Balancing Relevance and Redundancy: An Optuna-Tuned QUBO Formulation for Feature Selection in Recommender Systems

Bhavishya Prajapat[1,*], Alapan Kuila[1]

[1]Department of Computer Science and Engineering, Indian Institute Of Information Technology Design and Manufacturing Kurnool, India

Abstract

Feature selection is a critical step in recommender systems, where high-dimensional item representations often introduce redundancy, noise, and computational overhead. In this paper, we present Team Skywalkers' submission to QuantumCLEF Task $1B$, formulating feature selection as a Quadratic Unconstrained Binary Optimization (QUBO) problem that jointly optimizes feature relevance, redundancy reduction, and budget constraints. To address the challenge of manually tuning QUBO penalty weights, we propose a hybrid optimization framework that integrates Bayesian hyperparameter search through Optuna with quantum-inspired optimization. The resulting QUBO instances are solved using both Simulated Annealing (SA) and a D-Wave Quantum Processing Unit (QPU). Experiments on the sparse 100_ICM dataset identify optimal feature subsets at budgets of $k = 48$ and $k = 59$, where the selected features outperform the full 100-feature baseline, achieving a peak $NDCG@10$ of 0.0287. While hardware noise limits the recommendation accuracy obtained on the physical QPU, the quantum hardware demonstrates substantially faster optimization, solving QUBO instances up to 80 times faster than the classical simulator. These results highlight the potential of automated QUBO tuning and quantum-assisted optimization for scalable feature selection in recommender systems.

Keywords

Feature Selection, Quantum Annealing, Recommender Systems, Quadratic Unconstrained Binary Optimization, Hybrid Quantum-Classical Algorithms, Bayesian Optimization, QuantumCLEF

1. Introduction

Feature selection plays a critical role in Information Retrieval (IR) [1] and Recommender Systems (RS) [2, 3], where high-dimensional item metadata leads to sparse, noisy feature spaces. In content-based models, particularly Item-Based K-Nearest Neighbors (KNN) [4], including irrelevant or redundant features degrades similarity computations and reduces recommendation quality, typically measured by Normalized Discounted Cumulative Gain (NDCG) [5]. Effective feature selection is therefore critical for maintaining both ranking accuracy and computational tractability.

Classical feature selection techniques generally fall into two categories: filter [6] and wrapper methods [7, 8]. Filter methods evaluate features independently, failing to account for multi-way interactions and redundancy (multicollinearity) within the feature space. Conversely, wrapper methods, such as Bayesian search algorithms, directly evaluate feature subsets against the model's performance metric. While more accurate, classical wrappers struggle to scale computationally when navigating exponential combinatorial search spaces under strict feature budget constraints.

To overcome these limitations, we reformulate feature selection for the QuantumCLEF task [9, 10, 11] QUBO problem, which maps naturally to the energy minimization performed by Quantum Annealers. The QUBO objective encodes three additive terms: a relevance term rewarding informative features, a redundancy penalty discouraging correlated pairs, and a budget constraint enforcing a target feature count.

CLEF 2026: Conference and Labs of the Evaluation Forum, September 21–24, 2026, Jena, Germany

*Corresponding author.

prajapatbhavishya89@gmail.com (B. Prajapat); alapan.cse@iiitk.ac.in (A. Kuila)

https://github.com/Novaa-a (B. Prajapat)

0009-0006-1624-8766 (B. Prajapat)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

A key challenge in this formulation is the manual calibration of the penalty weights (Lagrange multipliers α, β, λ). We address this by wrapping the QUBO solver in a Bayesian optimization loop using Optuna [12], which automatically tunes these weights by treating the downstream NDCG@10 score as the [1]. The resulting pipeline runs on both digital Simulated Annealing (SA) [13] and physical D-Wave Quantum Processing Units (QPUs) [14], removing the need for manual constraint calibration.

In summary, this paper makes the following key contributions:

- We propose an automated Bayesian-quantum hybrid pipeline that eliminates manual QUBO hyperparameter tuning via Optuna-driven TPE optimization.
- We provide a systematic empirical analysis of optimal feature budget sizes for the qCLEF 100_ICM dataset, identifying two distinct performance peaks at $k = 48$ and $k = 59$.
- We validate our approach on a physical D-Wave QPU and quantify a 35–80$\times$ speed advantage over digital Simulated Annealing.
- We report a cross-team comparison on the qCLEF 2026 leaderboard, showing that our pipeline achieves the highest SA score among all teams while using the smallest feature budget.

