SPECTRUM OF ENGINEERING SCIENCES
ISSN (E) 3007-3138 (P) 3007-312X
VOLUME 4, ISSUE 3, 2026

A TF-IDF AND LOGISTIC REGRESSION PIPELINE FOR SCHOLARLY ARTICLE CLASSIFICATION AND RECOMMENDATION: IEEE XPLORE BENCHMARK STUDY

Ghazi Irfan[*1], Faraz Ali[2], Ghulam Mustafa[3] and Muhammad Kaleem Ullah Khan[4]

[*1,2,3]*Department of Computer Science, University of Central Punjab (UCP), Lahore, Pakistan*

[4]*Department of Software Engineering and IT, École de technologie supérieure (ÉTS), Montréal, Canada*

[*1]ghazi.irfan@ucp.edu.pk, [2]farazalich@yahoo.edu.pk, [3]ghulammustafa02@ucp.edu.pk, [4]ranakaleem109@gmail.com

DOI: http://doi.org/10.5281/zenodo.23725256

Keywords

Scholarly article classification, scholarly article recommendation, abstract-only text classification, TF-IDF, Logistic Regression, IEEE Xplore *benchmark, content-based recommendation, cosine similarity, calibrated probabilities, classifier-aware re-ranking, supervised machine learning.*

Article History

Received: 10 January 2026
Accepted: 26 February 2026
Published: 31 March 2026

Copyright @Author
Corresponding Author: *
Ghazi Irfan*

Abstract

The rapid growth of scholarly publications has made automated topical organization and recommendation essential for efficient literature search. However, most existing approaches treat classification and recommendation as two separate tasks with independent representations. This paper proposes a unified content-based framework in which a single TF-IDF representation of the article abstract drives both multi-class topical classification and top-k article recommendation. A new benchmark of 11,744 abstracts is constructed from the IEEE *Xplore* digital library in six topical queries. The abstract text and the topical query label are retained for every record, so the entire pipeline operates on abstracts alone without titles, author keywords, or indexer-supplied terms. A preliminary confusion analysis reveals that two queries (Big Data Analysis and Cloud Computing) exhibit near-complete vocabulary collapse and are consolidated into a single class, yielding a five-domain benchmark: Big Data & Cloud Computing, Data Science, Robotics, Wireless Communication, and Breast Cancer. On the classification side, five supervised learners (Logistic Regression, Linear SVM, SGD, k-Nearest Neighbours, and Decision Tree) are compared under identical 80/20 stratified hold-out and 10-fold cross-validation protocols. The grid-searched Logistic Regression attains 85.01% accuracy (weighted $F_1 = 0.850$), and a soft-voting ensemble of Logistic Regression, Linear SVM, and SGD reaches 85.57% (weighted $F_1 = 0.855$). On the recommendation side, the same TF-IDF representation powers a top-10 recommender that achieves MAP@10 $= 0.7664$ with pure cosine ranking. Reusing the classifier's calibrated class probabilities to re-rank cosine candidates lifts Precision@10 by $+12.5$ percentage points (to 0.7715) and MAP@10 by $+7.6$ points (to 0.8428), with consistent gains on NDCG@10 and MRR. The dataset, preprocessing pipeline, trained models, and replication scripts are released to support reproducibility.

Introduction

The volume of scientific publications has grown faster than any individual researcher can read or even survey. Public estimates place the cumulative number of indexed scholarly articles in the tens of millions, with hundreds of thousands of new papers added every month across digital libraries such as IEEE *Xplore*, ACM Digital Library, Springer, and arXiv [1, 2]. The result is the now-familiar problem of *information overload* [3, 4]: locating the small set of relevant papers in a large topical neighborhood is itself a research-time bottleneck.

A common building block of systems that address this overload, from faceted search to personalized paper recommenders, is a topical classifier that maps an abstract to one or more research domains [5, 6]. Reliable abstract classification is what allows downstream components, such as content-based filters or graph-based recommenders, to expand their set of candidates and inject domain priors into the ranking [7, 8]. The performance of these downstream components is therefore upper-bounded by the quality of the topical classifier and the rigor of its preprocessing pipeline.

Despite this, much of the published work on content-based scholarly classification reports headline accuracies on small, narrowly scoped datasets, with limited preprocessing rigor, and without a careful

classifier-by-classifier comparison on a single corpus. Bulut *et al.* [9], the most directly comparable prior study, evaluates a TF-IDF and cosine-similarity recommender on a corpus of 600 articles and reports an accuracy of 71%. Rahman *et al.* [10] extend the benchmarking to the arXiv corpus across many subject categories under a multi-label setting and reporting a best classification accuracy of 69% with TF-IDF and Logistic Regression. The wide variability of the dataset, label granularity, and evaluation protocol across these studies makes it difficult to attribute reported gains to the genuine methodological improvements rather than to the dataset characteristics. Furthermore, both classification and recommendation performance are typically reported in isolation, with few studies releasing a single end-to-end pipeline that performs both tasks on a controlled benchmark.

Three concrete use cases motivate the development of a robust classification-plus-recommendation pipeline for scholarly abstracts. The first is automated faceting of search results in digital libraries, where a topical classifier partitions retrieved articles into the user's domain of interest before re-ranking. The second is cold-start paper recommendation, in which a researcher's reading history is too short to support collaborative filtering, and a content-based pipeline must carry the load alone. The third is automatic indexing of newly published articles in domain-specific repositories where manual curation is not scalable. In all three cases, the practical quality of the downstream system is bounded by the joint quality of the classifier and the retrieval module, which makes a controlled end-to-end study directly valuable.

The present study is organized around four research questions. **RQ1:** On a controlled mid-sized academic benchmark, do the six topical queries used to construct the dataset yield linearly separable classes *from abstract text alone*, or does query-based labeling introduce structural confusion that no amount of preprocessing can resolve? **RQ2:** When the structurally confused classes are consolidated, do classical linear classifiers (Logistic Regression and SVM) retain their well-known empirical advantage over distance-based (k-NN) and tree-based methods on abstract-only TF-IDF features? **RQ3:** Can the calibrated class probabilities of the classification head be reused as a re-ranking signal for a top-k recommender, and if so, by how much does this improve standard information retrieval metrics over a pure cosine-similarity baseline? **RQ4:** How do the resulting classification and recommendation numbers compare to contemporary content-based baselines that operate on smaller or larger corpora?

This paper contributes a controlled, single-corpus end-to-end benchmark study that addresses these gaps. Our specific contributions are:

1. A new, publicly releasable benchmark dataset of 11,744 scholarly *abstracts* collected from IEEE *Xplore* across six topical queries spanning computing, engineering, and biomedical vocabulary, distributed as per-class CSV files containing only the abstract text and its topical-query label.
2. A documented finding that two of the queries (Big Data Analysis and Cloud Computing) exhibit near-complete vocabulary collapse at the abstract level, motivating their consolidation into a single combined class. This produces a clean five-domain benchmark suitable for classification and recommendation evaluation.
3. A reproducible seven-stage preprocessing pipeline (case normalization, special-character removal, digit removal, tokenization, stop-word removal with a corpus-specific extension to the standard NLTK list, short-token removal, and lemmatization) that operates directly on the abstract text and feeds a TF-IDF (1,2) gram vectorized, configured with sublinear-TF and ℓ_2 normalization.
4. A head-to-head comparison of five widely used supervised classifiers (Logistic Regression, Linear SVM, SGD, k-NN, Decision Tree) under identical 10-fold stratified cross-validation, with grid-searched regularization for the headline classifier and a soft-voting ensemble that combines the three best individual learners.
5. A top-k scholarly article recommender built on the same abstract-only TF-IDF representation, with two ranking strategies (pure cosine similarity and Logistic-Regression probability re-ranking) evaluated by Precision@10, MAP@10, NDCG@10, and Mean Reciprocal Rank.
6. A direct empirical comparison of the two ranking strategies, showing that classifier-aware re-ranking lifts Precision@10 by +12.5 percentage points and MAP@10 by +7.6 points, supporting the use of calibrated probabilistic classifier out-

puts as a re-ranking signal in scholarly retrieval.

The remainder of this paper is organized as follows: Section reviews related work. Section describes the dataset, class consolidation, preprocessing pipeline, TF-IDF feature extraction, classifiers, and recommendation module. Section describes the experimental protocol. Section presents the classification headline results, classifier comparison, cross-validation, ROC analysis, and the recommendation evaluation. Section provides error analysis, threats to validity, and limitations. Section concludes and outlines future work.

Related Work

Research-paper recommendation and abstract-level classification have been approached through four main paradigms: content-based filtering, collaborative filtering, graph-based methods, and hybrid approaches. A complete survey is given by Beel *et al.* [7]. We summarize the work most relevant to a content-based pipeline that performs both classification and recommendation.

