-Q framework is implemented in Python using Jupyter Notebook.

TABLE II
MODELS' PARAMETERS APPLIED TO DATASETS.

Models	**Parameters**
GRU4Rec	Epoch = 10, 50, lr = 0.08, layers = 100
NextItNet	Epoch = 10, 50, lr = 0.001, kernel size = 3, dilation = [1,4,1,4,1,4,1,4,1,4,1,4], dilation channel = 256
SR-GNN	Epoch = 10, 50, lr = 0.006, hidden_size = 100, out_size = 100
AR	minsup = 1, and no minconf applied
SKNN	k = 500, n = 5000
VSKNN	k = 500, n = 5000, sim = cosine, idf = 5
ART-Q-V1	p1, p2, p3, p4 = 0.25
ART-Q-V2 (Q1)	p1 = 0.7, p2 = 0.1, p3 = 0.1, p4 = 0.1
ART-Q-V3 (Q2)	p1 = 0.1, p2 = 0.7, p3 = 0.1, p4 = 0.1
ART-Q-V4 (Q3)	p1 = 0.1, p2 = 0.1, p3 = 0.7, p4 = 0.1
ART-Q-V5 (Q4)	p1 = 0.1, p2 = 0.1, p3 = 0.1, p4 = 0.7
ART-Q-V6	p1 = 0.01, p2 = 0.09, p3 = 0.5, p4 = 0.3

Note: Gated Recurrent Unit for Recommendation (GRU4Rec), Next Item recommendation Network (NextItNet), Session-based Recommendation with Graph Neural Networks (SG-GNN), Association Rule (AR), Session-Based k-Nearest Neighbor (SKNN), Vector Multiplication Session-Based k-Nearest Neighbor (VSKNN), Association Rule with Top-k Quartile Filtering (ART-Q), minimum confidence threshold (minconf), and Learning Rate (lr).

The configuration for each method follows the default settings provided by the respective authors, summarized in Table II. For neural models, training is performed with 10 and 50 epochs, and the highest-performing result is selected as the final outcome. The learning rate (lr) determines the step size during optimization. Parameters, such as layers, kernel size, and dilation, reflect the architectural depth and the model's ability to capture sequential dependencies [36, 37]. In the traditional AR method, a minsup of 1 means that any itemset appearing at least once is included. No minimum confidence threshold (minconf) threshold is applied, allowing even low-confidence rules to be considered [14]. Regarding the SKNN and VSKNN models within the KNN family, the parameter k represents the number of neighboring sessions used, while n specifies the number of candidate items considered for recommendation. VSKNN employs cosine similarity to measure session closeness and applies Inverse Document Frequency (IDF) weighting to reduce popularity bias and enhance recommendation relevance [4, 38]. All models are evaluated based on

HR@20, MRR@20, and NDCG@20. In addition, Q-HR@20 and A-HR@20 are used exclusively to assess ART-Q.

Three complementary evaluation approaches are conducted in the research. First, 6 variations in quartile proportions are applied to ART-Q across 4 segments. The V1 variant represents a uniform distribution with equal weights of 0.25 assigned to all quartiles. The V2 through V5 variants emphasize the highest proportion in Q1–Q4 at 0.7, with 0.1 assigned to the remaining quartiles. Then, the V6 variant uses a distribution derived from empirical observations, where a significant portion of target-item hits occurred in the third and fourth quartiles. This variation experiment is designed as a form of design-choice ablation, in which alternative quartile-weighting configurations are systematically tested to examine their impact on model performance. Similar ablation strategies have recently been employed in active learning and heuristic optimization to evaluate the effect of specific design decisions on overall effectiveness [39, 40].

Second, the evaluation of baselines and the best-performing ART-Q variants on each dataset follows established practices in session-based recommendation research, as demonstrated by widely used models such as GRU4Rec, SR-GNN, and NextItNet. In prior studies, they also evaluate recommendation performance using ranking-based metrics, particularly HR, MRR, and NDCG. Accordingly, the same metrics are adopted in this research to enable a consistent comparison with these baseline approaches.

Finally, descriptive statistical analysis of the quartiles that contribute most to metric improvements replaces statistical significance testing, thereby clarifying the behavior of ART-Q. Boxplots visualize the distributions of consequent sets and hit rates across quartiles to highlight how different strata influence ranking quality. This complementary analysis reinforces the ablation findings by demonstrating the effect of varying quartile weights and uncovering the underlying distributional patterns that explain the performance gains of ART-Q. Furthermore, this analysis identifies specific session cases in which ART-Q fails to achieve a hit, offering insight into system limitations.

The SREC suite and the ART-Q framework are executed on a laptop with an Intel Core i7 processor and a GeForce GTX 1650 GPU. In contrast, NextItNet is run on Google Colab using an A100 GPU to accelerate training on the YOO dataset, which proves slow on the local machine. Although the authors of NextItNet recommend splitting long sessions in the YOO dataset into sub-sessions [33], this strategy is ultimately not pursued due to excessive RAM usage on the Colab server.

III. Results and Discussion

A. Results

The ablation results comparison for ART-Q variants on the YOO and MPM datasets is shown in Figs. 2 and 3. The HR, Q-HR, and A-HR metrics are combined within a single chart. Meanwhile, MRR and NDCG are shown separately.

On the YOO dataset, the ablation study of ART-Q variants demonstrates how different quartile weightings affect performance. In the uniform distribution where each quartile receives a weighting of 0.25 in variant V1, ART-Q achieves a balanced baseline reflecting an HR@20 of 0.6357, a Q-HR@20 of 0.6638, and an A-HR@20 of 1.0081. When the weighting is skewed toward the early quartiles in variants V2 and V3, which emphasize Q1 or Q2 at 0.7, HR@20 declines slightly to 0.6052–0.6426. Furthermore, both Q-HR@20 and A-HR@20 fall below or close to the uniform baseline, indicating weaker generalization when recommendations rely heavily on top-ranked strata.