The remainder of this paper is organized as follows: Section 2 reviews related work in quantum-assisted feature selection. Section 3 formally defines the dataset topology and combinatorial optimization objective. Section 4 details the proposed QUBO methodology. Section 5 presents the experimental setup and hardware execution parameters. Section 6 provides our comparative evaluation results. Finally, Section 7 concludes the paper and outlines future directions.

2. Related Work

2.1. Quantum Feature Selection in Recommender Systems

Quantum Annealing (QA) has attracted growing attention as an alternative optimization paradigm for feature selection in Information Retrieval and Recommender Systems. Nikitin et al. [15] demonstrated that quantum-assisted feature selection is viable for sparse recommendation datasets. Ferrari Dacrema et al. [16] subsequently developed a family of QUBO formulations—MI-QUBO [17], QUBO-Correlation [17], and QUBO-Boosting [18]—that balance feature relevance against pairwise redundancy in a single energy function compatible with quantum hardware.

Within the qCLEF evaluation labs [9, 10, 11], several teams have applied these ideas to optimize Item-KNN [4] models. Team MALTO extracted feature importance scores from interpretable classifiers to populate the QUBO linear terms. CRUISE [19] similarly derives contribution scores from model explanations, while Performance-Driven QUBO (PDQUBO) [18] directly quantifies the recommendation impact of each feature before annealing. Collectively, these results show that QUBO-based dimensionality reduction of the Item Content Matrix (ICM) can preserve or improve NDCG [5] relative to the full feature set.

2.2. The QUBO Hyperparameter Bottleneck

A persistent challenge in QUBO-based feature selection is the sensitivity of the energy landscape to its penalty weights (Lagrange multipliers). Most qCLEF submissions either tune these manually or use static heuristics—such as the Auto-α estimator of Rodriguez-Lujan et al. [20]—to balance the linear and quadratic terms prior to annealing.

We address this bottleneck by embedding the QUBO solver inside a Bayesian optimization loop using Optuna [12], which tunes α, β, and λ automatically with respect to the downstream NDCG@10 score. This produces a well-calibrated energy landscape without manual intervention.

[1] https://github.com/Novaa-a/qCLEF-2026-skywalkers

3. Problem Definition

The goal of QuantumCLEF Task 1B [9, 21, 10, 11] is to identify a compact feature subset that maximizes downstream recommendation quality for an Item-KNN model. This section describes the dataset structure and formalizes the optimization objective.

3.1. Data Topology

The dataset provided for this task is a private music recommendation dataset encompassing both collaborative user interactions and descriptive item metadata. Let $\mathcal{U}$ represent the set of users and $\mathcal{I}$ represent the set of available items (songs).

The collaborative filtering signals are stored in a sparse User Rating Matrix, denoted as $\mathbf{URM} \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{I}|}$. Within this matrix, each tuple (u, i) indicates a confirmed interaction between user u and item i, serving as the implicit feedback mechanism for the system.

The content-based signals—comprising various audio descriptors, metadata, and tags associated with the songs—are represented by a sparse Item Content Matrix, $\mathbf{ICM} \in \mathbb{R}^{|\mathcal{I}|\times N}$. For our specific evaluation in this work, we strictly utilize the provided `100_ICM` dataset configuration, thereby fixing the total number of initially available features to $N = 100$. Because the dataset represents highly diverse tags, the matrix is inherently sparse; consequently, any missing data coordinates (missing feature values for a given song) are treated as zero. Some features within this raw matrix have been pre-normalized to ensure scale consistency across different descriptor types.

3.2. Feature Selection Objective

In high-dimensional Recommender Systems, using the full, unconstrained feature set often introduces noise and multicollinearity that degrades similarity-based models. A smaller subset of features helps to increase the efficiency of such systems and reduce storage space. The goal is therefore to identify the most informative subset of the available item features.

Mathematically, this feature selection task is framed as a combinatorial optimization problem. The goal is to find an optimal binary decision vector $\mathbf{x} \in \{0,1\}^N$. Each index in this vector acts as a discrete filter:

$$x_f = \begin{cases} 1, & \text{if feature } f \text{ is selected for the model} \\ 0, & \text{if feature } f \text{ is discarded as noise or redundancy} \end{cases} \tag{1}$$

By applying this binary mask to the ICM we obtain a reduced matrix $\mathbf{ICM}'$ containing only the selected features. The challenge is to search the 2^N possible subsets efficiently and find the one that maximizes NDCG@10 on unseen interactions.