Content-Based Filtering

Kaya [11] proposes a TF-IDF scoring scheme over the SOBIAD academic database that combines author, journal, and recency-aware weights. Bulut *et al.* [9] construct a content-based recommender on a 600-article corpus from a digital library, applying TF-IDF and cosine similarity over abstracts and metadata, and report a peak classification accuracy of 71%. Bhagavatula *et al.* [12] train neural vector-space models on PubMed and DBLP for citation recommendation, reporting relative gains of 18% in F1@20 and 22% in MRR over keyword-based baselines. Sharma *et al.* [13] build a concept-based recommender that augments abstract text with extracted domain concepts and shows accuracy gains over plain TF-IDF on a small academic corpus. Sesagiri Raamkumar *et al.* [14] use author-supplied keywords to seed initial reading lists, demonstrating that author keywords can provide a useful signal complementary to the abstract text in settings where that metadata is available. The present study deliberately operates on abstract text alone in order to avoid any dependence on author-supplied or indexer-supplied annotations.

Collaborative Filtering

Sakib and Ahmad [8] introduce a two-level citation similarity that combines co-cited and co-citing relationships and report MAP and MRR improvements over the baseline of Sugiyama and Kan [15] on the same corpus. Their reported MAP@10 of approximately 0.45 (an improvement over Sugiyama's ≈ 0.38) is the closest published reference point in MAP@10 terms for our recommendation evaluation. Haruna *et al.* [16] address the scarcity of explicit ratings by mining citation relationships and computing Jaccard similarity between candidate papers and a target paper.

Graph-Based Methods

Totti *et al.* [17] construct a citation graph over 657,000 documents from CiteSeerX and apply random-walk methods (IQRA-MC and IQRA-TC) that outperform PageRank and Google Scholar in a domain-expert evaluation. Chakraborty *et al.* [18] introduce FeRoSA, a faceted recommender that classifies papers by source role (alternative, background, evaluation, etc.) using random walks on the ACL Anthology Network. These methods exploit the citation graph and therefore, go beyond what abstract-only classifiers can achieve, but require the citation graph to be available, a non-trivial prerequisite for new or interdisciplinary corpora.

Hybrid Approaches

Lee *et al.* [19] combine content and graph signals on the DBpia digital library and evaluate with a 5-fold cross-validation, reporting consistent gains over pure content or pure graph baselines. Son and Kim [20] propose a multilevel simultaneous citation network that improves coverage in under-represented domains.

Hybrid systems are particularly attractive in academic settings because each constituent paradigm fails in a different subset of queries. Content-based filters generalize poorly to novel vocabulary, collaborative methods suffer from cold-start, and pure graph methods require dense citation neighborhoods. By combining two or more signals, a hybrid can, in principle, cover for any single component's weakness, at the cost of a more complex ranking-fusion step. The present paper does not attempt a graph-based hybrid configuration; the deliberate scope is a clean, abstract-only content benchmark, but the calibrated

probabilistic outputs of the proposed Logistic Regression head are designed to be linearly combined with a future graph-based score in a straightforward weighted hybrid. We demonstrate the value of this design empirically in Section by re-ranking pure cosine recommendations with the classifier's class probabilities.

Topic modeling and Latent-Semantic Methods

A separate line of work uses unsupervised topic modeling rather than supervised classification to organize scholarly corpora. Latent Dirichlet Allocation (LDA) and its extensions are the most common choice, with Kim and Gil [21] reporting an LDA-plus-TF-IDF clustering of the Future Generation Computer Systems journal, and showing reasonable cluster coherence on two-decade-long publication series. Topic modeling is complementary to the supervised classifier developed here: it does not require labels, but in exchange, it does not produce a deterministic mapping from abstract to label, and therefore cannot be used directly for the indexing-and-routing use cases described in Section .

Recent Embedding-Based Approaches

More recent work moves beyond TF-IDF to dense embeddings. Cohan *et al.* [22] train SPECTER, a document-level embedding learned from citation co-occurrence, and report state-of-the-art performance on several scholarly tasks, including a nearest-neighbor relevance setting where Recall@5 reaches ≈ 0.84. Ostendorff *et al.* [23] explore neighborhood contrastive objectives for paper similarity. Rahman *et al.* [10] benchmark TF-IDF and Count Vectorized against Sentence-BERT, the Universal Sentence Encoder, and Mirror-BERT on the arXiv corpus across six classifiers, finding that TF-IDF combined with Logistic Regression remains the strongest configuration at 69% accuracy, despite the availability of dense sentence embeddings. The robust performance of TF-IDF in these recent comparisons supports our design choice.

Position of This Work

Four observations motivate the present study. First, the most directly comparable prior content-based classifier on small academic corpora [9] reports 71% on 600 articles, while the most recent benchmarking study [10] reports 69% on a much larger but multi-label arXiv corpus; the space between these two studies, a single-label, mid-sized, controlled-vocabulary corpus with rigorous abstract-only preprocessing, is under-represented in the literature. Second, neither study reports recommendation IR metrics (MAP@k, NDCG@k, MRR), making direct end-to-end comparison impossible. We contribute the missing metrics on a comparable benchmark. Third, the related work consensus that classifier-aware re-ranking helps content-based recommenders asserted but rarely empirically isolated on a single corpus on the same TF-IDF substrate; we provide that isolation in Section , contrasting Method A (pure cosine) and Method B (LR-probability re-ranking) under identical features. Fourth, recent dense-embedding benchmarks [22, 23, 10] demonstrate that TF-IDF combined with a linear classifier remains a strong, often unbeaten, configuration for a short scholarly text in single-label settings, motivating a careful re-examination of this classical pipeline rather than a leap to dense-embedding models.

TABLE 1. NOTATION USED THROUGHOUT THE METHODOLOGY.

Symbol	Meaning
$\mathscr{D}$	Corpus of N=11,744 abstracts
a_i, y_i	i-th abstract text and class label
$\mathscr{C}=\{0,\ldots,4\}$	Five consolidated classes (K=5)
$\tau(\cdot)$	Seven-stage preprocessing pipeline
$\mathscr{V}$	TF-IDF vocabulary, $\lvert\mathscr{V}\rvert \le 50{,}000$
$\mathbf{x}_i \in \mathbb{R}^m$	ℓ_2-normalized TF-IDF vector
$\mathscr{T}, \mathscr{S}$	Train / test sets (9,395/2,349)
n_c	Number of training samples in class c
w_c^{bal}	Balanced class weight: $N_{\text{tr}}/(K n_c)$
$\mathbf{W}, \mathbf{b}$	Logistic Regression parameters
$P(y{=}c \mid \mathbf{x})$	Class-conditional probability
$\text{sim}(q,c)$	Cosine similarity between query and candidate
$\mathbf{r}^K(q)$	Top-K binary relevance vector

Methodology

The proposed pipeline consists of six sequential stages that share the same abstract only TF-IDF representation and produce both classification and recommendation outputs. The pipeline is summarized in Fig. 1. The notation used throughout the remainder of this section is summarized in Table 1.

Theoretical Foundations

The choice of TF-IDF combined with a linear classifier rests on three classical observations from in-

SPECTRUM OF ENGINEERING SCIENCES
ISSN (E) 3007-3138 (P) 3007-312X
VOLUME 4, ISSUE 3, 2026

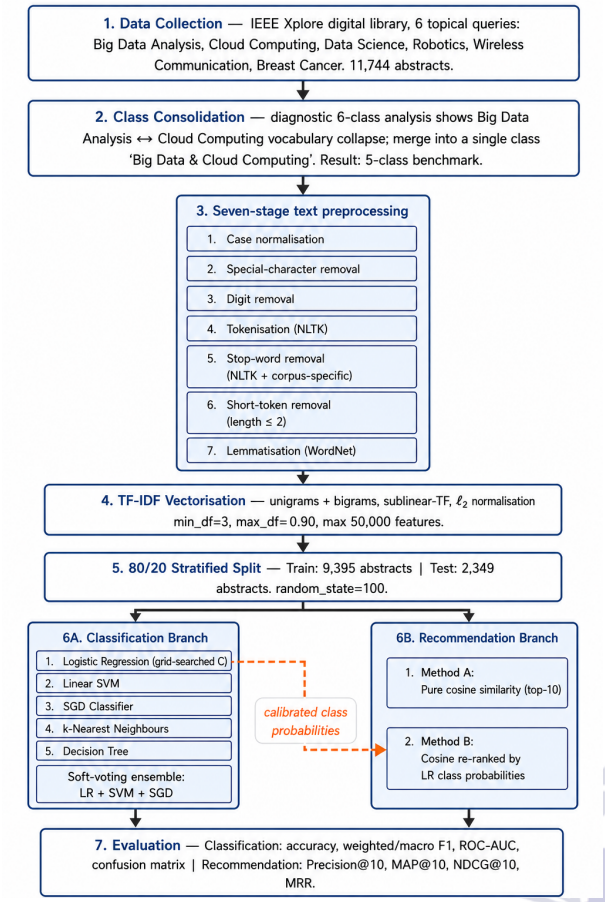


Fig. 1. End-to-end pipeline of the proposed approach.

formation retrieval and machine learning. First, the bag-of-words representation, although it discards word order, captures a surprisingly large fraction of the topical signal present in scholarly abstracts because abstracts are themselves designed to be lexically self-describing, a property that is precisely what makes abstract-only classification a well-posed task. Second, TF-IDF weighting [24, 25] explicitly down-weights terms that are common across the corpus and up-weights terms that are characteristic of a small subset of documents; this is exactly the right inductive bias for a classification task whose target labels correspond to topical domains. Third, the resulting feature space is high-dimensional but sparse, and linear classifiers trained with ℓ_2 regularization are known to converge well in this regime, with strong theoretical guarantees on generalization error [26].