Meanwhile, shifting emphasis toward the later quartiles in variants V4 and V5, where Q3 or Q4 is emphasized at 0.7, consistently improves HR@20 to between 0.6638 and 0.6744, Q-HR@20 to between 0.7136 and 0.7372, and A-HR@20 to between 1.0529 and 1.0822. Further, the empirically derived distribution in variant V6 using 0.01, 0.09, 0.5, and 0.3 delivers the best overall performance, attaining an HR@20 of 0.6899, a Q-HR@20 of 0.7958, and an A-HR@20 of 1.1183. This performance confirms that stronger weighting of deeper quartiles enables ART-Q to retrieve relevant items that would otherwise be overlooked in traditional top-k approaches.

An A-HR value exceeding 1.0 indicates that multiple antecedents contribute to improving the hit rate across multiple consequent sets, surpassing the number of cases in the test sessions. This contribution highlights a key distinction from traditional AR, which relies solely on the last single antecedent to determine the next item. By using multiple antecedents across quartiles, ART-Q achieves higher agreement with the ground-truth item.

On the MPM dataset, the ablation study shows that the uniform distribution in variant V1 delivers the highest overall performance, achieving the highest HR@20 (0.7498), MRR@20 (0.2717), and NDCG@20 (0.3260), while also achieving high Q-HR@20 (0.8157) and A-HR@20 (0.9676). Skewing the weighting toward Q1 through Q3 in variants V2 to V4 leads to consistent reductions across all metrics, indicating that overemphasizing the early quartiles weakens the model's ability to capture the broader dynamics of news consumption.

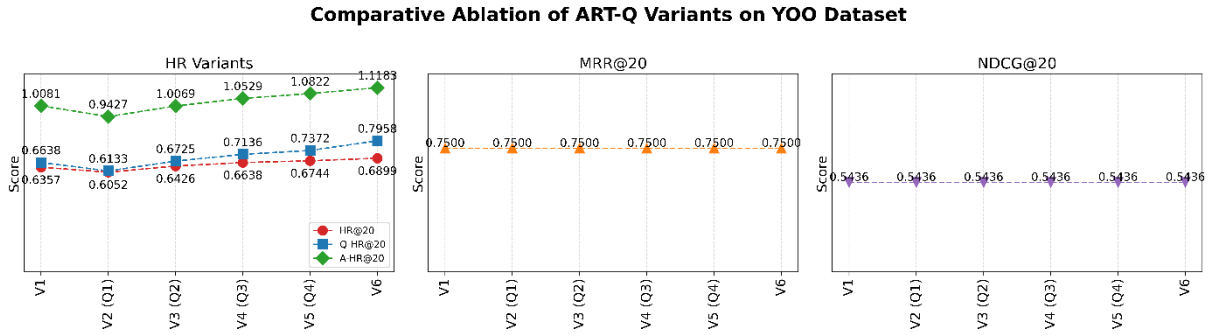

Fig. 2. Comparative ablation of Association Rule with Top-k Quartile Filtering (ART-Q) variants on YooChoose (YOO) dataset. Note: Hit Rate (HR), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG).

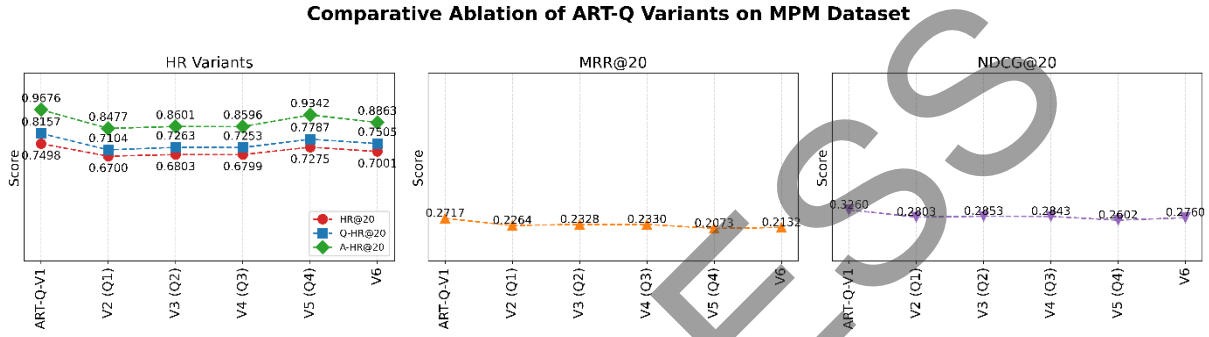

Fig. 3. Comparative ablation of Association Rule with Top-k Quartile Filtering (ART-Q) variants on Malang Posco Media (MPM) dataset. Note: Hit Rate (HR), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG).

Emphasizing Q4 in variant V5 improves HR@20 to 0.7275 and yields a relatively high Q-HR@20 of 0.7787 but suffers in MRR@20 at 0.2073 and NDCG@20 at 0.2602. This divergence suggests diminished ranking precision despite better retrieval. The empirically driven weighting in variant V6 produces moderate results, including an HR@20 of 0.7001, an MRR@20 of 0.2132, and an NDCG@20 of 0.2760, outperforming skewed early-quartile variants but still lagging behind variant V1.