4. Methodology

4.1. QUBO

QUBO is the standard mathematical framework underlying both quantum and simulated annealing [14]. A QUBO problem is defined by a binary vector $\mathbf{x} \in \{0,1\}^N$ and a symmetric matrix Q, and seeks the assignment that minimizes the scalar energy $E(\mathbf{x}) = \mathbf{x}^\top Q\mathbf{x}$.

Since quantum annealers operate without explicit hard constraints, all requirements—such as a fixed feature count—must be encoded as soft penalty terms added directly to the objective. The annealer then finds the configuration minimizing the total energy, implicitly trading off the reward terms against the penalties.

4.2. Mapping Feature Selection to QUBO

We map the feature selection problem to a QUBO energy function. Given the binary vector $\mathbf{x}$, where $x_i = 1$ denotes selection of feature i, the objective is:

$$\min_{\mathbf{x}\in\{0,1\}^N} E(\mathbf{x}) = -\alpha \sum_{i=1}^{N} R_i x_i + \beta \sum_{i=1}^{N} \sum_{j>i}^{N} S_{ij} x_i x_j + \lambda \left(\sum_{i=1}^{N} x_i - k \right)^2 \tag{2}$$

The three terms encode competing objectives:

- **Relevance** ($-\alpha \sum_i R_i x_i$): The linear coefficient R_i quantifies the marginal predictive contribution of feature i. Because the annealer minimizes energy, negating R_i rewards the selection of informative features by lowering the energy.
- **Redundancy penalty** ($+\beta \sum_i \sum_{j>i} S_{ij} x_i x_j$): The pairwise coefficient S_{ij} measures the similarity between features i and j. Jointly selecting two highly similar features incurs a positive energy penalty, discouraging redundant selections.
- **Budget constraint** ($+\lambda(\sum_i x_i - k)^2$): A squared penalty enforces the target feature count k. Any deviation from exactly k selected features increases the energy quadratically, making constraint violations energetically unfavorable.

The multipliers α, β, and λ control the relative importance of the three terms. When these are poorly calibrated—for example, if λ is too large—the annealer may satisfy the budget constraint at the expense of feature quality. Tuning these weights is therefore essential for obtaining meaningful solutions.

4.3. Computing QUBO Coefficients: Relevance and Redundancy

Both coefficient matrices are computed directly from the dataset prior to annealing.

Relevance (R_i): We estimate the marginal importance of each feature using a Random Forest [22, 23] trained on the dataset. The model assigns each feature a normalized Gini importance score reflecting its average contribution to reducing impurity across all trees. These scores form the linear coefficient vector $\mathbf{R}$.

Redundancy (S_{ij}): Pairwise redundancy is measured by cosine similarity between feature columns of the ICM. A high similarity between features i and j indicates that they provide overlapping information; this value is stored in S_{ij} and contributes a positive energy penalty when both features are selected jointly.

4.4. Hybrid Optimization: The Optuna-Tuned Pipeline

Sensitivity to the penalty weights α, β, and λ is a known difficulty in QUBO-based feature selection: poor calibration makes it hard to attribute performance changes to the feature subset rather than the energy landscape itself. We address this by embedding the QUBO solver in an automated Bayesian optimization loop using Optuna [12]. As illustrated in Figure 1, this architecture functions as a closed-loop feedback system divided into two primary layers: an Optimization Layer that proposes QUBO parameters, and an Execution Layer that evaluates them. The pipeline operates in two stages:

1. **Weight proposal:** Optuna's Tree-structured Parzen Estimator (TPE) [24] proposes a combination of α, β, and λ.
2. **Solve and evaluate:** The proposed weights define a QUBO matrix, which is minimized by a digital Simulated Annealing (SA) backend. The resulting feature subset is passed to the Item-KNN model, and the validation NDCG@10 score is returned to Optuna as feedback.

Using NDCG@10 as the objective, Optuna iteratively refines the weight combination over 1,000 trials, yielding a well-calibrated QUBO matrix for subsequent quantum hardware execution.