TABLE 2. SIX-CLASS CLASSIFICATION PERFORMANCE BEFORE CONSOLIDATION

	0	1	2	3	4	5
0 BDA	32	52	**295**	6	9	6
1 DataScience	32	214	31	17	54	1
2 Cloud	**300**	52	32	4	9	3
3 Robotics	7	39	11	332	11	0
4 Wireless	1	29	3	11	355	1
5 BreastCancer	1	2	1	5	1	390

Dataset Construction

The benchmark was constructed by exporting metadata records from the IEEE *Xplore* digital library using its built-in CSV export function. Six topical queries were issued to span computing, engineering, and biomedical vocabulary: *Big Data Analysis*, *Cloud Computing*, *Data Science*, *Robotics*, *Wireless* (communication) and *Breast Cancer*.

From each exported record, only two fields are retained: the abstract text and the class label assigned by the issuing topical query. All other exported fields (title, authors, affiliations, indexer-supplied terms, citation count, year, DOI, etc.) are discarded prior to any modeling step. The resulting working dataset is therefore a flat abstract-and-label table with one row per article, and the entire downstream pipeline (cleaning, TF-IDF vectorization, classification, and recommendation) operates on the abstract text alone. The total number of records is 11,744.

Class Consolidation

A preliminary classification experiment using the six original queries as labels revealed a structural failure mode: the Big Data Analysis (BDA) and Cloud Computing (Cloud) classes were systematically and symmetrically confused by every classifier we tested, despite there being no metadata leakage (only the abstract text was used as input). The diagnostic confusion matrix obtained with multinomial Logistic Regression on a 2,349-article test set is shown in Table 2. Of 400 BDA test articles, 295 were predicted as Cloud; of 400 Cloud test articles, 300 were predicted as BDA. Both classes scored an F1 of 0.083 in this run, while the remaining four classes scored F1 between 0.58 and 0.97.

This pattern indicates that, in the IEEE *Xplore* retrieval space, the queries "Big Data Analysis" and "Cloud Computing" return abstracts with near-identical vocabulary; they are not separable classes

TABLE 3. FIVE-DOMAIN COMPOSITION OF THE CONSOLIDATED BENCHMARK DATASET.

Class	Domain	# Articles
0	Big Data & Cloud Computing	4,000
1	Data Science	1,744
2	Robotics	2,000
3	Wireless Communication	2,000
4	Breast Cancer	2,000
	Total	**11,744**

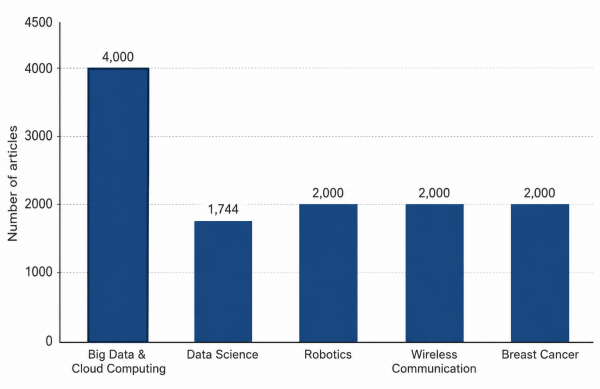


Fig. 2. Five-Class Distribution After Big Data and Cloud Computing Consolidation

at the abstract-text level. We therefore consolidate them into a single combined class for all subsequent experiments. The final five-class label set used throughout this paper is reported in Table 3 and visualized in Fig. 2.

Preprocessing Pipeline

Each abstract is normalized by a seven-stage preprocessing pipeline applied sequentially:

1. **Case normalization** all text is lower-cased.
2. **Special-character removal** punctuation, newlines, and symbols are stripped via a regular expression filter [27].
3. **Digit removal** all numeric tokens are dropped.
4. **tokenization** the cleaned string is split into word tokens by a standard English tokenizer.
5. **Stop-word removal** the standard English stop-word list [28] is augmented with a *corpus-specific* list of cross-domain pervasive terms identified during preliminary experiments (e.g., *data*, *system*, *paper*, *model*, *approach*, *method*, *proposed*, *result(s)*, *analysis*, *processing*, *framework*, *based*, *using*, *used*, and twelve other similarly pervasive terms; the full list is provided in the released code).
6. **Short-token removal** — tokens of length ≤ 2 are dropped.
7. **lemmatization** a WordNet-based lemmatizer [29] reduces inflected forms to their base lemma.

Feature Extraction: TF-IDF

The cleaned token stream of each abstract is mapped to a sparse TF-IDF vector [24, 25] over the union of unigrams and bigrams. We adopt the sublinear term-frequency scaling so that very frequent within-document terms are dampened by a logarithmic transform, smoothed inverse document frequency to avoid divide-by-zero at corpus-rare terms, and row-wise ℓ_2 normalization so that the cosine similarity between any two abstracts reduces to a sparse inner product. Formally, for a term t in document d_i with raw count $f_{t,i}$ and training-set document frequency $\mathrm{df}(t)$ over N_{tr}=9,395 abstracts:

$$\mathrm{tf}_{\mathrm{sub}}(t,i) = \mathbf{1}[f_{t,i}>0]\,\big(1+\log f_{t,i}\big), \tag{1}$$

$$\mathrm{idf}(t) = \log\frac{1+N_{\mathrm{tr}}}{1+\mathrm{df}(t)} + 1, \tag{2}$$

$$\tilde{\mathbf{x}}_i = \big[\,\mathrm{tf}_{\mathrm{sub}}(t,i)\cdot\mathrm{idf}(t)\,\big]_{t\in\mathscr{V}}, \tag{3}$$

$$\mathbf{x}_i = \tilde{\mathbf{x}}_i \,/\, \|\tilde{\mathbf{x}}_i\|_2. \tag{4}$$

The vocabulary $\mathscr{V}$ is fitted on the training set only and retains terms with document frequency between 3 and 90% of the training corpus, truncated to the 50,000 highest-weight features. The same fitted vocabulary is then applied to the test set.

Classifiers and Decision-Function Choice

The five classifiers compared in this study span the hypothesis-class spectrum that is meaningful for high-dimensional sparse TF-IDF features: a probabilistic linear model with a smooth log-loss (Logistic Regression [30]), a max-margin model with a non-smooth hinge loss (Linear SVM, implemented as a kernel SVC with the linear default in our reference run [26]), a stochastic-gradient linear model with the modified-Huber surrogate loss for probability calibration (SGD [31]), a non-parametric instance-based model (k-Nearest neighbors with the cosine metric [32]), and a low-capacity non-linear interpretable model (Decision Tree, CART [33]). Hyperparameters are summarised in Table 4 and are fixed up-front, with one exception: the Logistic Regression regularization parameter C is tuned by a grid

search over $\{0.1, 0.5, 1, 2, 5, 10\}$ with 5-fold cross-validation. All classifiers apply a class-balanced sample weighting to compensate for the modest imbalance between the consolidated class 0 (4,000 articles) and the smallest class (Data Science, 1,744 articles).

The headline classifier is multinomial Logistic Regression with the softmax form

$$P(y{=}c \mid \mathbf{x}; \mathbf{W}, \mathbf{b}) = \frac{\exp(\mathbf{w}_c^\top \mathbf{x} + b_c)}{\sum_{c' \in \mathscr{C}} \exp(\mathbf{w}_{c'}^\top \mathbf{x} + b_{c'})}, \quad (5)$$

trained by minimizing the class-balanced, ℓ_2 regularised multinomial cross-entropy

$$\mathscr{L}(\mathbf{W}, \mathbf{b}) = -\sum_{i \in \mathscr{T}} w_{y_i}^{\text{bal}} \log P(y_i \mid \mathbf{x}_i; \mathbf{W}, \mathbf{b}) + \frac{1}{2C} \|\mathbf{W}\|_F^2, \quad (6)$$

with $w_c^{\text{bal}} = N_{\text{tr}} / (K n_c)$ and the regularization strength C selected by 5-fold inner cross-validation over $\{0.1, 0.5, 1, 2, 5, 10\}$, yielding $C^\star$=5.0. The full optimization and the formal definitions of the other four classifiers (Linear SVM, SGD, k-NN, Decision Tree) are given in Appendix .