Overall, these findings confirm that balanced quartile weighting is effective in dynamic domains (e.g., news recommendation), where user interests are distributed more evenly across quartiles. Based on these results, variant V6 and V1 are selected for comparison with the respective baseline models on the YOO and MPM datasets. The comparative results are presented in Figs. 4 and 5, respectively. In the YOO dataset, non-neural models such as AR, SKNN, and VSKNN perform competitively, while neural models achieve slightly higher scores in some metrics but do not consistently dominate. The ART-Q-V6 achieves the best overall performance across the HR, MRR, and NDCG metrics. Meanwhile, in the MPM dataset, neural models show weak results, whereas non-neural methods perform better, with AR emerging as the best baseline. The ART-Q-V1, with uniform quartile distribution, achieves the best overall performance. The underlying reasons for these improvements are elaborated in the Discussion section.

B. Discussion

On the YOO dataset, non-neural models including AR, SKNN, and VSKNN perform competitively. VSKNN achieves the best results (HR@20 of 0.6429, MRR@20 of 0.2952, and NDCG@20 of 0.4281). Surprisingly, this performance on HR@20 matches or exceeds neural models exemplified by GRU4Rec, NextItNet, and SR-GNN, achieving HR@20 of 0.6223, 0.6690, and 0.6789, respectively.

Among the evaluated neural models, SR-GNN achieves the highest HR@20 of 0.6789 and NDCG@20 of 0.4579, demonstrating strong capacity to capture global session structure and long-range item transitions. NextItNet achieves the highest MRR@20 of 0.3581, indicating superior precision in ranking ground-truth items at early positions, a benefit enabled by a convolutional architecture designed for long-term dependencies. GRU4Rec, despite being introduced

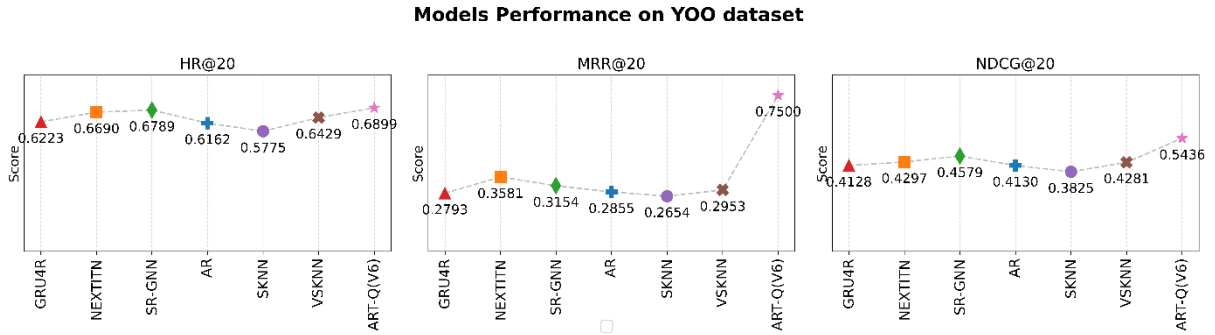

Fig. 4. Model performance on the YooChoose (YOO) dataset. Note: Hit Rate (HR), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG).

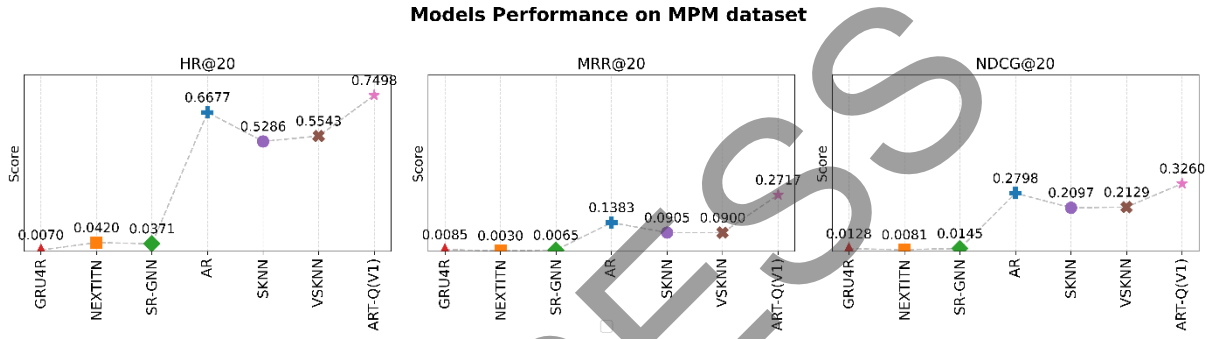

Fig. 5. Model performance on the Malang Posco Media (MPM) dataset. Note: Hit Rate (HR), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG).

earlier in the design, remains competitive across all metrics, achieving HR@20 of 0.6223, MRR@20 of 0.2793, and NDCG@20 of 0.4128, validating the ongoing utility of recurrent connections for modeling short-term sequential patterns. Overall, each neural model exhibits distinct strengths depending on the ranking objective, but none consistently dominates across all metrics, highlighting the structural trade-offs inherent in deep learning architectures.

In contrast, ART-Q-V6, configured with customized quartile proportions where p_1 is 0.01, p_2 is 0.09, p_3 is 0.5, and p_4 is 0.3, consistently outperforms all baselines, achieving HR@20 of 0.6899, NDCG@20 of 0.5436, and a fixed MRR@20 of 0.7500. Compared with the traditional AR method, this corresponds to a +11.9% gain in HR, a +31.6% gain in NDCG, and more than a 2-fold improvement in MRR. These substantial margins highlight ART-Q’s effectiveness in ranking relevant items earlier by successfully overcoming the limitations of single-antecedent rules through stratified attention.