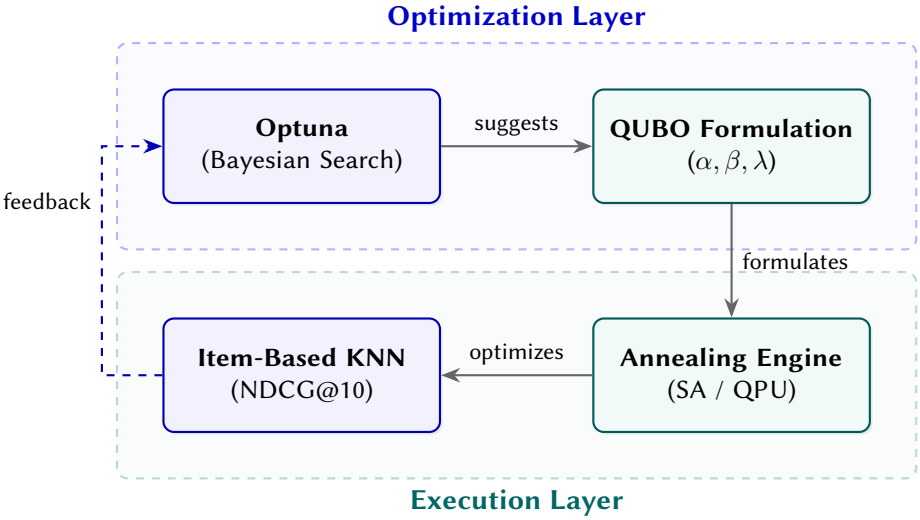

Figure 1: The Proposed Hybrid Bayesian-Quantum Pipeline for qCLEF. The Bayesian optimizer (*Optimization Layer*) tunes the QUBO energy landscape, while the *Execution Layer* solves for the optimal feature subset $(\mathbf{x})$ and validates performance via NDCG@10, creating a closed-loop feedback system.

5. Experimental Setup

To make sure our findings are robust and our performance metrics are not tainted by data leakage, we established a strict evaluation protocol directly within the official QuantumCLEF environment [9, 21, 10, 11]. This section details our data partitioning strategy, the hybrid hyperparameter search, and the quantum hardware execution parameters.

5.1. Datasets

Our experiments utilize the official music recommendation dataset, comprising a User Rating Matrix (URM) with 3,209,730 implicit interactions across 20,428 users and 14,607 items. To evaluate dimensionality reduction, we focus on the highly sparse 100_ICM variant, which represents items using 100-dimensional feature vectors.

To prevent the common methodological flaw of overfitting to the validation set—where tuning and final evaluation rely on the same holdout data—we implemented a strict two-tier partitioning strategy:

- **Optimization Split (80%):** Subdivided into *train_opt* and *val_opt*, this isolated environment exclusively guides the Optuna [12] Bayesian tuner to identify the optimal feature subsets.
- **Final Holdout Split (20%):** Reserved strictly as unseen evaluation data. The selected subsets are tested on this split only at the end of the pipeline to provide an unbiased estimate of real-world generalization.

The extreme sparsity of the 100_ICM matrix makes it an ideal benchmark for quantum-assisted feature selection. By strictly isolating the optimization process from the final holdout, we guarantee that our reported metrics reflect true predictive performance on novel interactions, completely free from data leakage.

Our **classical baseline** is defined as an Item-Based KNN model [4] evaluated on the complete, unfiltered $N = 100$ feature set of the 100_ICM matrix, without any dimensionality reduction. The model is configured with a neighbourhood size of top-$K = 100$, cosine item-item similarity, and a shrinkage parameter of $\tau = 5$ to regularize similarity estimates on sparse vectors. This full-feature configuration serves as the natural reference point against which all quantum-selected feature subsets are evaluated.

5.2. Hybrid Tuning and Simulated Annealing (SA)

Our feature selection budget was constrained to specific sizes (e.g., $k = 48$ and $k = 59$). We selected these targets because our early baseline tests showed they contained the most useful predictive information before the data became too sparse.

To map these budgets into our unconstrained QUBO model without manually guessing the penalty weights, we ran an automated Optuna study over 1,000 trials. In each trial, Optuna suggested configuration weights (such as the redundancy threshold and α scaling factor), built the matrix, and sent it to a digital Simulated Annealing (SA) sampler via the `dwave-neal` backend. The sampler performed 250 reads per matrix to find the lowest-energy state. If the resulting subset exactly matched our budget k, it was evaluated using the Item-Based KNN to get a validation NDCG@10 score. Optuna then used this score to iteratively improve the landscape for the next trial. This iterative feedback loop allowed the system to organically discover the optimal balance between feature relevance and penalty magnitude. Consequently, we avoided the computational bottleneck of an exhaustive grid search while ensuring strict adherence to the predefined budget constraints.

