Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0041

Binary text classification using machine learning and transformer models for small and imbalanced datasets

İsmail Şenyurt*[1], Sinan Toklu[2]
[1]Graduate Student, Department of Computer Engineering, Graduate School of Natural and Applied Sciences, Gazi University, Ankara, Turkiye
[2]Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Turkiye

Cite this article as: Şenyurt, İ., & Toklu, S. (2026). Binary text classification using machine learning and transformer models for small and imbalanced datasets. *J Comp Electr Electron Eng Sci, 4*(2), 83-91. https://doi.org/10.51271/JCEEES-0041

Received: 14.06.2026 ◆ **Accepted**: 22.07.2026 ◆ **Published**: 13.10.2026

ABSTRACT

Aims: The increasing speed with which platforms providing digital content have entered our lives has made the classification of the content on these platforms important. In this study, the binary classification problem of Netflix content as a movie or show was addressed, the effect of metadata features and text descriptions on an accurate classification process was examined and a new hybrid approach was presented.

Methods: Two Netflix datasets from open sources (Kaggle and Hugging-Face) were used in the study. Furthermore, so as to eliminate the imbalance in the datasets, machine learning (ML) and deep learning (DL) models with natural language processing (NLP) were used separately and hybridly on the two datasets downloaded for shows and movies, both individually and combined. In this way, the dataset was examined separately as imbalanced and balanced, and the imbalance was eliminated by merging two datasets.

Results: The proposed hybrid framework achieved the results of 0.970940 accuracy, 0.982801 precision, 0.936768 recall and 0.959233 F1-score.

Conclusion: It has been demonstrated through experimental results that the obtained results are better than those of other models. In the experiments, various machine learning methods and transformer-based language models were evaluated; it was observed that the metadata-based models showed high performance in terms of accuracy, whereas BERT and similar models were particularly successful in detecting the 'show' class. The study reveals that the structured metadata and textual data are complementary in nature, and it is shown that hybrid approaches combining machine learning with transformer models would produce more balanced and robust results.

Keywords: Text classification, natural language processing, ensemble learning, hybrid learning, ELECTRA

INTRODUCTION

In addition to the existence of many artificial intelligence applications containing text data, big data is required for the model to learn meaningful representations effectively. More consistent and accurate results can be obtained by utilizing the positive aspects of one or more datasets. Recent studies have demonstrated that the use of natural language processing and machine learning techniques is increasing day by day to improve text classification performance. The dataset of this study was previously examined by Salsabila (2024), who evaluated text classification using Decision Tree and Random Forest algorithms, providing a baseline for further investigation. In addition to the proposed hybrid model, several machine learning and transformer models were evaluated under the same experimental conditions to provide a comprehensive performance comparison. Furthermore, unlike the study by Salsabila (2024), model performance was assessed using not only accuracy but also recall, precision and F1-score metrics to provide a more comprehensive evaluation.

In this study, binary text classification experiments were conducted on a small and initially imbalanced dataset. To mitigate the effects of class imbalance, the original dataset was integrated with a second dataset. The study further investigates the effect of combining transformer-based text representations with both traditional machine learning and hybrid approaches under a unified experimental framework. In addition, the performance of these approaches was examined on both the original and the merged datasets to assess the impact of dataset balancing.

Despite the remarkable advances in transformer-based natural language processing, hybrid and ensemble learning for small and imbalanced binary text classification datasets remain limited. To address this gap, this study presents a unified experimental framework that systematically compres these approaches under identical preprocessing, training and evaluation conditions.

***Corresponding Author:** İsmail Şenyurt, 24802002006@gazi.edu.tr

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

The main contributions of this study are summarized as follows;

A comprehensive framework is presented to compare machine learning, transformer-based, hybrid and ensemble learning approaches for binary text classification on small and initially imbalanced datasets.

The effectiveness of integrating efficiently learning an encoder that classifies token replacements accurately (ELECTRA)-based contextual text representations with conventional machine learning algorithms is investigated through a hybrid framework, providing a detailed comparison with standalone transformer models.

The impact of balanced datasets is analyzed systematically to show its influence on the classification performance of different learning approaches.

An ensemble learning strategy is evaluated to investigate whether combining multiple classifiers can further improve the predictive performance of transformer-based hybrid models.

The framework is established by reporting both overall and class-wise performance metrics (accuracy, precision, recall, F1-score), providing a more reliable assessment of classifier behaviour, particularly for small and imbalanced datasets.

RELATED WORKS

One of the studies conducted in this field is the work of Shafiee (2024). Although this study examines in detail how machine learning techniques are used in recommendation systems, it does not provide systematic experimental comparisons regarding the text-based classification performance of transformer-based large language models (**Table 1**).

Additionally, in another study, Motamedi et al. (2024) applied a ML method based on metadata, audio and visual features to predict movie scores. In their study, the problem was treated as two separate binary classifications (high/low E and H) and classical ML algorithms such as logistic regression (LR), support vector machines (SVM), decision trees (DT), k-nearest neighbours and random forests (RF) were evaluated using the nested cross-validation method. The results showed that metadata features, particularly those containing genre information, provided the highest performance. DL models were also tested, but they did not yield better results than classical ML methods due to the relatively small size of the dataset (709 films). In another important study in this field, Shaikh and Singh (2025) used the cancer-risk-factors.csv dataset, which includes demographic, lifestyle, genetic and environmental factors to comparatively evaluate SVM, LR, RF and XGBoost algorithms for the purpose of predicting