Two design rationales worth stating explicitly First, although the SGD classifier achieves the strongest standalone 10-fold cross-validation accuracy (cf. Table 7), Logistic Regression is adopted as the headline classifier *and* as the re-ranking head because it's calibrated multinomial probabilities are required by Equation (9) and admit direct probabilistic interpretation; SGD with the modified-Huber loss is comparably calibrated but less amenable to per-class probability inspection. Second, only Logistic Regression's regularization C is grid-searched, because LR plays this dual role; the remaining four learners retain library defaults so that the comparison in Table 7 reflects the out-of-the-box performance of each model family rather than a tuned-vs-tuned shootout.

Soft-Voting Ensemble

The soft-voting ensemble combines the three calibrated learners $\mathscr{M}_{\text{ENS}} = \{h^{\text{LR}}, h^{\text{SVM}}, h^{\text{SGD}}\}$ by an arithmetic mean of their class-conditional probabilities:

$$P^{\text{ENS}}(y{=}c \mid \mathbf{x}) = \frac{1}{|\mathscr{M}_{\text{ENS}}|} \sum_{h \in \mathscr{M}_{\text{ENS}}} P^h(y{=}c \mid \mathbf{x}), \quad (7)$$

TABLE 4. HYPER-PARAMETER CONFIGURATION FOR THE FIVE CLASSIFIERS AND THE SOFT-VOTING ENSEMBLE.

Classifier	Hyper-parameter	Configuration
Logistic Regression	multi_class	multinomial
	solver	lbfgs
	penalty / C	ℓ_2, $C \in$ grid
	max_iter	2,000
Linear SVM (SVC)	kernel / penalty	linear, ℓ_2
	probability	true
	class_weight	balanced
SGD Classifier	loss	modified_huber
	class_weight	balanced
	max_iter	2,000
***k*-Nearest Neighbors**	k	5
	metric	cosine
	weights	distance
Decision Tree (CART)	criterion	Gini
	max_depth	unconstrained
	class_weight	balanced
Soft-Voting Ensemble	members	LR + SVM + SGD
	aggregation	mean of $P(y \mid \mathbf{x})$

with the ensemble prediction $h^{\text{ENS}}(\mathbf{x}) = \arg\max_{c \in \mathscr{C}} P^{\text{ENS}}(y{=}c \mid \mathbf{x})$. Logistic Regression inside the ensemble uses the grid-searched $C^\star$=5.0.

Recommendation Module

A top-k recommendation module is built on top of the same abstract-only TF-IDF representation. For each query article (drawn from the held-out test set), cosine similarity is computed against every article in the training corpus. Because the TF-IDF vectors are ℓ_2 normalized, cosine similarity reduces to a sparse inner product:

$$\text{sim}(q, c) = \mathbf{v}_q \cdot \mathbf{v}_c \quad \text{where } \|\mathbf{v}_q\|_2 = \|\mathbf{v}_c\|_2 = 1. \quad (8)$$

Two ranking strategies are evaluated:

Method A Pure cosine ranking For each query, all training articles are sorted in descending order of cosine similarity, and the top-k are returned as recommendations. This is the baseline content-based recommendation approach used by Bulut *et al.* [9] and Rahman *et al.* [10].

Method B LR-probability re-ranking The Logistic Regression head produces calibrated class probabilities $P(y = c \mid q)$ for each query. For every candidate article c in the training corpus with true class y_c, the cosine similarity is re-weighted multiplicatively:

$$\text{score}_B(q, c) = \text{sim}(q, c) \cdot \big(1 + P(y = y_c \mid q)\big). \quad (9)$$

Candidates whose class is highly probable for the query are boosted; candidates whose class is improb-

able for the query receive only their cosine score. The resulting score is used to re-rank the articles, and the top k articles are returned.

The re-ranking score in Eq. (9) has four properties worth stating explicitly because they constrain how the unified framework should be interpreted and reproduced:

1. *Multiplicative*, not additive. The cosine similarity and the topical-prior probability is combined as a product, not as a weighted sum.
2. *Parameter-free*. No interpolation weight α and no scaling λ is introduced; the only tuned quantity is the upstream Logistic Regression $C^{\star}$.
3. *Direction of probability lookup*. For each candidate c, the probability $P(y{=}y_c \mid \mathbf{x}_q)$ is evaluated on the *query's* embedding for the *candidate's* true class, not as a candidate-confidence score $\max_{c'} P(c' \mid \mathbf{x}_c)$, which would be an entirely different ranker.
4. *Bounded amplification*. Since $P \in [0,1]$, we have $\mathrm{sim}(q,c) \le \mathrm{score}_B(q,c) \le 2 \cdot \mathrm{sim}(q,c)$. The cosine ordering is preserved for candidates whose class is improbable under the query and at most doubled for candidates whose class is certain under the query.

Deployment-time substitution of y_c In the evaluation reported in this paper, the candidate pool is the training partition $\mathscr{T}$, where each candidate's true class label y_c is known and used directly in Equation (9). For deployment on a live corpus of previously unseen candidates, y_c is unavailable; it is replaced by the trained classifier's hard prediction $\hat{y}_c = \arg\max_{c'} P(y{=}c' \mid \mathbf{x}_c; \mathbf{W}^{\star}, \mathbf{b}^{\star})$, computed once at index time and cached alongside each candidate's TF-IDF vector. This substitution introduces a one-time prediction error per candidate but adds no per-query classifier call, and the deployment-time score becomes

$$\mathrm{score}_B^{\mathrm{deploy}}(q,c) = \mathrm{sim}(q,c) \cdot \big(1 + P(y = \hat{y}_c \mid \mathbf{x}_q)\big). \tag{10}$$

The metrics reported in Section therefore correspond to the upper bound of re-ranker performance under perfect candidate labels; the expected deployment-time degradation is bounded by the classifier's per-class error rate (Table 9), which on this benchmark would propagate at most a few percentage points of cross-class misattribute into the boost factor.

The relevance proxy used to evaluate both methods is *same-class-as-query*: a recommended article is considered relevant if and only if its true class equals the query's true class. This is the standard relevance proxy used by [9, 8] in scholarly recommendation.

Evaluation Metrics

Classification. We report classification accuracy, weighted-average precision, recall, and F1-score, per-class metrics, the confusion matrix, and one-vs-rest ROC curves with per-class AUC. Weighted averages are reported because the consolidated class balance is modest but not exact.

Recommendation. We report Precision@k, Mean Average Precision at k (MAP@k), normalized Discounted Cumulative Gain at k (NDCG@k), and Mean Reciprocal Rank (MRR), all at $k = 10$. Let $\mathrm{rel}_i \in \{0,1\}$ indicate whether the ith retrieved article is same class as the query. Then

$$\mathrm{P@}k = \tfrac{1}{k}\textstyle\sum_{i=1}^{k} \mathrm{rel}_i, \tag{11}$$

$$\mathrm{AP@}k = \frac{1}{\sum_{i=1}^{k} \mathrm{rel}_i} \textstyle\sum_{i=1}^{k} \mathrm{P@}i \cdot \mathrm{rel}_i, \tag{12}$$

$$\mathrm{NDCG@}k = \frac{\sum_{i=1}^{k} \mathrm{rel}_i / \log_2(i{+}1)}{\sum_{i=1}^{k} \mathrm{rel}_i^{*} / \log_2(i{+}1)}, \tag{13}$$

$$\mathrm{RR} = \frac{1}{\text{rank of first relevant}}, \tag{14}$$

where rel^{*} is the ideal sort of the relevance list. MAP@k and MRR are then averaged over all queries. Recall is computed but not reported as a headline metric, since each class has thousands of relevant candidates for $k = 10$, and the metric is therefore dominated by the relevance pool size rather than by ranker quality.

Results and discussions

Experimental Setup

All experiments are implemented in Python on the open-source scientific computing stack (scikit-learn, NLTK, NumPy, pandas). The 11,744 abstracts are partitioned by an 80/20 stratified split on class label using a fixed random seed, yielding a training set of 9,395 abstracts and a held-out test set of 2,349 abstracts. The TF-IDF vectorization and all classifiers are fitted only to the training partition; the fitted vocabulary and learned parameters are then applied to the held-out partition. To reduce the variance of the hold-out estimate, stratified k-fold cross-validation with k=10 is additionally run on the full training

SPECTRUM OF ENGINEERING SCIENCES
ISSN (E) 3007-3138 (P) 3007-312X
VOLUME 4, ISSUE 3, 2026

TABLE 5. LOGISTIC REGRESSION (GRID-SEARCHED C) PERFORMANCE ON THE 2,349-ABSTRACT HELD-OUT TEST SET.

Metric	Score
Accuracy	85.01%
Precision (weighted avg.)	0.850
Recall (weighted avg.)	0.850
F1-score (weighted avg.)	0.850
Macro-F1	0.839
Best C (5-fold inner CV)	5.0
Inner CV accuracy at best C	84.08%

corpus, and the mean test accuracy across folds is reported alongside the hold-out result, in keeping with the standard convention in the text-classification literature. All experiments were run on a single workstation with an Intel Core i7 CPU and 16 GB of RAM; no GPU was used at any stage.