On the MPM dataset, neural models perform notably weak performance across all metrics. GRU4Rec achieves only HR@20 of 0.0070, MRR@20 of 0.0085, and NDCG@20 of 0.0128, indicating limited ability to capture relevant next-item signals in a non-sequential, rapidly evolving domain. Although NextItNet is designed for long-range dependencies, it records the highest HR@20 of 0.0420 among the three, but also has the lowest MRR@20 (0.0030). The results suggest that the recommended items are rarely positioned near the top. SR-GNN achievs a slightly better NDCG@20 of 0.0145, reflecting modest improvements in ranking quality, yet remains suboptimal in both retrieval and ranking precision. These results highlight the profound challenge that deep learning models face in dynamic domains, such as news consumption, where rigid sequential assumptions and static embeddings often fail to generalize to transient user interests and shifting content landscapes.

Non-neural models deliver markedly stronger performance than their neural counterparts on the MPM dataset. The AR method achieves the best results across all metrics, yielding HR@20 of 0.6677, MRR@20 of 0.1383, and NDCG@20 of 0.2798, highlighting an inherent robustness in capturing meaningful item transitions in dynamic content environments such as digital news. While SKNN and VSKNN score lower in retrieval, posting HR@20 values of 0.5286 and

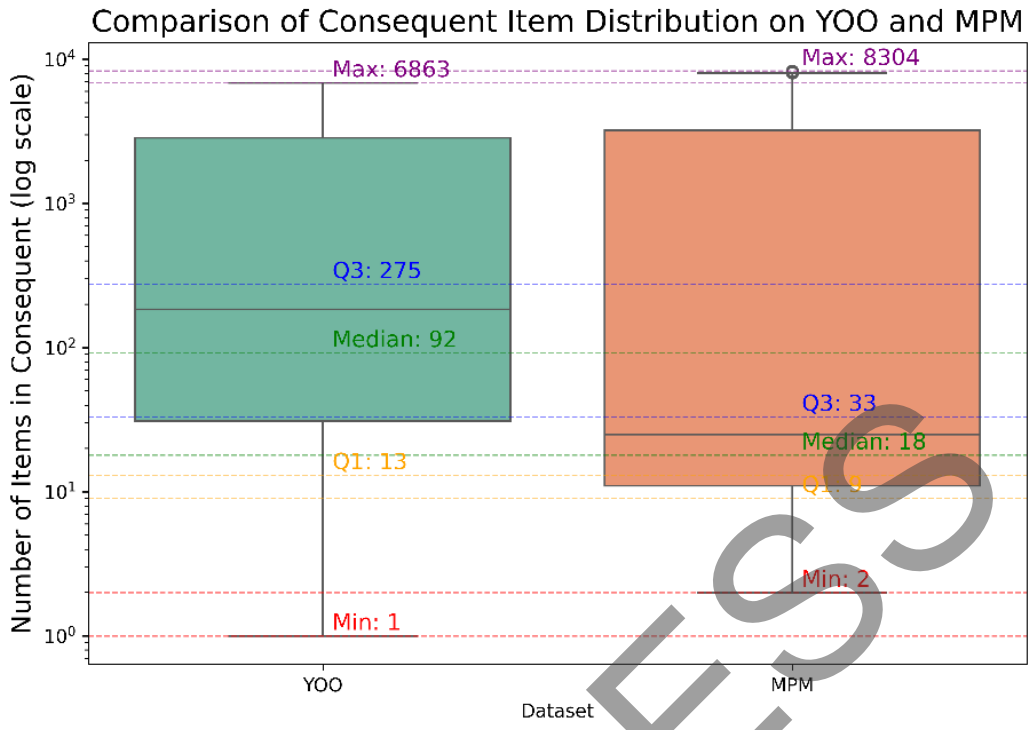

Fig. 6. Comparison of consequent items distribution in (A) YooChoose (YOO) and (B) Malang Posco Media (MPM).

0.5543, respectively, both models perform comparably in ranking quality, maintaining MRR@20 near 0.09 and NDCG@20 near 0.21. Overall, these results confirm that traditional rule-based approaches remain highly competitive for session-based news recommendation, especially when responsiveness and ranking consistency are prioritized over long-range sequence modeling.

Within the ART-Q framework, the uniform distribution of variant V1 produces the most balanced results, establishing HR@20 of 0.7498, MRR@20 of 0.2717, and NDCG@20 of 0.3260. This outcome suggests that user interests in news consumption are best captured when candidate selection is evenly distributed across quartiles rather than skewed toward specific strata. It highlights the importance of quartile design choices for domains characterized by rapidly changing content.

The novel metrics of Q-HR and A-HR in the research are proven effective at capturing the architectural ability of ART-Q to leverage both quartile-level and antecedent-level contributions. Specifically, Q-HR quantifies retrieval success across quartiles, thereby highlighting how relevant items can be ranked within different quartile strata. Meanwhile, A-HR reflects the contribution of multiple antecedents within a session to improve overall ranking consistency.

The underlying mechanisms driving these achievements are revealed through analyses of consequent set distributions, rule weights, and hit-rate allocation across quartiles. As shown in Fig. 6, subsequent set sizes vary substantially, with the YOO dataset reaching 6,863 items (median 92) and the MPM dataset reaching 8,304 items (median 18). This variance underscores that a single antecedent may generate candidate pools spanning several orders of magnitude, rendering naive top-k selection prone to overfitting toward popular consequents. Complementing this, Fig. 7 shows that rule weights are heavily skewed in YOO, where Q3 is 3,199, but more evenly distributed in MPM, where Q3 measurement is 492.6, and the median is 197.1. This contrast reflects distinct domain-specific dynamics, i.e., high popularity bias in e-commerce versus broader topical spread in news. These findings directly motivate the quartile-based filtering of ART-Q, which distributes attention across multiple strata to reduce overconcentration on popular items.

