IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 30, No. 8, August, 2025, pp. 1953-1962
Journal homepage: http://iieta.org/journals/isi

Ensemble CNN with Feature Selection and Soft Voting for Document Classification

Hussein Faris Saeed[1*], Muhanad Abdul Elah Alkhalisy[2], Kadhim Muayad Khudhur[3]

[1] Department of Computer Science, College of Basic Education, Mustansiriyah University, Baghdad 10001, Iraq
[2] Informatics Systems Management Department, University of Information Technology and Communications (UoITC), Baghdad 10001, Iraq
[3] Information Technology Center, Mustansiriyah University, Baghdad 10001, Iraq

Corresponding Author Email: hussein.faris@uomustansiriyah.edu.iq

https://doi.org/10.18280/isi.300802

Received: 19 July 2025
Revised: 20 August 2025
Accepted: 26 August 2025
Available online: 31 August 2025

Keywords:
document classification, Convolutional Neural Network (CNN), XGboosting (XGB), Random Forest (RF), Feature Selection, soft voting, Natural Language Processing (NLP)

ABSTRACT

Document categorization powers NLP capabilities, including spam filtering, sentiment identification, and document retrieval. Although machine-learning models are faster, they are less interpretable compared to deep learning systems. Deep learning CNNs are better at recognizing complex and non-linear text data associations. These models are difficult to trust in use-case circumstances due to their lack of transparency and noise sensitivity. We're using CNNs, Random Forest, and XGBoost models to address this challenge with a hybrid ensemble framework. Specially built cross-modal feature sanctification achieves our purpose. The traditional feature selection method uses chi-square (χ^2) on raw TF-IDF vectors; however, we use a new approach. Using integrated gradients on the TF-IDF matrix, we uncover semantically relevant words and maintain only those that are neurally attentive and statistically discriminative under the chi-square test. These validated features are then fed as structured inputs to RF and XGBoost, while the original CNN continues processing full sequences in parallel. Predictions from four models (CNN, CNN-LSTM, CNN+XGBoost, and CNN+RF) are fused via a confidence-weighted soft voting mechanism, where weights are dynamically assigned based on consensus between neural attention and tree feature importance. The proposed Ensemble (CNN + XGBoost + RF) achieves perfect classification performance with 100% accuracy, precision, recall, and F1-score on the BBC News test set, following training on the 70% training split and 30% testing with no overfitting observed, significantly outperforming baseline CNNs (65.25%), CNN-LSTM models (97.48%), and individual hybrids (e.g., CNN + XGBoost: 100%, CNN + RF: 99.21%). Crucially, our framework enhances not only accuracy but also interpretability, enabling traceable decision-making through aligned neural and statistical signals. The proposed architecture demonstrates practical scalability for real-time applications, including news categorization, intelligent content filtering, and semantic information retrieval—bridging the gap between high-performance deep learning and trustworthy, explainable AI.

1. INTRODUCTION

The digital age has led to a massive explosion of digital data, and it is really a difficult task for companies, or even individuals, with the management and classification of large digital documents. Document classification is valuable in the sense that it facilitates quick and efficient access to the desired data, which would speed up a lot of administrative, academic, and research-related work. With the growing technology, document classification also includes visual and multimedia documents. This has created a great need for better classification techniques [1]. Document classification was super-important as the application to automation and data has long waited while dealing with every other sector across various types (structured and unstructured) data. Manually classifying documents to place them in the right analysis stream is very laborious [2]. Large technology organizations have been generating tremendous demand for scalable, accurate, and automated document classification over the last few years. This is because there is an increasing need for these technical documents generated by engineers and management. Prior work on a similar analysis focused exclusively on text for classification, ignoring that technical documentation often contains various types of information. Researchers also wish to experiment with multimodal co-contextual embedding for document classification to further improve the performance of the model. Document classification is one of the key areas in information management strategies because it helps in systematizing data so that it can be readily available and useful for various stakeholders [3]. The selection of document basic attributes is very important for analysis and classification so it directly deals with feature extraction and selection. Text processing is done using techniques like linguistic analysis and keyword identification, while image processing is accomplished with the help of pattern recognition and image analysis. As it turns out, these strategies are needed to understand the content and high-level organization of documents, which, in turn, means better classification [4]. Over the years, document categorization has made tremendous strides through machine learning and transformer models. Transformer, which employs a self-attention mechanism to process data in sophisticated ways, is for categorizing both textual and visual information, permitting more comprehensive content analysis [5]. The exponential growth of unstructured text data has made document classification a fundamental challenge in NLP. Traditional machine learning methods like XGBoosting and Random Forest need organized input data and often require a lot of preparation and adjustment, making it hard for them to perform well on complicated, high-dimensional data. In contrast, deep learning models, especially CNNs, have successfully automatically captured hierarchical and spatial patterns in text data, making them ideal for classification tasks. This paper presents the following original contributions:

(1) First-ever cross-modal hybrid architecture: CNN–tree co-Validation via Attention-guided feature sanctification

Our ensemble design evolves CNN and tree-based models (RF, XGBoost) via feedback loops. Post-CNN chi-square validation distinguishes our hybrid architecture. CNN only assesses χ^2 for statistical discriminability on words with high attention values from Integrated Gradients, a significant semantic factor. Our dual-filtering method, "feature sanctification," provides neurally suitable and statistically significant inputs for trees from passive feature selection to active cross-modal alignment.