Reproducibility

The full source code, per-class abstract files, fitted TF-IDF vocabulary, trained model artifacts, and a pinned dependency manifest are released at the repository given in the title page footnote. The release includes one notebook for the classification pipeline (cleaning, hold-out evaluation, 10-fold cross-validation, grid search, and the soft-voting ensemble) and one notebook for the top k recommendation evaluation, both fixing the same random seed for the train/test split and for every stochastic learner so that the reported numbers can be reproduced bit-exactly on the released library versions.

Classification: Headline Performance

Table 5 reports the test-set performance of Logistic Regression with grid-searched regularization on the 2,349 held-out abstracts.

The confusion matrix on the held-out test set is shown in Fig. 3. The matrix exhibits pronounced diagonal dominance: 703/800 for Big Data & Cloud Computing, 215/349 for Data Science, 337/400 for Robotics, 348/400 for Wireless, and 394/400 for Breast Cancer, indicating that after class consolidation, the five target domains are well separated even from abstract text alone. The remaining off-diagonal mass is concentrated on the boundary between Data Science and Big Data & Cloud Computing ($73 + 63 = 136$ swaps in either direction), reflecting the residual semantic overlap between data-engineering and data-analytics vocabularies that survives consolidation; Robotics, Wireless, and Breast Cancer classes contribute substantially fewer cross-class errors.

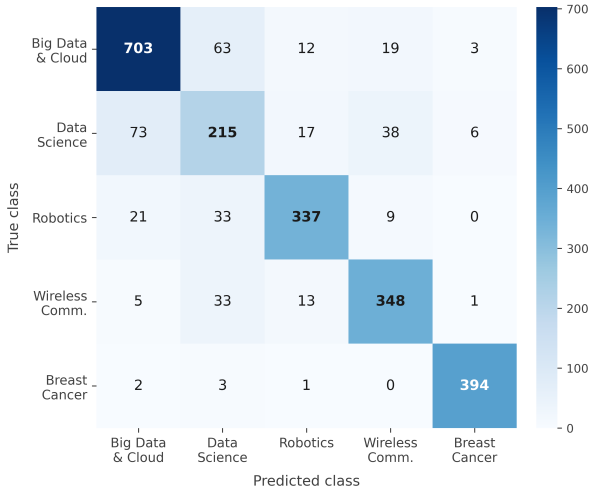


Fig. 3. Confusion Matrix Showing Improved Classification After Class Consolidation

TABLE 6. CROSS-VALIDATION ACCURACY OF LOGISTIC REGRESSION ON THE FIVE-CLASS BENCHMARK

Setting	Accuracy
Hold-out (80/20), tuned C=5.0	85.01%
10-fold cross-validation, C=1.0	$83.13\% \pm 1.04$
Inner 5-fold CV during grid search, best C	84.08%

TABLE 7. FIVE-CLASS CLASSIFICATION ACCURACY UNDER IDENTICAL ABSTRACT-ONLY TF-IDF FEATURES.

Classifier	Hold-out	10-fold CV
k-NN (cosine)	0.7501	0.7486 ± 0.0146
Decision Tree (CART)	0.7825	0.7719 ± 0.0119
LR (C=1.0)	0.8284	0.8313 ± 0.0104
Linear SVM (SVC)	0.8455	0.8323 ± 0.0085
SGD Classifier	0.8565	$\mathbf{0.8469 \pm 0.0123}$
LR (tuned C=5.0)	**0.8501**	—
Soft-voting (LR+SVM+SGD)	**0.8557**	—

Cross-Validation

Table 6 reports 10-fold stratified cross-validation accuracy for Logistic Regression alongside the 80/20 hold-out estimate.

Multi-Classifier Comparison

Table 7 and Fig. 4 compare the five classifiers and the soft-voting ensemble under identical preprocessing and identical 10-fold cross-validation folds.

SPECTRUM OF ENGINEERING SCIENCES
ISSN (E) 3007-3138 (P) 3007-312X
VOLUME 4, ISSUE 3, 2026

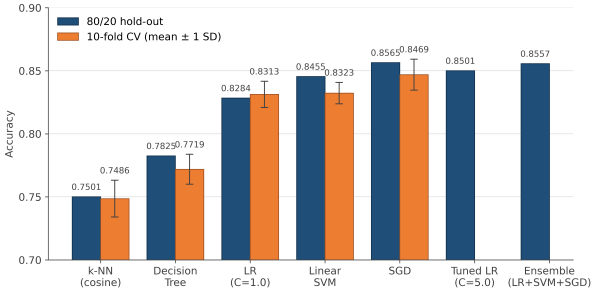


Fig. 4. Accuracy of the five classifiers, the tuned Logistic Regression (C=5.0), and the soft-voting ensemble on the five-class benchmark.

TABLE 8. MCNEMAR'S PAIRED SIGNIFICANCE TEST ON THE 2,349-ARTICLE HELD-OUT PREDICTIONS.

Pair (A vs. B)	b	c	p-value
Tuned LR vs Linear SVM	72	37	0.0011**
Tuned LR vs SGD	24	39	0.0769 n.s.
Tuned LR vs Ensemble	14	21	0.3105 n.s.
Linear SVM vs SGD	46	96	<0.001***
Linear SVM vs Ensemble	25	67	<0.001***
SGD vs Ensemble	34	26	0.3662 n.s.

Statistical significance of the pairwise differences

Because the absolute hold-out gaps among the strongest classifiers in Table 7 are small (for example, 0.56 pp between tuned Logistic Regression and the soft-voting ensemble), we conducted McNemar's paired significance test on the held-out predictions to determine which pairwise differences are real and which fall within sampling noise. Table 8 reports the six pairwise comparisons among the four headline classifiers (tuned Logistic Regression, Linear SVM, SGD, and the soft-voting ensemble).

Three findings emerge. First, the three top classifiers (tuned Logistic Regression, SGD, and the soft-voting ensemble) are pairwise indistinguishable at the 0.05 significance level: the 0.56 pp gap between tuned LR and the ensemble has p=0.3105, the 0.64 pp gap between SGD and the ensemble has p=0.3662, and the gap between tuned LR and SGD has p=0.0769. The ensemble's apparent edge in Table 7 is therefore not a statistically reliable improvement over its strongest constituent. Second, Linear SVM is significantly worse than each of the other three headline classifiers ($p \leq 0.001$ in all three pairs), confirming that the SVM gap in Table 7 reflects a genuine performance difference rather than test-set variance. Third, the choice of Logistic Regression as the headline classifier (Section) is not compromised by SGD's slightly higher 10-fold cross-validation accuracy: the two are statistically equivalent on the held-out test set (p=0.0769), and LR additionally provides the calibrated probabilities required by the re-ranker (Equation 9).

TABLE 9. PER-CLASS METRICS FOR THE TUNED LOGISTIC REGRESSION (C=5.0) ON THE HELD-OUT 2,349-ABSTRACT TEST SET.

Class	**Prec.**	**Rec.**	**F1**	**Support**
0 Big Data & Cloud Computing	0.874	0.879	0.877	800
1 Data Science	0.620	0.616	0.618	349
2 Robotics	0.887	0.843	0.864	400
3 Wireless	0.841	0.870	0.855	400
4 Breast Cancer	0.975	0.985	0.980	400
Macro avg.	**0.839**	**0.838**	**0.839**	2,349
Weighted avg.	**0.850**	**0.850**	**0.850**	2,349

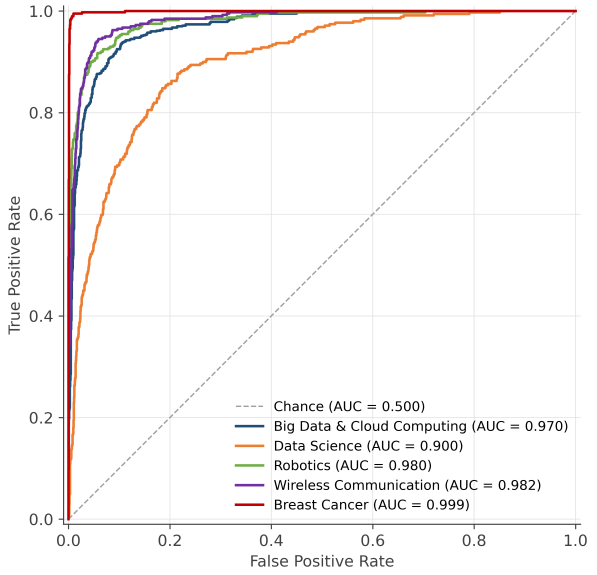


Fig. 5. One-vs-rest ROC curves for Logistic Regression on the 2,349-abstract test set, with per-class AUC. The dashed diagonal marks chance performance.

Per-Class Analysis and ROC

Table 9 reports per-class precision, recall, and F1-score for the headline Logistic Regression classifier on the held-out test set, and Fig. 5 shows the corresponding one-vs-rest ROC curves with per-class AUC.