Table 1. Similar studies in the literature

Study	Preferred dataset	Models	Positive ways	Limitations	Accuracy-precision-recall-F1-score
Motamedi et al. (2024)	IMDb	LR, SVC, DT, RF,	Using Cross validation technique	Dataset is limited, contains only 709 films	0.900-0.894-0.894-0.895
Salsabila (2024)	Netflix dataset	RF, DT	Feature extraction methods were added by using DT and RF algorithms	Study is limited with one dataset. Evaluations were done with limited features	0.968, -,-,-
Proposed study	Two different datasets from Kaggle and hugging face	LR, SVM, RF, DT, BERT, RoBERTa, DeBERTa, DistillBERT, ELECTRA	Utilizing both ML and DL methods	Datasets are not enough for DL models	0.970-0.982-0.936-0.939
Shaikh and Singh (2025)	cancer-risk-factors.csv	SVM, LR, RF, XGBoost	Multi-label prediction and evaluation of all metrics	Class imbalance	0.900-0.894-0.900-0.896
Dritsas and Trigka (2025)	NYC Taxi Trip Duration, Netflix Prize, Higgs Boson	LinR, LR, RF, GBT, SVR, KNN	Implementing more than one dataset, evaluation and monitoring of CPU/RAM computations	Limited with only ML models	0.900-0.911-0.892-0.901
Galke et al. (2025)	20NG, r8, r52, OHSUMED, MR	RNN, SGM, LSTM, BERT, GPT, RoBERTa, DistillBERT, DeBERTa	Using several ways such as LLM and SLM and graph based approaches	Only comparison, not available any new method	0.872,-,-,-
Nilla and Setiawan (2024)	IMDb data from Twitter	CNN, RoBERTa	Merging RoBERTa and CNN1d and using SMOTE	Only accuracy evaluation	0.864,-,-,-
Chaudhary (2024)	Short Jokes(kaggle), non humorous dataset (Reuters)	SBERT, LR, SVC, NB, BERT, DistillBERT, RoBERTa	Using both ML and DL methods	Results obtained from DL models are lower than ML models, that's why has limited generalization power	0.784,-,-, 0.781
Ammar and Rania (2021)	Arabic Corpus, AJGT, IMDB review, SemEval, COVID19-Fake, ArSarcasm	GB, SVM, NB, LR, RF, LSTM, GRU, CNN, LSTM	Using ensemble method with six datasets	Only accuracy evaluation	0.781,-,-,-
Wang et al. (2024)	COVID-19-related Tweets, Economiz Texts, E-Commerce Texts, SMS Spam Collection	MNB, LR, RF, DT, KNN, GRU, LSTM, RNN, DeBERTa, GPT 3.5, GPT 4, Gemini-pro, Llama 3, Qwen 14b	Using both ML and DL methods	The success of GRU model depends on fine tuning and computation capacity	0.938,-,-,0.938

IMDb: Internet movie database, LR: Logistic regression, SVC: Support vector classifier, DT: Decision tree, RF: Random forest, SVM: Support vector machine, BERT: Bidirectional encoder representations from transformers, RoBERTa: Robustly optimized BERT pretraining approach, DeBERTa: Decoding-enhanced BERT with disentangled attention, DistilBERT: Distilled BERT, ELECTRA: Efficiently learning an encoder that classifies token replacements accurately, XGBoost: Extreme gradient boosting, NYC: New York city, LinR: Linear regression, GBT: Gradient-boosted trees, SVR: Support vector regressor, KNN: K-nearest neighbors, CPU: Central processing unit, RAM: Random access memory, 20NG: 20 newsgroups, R8: Reuters-8, R52: Reuters-52, OHSUMED: OHSUMED medical literature dataset, MR: Movie reviews, RNN: Recurrent neural network, SGM: Sequence generation model, LSTM: Long short-term memory, GPT: Generative pre-trained transformer, CNN: Convolutional neural network, SMOTE: Synthetic minority over-sampling technique, SBERT: Sentence-BERT, NB: Naive bayes, GB: Gradient boosting, GRU: Gated recurrent unit, MNB: Multinomial naive bayes, LLM: Large language model, SLM: Small language model, DL: Deep learning, ML: Machine learning

cancer risk levels and the problem was treated as a multi-class classification problem. According to the results obtained, the LR model showed the highest performance with an accuracy rate of 90%, followed by the SVM, XGBoost and RF models. Precision, recall and F1-score metrics were also reported and model performance was detailed through confusion matrix analysis. However, the study primarily focused on the accuracy metric, did not comprehensively address the class imbalance problem, and did not evaluate DL based methods.

In their study on the Apache Spark ecosystem, Dritsas and Trigka (2025) addressed a multi-class classification problem aimed at categorizing movie ratings using a large-scale movie dataset. In the study, a classical ML approach was adopted and by using the RF algorithm, values of 0.90 accuracy, 0.91 precision, 0.892 recall and 0.901 F1-score were obtained. These results demonstrated that ensemble-based methods can be effective in multi-classification problems within big data environments and DL algorithms and transformer-based architectures were not included in the study. Additionally, in another study, Galke et al. (2025) systematically examined the progress in the field of text classification, comparing a wide family of methods ranging from BoW-based classical ML models to transformer-based language models and graph-based approaches. The study revealed that fine-tuned SLMs in particular, exhibited more stable and higher performance compared to in-context LLMs. It was reported that graph-based methods failed to surpass strong baselines in most cases.

In a study similar to these topics conducted by Nilla and Setiawan (2024), a hybrid movie recommendation system was developed by combining the content-based filtering method with a RoBERTa-based word embedding approach and 1D convolutional neural network (CNN) classification. In the study, 845 movies and 34.086 Twitter comments were used and data imbalance was addressed using the SMOTE method. As a result of a three-stage experimental design, the CNN model achieved 86.41% accuracy with SGD optimization and an appropriate learning rate. Although this study presents a structure that integrated deep learning classification with content-based filtering, it did not directly fine-tune transformer models and did not report detailed performance metrics such as precision, recall and F1-score.