Figure 8 presents the heatmap of quartile contributions in retrieving the ground-truth item during inference. Each row corresponds to a test session, while columns represent the four quartiles. Blue cells

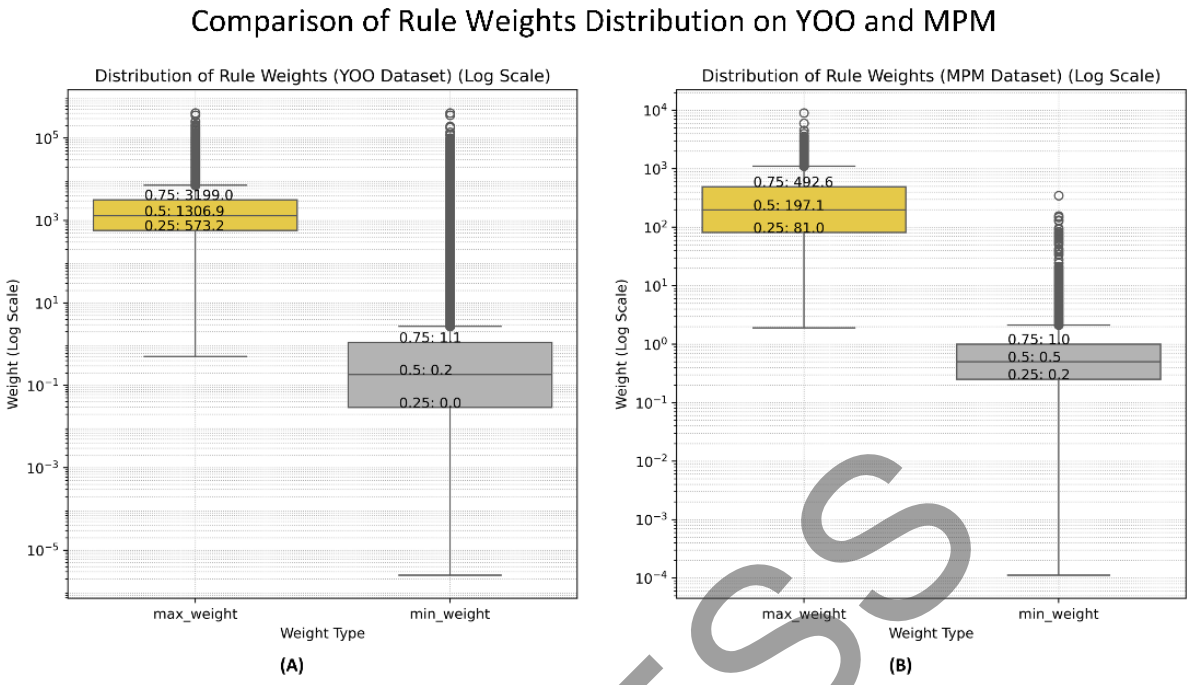

Fig. 7. Comparison of rule weights distribution in (A) YooChoose (YOO) and (B) Malang Posco Media (MPM).

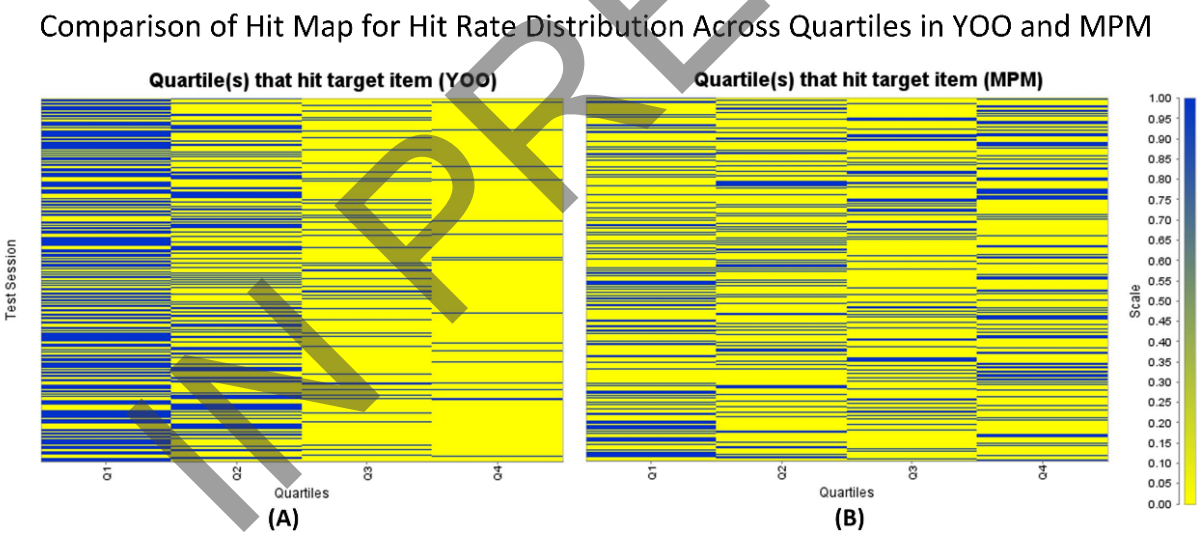

Fig. 8. Comparison of hit map for hit rate distribution across quartiles in (A) YooChoose (YOO) and (B) Malang Posco Media (MPM).

denote successful hits and yellow cells indicate misses. The visualization highlights how the hit distribution is spread across quartiles, depending on the dataset and the ART-Q variant used. For the YOO dataset, results are shown for ART-Q-V6, while for MPM the visualization is based on ART-Q-V1, both being the best-performing configurations for the respective domains.

In the YOO dataset using variant V6, while a majority of hits occur in Q1 (56.8%), over 40% arise from Q2–Q4, indicating that relevant items often lie beyond the highest-weighted rules. In the MPM dataset using variant V1, hits are balanced across all quartiles, with nearly half retrieved from Q3 and Q4, highlighting the importance of deeper strata in dynamic domains. Interpreted through ablation, these patterns confirm that relying solely on Q1 reduces ART-Q to behavior resembling the traditional AR method, thereby

overlooking valuable signals from lower quartiles and leading to weaker retrieval and ranking quality.