5.3. Quantum Hardware Execution

After the Optuna engine found the best QUBO weights for our target budgets, we moved from classical simulation to physical quantum hardware. The finalized matrices were submitted through the official wrapper to the D-Wave Leap Cloud.

We mapped the QUBO models onto the physical Quantum Processing Unit (QPU) using the `EmbeddingComposite(DWaveSampler())` architecture. Because physical quantum computers experience analog noise and thermal interference, we performed 1,000 quantum reads for each run. We extracted the energy states from all reads, removed any subsets that did not meet the strict size constraint k, and kept the valid combinations with the lowest energy.

Since physical quantum annealers have random variations, a single run cannot guarantee the absolute best result. To check if our method was stable, we repeated the entire pipeline multiple times. We then combined the results to find the five most stable and lowest-energy feature subsets, which we used as our final models for evaluation. Aggregating the results across these independent runs effectively mitigated the inherent stochasticity of the quantum annealing process. This ensemble-like approach ensured that our final selected features were not statistical anomalies but genuinely robust minimum-energy solutions.

Our five submitted configurations are summarized in Table 2. Submissions 003, 010, 011, and 012 all target a budget of $k = 59$ features and correspond to the four lowest-energy, valid subsets found across independent QPU and SA runs with the $k = 59$ QUBO matrix. Submission 016 targets a budget of $k = 48$ features and uses the best subset found with the $k = 48$ configuration. The run numbering reflects the chronological order of Optuna trial selection. All reported NDCG@10 scores correspond to official results as evaluated by the qCLEF 2026 assessment infrastructure on the held-out test set.

6. Results

This section presents the empirical findings of our feature selection pipeline, detailing the discovery of optimal dimensionality budgets and evaluating the comparative performance of digital simulation versus physical quantum execution.

6.1. Dimensionality Exploration and Budget Selection

We ran a 500-trial Bayesian search to find the optimal number of features (k). As shown in Table 1, the search algorithm proved that useful information is concentrated in specific zones. The optimizer completely avoided ranges below 31 and above 80, confirming that these areas represent underfitting

(not enough signal) and feature noise (too much bad data). Based on this, we targeted the two main information pockets: $k = 59$ and $k = 48$.

The data demonstrates that $k = 59$ represents the **tuning-phase optimum**: the Bayesian search aggressively concentrated 35.6% of its total trials on this size, achieving the highest optimization-phase NDCG of 0.0304. However, with a test NDCG of 0.0309, it is slightly outperformed on the final holdout by $k = 48$ (0.0312). This indicates that $k = 59$ is where the optimizer converges most confidently, while $k = 48$ generalizes marginally better to unseen interactions.

Meanwhile, $k = 48$ acts as the **generalization optimum** (Optimal Peak I): it achieves the highest test NDCG across all budgets (0.0312) while using approximately 18% fewer features than $k = 59$. Models at $k = 48$ retain only the most essential core signals, successfully shedding features that contribute noise without genuine predictive value. By mathematically isolating these optimal subsets, we maximize the predictive signal while inherently regularizing the model against structural noise. The hyphen (–) entries in Table 1 indicate budget ranges where insufficient trials were sampled during the 500-trial search to yield a reliable mean test NDCG; the non-zero test values reported for ranges with 0% search density (e.g., 71–80, 81+) were obtained from a fixed initial grid evaluation conducted prior to the adaptive Bayesian phase.

Table 1: Empirical evaluation of feature space dimensionality (k) on 100_ICM dataset. Results aggregated from a 500-trial Bayesian search to identify optimal density zones.

