EURASIAN JOURNAL OF ACADEMIC RESEARCH
Innovative Academy Research Support Center
IF = 9.3 www.in-academy.uz/index.php/ejar

HYBRID TEXT CLASSIFICATION BASED ON TF-IDF AND ADAPTIVE ALSHE ENSEMBLE

Rakhmanov Askar
Department of "System and Applied Programming" Tashkent University of Information Technologies named after Muhammad al Khwarizmi. asqartr1.2.3dipu@gmail.com
Abduvalieva Zebiniso
Department of "System and Applied Programming" Tashkent University of Information Technologies named after Muhammad al-Khwarizmi. zebinosoabduvaliyeva@gmail.com
Murodov D.D
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi. dilimmurod@gmail.com
https://doi.org/10.5281/zenodo.20053663

ARTICLE INFO

Received: 28th April 2026
Accepted: 05th May 2026
Online: 06th May 2026

KEYWORDS

Text classification, TF-IDF, machine learning, NLP, LinearSVC, Naive Bayes, ensemble, ALSHE, Macro F1.

ABSTRACT

This article investigates the task of multi-class classification of technical texts. The experiments utilized the LocalDocs-10 corpus, compiled from software package descriptions and partitioned into 10 thematic classes. Texts were represented via word-level and character-level TF-IDF n-grams, alongside compact SVD-derived features. A comparative evaluation was conducted between classical machine learning algorithms and several hybrid approaches. Special emphasis was placed on the adaptive ALSHE-Gated model, which integrates Complement Naive Bayes and LinearSVC through a confidence-driven switching mechanism. The Passive-Aggressive Classifier achieved the highest performance among baseline models, attaining an Accuracy of 0.855 *and a Macro F1-score of* 0.836*. These findings affirm that lightweight TF-IDF models constitute a viable alternative to computationally intensive neural networks for small- to medium-sized corpora.*

Introduction. Text classification represents one of the most sought-after tasks in natural language processing. It is employed for the automatic categorization of news articles, user inquiries, emails, reviews, scientific materials, and technical documentation. Unlike tabular data, text lacks a predefined set of numerical features: documents vary in length, style, terminology, and the prevalence of rare words. Consequently, prior to classification, text must be transformed into a numerical representation [1].

Contemporary research frequently leverages deep language models; however, these demand substantial computational resources and extensive training data. For conference proceedings, educational initiatives, and applied projects, classical methods remain essential: TF-IDF, Naive Bayes,

linear SVMs, logistic regression, and simple ensembles. These approaches train rapidly, perform effectively on sparse features, and offer interpretability through word weights.

The objective of this study is to evaluate the efficacy of classical models for classifying technical texts and to demonstrate their potential integration into hybrid solutions[2]. The core idea involves combining a probabilistic classifier, a margin-based model, and a character-word text representation.

Materials and Methods

The experimental corpus, LocalDocs-10, was assembled from Debian/Ubuntu software package descriptions, with package sections serving as document classes. A total of 10 classes were utilized: *graphics*, *libs*, *libdevel*, *utils*, *java* , *python*, *admin*, *devel*, *misc*, and *fonts*. This selection is well-suited for model evaluation, as it incorporates authentic technical lexicon and semantically overlapping categories [1-5].

Preprocessing entailed removing extraneous whitespace and line breaks, converting text to lowercase, and excluding standard English stop words. Documents were represented using word-level TF-IDF n-grams of length 1–2 and character-level TF-IDF n-grams (*char_wb*) of length 3–5. Character-level features prove particularly advantageous for technical texts, capturing fragments of library names, package identifiers, file extensions, and abbreviations.

For models challenged by large sparse matrices, dimensionality reduction was applied via TruncatedSVD to 32 latent features, followed by normalization. The dataset was split into training and test sets at a 70/30 ratio with class stratification. Macro F1-score was selected as the primary metric, as it equally weights performance across all classes.

Таблица 1. Краткое распределение документов в корпусе LocalDocs-10

Class	Documents	Class	Documents
graphics	60	libs	60
libdevel	60	utils	55
java	53	python	52
admin	37	devel	36
misc	24	fonts	23

3. Evaluated Models and the Hybrid Approach

In this work, we compared Bayesian methods, linear classifiers, prototype-based models, decision trees, ensembles, and the proposed hybrid variants. The models include: MultinomialNB, ComplementNB, BernoulliNB, LinearSVC, SGD-SVM, SGD-LogLoss, Passive-Aggressive, RidgeClassifier, Perceptron, NearestCentroid, KNN-SVD, DecisionTree-SVD, RandomForest-SVD, ExtraTrees-SVD, and AdaBoost-SVD.

The hybrid Char+Word LinearSVC combines word-level and character-level features. Word n-grams capture the semantic meaning of terms, while

character-level features improve robustness to rare words and technical designations. The Hybrid HardVote approach performs voting over several models: ComplementNB, SGD-SVM, Passive-Aggressive, and RidgeClassifier [2–6]. This ensemble scheme reduces the dependence of the final prediction on any single algorithm.