On the other hand, ART-Q also encounters cases where test sessions fail to hit the ground-truth item. Although the training and testing sets are prepared that every test item appears in the training set to avoid cold-start scenarios, three conditions may still lead to misses. In the first scenario, an antecedent x may lack a consequent set containing the ground-truth y because the pattern (x, y) never co-occurs in the training data. In the second scenario, the consequent set may contain uniform weights, with the ground-truth item falling outside the top-k. Thus, it is excluded from any quartile. In the third scenario, although the ground-truth y is in the consequent set, the relatively small weight places it beyond the top-k. This condition is confirmed when k is increased, for instance to k = 50, and the ground-truth item is successfully covered.

Intuitively, the second and third cases can be mitigated by adaptive quartile proportions. However, the current version of ART-Q uses a static and uniform configuration across sessions, resulting in unavoidable misses. The first case corresponds to a classical cold-start scenario, which produces empty recommendations [13, 41]. In practice, this can be addressed by still offering the available top-k items within each quartile, since the system does not yet know the ground-truth next item in real-world applications. Nevertheless, the top-k consequent items retain contextual relevance with the antecedent since associated weights are derived from confidence and support measures. Even when the ground-truth item is absent, these candidates remain semantically plausible, reflecting observed co-occurrence patterns in the training data. This property suggests that ART-Q can still provide meaningful recommendations even when perfect accuracy is not achieved.

While the research primarily focuses on ranking quality using standard metrics including HR, MRR, and NDCG, aspects of recommendation diversity are outside the current scope. Because the datasets used here consist only of anonymized item IDs, diversity cannot be directly assessed. A more comprehensive evaluation will require incorporating item metadata, such as product names and descriptions in the e-commerce domain or article titles and contents in the news domain. Such metadata will enable the measurement of content-level diversity, novelty, and long-tail exposure, complementing the ranking-focused evaluation presented in this research.

From a resource utilization perspective, neural-based methods require significantly longer training times. GRU4Rec and SR-GNN demand several hours on a standard laptop GPU, while NextItNet requires substantial computational resources, taking approximately 3.5 hours on an A100 GPU within Google Colab. In contrast, non-neural approaches, including ART-Q, complete training and inference within less than two minutes on a standard laptop CPU, rendering the framework highly efficient and practical for large-scale deployments.

IV. Conclusion

The research introduces ART-Q, a quartile-based ranking framework designed to improve recommendation ranking quality in session-based recommendation systems. Across two domains, i.e., YOO e-commerce and MPM digital news, ART-Q consistently outperforms classical rule-based and KNN baselines and achieves competitive results relative to neural models, e.g., GRU4Rec, NextItNet, and SR-GNN. By partitioning consequent sets into quartiles and applying multiple top-k recommendations across these strata, ART-Q reduces ranking ambiguity while increasing the likelihood that relevant items appear earlier in the list. This strategy improves HR, MRR, and NDCG while maintaining a lightweight and computationally efficient design suitable for real-time applications. Furthermore, new evaluation metrics at the quartile and antecedent levels are introduced, allowing for more fine-grained analysis aligned with the multiple top-k recommendation approach.

However, several limitations must be acknowledged. First, the current implementation relies on static quartile partitioning, which may limit adaptability to evolving session dynamics. Second, the datasets used only include anonymized item IDs, making it impossible to evaluate content-level diversity or novelty beyond ranking-based metrics. Third, while training and testing splits ensure item overlap, ART-Q still encounters cases where ground-truth items are not retrieved, indicating challenges with cold-start and sparse co-occurrence patterns.

ART-Q has the potential to be expanded in a variety of ways through future research. One potential approach is to develop adaptive quartile weighting schemes that dynamically adjust to session characteristics or item popularity trends. To more explicitly assess diversity, novelty, and long-tail exposure, another method that can be applied is to incorporate content metadata (e.g., product descriptions and article titles), as these aspects are beyond the scope of this study.

Acknowledgement

The authors gratefully acknowledge the financial support from the Directorate of Research, Technology, and Community Service (DRTPM), Ministry of Education, Culture, Research, and Technology of the

Republic of Indonesia. This work was funded through the 2024 Regular Fundamental Research Grant (No. 109/E5/PG.02.00.PL/2024). The authors also gratefully acknowledge Malang Posco Media (MPM) for its valuable support as a research partner, particularly for providing the data and facilitating effective coordination throughout the research process.

Author Contribution

Conceived and designed the analysis, T. M. A.; Collected data, K. A.; Contributed the data and analysis tools, K. A.; Performed the analysis, T. M. A. and W. A. D.; and Wrote the paper, T. M. A. and W. A. D.

Data Availability

The YOO dataset used in this study is publicly available online for research purposes. Meanwhile, the MPM dataset is not publicly available due to proprietary restrictions by the research partner, Malang Posco Media (MPM). However, the MPM dataset is available from the corresponding author, Tubagus Mohammad Akhriza, upon reasonable request for academic and non-commercial research purposes.

References

[1] S. Raza and C. Ding, "News recommender system: A review of recent progress, challenges, and opportunities," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 749–800, 2022.

[2] D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," *Journal of Big Data*, vol. 9, pp. 1–36, 2022.

[3] X. Huang, Y. He, B. Yan, and W. Zeng, "Fusing frequent sub-sequences in the session-based recommender system," *Expert Systems with Applications*, vol. 206, 2022.

[4] M. Ludewig, N. Mauro, S. Latifi, and D. Jannach, "Empirical analysis of session-based recommendation algorithms," *User Modeling and User-Adapted Interaction*, vol. 31, pp. 149–181, 2021.