Another study in the literature is the one conducted by Chaudhary (2024). In this study, classical ML methods and DL models were compared for the classification of social media texts. Traditional methods such as Naive Bayes, SVM and LR as well as CNN, LSTM and BiLSTM architectures, were conducted in the study. Furthermore, an accuracy of 0.7767 and F1-score of 0.7819 were obtained with DistilBERT; an accuracy of 0.7823 and F1-score of 0.7772 with the BERT model; an accuracy of 0.7841 and F1-score of 0.7778 with the RoBERTa model and these values lagged behind the results obtained from ML methods. In another study, Ammar and Rania (2021) applied both ML and DL techniques using the IMDB dataset, one of their utilized datasets and achieved accuracy values of 0.7815 with LSTM1 and 0.7841 with GRU1. In the study, individual models were compared with an ensemble structure and it was demonstrated that the ensemble model exhibited higher and more stable performance in terms of accuracy and F1-score, particularly on the IMDB dataset. Additionally, Wang et al. (2023) comparatively evaluated traditional ML algorithms, DL models and LLMs in the problem of classifying e-commerce product texts. In the study, the GRU model showed the highest performance among models without fine-tuning, achieving 0.9387 accuracy and 0.9383 F1-score; this revealed that GRU's sequence and context modeling capacity is effective in product text classification.

Although previous studies have demonstrated the effectiveness of both conventional machine learning and transformer-based approaches for text classification, most studies focus on a single learning paradigm. Furthermore, few studies systematically compare machine learning, transformer-based, hybrid, and ensemble learning approaches under identical experimental conditions while simultaneously addressing class imbalance through dataset enrichment and reporting class-wise performance metrics. These limitations constitute the primary motivation for the proposed framework.

METHODS

Ethics

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data. All procedures were carried out in accordance with the ethical rules and principles.

Dataset Description

This study aims to provide a comprehensive comparison of traditional machine learning and transformer-based models for binary text classification. Two publicly available Netflix datasets were employed in this study. The primary dataset was obtained from Kaggle (https://www.kaggle.com/datasets/victorsoeiro/netflix-tv-shows-and-movies?select=titles.csv), while a second dataset was collected from Hugging Face (https://huggingface.co/datasets/hugginglearners/netflix-shows) to support the dataset balancing process described in section 3.2.

The datasets contain both numerical and textual information describing Netflix content. Each row includes attributes such as content title, description, release year and content type (show or movie). The combination of heterogeneous feature types enables the evaluation of both machine learning algorithms and transformer-based language models within the same experimental framework.

The primary Kaggle dataset consists of 5.849 instances represented by 15 features. The target feature is the content type, where each instance is classified as either a movie or a show. The original dataset has an imbalanced distribution with nearly 64% of the rows belonging to the movie class and 36% to the show class. The detailed preprocessing and balancing procedures are presented in section 3.2.

Data Preprocessing and Dataset Balancing

Prior to model development, the datasets underwent a comprehensive preprocessing procedure to improve data quality and minimize bias. Missing parts of the dataset were examined first and replaced with the value 'unknown' to preserve the available samples. In addition, the 'runtime' and 'seasons' features were excluded because they might point the target class directly. Duplicate records were identified and removed in order to improve the generalization power of the models. Furthermore, textual descriptions containing clue words such as 'movie, show, series, television and mini-series' were eliminated to prevent models from making biased

predictions. Finally, after the dataset was divided into training and test subsets an additional duplicate check was performed to prevent potential data leakage.

The original dataset (DF1) consisted of 3.744 movie instances and 2.106 show instances, resulting in an imbalanced class distribution. To investigate the most appropriate balancing strategy, DF1 was merged with another Netflix dataset (DF2), which contained 6.131 movie instances and 2.676 show instances. The dataset consisted of 9875 movie samples and 4782 show samples after combining. Four approaches were considered: (1) increasing the number of show class to 9875 by oversampling, (2) reducing the movie class to 4782 by downsampling, (3) using the merged dataset without doing anything. (4) combining show samples (reaching 4782 rows) from two datasets and adding 1038 movie samples of second dataset to first dataset (containing 3744 movie sample) to balance each class. The relatively small size of the dataset was the main reason why oversampling and downsampling approaches did not provide the expected improvements. During oversampling, the minority class had to be augment with nearly its original size, while downsampling reduced limited training data. Therefore, the fourth strategy, which preserved minimal sampling process was selected for the experiments. The comparative performance of these ways is presented in the result section.

Experimental Workflow

Figure 1 illustrates the whole experimental workflow adopted in this study. Data collection, preprocessing and balancing parts were described in previous sections. The dataset was divided into test and training sets with an 80:20 ratio. In addition, 5-fold-cross-validation was employed during the model development phase to reduce the risk of overfitting. The experimental framework consists of four consecutive stages. In the first stage, traditional machine learning algorithms were trained and in the second stage, several transformer-based models were fine-tuned on the same dataset. In the third stage, transformer models were integrated with machine learning classifiers to construct a hybrid model and ensemble techniques were employed for further performance. Finally, all models were evaluated using the same experimental processes and detailed descriptions of model architectures are presented in the following sections.

Machine Learning Models

Machine learning algorithms were employed as the first stage of model development to set a baseline performance. Owing to their relatively low computational complexity, short training time and effectiveness of using small datasets, conventional machine learning models remain a suitable choice for text classification tasks. All machine learning models were trained on the same preprocessed dataset using an 80:20 ratio. This protocol is widely adopted in the literature for text classification cases. For example, Agrawal et al. (2025) reported that an 80:20 train-test split produced the highest classification accuracy, supporting the selection of the same partitioning strategy in the present study. Several machine learning algorithms were evaluated and these models include comprehensive comparison strategies. Their performance serves as a foundation for the hybrid and ensemble learning framework introduced in the subsequent sections.