The main proposed solution is ALSHE-Gated. The model first applies ComplementNB and estimates the prediction confidence. If the maximal probability exceeds the threshold $\tau=0.62$, the Naive Bayes decision is accepted. If the confidence is below the threshold, the document is passed to LinearSVC. Thus, the fast probabilistic classifier is used for clear-cut cases, while the margin-based model handles more ambiguous documents.

In simplified form, the rule can be described as follows: when ComplementNB exhibits high confidence, the model adopts its prediction; otherwise, it uses the prediction of LinearSVC. The advantage of this approach lies in its simplicity, reproducibility, and the possibility of further extension without resorting to heavy neural networks.

4. Results and Discussion

The experimental results show that, for the LocalDocs-10 corpus, the strongest performers are linear models. The best result was achieved by the Passive-Aggressive Classifier: Accuracy = 0.855 and Macro F1 = 0.836. This can be explained by the fact that TF-IDF features are often well separated by linear decision boundaries, especially in the thematic classification of technical documents.

Table 2. Main model results on the test set

Модель	Accuracy	Macro F1	Краткий вывод
PassiveAggressive	0.855	0.836	best overall performance
NearestCentroid	0.841	0.824	strong and simple model
RidgeClassifier	0.848	0.823	robust linear method
SGD-LogLoss	0.833	0.820	competitive quality
LinearSVC	0.833	0.809	strong baseline
ALSHE-Gated	0.833	0.804	hybrid with adaptive selection
HardVote	0.826	0.800	simple voting over models
ComplementNB	0.790	0.774	fast training
AdaBoost-SVD	0.616	0.519	weaker on SVD features

Bayesian and SVD-based models. Bayesian methods demonstrated high speed but were outperformed by linear classifiers in terms of Macro F1. ComplementNB turned out to be more practical than MultinomialNB, which is related to its robustness to class imbalance. BernoulliNB was weaker because the binary representation loses information about term weights.

Models based on SVD features yielded mixed results. KNN-SVD and

ExtraTrees-SVD remained competitive, while DecisionTree-SVD and AdaBoost-SVD performed noticeably worse. This indicates that dimensionality reduction is not always beneficial: some of the important sparse structure of the TF-IDF representation can be lost.

Hybrid models did not surpass the leader but showed comparable quality and a more interesting architecture. ALSHE-Gated is particularly valuable as a basis for further development: one can automatically tune the confidence threshold, introduce uncertainty-related features, take document length into account, and incorporate embeddings.

5. Practical application and limitations

The obtained results can be applied in systems for automatic document sorting, university electronic archives, technical support, request classification, and processing of descriptions of software modules. In such tasks, one often needs not only high accuracy but also fast training, ease of deployment, and the ability to explain why a document was assigned to a particular category.

Classical TF-IDF models are convenient because they do not require GPUs, large servers, or extensive fine-tuning. This is especially important for small organizations, educational laboratories, and local information systems where continuous use of cloud-based neural models is not feasible. In addition, feature weights can be analyzed manually, for example, to see which words or character substrings most strongly influenced the assignment to classes such as *graphics*, *java*, or *python*.

A limitation of the study is the corpus size. LocalDocs-10 is suitable for a reproducible experiment but does not replace large benchmark datasets. The work also did not consider deep sentence-level semantics: models mainly relied on frequency-based and character-level features. Therefore, in future research it would be reasonable to test ALSHE-Gated on larger corpora and to add sentence embeddings as an additional source of features.

Despite these limitations, the work delivers an important practical insight: before applying complex neural architectures, it is useful to build a strong classical baseline. In many tasks such a baseline can provide sufficient accuracy at lower cost and serve as a reliable foundation for subsequent hybridization.

6. Conclusion

The conducted experiment confirms that classical machine learning methods remain effective for classifying small and medium-sized text corpora. On the LocalDocs-10 corpus the best method was the Passive-Aggressive Classifier with Macro F1 = 0.836. At the same time, hybrid approaches—especially ALSHE-Gated—demonstrate potential for building fast, interpretable, and adaptive classification systems.

The main scientific-practical result is that confidence-based switching between ComplementNB and LinearSVC makes it possible to construct a simple adaptive architecture without using heavy neural networks. Although ALSHE-Gated did not outperform the best base model in the current experiment, it provides a clear

pathway for further improvement: tuning the threshold τ, adding meta-features of confidence, analyzing errors, and expanding the feature space.

The practical significance of the work lies in the fact that the proposed scheme can be used in educational, research, and applied document-processing systems. In the future, we plan to evaluate the model on larger datasets, automatically optimize the threshold τ, and compare TF-IDF hybrids with transformer-based sentence embeddings.

References:

1. Scikit-learn developers. Working With Text Data. Scikit-learn documentation.
2. Pedregosa F. et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 2011, 12, pp. 2825-2830.
3. Cortes C., Vapnik V. Support-vector networks. Machine Learning, 1995, 20, pp. 273-297.
4. Sebastiani F. Machine learning in automated text categorization. ACM Computing Surveys, 2002, 34(1), pp. 1-47.
5. Joulin A., Grave E., Bojanowski P., Mikolov T. Bag of Tricks for Efficient Text Classification. EACL, 2017.
6. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT, 2019.