Budget (k)	**Search Density**	**Max Tune NDCG**	**Mean Test NDCG**	**Information State**
$10 - 30$	0.0%	-	0.0248	Underfitting (Insufficient Signal)
$31 - 40$	12.4%	0.0267	0.0285	Rapid Convergence
$41 - 47$	10.4%	0.0292	-	Transition Zone
48	**0.4%**	**0.0256**	**0.0312**	**Optimal Peak I (Generalization Optimum)**
$49 - 55$	4.0%	0.0291	0.0298	Stability Plateau
$56 - 58$	5.8%	0.0300	-	Secondary Convergence
59	**35.6%**	**0.0304**	0.0309	**Optimal Peak II (Tuning Optimum)**
$60 - 70$	31.4%	0.0297	0.0275	Overfitting (Redundancy Accumulation)
$71 - 80$	0.0%	-	0.0275	Redundancy Exhaustion
$81+$	0.0%	-	0.0251	Feature Noise

6.2. Comparative Performance: Simulated vs. Quantum Annealing

Table 2 summarizes the performance of our five final submissions. We compare the digital Simulated Annealing (SA) sampler against the physical Quantum Processing Unit (QPU) using the same Optuna-tuned QUBO matrices.

Table 2: Performance evaluation of individual configurations, detailing NDCG@10, annealing time, and feature space size. Simulated Annealing (SA) and Quantum Annealing (QA) runs are presented independently to highlight hardware variance.

Submission ID	**NDCG@10**	**Annealing Time(μs)**	**Type (S/Q)**	**Features (k)**
012	0.0287	9653001	S	59
010	0.0277	9377061	S	59
011	0.0269	9380512	S	59
016	0.0264	20508778	S	48
003	0.0254	9861894	S	59
Baseline	0.0226	-	-	100
010	0.0211	229306	Q	59
003	0.0185	271926	Q	59
012	0.0169	273842	Q	59
011	0.0166	270101	Q	59
016	0.0155	255336	Q	48

The experimental results validate that our Optuna-tuned QUBO formulations successfully capture the optimization landscape. The digital Simulated Annealing (SA) executions consistently outperformed the dense 100-feature classical baseline (0.0226), reaching a peak NDCG@10 score of 0.0287. This confirms that our targeted dimensionality budgets ($k = 48$ and $k = 59$) effectively filter out feature noise while retaining the most critical predictive signals.

When deployed on physical quantum hardware (QA), we observed a performance pattern typical of the Noisy Intermediate-Scale Quantum (NISQ) era [25]. The physical QPU achieved a maximum NDCG@10 of 0.0211. This decrease in accuracy compared to the digital simulator is partially expected due to analog noise, thermal decoherence, and the limitations of mapping the matrices onto the physical QPU topology through minor embedding. Additionally, because our fine-tuning pipeline was optimized primarily for SA, the approach likely overfits the digital simulator to some extent. Physical quantum annealers involve distinct hardware-native hyperparameters—such as chain strength and annealing schedules—that require independent tuning, likely compounding this performance gap. Notably, all QPU executions in this evaluation fall below the classical baseline of 0.0226, a finding consistent with current NISQ-era hardware limitations and corroborated by the results of all pure-QPU (Q-type) submissions across competing teams in Table 3.

However, the execution metrics reveal a fundamental divergence in time-to-solution. The QPU resolved the energy landscapes in approximately 2.5×10^5 microseconds ($\sim$250 ms). Comparing matched submission pairs, the physical quantum hardware operated roughly 35 to 80 times faster than the digital SA sampler, which required between 9.4×10^6 and 2.1×10^7 microseconds ($\sim$9–21 seconds) to converge on a solution. This speed-accuracy tradeoff is a characteristic hallmark of the NISQ era, where hardware fidelity remains a limiting factor; however, the demonstrated acceleration represents a concrete step toward practical quantum advantage as qubit coherence times and embedding efficiency improve with next-generation QPU architectures.

6.3. Cross-Team Comparison on qCLEF 2026 Leaderboard

To situate our approach within the broader competitive landscape, Table 3 presents a cross-team comparison against all participating groups in the official qCLEF 2026 Task 1B evaluation [10, 11]. For each team, we report the best achieved NDCG@10 score for the Simulated Annealing (SA) and Quantum Annealing (QA) or Hybrid (H) solver type independently, along with the number of features used. To ensure a methodologically sound comparison, we distinguish between pure quantum (Q) and hybrid (H) solver types, as these represent fundamentally different execution paradigms.

Table 3: Cross-team performance comparison on the official qCLEF 2026 Task 1B leaderboard (100_ICM dataset). Best result per team per solver type is reported. QA Type: Q = pure QPU execution, H = Hybrid solver. Bold values denote the overall best in each metric column.