[5] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A survey on session-based recommender systems," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, pp. 1–38, 2021.

[6] K. Wu and K. Chi, "Enhanced e-commerce customer engagement: A comprehensive three-tiered recommendation system," *Journal of Knowledge Learning and Science Technology*, vol. 2, no. 3, pp. 348–359, 2023.

[7] Z. Zhu *et al.*, "Understanding or manipulation: Rethinking online performance gains of modern recommender systems," *ACM Transactions on Information Systems*, vol. 42, no. 4, pp. 1–32, 2024.

[8] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2016. [Online]. Available: https://arxiv.org/abs/1511.06939

[9] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, "A simple convolutional generative network for next item recommendation," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. Melbourne, VIC, Australia: Association for Computing Machinery, Feb. 11–15 2019, pp. 582–590.

[10] S. Wu *et al.*, "Session-based recommendation with graph neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 346–353, 2019.

[11] Z. Wang, "Empowering few-shot recommender systems with large language models-enhanced representations," *IEEE Access*, vol. 12, pp. 29 144–29 153, 2024.

[12] W. Li, Y. Cai, M. H. Hanafiah, and Z. Liao, "An empirical study on personalized product recommendation based on cross-border e-commerce customer data analysis," *Journal of Organizational and End User Computing (JOEUC)*, vol. 36, no. 1, pp. 1–16, 2024.

[13] S. Z. U. Hassan, M. Rafi, and J. Frnda, "GCZRec: Generative collaborative zero-shot framework for cold start news recommendation," *IEEE Access*, vol. 12, pp. 16 610–16 620, 2024.

[14] T. N. Dinh, B. D. Nguyen, X. D. Vu, and Q. D. Truong, "Improve medicine prescribing performance using recommendation systems," in *International Conference on Intelligent Systems and Data Science*. Can Tho, Vietnam: Springer, Nov. 11–12, 2023, pp. 304–312.

[15] M. Ludewig and D. Jannach, "Evaluation of session-based recommendation algorithms," *User Modeling and User-Adapted Interaction*, vol. 28, pp. 331–390, 2018.

[16] H. S. Sheu and S. Li, "Context-aware graph embedding for session-based news recommendation," in *Proceedings of the 14th ACM Conference on Recommender Systems*. Virtual: Association for Computing Machinery, Sep. 22–26, 2020, pp. 657–662.

[17] R. Yeganegi, S. Haratizadeh, and M. Ebrahimi, "STAR: A session-based time-aware recommender system," *Neurocomputing*, vol. 573, 2024.

[18] T. M. Akhriza and I. D. Mumpuni, "A time-window approach to recommending emerging and on-the-rise items," in *2022 Seventh Interna-*

tional Conference on Informatics and Computing (ICIC). Bali, Indonesia: IEEE, Dec. 8–9, 2022, pp. 1–8.

[19] F. Antonello, P. Baraldi, E. Zio, and L. Serio, "A novel metric to evaluate the association rules for identification of functional dependencies in complex technical infrastructures," *Environment Systems and Decisions*, vol. 42, pp. 436–449, 2022.

[20] J. H. Chowdhury, M. B. Hossain, M. S. Kaiser, and M. S. Arefin, "Performance comparisons in association rule mining over public datasets," in *Proceedings of the International Conference on Big Data, IoT, and Machine Learning: BIM 2021*. Cox's Bazar, Bangladesh: Springer, 2022, pp. 761–775.

[21] D. Liu, "Design of data mining system for sports training biochemical indicators based on artificial intelligence and association rules," *International Journal of Data Mining and Bioinformatics*, vol. 28, no. 3-4, pp. 236–256, 2024.

[22] L. Wang, L. Gui, and P. Xu, "Incremental sequential patterns for multivariate temporal association rules mining," *Expert Systems with Applications*, vol. 207, 2022.

[23] J. Li and S. Bao, "Personalized recommendation method of e-commerce products based on in-depth user interest portraits," *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 19, no. 1, pp. 1–15, 2024.

[24] Z. Chen, W. Gan, J. Wu, K. Hu, and H. Lin, "Data scarcity in recommendation systems: A survey," *ACM Transactions on Recommender Systems*, vol. 3, no. 3, pp. 1–31, 2025.

[25] N. Liu and J. Zhao, "Recommendation system based on deep sentiment analysis and matrix factorization," *IEEE Access*, vol. 11, pp. 16 994–17 001, 2023.

[26] S. Kanwal, S. Nawaz, M. K. Malik, and Z. Nawaz, "A review of text-based recommendation systems," *IEEE Access*, vol. 9, pp. 31 638–31 661, 2021.

[27] J. Polohakul, E. Chuangsuwanich, A. Suchato, and P. Punyabukkana, "Real estate recommendation approach for solving the item cold-start problem," *IEEE Access*, vol. 9, pp. 68 139–68 150, 2021.

[28] A. Gharahighehi and C. Vens, "Diversification in session-based news recommender systems," *Personal and Ubiquitous Computing*, vol. 27, pp. 5–15, 2023.

[29] G. De Souza Pereira Moreira, S. Rabhi, J. M. Lee, R. Ak, and E. Oldridge, "Transformers4rec: Bridging the gap between NLP and sequential/session-based recommendation," in *Proceedings of the 15th ACM conference on recommender systems*. Amsterdam, Netherlands: Association for Computing Machinery, Sep. 27–Oct. 1, 2021, pp. 143–153.

[30] P. Symeonidis, L. Kirjackaja, and M. Zanker, "Session-based news recommendations using SimRank on multi-modal graphs," *Expert Systems with Applications*, vol. 180, 2021.

[31] M. Karimi, B. Cule, and B. Goethals, "Leveraging sequential episode mining for session-based news recommendation," in *International Conference on Web Information Systems Engineering*. Melbourne, VIC, Australia: Springer, Oct. 25–27, 2023, pp. 594–608.