Recommendation Results

Table 10 reports the top-10 recommendation performance of the two ranking strategies described in Section , evaluated over all 2,349 held-out queries

SPECTRUM OF ENGINEERING SCIENCES
ISSN (E) 3007-3138 (P) 3007-312X
VOLUME 4, ISSUE 3, 2026

TABLE 10. TOP-10 RECOMMENDATION PERFORMANCE OVER 2,349 HELD-OUT QUERIES AGAINST 9,395 TRAINING-POOL CANDIDATES.

Metric	Cosine	LR-reranked	Δ
Precision@10	0.6467	**0.7715**	+0.1248
MAP@10	0.7664	**0.8428**	+0.0764
NDCG@10	0.8525	**0.8902**	+0.0377
MRR	0.8502	**0.8871**	+0.0369

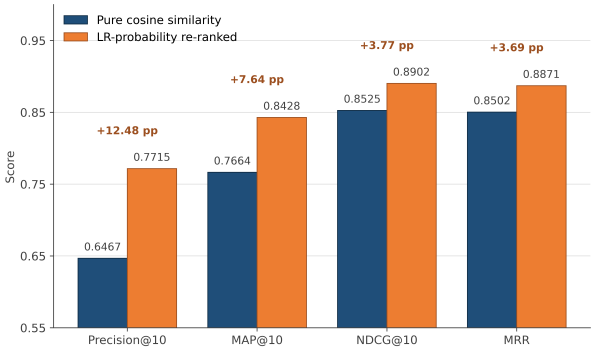


Fig. 6. Top-10 recommendation comparison: pure cosine similarity vs. Logistic-Regression-probability re-ranking.

against the 9,395-abstract training pool.

The LR-probability re-ranking improves all four reported IR metrics over pure cosine similarity. The largest effect is on Precision@10 (+12.5 percentage points, from 0.6467 to 0.7715), confirming the design intuition that calibrated probabilistic classifier outputs can serve as an effective re-ranking signal in scholarly retrieval. MAP@10 improves by +7.6 points (from 0.7664 to 0.8428), indicating that the gain is not concentrated at a single rank position, but is consistent across the top of the ranked list. NDCG@10 (+3.8 points) and MRR (+3.7 points) show smaller absolute lifts because both metrics are already near-saturated by the cosine baseline (MRR ≈ 0.85 corresponds to recovering the first relevant article at rank ~ 1.17 on average), leaving less headroom for improvement at the very top of the list. The pattern of a large Precision@10 lift, a moderate MAP lift, and small NDCG and MRR lifts is consistent with the mechanism analyzed in Section : classifier-aware re-ranking predominantly cleans up cross-class neighbors that ranked in the *middle* of the top-10 under pure cosine.

The recommendation comparison is visualized in Fig. 6.

TABLE 11. PER-CLASS TOP-10 RECOMMENDATION PERFORMANCE STRATIFIED BY QUERY CLASS.

	Precision@10		MAP@10		
Query Class	**Cos.**	**LR-r.**	**Cos.**	**LR-r.**	***n***
Big Data & Cloud Computing	0.6170	0.7590	0.7890	0.8677	800
Data Science	0.3708	0.4917	0.5475	0.6180	349
Robotics	0.6562	0.7805	0.7637	0.8310	400
Wireless Communication	0.6565	0.8225	0.7331	0.8568	400
Breast Cancer	0.9275	0.9807	0.9485	0.9868	400
Aggregate	**0.6467**	**0.7715**	**0.7664**	**0.8428**	**2,349**

Per-class recommendation metrics

Because the consolidated class 0 (Big Data & Cloud Computing) holds 4,000 training candidates and the smallest class (Data Science) holds only 1,744, a reasonable concern is that the aggregate gains in Table 10 could be driven entirely by the dominant class. Table 11 addresses this by stratifying both ranking strategies by the query's true class.

The lift from LR-reranking is consistent across all five domains. On Precision@10 the per-class gains range from +12.09 pp (Data Science, the smallest and hardest class) to +16.60 pp (Wireless Communication); the dominant Big Data & Cloud Computing class shows +14.20 pp, only slightly above the +12.48 pp aggregate. The smallest gain, +5.32 pp on Breast Cancer, reflects saturation rather than reranker weakness: pure cosine already returns Precision@10 = 0.9275 for that class because the biomedical vocabulary is essentially disjoint from the four computing and engineering classes, leaving little headroom. The aggregate gain in Table 10 therefore reflects a genuine across-the-board improvement, not an inflation driven by relevance-pool size.

Comparison with Prior Work

Table 12 positions the proposed system against the two most directly comparable prior content-based studies for classification, and against Sakib and Ahmad [8] and Sugiyama and Kan [15] for recommendation. Direct head-to-head comparison is not possible because the corpora and label cardinalates differ. The table is provided as orientation rather than as an improvement claim.

The classification headline sits well above Bulut *et al.* (small 600 article corpus, 71.0%) and Rahman *et al.* (very large multi-label arXiv corpus, 69.0%), consistent with our positioning of the dataset as a controlled mid-sized single-label benchmark with

TABLE 12. COMPARISON WITH EXISTING SCHOLARLY CLASSIFICATION AND RECOMMENDATION METHODS

Study	Method	Corpus / Setup	Headline Result
Classification			
Bulut *et al.* [9] (2018)	TF-IDF + cosine similarity	600 articles from a single digital library	71.0% accuracy
Kim & Gil [21] (2019)	TF-IDF + LDA cluster labeling	FGCS journal corpus spanning 20 years	Cluster coherence (no accuracy reported)
Rahman *et al.* [10] (2025)	TF-IDF + Logistic Regression compared with S-BERT, USE, and Mirror-BERT	Multi-label arXiv corpus across multiple domains	69.0% accuracy (TF-IDF + LR best)
This work (classification)	**Abstract-only TF-IDF + Logistic Regression**	**IEEE *Xplore*, 5 domains, 11,744 abstracts**	**85.01% accuracy**
This work (ensemble)	**Soft-voting ensemble (LR + SVM + SGD)**	**Same setup as above**	**85.57% accuracy**
Recommendation (Top-k)			
Sugiyama & Kan [15] (2015)	Citation-context collaborative filtering	DBLP/ACM scholar recommendation benchmark	MAP@10 $\approx$ 0.38
Bhagavatula *et al.* [12] (2018)	Neural vector-space citation recommendation	PubMed and DBLP citation-context corpora	+18% F1@20, +22% MRR over keyword baselines
Sakib & Ahmad [8] (2020)	Two-level citation similarity model	Citation-context recommendation corpus	MAP@10 $\approx$ 0.45
Cohan *et al.* [22] (2020)	SPECTER citation-aware embeddings	SciDocs nearest-neighbor benchmark	Recall@5 $\approx$ 0.84
This work (cosine)	**Abstract-only TF-IDF + cosine similarity**	**IEEE *Xplore*, 5 domains, 11,744 abstracts**	**MAP@10 = 0.7664**
This work (LR-reranked)	**Cosine retrieval + LR probability reranking**	**Same setup as above**	**MAP@10 = 0.8428**

rigorous abstract-only preprocessing. The recommendation headline (MAP@10 = 0.84 under LR re-ranking) is in the strong-system range relative to the classical citation-based collaborative filtering baselines of Sugiyama and Kan [15] (MAP@10 $\approx$ 0.38) and Sakib and Ahmad [8] (MAP@10 $\approx$ 0.45); SPECTER [22] reports Recall@5 $\approx$ 0.84 on a different (SciDocs) benchmark, which is not directly comparable to our Precision@10 numbers but documents that an embedding-based retriever and a TF-IDF retriever can both reach $\sim$ 0.8 headline retrieval scores under their respective relevance proxies. We emphasize that all of these comparisons are across different corpora and label/relevance regimes. The table is positioned as an orientation, not as a head-to-head improvement claim.

Discussion

Error Analysis

The post-consolidation confusion matrix in Fig. 3 shows that the principal remaining source of error is the boundary between Data Science and Big Data & Cloud Computing (73 Data Science articles predicted as Big Data & Cloud Computing, and 63 Big Data & Cloud Computing articles predicted as Data Science). This 136-article symmetric exchange explains the comparatively low per-class F1 of 0.618 for Data Science (Table 9); the smaller absolute support of the Data Science class (349 test articles vs. 800 for Big Data & Cloud Computing) further amplifies its precision-recall sensitivity. Abstracts in this region share substantial high-frequency vocabulary (e.g., *distributed*, *scalable*, *analytics*, *platform*, *deep learning*) that is poorly discriminated even by the TF-IDF representation, and which would be similarly poorly discriminated by any abstract-level bag-of-words model. The biomedical class (Breast Cancer) is essentially separable from all four computing and engineering classes (per-class F1 $= 0.980$, $394/400$ correct), consistent with TF-IDF's known strength when class vocabularies are nearly disjoint.