Team	Best SA NDCG@10	SA k	Best QA/H NDCG@10	QA Type	QA k
Skywalkers (Ours)	**0.0287**	**59**	0.0211	Q	59
R2AI	0.0256	97	**0.0245**	H	97
SYZYGY	0.0243	85	0.0152	Q	62
FAST-NUCES	0.0229	75	0.0218	Q	72
Baseline (All Features)	0.0226	100	—	—	—

The leaderboard results confirm two key strengths of our proposed framework. First, our Optuna-tuned QUBO pipeline achieves the highest SA score across all competing teams, reaching an NDCG@10 of 0.0287—a relative improvement of 12.0% over the second-ranked team (R2AI, 0.0256) and 27.0% over the dense 100-feature classical baseline (0.0226). Critically, this leading performance is achieved with only $k = 59$ features, representing a $\mathbf{39}\%$ reduction from the feature space used by R2AI ($k = 97$) and far fewer features than any other competing system. This demonstrates that our automated Bayesian constraint tuning does not simply trade accuracy for efficiency—it achieves superior accuracy while simultaneously enforcing a far more aggressive dimensionality reduction.

On physical quantum hardware (pure QPU execution), our best result of 0.0211 ranks competitively within the Q-type submissions. FAST-NUCES achieves 0.0218 using pure QPU execution with a larger feature budget ($k = 72$), while the top overall QA/H score of 0.0245 belongs to R2AI, whose submissions employ a Hybrid (H) solver rather than pure quantum annealing—a distinction that makes a direct accuracy comparison with our pure QPU executions methodologically inequivalent. The QPU accuracy gap observed in our own results, relative to our SA simulations, is consistent with the hardware noise characteristics discussed in Section 6 and is a shared challenge across all pure-QPU submissions in this evaluation.

7. Conclusion and Future Work

This study demonstrates the viability of utilizing quantum-assisted optimization for feature selection in highly sparse recommender systems. By mapping the feature selection problem into a QUBO formulation and tuning the penalty constraints via a hybrid Bayesian approach, we successfully isolated optimal dimensionality pockets that consistently outperformed dense classical baselines in simulation.

While current hardware noise temporarily limits the absolute recommendation accuracy of physical Quantum Processing Units (QPUs), the massive acceleration in sampling time proves that our learned QUBO parameters reliably and rapidly guide the quantum system toward high-quality minimal energy states. We acknowledge that this SA-to-QPU accuracy gap represents a current limitation of our approach. This discrepancy is likely driven not only by NISQ-era hardware constraints, such as qubit coherence times and embedding fidelity, but also by algorithmic bias. Because our fine-tuning pipeline was optimized primarily via SA feedback, the approach may inherently overfit the digital simulator.

Physical quantum annealers involve a distinct set of hardware-native hyperparameters, such as chain strength and annealing schedules, that require their own independent tuning

Future work will focus on bridging the gap between simulated accuracy and physical hardware execution. A primary objective will be to implement QPU-specific hyperparameter optimization, allowing us to accurately isolate the impact of hardware noise from sub-optimal parameterization. As quantum annealing architectures mature and error-mitigation techniques improve, the analog noise that currently degrades QPU performance is expected to diminish, potentially unlocking both the speed and accuracy advantages simultaneously. Additionally, exploring hybrid quantum-classical solvers will allow us to scale this methodology to significantly larger feature spaces, facilitating the near-instantaneous optimization of massive, enterprise-scale recommendation matrices.

Acknowledgments

We thank the QuantumCLEF 2026 organizing committee—Andrea Pasin, Maurizio Ferrari Dacrema, Washington Cunha, Marcos André Gonçalves, Paolo Cremonesi, and Nicola Ferro—for designing the evaluation infrastructure and providing the dataset used in this work. We also thank the D-Wave Leap team for providing access to quantum hardware through their cloud platform. The code for our pipeline (SA-compatible) is publicly available at https://github.com/Novaa-a/qCLEF-2026-skywalkers.

Generative AI Declaration

The authors declare that Generative AI technologies were utilized during the preparation of this manuscript. Specifically, Large Language Models (LLMs) were used solely as writing assistants to refine grammar, enhance phrasing clarity, and format text for readability. All core methodologies, experimental designs, coding, data analysis, and scientific conclusions are entirely the original work of the authors.

References

[1] O. Egozi, E. Gabrilovich, S. Markovitch, Concept-based feature generation and selection for information retrieval., in: AAAI, volume 8, 2008, pp. 1132–1137.