[32] A. Smets, J. Hendrickx, and P. Ballon, "We're in this together: A multi-stakeholder approach for news recommenders," *Digital Journalism*, vol. 10, no. 10, pp. 1813–1831, 2022.

[33] F. Yuan, "Generative model for sequential recommendation based on Convolution Neural Networks (CNN)," 2024. [Online]. Available: https://github.com/fajieyuan/WSDM2019-nextitnet

[34] Malte, N. Mauro, S. Latifi, and T. A. Motta, "Python-based framework for building and evaluating session-based and session-aware recommender systems," 2021. [Online]. Available: https://github.com/rn5l/session-rec

[35] S. Latifi, N. Mauro, and D. Jannach, "Session-aware recommendation: A surprising quest for the state-of-the-art," *Information Sciences*, vol. 573, pp. 291–315, 2021.

[36] Y. Ouyang, H. Long, R. Gao, and J. Liu, "Course recommendation model based on layer dropout graph differential contrastive learning," *IEEE Access*, vol. 12, pp. 7762–7774, 2024.

[37] L. H. Baniata, S. Kang, M. A. Alsharaiah, and M. H. Baniata, "Advanced deep learning model for predicting the academic performances of students in educational institutions," *Applied Sciences*, vol. 14, no. 5, pp. 1–19, 2024.

[38] M. Ferrari Dacrema, S. Boglio, P. Cremonesi, and D. Jannach, "A troubling analysis of reproducibility and progress in recommender systems research," *ACM Transactions on Information Systems (TOIS)*, vol. 39, no. 2, pp. 1–49, 2021.

[39] R. M. Fajri, A. Saxena, Y. Pei, and M. Pechenizkiy, "FAL-CUR: Fair active learning using uncertainty and representativeness on fair clustering," *Expert Systems with Applications*, vol. 242, pp. 1–12, 2024.

[40] Y. Koguma, "Tabu search-based heuristic solver

for general integer linear programming problems," *IEEE Access*, vol. 12, pp. 19 059–19 076, 2024.

[41] J. Ahamed, M. N. Noori, and M. Ahmed, "Matrix factorization and cosine similarity based recommendation system for cold start problem in e-commerce industries," *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 775–787, 2024.

APPENDIX

The Appendix can be seen in the next page.

Pseudocode: ART-Q Recommendation Framework

Input:
- T = Training dataset of user sessions
- U = Test dataset of sessions
- k = top-k recommendation required
- p1, p2, p3, p4 = proportion of quartiles, with sum of pi = 1

Output:
- REC # dictionary of all recommendations
- HR@k, QHR@k, AHR@k, MRR@k, NDCG@k # Evaluation metrics

----- Modeling Phase -----
incremental rule generation
1. Initialize:
   FI = {} #frequent itemset dictionary
   R = {} # rule dictionary
2. For each session s in T do
   # find all frequent 1-itemsets FI
   For each single item x in s:
   if {x} not in FI: FI[{x}] = 1
   else: FI[{x}] += 1
   # find all frequent 2-itemsets FI
   For each pair {x, y} in s:
   if {x, y}in FI: FI[{x, y}] = 1
   else FI[{x, y}] += 1

Rule dictionary functions as a data model built from training data

3. For 2-itemsets (x, y) in FI:
   support = FI[{x, y}]
   confidence = support / FI[{x}]
   weight w = confidence * FI[{y}]
   if x not in R : R[x] = {}
   R[x][y] = w
4. Return R and FI

----- Inference and Evaluation Phase -----
5. Initialize:
   HR=0, QHR=0, AHR=0, MRR=0, NDCG=0
   REC = {} # REC[sid][x][Qi] = Topk_Q, sid is the session ID

6. # Session loop
   For each session u in U do
   a. Split u into u_x = {x1,...,xi}, y = xi+1
   b. REC[sid] = {} # dict. for each sid
   c. # Reset per-session counters:
   REC[sid][x] = {} # dict. for each x
   quartile_hit = 0
   antecedent_hit = 0
   session_MRR = 0
   session_NDCG = 0
   d. # Antecedent loop
   For each antecedent x in u_x do
   Y = R[x] # consequents of x
   Sort Y by w(x,y) in descending order
   Partition Y into quartiles Q = [Q1, Q2, Q3, Q4] with proportions p1..p4
   e. # Quartile loop, for generating multiple top-k items across quartiles
   For each Qi in Q do
   Topk_Q = select top-k items in Qi
   REC[sid][x][Qi] = Topk_Q

   if y in Topk_Q:
   quartile_hit = 1
   antecedent_hit += 1

   # Quartile-specific MRR (Eq. 13)
   rank = position(y in Topk_Q)
   MRR_Qi = 1 / rank
   # Quartile-specific NDCG (Eq. 14, 9)
   DCG = DCG(Topk_Q, y)
   IDCG = IDCG(y)
   NDCG_Qi = DCG / IDCG

   # Keep the maximum quartile score for this session
   session_MRR = max(session_MRR, MRR_Qi)
   session_NDCG = max(session_NDCG, NDCG_Qi)

   # End of Quartile loop
   QHR += quartile_hit
   AHR += antecedent_hit
   MRR += session_MRR
   NDCG += session_NDCG

   # End Antecedent loop
   if quartile_hit == 1: HR += 1
#End of session loop

7. Normalize evaluation metrics:
   HR = HR / |U| # maximum = 1
   QHR = QHR / |U| # may exceed 1
   AHR = AHR / |U| # may exceed 1
   MRR = MRR / |U| # maximum = 1
   NDCG = NDCG / |U| # maximum = 1
8. Return REC and all evaluation metrics