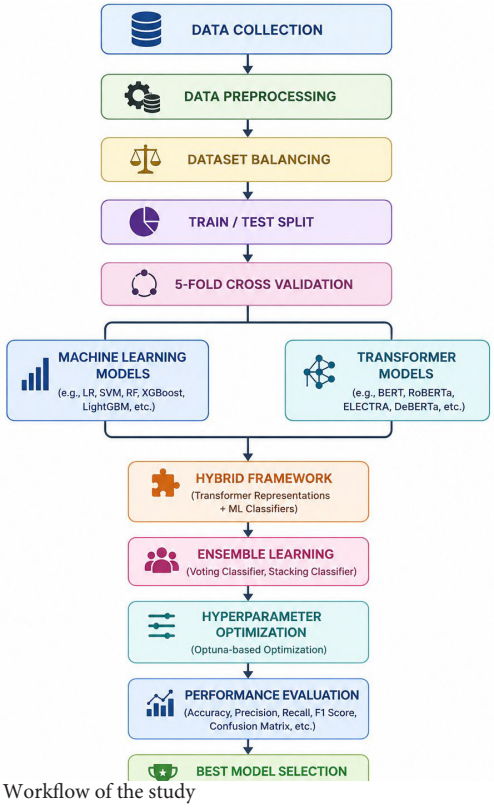


Figure 1. Workflow of the study

Transformer-Based Models

Transformer-based models were employed in the second stage of the experimental framework to investigate the contribution of contextual representations to binary text classification. Unlike conventional machine learning algorithms, transformer architectures are capable of capturing semantic relations and long-range text dependencies within the data by using attention mechanisms. Since the 'description' feature contains more semantic information defining each Netflix title, transformer models were expected to provide more information than traditional approaches. To provide a comprehensive comparison, six publicly available pretrained transformer models were evaluated, namely BERT-base-uncased, DistilBERT, RoBERTa, DeBERTa, ALBERT, ELECTRA. These models have different architectures such as computational complexity and pretraining strategy. Among them, ELECTRA was selected as the primary model because it achieved the most competitive overall performance during experiments and also it was used to construct a hybrid model in section 3.6. This hybrid design enables to evaluate a standalone classification and to integrate as a component within hybrid and ensemble approaches.

Hybrid and Ensemble Framework

Instead of relying solely transformer Electra model, a hybrid design was enabled to combine efficient ways of machine learning approaches. Within the proposed framework, the textual descriptions were first processed by the ELECTRA model to generate contextual embedding vectors. These feature representations were subsequently used as input for several machine learning classifiers including logistic regression, naive bayes, support vector machine, random forest, XGBoost and LightGBM. To further enhance predictive performance and improve robustness, ensemble learning techniques (voting and stacking classifiers) were incorporated into the proposed framework. This two-stage framework enables a systematic

comparison between transformer models, hybrid models and ensemble approaches under identical experimental conditions.

Hyperparameter Optimization

Hyperparameter optimization was performed for the ELECTRA model using the Optuna framework. Optuna was selected due to enable automatic exploration of the hyperparameter space while reducing unnecessary evaluations. A total of 20 trials were conducted, and during each trial, the model was evaluated on a dedicated validation set.

The optimization process included the tuning of several key hyperparameters, including the learning rate, batch size, number of training epochs, maximum sequence length, weight decay, warm-up ratio and dropout rate. The weighted F1-score obtained on the validation set to provide a more reliable performance score than accuracy. The highest validation score was selected as the final ELECTRA model and implemented subsequent experiments.

Performance Evaluation

The performance of all models was evaluated using a comprehensive set of metrics to ensure a reliable comparison. Therefore, in addition to accuracy, precision, recall and F1-score were adopted in order to assess the classification performance of models. To provide a more detailed analysis of class behaviours, class-wise performance metrics were also reported for both show and movie classes. Class-specific performance metrics enable the effectiveness of each model to be assessed individually and also help to understand the effect of dataset balancing and different learning strategies.

RESULTS

Baseline Comparison and Reproduction

The experimental configuration proposed by Salsabila (2024) was first reproduced using the same dataset and machine learning algorithms to establish a reliable baseline. The original study employed four features (release year, age certification, genres and production countries) and evaluated the performance of decision tree and random forest algorithms using different numbers of decision trees. The reproduced baseline results are presented in Table 2.

Table 2. Results obtained in the reference study conducted by Salsabila (2024)

	Accuracy	Correct prediction	Wrong prediction
Decision tree	0.9581	1121	49
Random forest (n=10)	0.9675	1132	38
Random forest (n=50)	0.9684	1133	37
Random forest (n=100)	0.9675	1132	38
Random forest (n=200)	0.9667	1131	39

To investigate the influence of categorical feature encoding on classification performance, the label encoder used in the reference study by Salsabila (2024) was replaced with one-hot encoding with same four features. One-hot encoding was preferred because most of the features in the dataset are categorical and do not possess a natural ordinal relationship. One-hot encoding represents each category independently, providing a more suitable feature representation for classifiers. The comparative results are presented in Table 3.

Table 3. Comparative results derived from differences during the preprocessing stages

Model	Encoding	Accuracy	Precision	Recall	F1-score
Decision tree	Label encoder	0.9581	0.9582	0.9581	0.9582
	One-hot encoder	0.9615	0.9548	0.9391	0.9469
Random forest (n=10)	Label encoder	0.9675	0.9676	0.9675	0.9674
	One-hot encoder	0.9658	0.9801	0.9251	0.9518
Random forest (n=50)	Label encoder	0.9684	0.9685	0.9684	0.9682
	One-hot encoder	0.9709	0.9828	0.9368	0.9592
Random forest (n=100)	Label encoder	0.9675	0.9676	0.9675	0.9674
	One-hot encoder	0.9701	0.9804	0.9368	0.9581
Random forest (n=200)	Label encoder	0.9667	0.9668	0.9667	0.9665
	One-hot encoder	0.9692	0.9803	0.9344	0.9568

As shown in Table 3, replacing the encoding technique led to an increase in accuracy and precision, whereas recall decreased slightly. These results suggest that representing categorical variables independently provides a more informative feature representation, resulting in improved overall classification performance.

While classifying the 'description' feature was incorporated into the four metadata features in the Salsabila's (2024) study to investigate the contribution of textual information. The proposed configuration additionally exploited semantic information contained in textual descriptions in addition to baseline model. The comparative results are presented in Table 4.