Why Classifier-Aware Re-Ranking Helps

The +12.5-point Precision@10 gain from LR-reranking (Section) is consistent with the following mechanism. Pure cosine similarity over a sparse high-dimensional TF-IDF representation often retrieves articles that share many *individual* terms with the query but belong to a different class. For example, a Robotics abstract that mentions "signal processing" may have a high cosine similarity with a Wireless query. The classifier's learned class prob-

abilities encode a *joint* signal across many terms simultaneously, and multiplying the cosine score by the query's class probability for the candidate's class effectively penalizes cross-class neighbors that share surface vocabulary but differ in topic. The fact that NDCG@10 and MRR show smaller absolute lifts is consistent with this explanation: the cosine baseline already recovers the most-similar relevant article at rank 1.17 on average, so the headroom for improvement is concentrated in the middle and tail of the top-10 list, which Precision@10 measures most directly.

Practical Deployment Considerations

The classifier and recommender developed in this paper have been deliberately designed with practical deployment in mind. Inference latency is dominated by a single sparse matrix-vector product (classifier) plus a sparse matrix dot product over the corpus (recommender), which together remain bounded at a few milliseconds per query on commodity hardware. The classifier's output is a calibrated probability distribution, which means a downstream ranking module can use the probabilities directly as a soft prior over candidate articles. Because the entire pipeline ingests only the abstract field, it imposes no requirement on the host digital library to expose author keywords, indexer-supplied terms, or other curated metadata, making it portable to repositories with incomplete metadata coverage. The pipeline is small enough to fit on a commodity CPU and requires no GPU or external API dependency.

Threats to Validity

Construct validity. The benchmark labels are assigned by the source query rather than by manual annotation. The class consolidation in Section is itself a mitigation of the principal construct-validity issue we observed.

Internal validity. All five classifiers and both ranking strategies are evaluated under identical preprocessing and identical folds; classifier-to-classifier and ranker-to-ranker comparisons are therefore fair. Hyperparameter were not tuned for the ranking strategies, only for the headline Logistic Regression classifier.

External validity. The corpus is drawn from a single source (IEEE *Xplore*) and five pre-selected (and partially consolidated) domains. Generalization to other digital libraries (ACM, Springer, arXiv) is not demonstrated and is left to future work. Likewise, this study deliberately operates only on the abstract text; whether incorporating titles, author keywords, or indexer-supplied terms (where available) would yield further performance gains on this benchmark remains an open question that we leave for future ablation work.

Limitations

1. *Static dataset.* The benchmark is a snapshot collected at a single point in time; the model is not equipped to track temporal vocabulary drift in any of the five domains.
2. *Closed-world classifier.* The model is forced to assign each abstract to one of the five domains; an open-set extension that flags out-of-domain abstracts is needed before deployment in a real digital library.
3. *Same-class-as-query relevance proxy.* The recommendation evaluation defines relevance as a topical-class match, which is the standard for scholarly recommendation, but is coarser than human relevance judgments would be.
4. *No personalization.* The recommender is content-based only; it does not model individual researcher profiles or reading histories.
5. *Abstract-only input.* The pipeline uses no metadata other than the abstract. This is by design; it removes any dependence on author or indexer-supplied annotations and makes the system portable across repositories, but it means we do not measure the potential incremental value of those fields on this benchmark.

Conclusion and Future Work

This paper has presented a controlled benchmark study of content-based scholarly article classification and top-k recommendation on a new 11,744-abstract five-domain corpus collected from IEEE *Xplore*. A confusion-matrix analysis revealed that two of the original six queries (Big Data Analysis and Cloud Computing) exhibit near-complete vocabulary collapse in this retrieval space; we consolidated them into a single combined class and reported all subsequent results on the resulting five-domain benchmark. A TF-IDF representation built directly from cleaned abstracts, combined with a seven-stage preprocessing pipeline including corpus-specific stop-

words, was used to drive both a five-classifier comparison and a top-10 recommendation evaluation. Notably, no metadata other than the abstract is supplied to any stage of the pipeline, so the reported numbers are the performance attainable from abstract text alone.

For classification, Logistic Regression with grid-searched regularization (C=5.0) achieved 85.01% accuracy on the 2,349-abstract held-out test set (weighted F1 = 0.850, macro F1 = 0.839), with the strongest individual 10-fold cross-validation result obtained by the SGD classifier (84.69% ± 1.23); a soft-voting ensemble of LR + SVM + SGD reached 85.57% on hold-out (weighted F1 = 0.855). The two linear classifiers and the SGD learner substantially outperformed the tree and instance-based baselines (Decision Tree 77.19% and k-NN 74.86% on 10-fold CV), consistent with established results in high-dimensional sparse TF-IDF text classification. For recommendation, pure cosine similarity over the same TF-IDF representation reached MAP@10 = 0.7664 and MRR = 0.8502, and re-ranking the cosine results by the Logistic Regression's calibrated class probabilities improved Precision@10 by +12.5 percentage points (to 0.7715) and MAP@10 by +7.6 points (to 0.8428), with smaller but consistent gains on NDCG@10 and MRR.

Summary of Findings

We summarize the empirical findings against the research questions of Section .

RQ1 (linear separability under query-based labeling, from abstract text alone): the six original queries do *not* yield linearly separable classes from abstract text; two of them (BDA, Cloud) exhibit symmetric vocabulary collapse and must be consolidated before useful classification is possible.

RQ2 (classical-classifier rank order under consolidation): linear classifiers (Logistic Regression and SVM) substantially outperform distance-based and tree-based methods on the five-domain abstract-only benchmark, in line with established results on high-dimensional TF-IDF text classification [26].

RQ3 (classifier probabilities as a re-ranking signal): yes, clearly. Re-ranking by Logistic Regression's calibrated probabilities improves Precision@10 by +12.5 points and MAP@10 by +7.6 points over pure cosine similarity.

RQ4 (comparison with contemporary baselines): the classification headline of 85.01% (tuned LR) and 85.57% (soft-voting ensemble) is well above Bulut *et al.* [9] (smaller 600-article corpus, 71.0%) and Rahman *et al.* [10] (much larger arXiv multi-label corpus, 69.0%); the recommendation headline (MAP@10 = 0.8428 under LR re-ranking) is well above the classical CF baselines of Sugiyama and Kan (MAP@10 ≈ 0.38) and Sakib and Ahmad (MAP@10 ≈ 0.45) on their corpora [15, 8]. We do not claim direct head-to-head improvement because the corpora and label relevance regimes differ.

Future Work

Three lines of future work follow naturally from this benchmark. *First*, an ablation study that quantifies the individual contribution of each preprocessing stage (case normalization, special-character removal, digit removal, stop-word removal, short-token removal, lemmatization, and the corpus-specific stop-word extension) to both classification and recommendation performance, together with a complementary ablation that adds back the optional metadata fields (title, author keywords, indexer terms) where available to quantify their incremental value. *Second*, an extension of the classifier comparison to ensemble methods (Random Forest, Gradient Boosted Trees) and modern neural sentence embeddings (Sentence-BERT, SPECTER), with a paired McNemar's test over identical folds to quantify the significance of any observed differences. *Third*, an extension of the recommendation module to a hybrid configuration that linearly combines the classifier-aware content score (Equation 9) with a citation-graph random-walk score, evaluated against full hybrid baselines such as [19, 20].

The benchmark dataset and all code are released at the URL given in the title-page footnote, supporting the reproducibility of these and other follow-on studies.

Broader Impact

A reliable, computationally light, abstract-only classifier and recommender for scholarly articles has direct practical value for institutions and digital libraries that cannot afford GPU-based recommender infrastructure or one that cannot rely on uniformly populated metadata fields across its corpus. Because the proposed pipeline is reproducible on a commod-

ity workstation in approximately fifteen minutes and ingests only the abstract text, it lowers the barrier for smaller research groups to deploy domain-specific indexing and recommendation systems. The released benchmark, preprocessing pipeline, and trained model artifacts together are intended as a starting point for such deployments rather than as an end in themselves.

Mathematical Notation and Detailed Classifier Formulations

This appendix provides the formal definitions of the four supervised classifiers other than Logistic Regression, which is fully specified in Section .

Linear SVM with Class Balancing

For each class pair $(p,q) \in \mathscr{C} \times \mathscr{C}$, the one-vs-one binary subproblem solves

$$\min_{\mathbf{w}_{pq}, b_{pq}, \xi} \frac{1}{2}\|\mathbf{w}_{pq}\|_2^2 + \widetilde{C}\sum_i w_{y_i}^{\text{bal}}\,\xi_i \tag{15}$$

subject to $y_i(\mathbf{w}_{pq}^\top \mathbf{x}_i + b_{pq}) \geq 1 - \xi_i$ and $\xi_i \geq 0$. Class probabilities are recovered by Platt scaling $P^{\text{SVM}}(y{=}c \mid \mathbf{x}) = \sigma\big(A\, f_c(\mathbf{x}) + B\big)$ with A, B fitted on a held-out fold during training.