[2] R. Nembrini, M. Ferrari Dacrema, P. Cremonesi, Feature selection for recommender systems with quantum computing, Entropy 23 (2021) 970.

[3] Y. Afoudi, M. Lazaar, M. Al Achhab, Impact of feature selection on content-based recommendation system, in: 2019 International conference on wireless technologies, embedded and intelligent systems (WITS), IEEE, 2019, pp. 1–6.

[4] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th international conference on World Wide Web, 2001, pp. 285–295.

[5] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of ir techniques, ACM Transactions on Information Systems (TOIS) 20 (2002) 422–446.

[6] N. Sánchez-Maroño, A. Alonso-Betanzos, M. Tombilla-Sanromán, Filter methods for feature selection–a comparative study, in: International conference on intelligent data engineering and automated learning, Springer, 2007, pp. 178–187.

[7] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of machine learning research 3 (2003) 1157–1182.

[8] S. Maldonado, R. Weber, A wrapper method for feature selection using support vector machines, Information Sciences 179 (2009) 2208–2217.

[9] A. Pasin, M. Ferrari Dacrema, P. Cremonesi, N. Ferro, et al., Quantumclef 2024: Overview of the quantum computing challenge for information retrieval and recommender systems at clef, in: CEUR WORKSHOP PROCEEDINGS, volume 3740, CEUR-WS, 2024, pp. 3032–3053.

[10] A. Pasin, M. F. Dacrema, W. Cunha, M. A. Gonçalves, P. Cremonesi, N. Ferro, Overview of quantumclef 2026: The third quantum computing challenge for information retrieval and recommender systems at CLEF, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction - 17th International Conference of the CLEF Association, CLEF 2026, Jena, Germany, September 21-24, 2026, Proceedings, Lecture Notes in Computer Science, Springer, 2026.
[11] A. Pasin, M. F. Dacrema, W. Cunha, M. A. Gonçalves, P. Cremonesi, N. Ferro, Quantumclef 2026: Overview of the third quantum computing challenge for information retrieval and recommender systems at CLEF, in: Working Notes of the Conference and Labs of the Evaluation Forum, CLEF 2026, Jena, Germany, 21-24 September 2026, CEUR Workshop Proceedings, CEUR-WS.org, 2026.
[12] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2623–2631.
[13] S. Kirkpatrick, C. D. Gelatt Jr, M. P. Vecchi, Optimization by simulated annealing, science 220 (1983) 671–680.
[14] C. C. McGeoch, Adiabatic quantum computation and quantum annealing: Theory and practice, Springer Nature, 2022.
[15] A. Nikitin, A. Chertkov, R. Ballester-Ripoll, I. Oseledets, E. Frolov, Are quantum computers practical yet? a case for feature selection in recommender systems using tensor networks, arXiv preprint arXiv:2205.04490 (2022).
[16] M. Ferrari Dacrema, F. Moroni, R. Nembrini, N. Ferro, G. Faggioli, P. Cremonesi, Towards feature selection for ranking and classification exploiting quantum annealers, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 2814–2824.
[17] T. M. Almeida, S. Matos, Towards a hyperparameter-free qubo formulation for feature selection in ir., in: CLEF (Working Notes), 2024, pp. 3054–3063.
[18] J. Niu, J. Li, K. Deng, M. Sanderson, N. Ferro, Y. Ren, Performance-driven qubo for recommender systems on quantum annealers, ACM Transactions on Recommender Systems (2024).
[19] F. Giobergia, C. Savelli, A. Koudounas, E. Baralis, et al., Quantum feature selection from interpretable models using qubo formulation, Working Notes of CLEF (2025).
[20] I. Rodriguez-Lujan, R. Huerta, C. Elkan, C. S. Cruz, Quadratic programming feature selection, The Journal of Machine Learning Research 11 (2010) 1491–1516.
[21] A. Pasin, M. Ferrari Dacrema, W. Cunha, M. A. Gonçalves, P. Cremonesi, N. Ferro, et al., Quantumclef 2025: Overview of the second quantum computing challenge for information retrieval and recommender systems at clef, in: CEUR WORKSHOP PROCEEDINGS, volume 4038, CEUR-WS, 2025, pp. 4086–4104.
[22] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.
[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, the Journal of machine Learning research 12 (2011) 2825–2830.
[24] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, Advances in neural information processing systems 24 (2011).
[25] J. Preskill, Quantum computing in the nisq era and beyond, Quantum 2 (2018) 79.