As shown in Table 4, incorporating the 'description' feature significantly improved the recall and F1-score, while only a slight reduction in accuracy and precision was observed. These results indicate that the semantic information contained in textual descriptions improves the model's ability of classifying.

Performance of Machine Learning Models

The performance of the Naïve Bayes classifier was strongly influenced by the selected feature set among the other machine learning algorithms. Although the additional features improved the performance of the Naïve Bayes classifier, other algorithms, particularly random forest, achieved superior overall results. This observation suggests that while Naïve Bayes provides a competitive baseline for binary text classification, models capable of capturing more complex relationships between textual and structured features are better suited for this classification task.

Table 4. Effect of using different features in data preprocessing on model performance

Case	Accuracy	Precision	Recall	F1-score
The results of Salsabila's (2024) study with 4 columns (release year, age certification, genres, production countries)	0.9684	-	-	-
The results of 5 features (4 features of Salsabila study + description column) (proposed method)	0.9605	0.9446	0.9985	0.9708

Performance of Deep Learning Models

Conventional machine learning models could achieve competitive performance, but they rely primarily on special feature representations and have limited capability in learning deep contextual relationships within textual data. Therefore, transformer-based deep learning models were further preferred to exploit their stronger representation learning capability and higher computational capacity. Before model training, the 'runtime' and 'seasons' features were excluded because they directly revealed the target class and could introduce data leakage.

Zero-shot learning, few-shot learning, full fine-tuning of transformer architectures and embedding-based hybrid approaches were investigated to elaborately evaluate different deep learning strategies. Six pretrained transformer models, namely BERT-base, DistilBERT, RoBERTa, DeBERTa, ALBERT, and ELECTRA, were evaluated under identical experimental settings to provide a fair comparison.

As presented in Figure 2, zero-shot learning produced the lowest performance as expected, because no task-specific adaptation was performed. Fine-tuning with a limited number of labeled samples substantially improved the results in the few-shot setting. Among the fully fine-tuned transformer architectures, ELECTRA achieved the highest weighted F1-score.

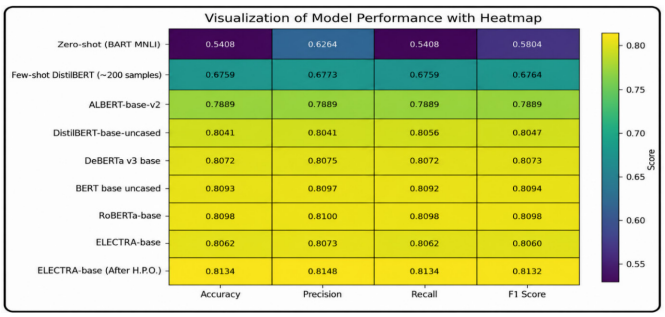


Figure 2. Comparison of values obtained with deep learning models

Class-wise performance results for ELECTRA are presented in Table 5 to provide a detailed evaluation. While the Show class obtained higher recall (0.8370), the movie class achieved higher precision and F1-score (0.8120), indicating that the proposed framework successfully identified bpth classes without bias toward either category. These findings demontrate that balancing and preprocessing strategies contributed to improving generalization power of models.

Table 5. Class-wise performance of ELECTRA model

Class	Precision	Recall	F1-score	Support
Movie	0.8263	0.7753	0.8000	957
Show	0.7884	0.8370	0.8120	957
Macro average	0.8073	0.8062	0.8060	1914
Weighted average	0.8073	0.8062	0.8060	1914

ELECTRA: Efficiently learning an encoder that classifies token replacements accurately

Figure 3 Comparison of values obtained with deep learning models presents the training and validation performance of the ELECTRA model throughout the training process. While the training loss continuously decreased with each epoch, the validation loss began to increase after the third epoch, indicating a sign of overfitting. A similar trend was observed for the accuracy curves, where training accuracy continued to improve whereas validation accuracy plateaued. To preserve the model's generalization capability, dropout regularization and early stopping were employed, preventing further performance degradation on the validation set.

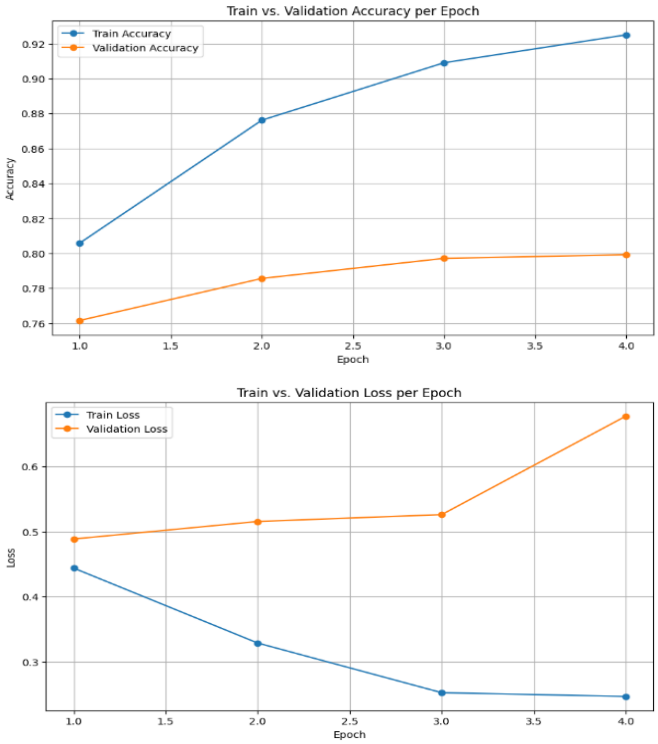


Figure 3. Performance during validation process

Performance of Hybrid and Ensemble Models

Following the comparative evaluation of the transformer architectures, ELECTRA was selected for the proposed hybrid framework because it achieved the highest overall performance among the transformer models. The contextual embeddings generated by ELECTRA were subsequently combined with conventional machine learning classifiers to investigate whether transformer-based feature representations could further improve classification performance.