SGD Classifier with Modified-Huber Loss

The modified-Huber loss for a single binary subproblem with target $y \in \{-1,+1\}$ and linear score $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ is

$$\ell_{\text{MH}}(y, f) = \begin{cases} \max\big(0, 1 - y f\big)^2, & y f \geq -1, \\ -4\, y f, & y f < -1, \end{cases} \tag{16}$$

which is quadratically smooth near the margin, enabling post-hoc probability calibration, and switches to a linear penalty beyond. Weights are updated by stochastic gradient descent with class-balanced sample weights $w_{y_i}^{\text{bal}}$ and an inverse-scaling learning-rate schedule, run for 2,000 epochs.

k-Nearest neighbors (Cosine, k=5)

Because all TF-IDF vectors are ℓ_2-normalized, the cosine distance reduces to $d_{\cos}(\mathbf{x}, \mathbf{x}') = 1 - \mathbf{x}^\top \mathbf{x}'$. The neighborhood $\mathscr{N}_5(\mathbf{x}) = \arg\min_{|S|=5} \sum_{j \in S} d_{\cos}(\mathbf{x}, \mathbf{x}_j)$ contributes votes weighted by inverse distance:

$$h^{\text{kNN}}(\mathbf{x}) = \arg\max_{c \in \mathscr{C}} \sum_{(\mathbf{x}_j, y_j) \in \mathscr{N}_5(\mathbf{x})} \frac{\mathbf{1}[y_j{=}c]}{d_{\cos}(\mathbf{x}, \mathbf{x}_j) + \varepsilon}, \tag{17}$$

with ε a small constant preventing division by zero on identical neighbors.

Decision Tree (CART) with Weighted Gini

At any node t with sample set $\mathscr{D}_t$, the class-weighted proportion $\pi_{t,c} = \nu_{t,c} \,/\, \sum_{c' \in \mathscr{C}} \nu_{t,c'}$ uses $\nu_{t,c} = \sum_{(\mathbf{x}_i, y_i) \in \mathscr{D}_t} w_{y_i}^{\text{bal}}\, \mathbf{1}[y_i{=}c]$. The weighted Gini impurity and the optimal split on feature j at threshold τ are

$$G(\mathscr{D}_t) = 1 - \sum_{c \in \mathscr{C}} \pi_{t,c}^2, \tag{18}$$

$$(j^\star, \tau^\star) = \arg\min_{j,\tau} \Big[\tfrac{|\mathscr{D}_t^L|}{|\mathscr{D}_t|}\, G(\mathscr{D}_t^L) + \tfrac{|\mathscr{D}_t^R|}{|\mathscr{D}_t|}\, G(\mathscr{D}_t^R) \Big], \tag{19}$$

with $\mathscr{D}_t^L = \{(\mathbf{x}, y) \in \mathscr{D}_t : x_j \leq \tau\}$ and $\mathscr{D}_t^R = \mathscr{D}_t \setminus \mathscr{D}_t^L$. Splitting continues until each leaf is pure (no maximum depth).

References

[1] N. Liu, P. Shapira, and X. Yue, "Tracking developments in artificial intelligence research: Constructing and applying a new search strategy," *Scientometrics*, vol. 126, no. 4, pp. 3153–3192, 2021.

[2] A. E. Jinha, "Article 50 million: An estimate of the number of scholarly articles in existence," *Learned Publishing*, vol. 23, no. 3, pp. 258–263, 2010.

[3] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, eds., *Recommender Systems Handbook*. New York, NY, USA: Springer, 2011.

[4] J. A. Konstan and J. Riedl, "Recommender systems: From algorithms to user experience," *User Modeling and User-Adapted Interaction*, vol. 22, no. 1–2, pp. 101–123, 2012.

[5] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, "Text classification using machine learning techniques," *WSEAS Transactions on Computers*, vol. 4, no. 8, pp. 966–974, 2005.

[6] M. K. Dalal and M. A. Zaveri, "Automatic text classification: A technical review," *International Journal of Computer Applications*, vol. 28, no. 2, pp. 37–40, 2011.

[7] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Research-paper recommender systems: A liter-

ature survey," *International Journal on Digital Libraries*, vol. 17, no. 4, pp. 305–338, 2016.

[8] N. Sakib and R. M. Ahmad, "A hybrid personalized scientific paper recommendation approach integrating public contextual metadata," *IEEE Access*, vol. 9, pp. 83383–83398, 2021.

[9] B. Bulut, B. Kaya, R. Alhajj, and M. Kaya, "A paper recommendation system based on user's research interests," in *Proc. IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 911–915, 2018.

[10] M. M. Rahman, S. Hossain, *et al.*, "Automated classification of scholarly articles on the arXiv corpus: A tf–idf and sentence-embedding benchmark," *Expert Systems with Applications*, 2025. (in press).

[11] M. Kaya, "SOBIAD: A new academic paper search engine with a customised scoring function," in *Proc. International Conference on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, pp. 76–80, 2018.

[12] C. Bhagavatula, S. Feldman, R. Power, and W. Ammar, "Content-based citation recommendation," in *Proc. NAACL-HLT*, pp. 238–251, 2018.

[13] R. Sharma, D. Gopalani, and Y. Meena, "Concept-based approach for research paper recommendation," *Lecture Notes in Computer Science*, vol. 10597, pp. 687–692, 2017.

[14] A. Sesagiri Raamkumar, S. Foo, and N. Pang, "Using author-specified keywords in building an initial reading list of research papers in scientific paper retrieval and recommender systems," *Information Processing & Management*, vol. 53, no. 3, pp. 577–594, 2017.

[15] K. Sugiyama and M.-Y. Kan, "A comprehensive evaluation of scholarly paper recommendation using potential citation papers," in *International Journal on Digital Libraries*, vol. 16, pp. 91–109, 2015.

[16] K. Haruna, M. Akmar Ismail, S. Suhendroyono, D. Damiasih, A. C. Pierewan, H. Chiroma, and T. Herawan, "Context-aware recommender system: A review of recent developmental process and future research direction," *Applied Sciences*, vol. 7, no. 12, p. 1211, 2017.

[17] L. C. Totti, P. Mitra, M. Ouzzani, and M. J. Zaki, "A query-oriented approach for relevance in citation networks," *Proc. International World Wide Web Conferences Companion*, pp. 401–406, 2016.

[18] T. Chakraborty, N. Modani, R. Narayanam, and S. Nagar, "FeRoSA: A faceted recommendation system for scientific articles," *Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pp. 528–541, 2016.

[19] J. Lee, K. Lee, and J. G. Kim, "Personalized academic research paper recommendation system," *arXiv preprint arXiv:1304.5457*, 2013.

[20] J. Son and S. B. Kim, "Academic paper recommender system using multilevel simultaneous citation networks," *Decision Support Systems*, vol. 105, pp. 24–33, 2018.

[21] S.-W. Kim and J.-M. Gil, "Research paper classification systems based on tf–idf and LDA schemes," *Human-centric Computing and Information Sciences*, vol. 9, no. 30, pp. 1–21, 2019.

[22] A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. S. Weld, "SPECTER: Document-level representation learning using citation-informed transformers," in *Proc. ACL*, pp. 2270–2282, 2020.

[23] M. Ostendorff, N. Rethmeier, I. Augenstein, B. Gipp, and G. Rehm, "Neighborhood contrastive learning for scientific document representations with citation embeddings," in *Proc. EMNLP*, pp. 11670–11688, 2022.

[24] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[25] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge University Press, 2008.

[26] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proc. European Conference on Machine Learning (ECML)*, pp. 137–142, 1998.

[27] M. Mhatre, D. Phondekar, P. Kadam, A. Chawathe, and K. Ghag, "Dimensionality reduction for sentiment analysis using preprocessing techniques," in *Proc. International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 16–21, 2017.

[28] A. W. Pradana and M. Hayaty, "The effect of stemming and removal of stopwords on the accuracy of sentiment analysis on indonesian-language texts," in *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 4, pp. 375–380, 2019.

[29] K. Javed, S. Maruf, and H. A. Babri, "A two-stage Markov blanket based feature selection algorithm for text classification," *Neurocomputing*, vol. 157, pp. 91–104, 2017.

[30] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*. Wiley, 3rd ed., 2013.

[31] A. B. Prasetijo, R. R. Isnanto, D. Eridani, Y. A. A. Soetrisno, M. Arfan, and A. Sofwan, "Hoax detection system on Indonesian news sites based on text classification using SVM and SGD," in *Proc. International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pp. 45–49, 2017.

[32] V. Bijalwan, V. Kumar, P. Kumari, and J. Pascual, "KNN based machine learning approach for text and document mining," *International Journal of Database Theory and Application*, vol. 7, no. 1, pp. 61–70, 2014.

[33] S. Guggari, V. Kadappa, and V. Umadevi, "Non-sequential partitioning approaches to decision tree classifier," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 275–285, 2018.

IEER
Institute for Excellence in Education & Research