To evaluate the influence of dataset composition, the proposed hybrid framework was trained separately using DF1, DF2, and the merged DF3 dataset. The corresponding results are presented in Table 6-8. Among the evaluated datasets, the merged dataset (DF3) consistently achieved the highest classification performance, indicating that the enriched dataset provided a more suitable foundation for learning robust feature representations. Consequently, all subsequent hybrid and ensemble experiments were conducted using DF3.

Table 6. Results of DF1 dataset with several ML models

Model	Accuracy	Precision	Recall	F1-score
SVM	0.8259	0.8063	0.8581	0.8314
Multinomial NB	0.8039	0.7597	0.8889	0.8192
Logistic regression	0.7871	0.7705	0.8179	0.7935
Random forest	0.7845	0.8305	0.7149	0.7683
LightGBM	0.7811	0.7933	0.7604	0.7765
VotingClassifier	0.8293	0.8263	0.8340	0.8301
StackingClassifier	0.8400	0.8396	0.8407	0.8401

SVM: Support vector machine, NB: Naive bayes, LightGBM: Light gradient boosting machine

Table 7. Results of DF2 dataset with several ML models

Model	Accuracy	Precision	Recall	F1-score
SVM	0.8256	0.8108	0.8493	0.8296
Multinomial NB	0.8038	0.7704	0.8657	0.8153
Logistic regression	0.7957	0.7940	0.7985	0.7962
Random forest	0.8391	0.9500	0.7158	0.8164
LightGBM	0.8112	0.8661	0.7363	0.7959
VotingClassifier	0.8542	0.8771	0.8239	0.8497
StackingClassifier	0.8726	0.8957	0.8436	0.8688

SVM: Support vector machine, NB: Naive bayes, LightGBM: Light gradient boosting machine

Table 8. Results of DF3 dataset with several ML models

Model	Accuracy	Precision	Recall	F1-score
SVM	0.8463	0.8277	0.8747	0.8505
Multinomial NB	0.8187	0.7864	0.8752	0.8284
Logistic regression	0.8131	0.8087	0.8201	0.8144
Random forest	0.8266	0.8848	0.7509	0.8124
LightGBM	0.8039	0.8509	0.7369	0.7898
VotingClassifier	0.8587	0.8640	0.8514	0.8577
StackingClassifier	0.8803	0.8915	0.8660	0.8786

SVM: Support vector machine, NB: Naive bayes, LightGBM: Light gradient boosting machine

The comparative performance of the transformer, hybrid, and ensemble approaches is summarized in Table 9.

Table 9. Results of DF3 dataset with deep learning, hybrid and ensemble methods

Model	Accuracy	Precision	Recall	F1-score
ELECTRA (full fine-tune)	0.8177	0.8191	0.8177	0.8174
ELECTRA+ML (best RF)	0.8223	0.8223	0.8223	0.8223
ELECTRA+ML (stacking)	0.8228	0.8229	0.8228	0.8228

ELECTRA: Efficiently learning an encoder that classifies token replacements accurately, ML: Machine learning, RF: Random forest

ELECTRA+random forest achieved the highest performance among the hybrid models. Furthermore, the Stacking ensemble slightly outperformed the best hybrid model, increasing the weighted F1-score from 82.23% to 82.28%. Although the numerical improvement is modest, the result indicates that ensemble learning could provide a performance gain.

Hyperparameter Optimization Results

The influence of hyperparameter optimization on the ELECTRA model is summarized in Table 10. Compared with the default configuration, the optimized parameter setting resulted in improved classification performance and provided the final configuration used throughout the hybrid and ensemble experiments. The optimization process identified suitable values for the learning rate, weight decay, warm-up ratio, batch size, maximum sequence length, and dropout probabilities, enabling the model to achieve a better balance between fitting capability and generalization performance.

In this study, when the computational costs of experiments conducted on virtual servers (Google Colab) were compared-ranging from data preprocessing to classical ML, DL and hybrid/ensemble architectures. It was found that the preprocessing stages had a negligible duration, while each transformer-based model incurred a computational costof approximately 5 hours for training. Additionally, the hyperparameter optimization process performed on the

Table 10. Hyperparameter tuning values

Hyperparameter	Search space	Type	Optimal value (selected)
Learning rate	1×10^{-6}-5×10^{-5} (log-uniform)	Continuous	1.7048×10^{-5}
Batch size	{8, 16, 32}	Categorical	16
Number of epochs	{3, 4, 5}	Categorical	4
Max sequence length	{128, 192, 256}	Categorical	256
Weight decay	0-0.1 (log-uniform)	Continuous	0.0611
Warm-up ratio	0-0.1 (uniform)	Continuous	0.074
Hidden dropout	0-0.3 (uniform)	Continuous	0.039
Attention dropout	0-0.3 (uniform)	Continuous	0.152
Optimizer	AdamW	Fixed	AdamW
Loss function	Binary cross-entropy	Fixed	Binary cross-entropy
Evaluation metric	Weighted F1-score	Fixed	Weighted F1-score

ELECTRA model was the most costly stage and required approximately 6 to 8 hours of additional processing time (experiments with other models were not included into these calculations because of failure/missing results). On the other hand, while the CLS embedding extraction and experiments conducted with five different ML models in the Electra-based hybrid approach on the combined dataset were completed in nearly 1 hour, testing the ML methods as a hybrid took 10 minutes and the stacking ensemble structure required an additional 30 minutes of computation time. Overall, results indicate that while transformer-based models provide high accuracy, they require significant computational cost; conversely, hybrid and classical methods can offer competitive performance with much lower resource consumption.

DISCUSSION

This study presented a comprehensive comparison of conventional machine learning algorithms, transformer-based language models, and hybrid and ensemble learning frameworks for binary text classification on small and initially imbalanced datasets. Unlike previous studies that relied primarily on metadata features or evaluated a limited number of models, this work systematically investigated the contribution of data preprocessing, dataset balancing, transformer-based representation learning, and hybrid learning strategies within a unified experimental framework. The results demonstrate that each stage of the proposed workflow contributed to improving the robustness and reliability of the classification models.

The experimental findings show that preprocessing decisions substantially influenced classification performance. Replacing Label Encoding with One-Hot Encoding provided a more appropriate representation for categorical variables. Furthermore, incorporating the description feature significantly improved the model's ability to capture semantic information contained in textual data, resulting in higher recall and weighted F1-score values. These findings confirm that combining structured metadata with textual information produces more informative feature representations than relying on metadata alone.

The comparative evaluation of transformer architectures demonstrated that pretrained language models can effectively

exploit contextual semantic information for binary text classification. Among the evaluated architectures, ELECTRA consistently achieved the best overall performance and was therefore selected as the backbone of the proposed hybrid framework. Rather than using the conventional transformer classification head alone, the contextual embeddings generated by ELECTRA were successfully integrated with conventional machine learning classifiers. The results further demonstrated that the ELECTRA+random forest hybrid model outperformed the standalone transformer model, while the Stacking ensemble achieved the highest overall performance, indicating that combining complementary learning strategies provides a consistent improvement over individual model.

Another important finding concerns dataset balancing. Instead of relying solely on oversampling or downsampling techniques, a second publicly available dataset with similar characteristics was incorporated to enrich the training data and reduce class imbalance. The merged dataset (DF3) consistently produced better classification performance than the individual datasets and improved model generalization. In addition, reporting class-wise precision, recall, and F1-score enabled a more comprehensive evaluation of classifier behaviour for both the majority and minority classes, providing a more reliable assessment than overall accuracy alone.

Limitations

Although the proposed framework achieved competitive classification performance, several limitations should be acknowledged. First, despite dataset enrichment, the overall training corpus remains relatively small compared with the large-scale datasets typically used to fully exploit transformer architectures. Second, only English-language Netflix content was considered, which may limit the generalizability of the findings to multilingual or domain-specific datasets. Finally, the computational cost of transformer fine-tuning remains substantially higher than that of conventional machine learning models, although the proposed hybrid framework considerably reduces the additional computational overhead after feature extraction.

CONCLUSION

Future research may extend this work by evaluating the proposed framework on larger and multilingual text collections, investigating more recent large language models and encoder architectures. Furthermore, adaptive ensemble strategies and multimodal learning approaches that jointly utilize textual and non-textual information represent promising directions for improving binary text classification performance.

ETHICAL DECLARATIONS

Ethics Committee Approval

Ethical approval was not required for this study as it does not involve human participants, animal subjects, or identifiable personal data.

Peer Review Process

This manuscript was subject to external peer review.

Conflict of Interest

The authors declare no conflicts of interest related to this study.

Financial Disclosure

The authors received no financial support for the conduct or publication of this research.

Author Contributions

Concept: İŞ, ST; Design: İŞ, ST; Control: İŞ, ST; Resources: İŞ, ST; Materials: İŞ, ST; Data Collection and/or Processing: İŞ, ST; Analysis and/or Interpretation: İŞ, ST; Literature Review: İŞ, ST; Writing the Article: İŞ, ST; Critical Review: İŞ, ST.

Editor Note

Sinan Toklu is a member of the Editorial Board of this journal. He was not involved in the editor assignment, reviewer selection, peer review process, or editorial decision for this manuscript. The peer review process was managed independently by another editor.

REFERENCES

Aburass, S., Dorgham, O., & Abu Rumman, M. (2024). An ensemble approach to question classification: integrating ELECTRA transformer, GloVe, and LSTM. *International Journal of Advanced Computer Science and Applications, 15*(1), 507-514. https://doi.org/10.14569/IJACSA.2024.0150148

Agrawal, R., Gupta, T., Gupta, S., Chauhan, S., Patel, P., & Hamdare, S. (2025). Fostering trust and interpretability: integrating explainable AI (XAI) with machine learning for enhanced disease prediction and decision transparency. *Diagnostic Pathology, 20*, 105. https://doi.org/10.1186/s13000-025-01686-3

Behmanesh, S., Talebpour, A., Shamsfard, M., & Jafari, M. M. (2023). Improved relation span detection in question answering systems over extracted knowledge bases. *Expert Systems with Applications, 224*, 119973. https://doi.org/10.1016/j.eswa.2023.119973

Biswas, D., Byun, T. Y., & Gil, J. M. (2025). Research paper classification based on CNN and BiLSTM models utilizing word embedding methods. *Human-Centric Computing and Information Sciences, 15*, 24. https://doi.org/10.22967/HCIS.2025.15.024

Chaudhary, T. (2024). Humorous and professional: a dive into social media text classification. Master's thesis, International Institute of Information Technology Hyderabad.

Das, S. D., Vadi, Y. A., & Yadav, K. (2024). Transformer-based joint modelling for automatic essay scoring and off-topic detection. In Calzolari, N., Kan, M. Y., Hoste, V., Lenci, A., Sakti, S., & Xue, N. (Eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (pp. 16751-16761).

Filali, A., Shorfuzzaman, M., Abdellaoui Alaoui, E. A., Merras, M., Es-Sabry, M., Berrajaa, A., ..., & Yousef, A. (2026). Cross-lingual SMS spam detection using GAN-based augmentation for imbalanced datasets. *Scientific Reports, 16*, 7128. https://doi.org/10.1038/s41598-026-37769-4

Gasparetto, A., Marcuzzo, M., Zangari, A., & Albarelli, A. (2022). A survey on text classification algorithms: from text to predictions. *Information, 13*(2), 83. https://doi.org/10.3390/info13020083

Krishnan, A. (2023). Exploring machine learning and transformer-based approaches for deceptive text classification: a comparative analysis. *Inteligencia Artificial, 20*(59), 123-127.

Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., ..., & He, L. (2022). A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology, 13*(2), 1-41. https://doi.org/10.1145/3495162

Lu, H., Ehwerhemuepha, L., & Rakovski, C. (2022). A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance. *BMC Medical Research Methodology, 22*, 181. https://doi.org/10.1186/s12874-022-01665-y

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning-based text classification: a comprehensive review. *ACM Computing Surveys, 54*(3), 1-40. https://doi.org/10.1145/3439726

Mohammed, A., & Kora, R. (2022). An effective ensemble deep learning framework for text classification. *Journal of King Saud University-Computer and Information Sciences*, *34*(10), 8825-8837. https://doi.org/10.1016/j.jksuci.2021.11.001

Mu, Y., Jin, M., Song, X., & Aletras, N. (2024). Enhancing data quality through simple de-duplication: navigating responsible computational social science research. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (pp. 12477-12492). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-main.694

Nilla, A., & Setiawan, E. B. (2024). Film recommendation system using content-based filtering and the convolutional neural network (CNN) classification methods. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika, 10*(1), 17-29. https://doi.org/10.26555/jiteki.v10i1.28113

Pham, P., Nguyen, L. T. T., Pedrycz, W., & Vo, B. (2023). Deep learning, graph-based text representation and classification: a survey, perspectives and challenges. *Artificial Intelligence Review, 56*(6), 4893-4927. https://doi.org/10.1007/s10462-022-10265-7

Rizvi, S. M. H., Imran, R., & Mahmood, A. (2025). Text classification using graph convolutional networks: a comprehensive survey. *ACM Computing Surveys, 57*(8), 1-38. https://doi.org/10.1145/3714456

Shaukat, F., Ejaz, N., Ashraf, Z., Alnfiai, M. M., Alotaibi, N. N., & Alnefaie, S. M. M. (2025). An interpretable multi-transformer ensemble for text-based movie genre classification. *PeerJ Computer Science, 11*, e2945. https://doi.org/10.7717/peerj-cs.2945

Shaukat, F., Ejaz, N., Kamal, R., Alkhalifah, T., Aslam, S., & Mu, M. (2025). Multi-label movie genre classification with attention mechanism on movie plots. *Computers, Materials & Continua, 83*(3), 5595-5622. https://doi.org/10.32604/cmc.2025.061702

Sy, C. Y., Maceda, L. L., Canon, M. J. P., & Flores, N. M. (2024). Beyond BERT: exploring the efficacy of RoBERTa and ALBERT in supervised multiclass text classification. *International Journal of Advanced Computer Science and Applications, 15*(3), 223-233. https://doi.org/10.14569/IJACSA.2024.0150323

Taha, K., Yoo, P. D., Yeun, C., Homouz, D., & Taha, A. (2024). A comprehensive survey of text classification techniques and their research applications: observational and experimental insights. *Computer Science Review, 54*, 100664. https://doi.org/10.1016/j.cosrev.2024.100664

Tahir, T., Maqbool, K. Q., Zafar, S., Hasan, M. Z., Hussain, M. Z., & Yaqub, M. A. (2025). A global analysis of Netflix content production: unveiling dominant countries and industry trends. *Dialogue Social Science Review, 2*(3), 519-534.

Timoneda, J. C., & Vallejo Vera, S. (2025). BERT, RoBERTa, or DeBERTa? Comparing performance across transformers models in political science text. *The Journal of Politics, 87*(1), 347-364. https://doi.org/10.1086/730737

Tuteja, M., & González Juclà, D. (2023). Long text classification using transformers with paragraph selection strategies. In Proceedings of the Natural Legal Language Processing Workshop 2023 (pp. 17-24). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.nllp-1.3

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ..., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998–6008.

Wang, Z., Pang, Y., Lin, Y., & Zhu, X. (2024). Adaptable and reliable text classification using large language models. In 2024 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 67-74). IEEE. https://doi.org/10.1109/ICDMW65004.2024.00015

Yao, X., Du, W., Sun, L., & Hu, B. (2025). Automatic trailer generation for movies using convolutional neural network. *Scientific Reports, 15*, 7819. https://doi.org/10.1038/s41598-025-91084-y

Yildirim, S. (2024). Fine-tuning transformer-based encoder for Turkish language understanding tasks. arXiv. https://doi.org/10.48550/arXiv.2401.17396

Yin, L., Wang, L., Cai, Z., Lu, S., Wang, R., AlSanad, A., ..., & Zheng, W. (2024). DPAL-BERT: a faster and lighter question answering model. *Computer Modeling in Engineering & Sciences, 141*(1), 771-786. https://doi.org/10.32604/cmes.2024.052622

Zeidi, F., Amasyali, M. F., & Erol, Ç. (2024). LegalTurk: optimized BERT for multi-label text classification and NER. arXiv. https://doi.org/10.48550/arXiv.2407.00648

Zhang, M.-L., & Zhou, Z.-H. (2006). Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering, 18*(10), 1338-1351. https://doi.org/10.1109/TKDE.2006.162

Zhang, T., Patil, S. G., Jain, N., Shen, S., Zaharia, M., ..., & Gonzalez, J. E. (2024). RAFT: adapting language model to domain specific RAG. arXiv. https://doi.org/10.48550/arXiv.2403.10131

Zheng, W., Lu, S., Cai, Z., Wang, R., Wang, L., & Yin, L. (2024). PAL-BERT: an improved question answering model. *Computer Modeling in Engineering & Sciences, 139*(3), 2729-2745. https://doi.org/10.32604/cmes.2023.046692

Zhu, W., Qiu, R., & Fu, Y. (2024). Comparative study on the performance of categorical variable encoders in classification and regression tasks. arXiv. https://doi.org/10.48550/arXiv.2401.09682

Ziyaden, A., Yelenov, A., Hajiyev, F., Rustamov, S., & Pak, A. (2024). Text data augmentation and pre-trained language model for enhancing text classification of low-resource languages. *PeerJ Computer Science, 10*, e1974. https://doi.org/10.7717/peerj-cs.1974