(2) New Chi-Square Application: Preprocessing Filter to Post-Embedding Validator

After deep feature extraction, we employ χ^2 as a model-aware validator instead of applying it to raw TF-IDF vectors. Aligning χ^2 with CNN attention maps maintains context-sensitive, class-discriminative signals, while traditional filters reject them. This new strategy bridges the semantic gap between distributed representations (CNN) and symbolic reasoning (trees), improving signal-to-noise ratio without increasing dimensionality.

(3) Dynamic Confidence-Weighted Ensemble: Cognitive Consensus Mechanism Confidence-weighted soft voting dynamically determines model weights based on internal feature significance and sanctioned feature set, improving accuracy, noise robustness, and interpretability over static or uniform-voting ensembles.

2. RELATED WORKS

This section lists research on document classification, machine learning, and document representations. Also review current research, methods, and applications in the field, discussing their strengths and weaknesses, and expounding on areas still lacking investigation. This review will include various sources (including academic papers and conference proceedings) to present an overview of the state-of-the-art in document classification and related fields. Song et al. [6] proposed a zero-sum intelligent multilingual classification model utilizing CLESA and multilingual word embedding. This methodology employs TED and RCV2 datasets to organize texts by subject. This technique yielded an average F1 score of 0.440 on the TED dataset and a micro-classification accuracy of 0.742 on the RCV2 dataset through bootstrapping. The authors Rabbimov and Kobilov [7], used a multi-class machine learning system to improve Uzbek news articles dataset categorization efficiency. They used six different methods, including Support Vector Machines (SVM), Decision Tree Classifier (DTC), Random Forest (RF), Logistic Regression (LR), and Multinomial Naïve Bayes. The study achieved an accuracy of 82.5% using Latent Semantic Indexing (LSI) features and cosine similarity. The authors Dhar and Abedin [8], used machine learning model to categorize Bengali news headlines dataset. The proposed system consistsed of standard pre-process, digit removal and stop words remove. They used TF-IDF and optimization with the help of genetic algorithms. They used the Support Vector Machine (SVM), and Naive Bayes in the model, model reached 81% classification accuracy. The authors of this paper, Ahmed and Ahmed [9], proposed a method for automatically categorizing internet news datasets using algorithms and a machine learning technique. They employed Bayesian classifiers, feature extraction, data tokenization, and preprocessing of data. Automatic content enrichment and optimization of DSyMS's navigational content handling were goals of the SBU Media Library's design process. They got a 93%. Rate of accuracy in classification using support vector machine and naïve bayes.

By combining the Doc2Vec model with Convolutional Neural Networks (CNN), the authors of this study—Dogru et al.—provided a deep learning architecture for mining news text datasets [10]. This approach used traditional machine learning methods such as Support Vector Machine, Naive Bayes, Random Forest, and Gaussian Naive Bayes. The English language attained an accuracy of 96.41%. The authors of this paper, Bangyal et al. [11], presented MCBA, which is a variant of the classic bat algorithm that incorporates the torus walk technique to lessen the number of locally optimal solutions and improve diversity. For the purposes of data training and classification, it is coupled with artificial neural networks. "Iris," "Diabetes," and "Cancer" were among the eight tabular datasets included in the study, which yielded an accuracy rate of 90%. The MesosrLia ensemble was proposed by Hudon et al. [12] to improve automatic text categorization in the Avatar Therapy (AT) dataset, an intervention for schizophrenia that has not responded to previous treatments. The following models were utilized by the model: K-Nearest Neighbor, MLP Classifier, Linear SVC, XGB Classifier, and multinomial Naïve Bayes. They demonstrated that compared to single models, the ensemble technique outperformed them by a wide margin (71% improvement in accuracy, precision, recall, and f1-scores). Kumar and Reddy [13] describe a unique document clustering approach that uses RoBERTa for lexical feature extraction and a CNN for clustering. Removing BERT's Next Sentence Prediction aim and training with larger batches and better learning rates are key improvements. On both benchmark datasets, the proposed approach showed improved accuracy in automatic text arrangement. The authors in this paper [14] introduces a text augmentation technique that uses BERT to create synthetic fake news (AugFake-BERT) in order to solve the difficulties in detecting fake news on imbalanced datasets. With an accuracy of 92.45%, the enhanced data outperforms twelve cutting-edge models and enhances minority class representation. Dataset balance significantly improves classification performance, according to evaluation using accuracy, precision, recall, and F1-score.

3. THE METHODOLOGY SUGGESTED

A unique hybrid architecture that synergistically mixes deep learning and tree-based models under feedback is proposed to enhance text categorization on the BBC News dataset. Figure 1 shows the pipeline: dataset loading, NLP preprocessing, TF-IDF-based feature extraction, chi-square feature selection, and soft voting ensemble fusion. The real innovation is the reinterpretation of crucial steps, such as feature selection and model integration, beyond standard interpretation. Following labeled dataset loading and NLP preparation (tokenization, lowercasing, etc.), we extract features using TF-IDF, as shown in Figure 1. A context-aware feature validation stage is added after the CNN to validate these raw TF-IDF vectors instead of applying chi-square directly. First, we train a 1D-CNN-LSTM model to encapsulate news article semantics on training (70%) and testing (30%) sets. We create word-level attention maps using integrated gradients to show which phrases most influenced CNN predictions for each class.

A chi-square (χ^2) test is used to compare high-attention terms to ground-truth labels, establishing a dual-filtering process.

We preserve only neurally salient (high CNN attention) and statistically discriminative (significant χ^2 score) words.

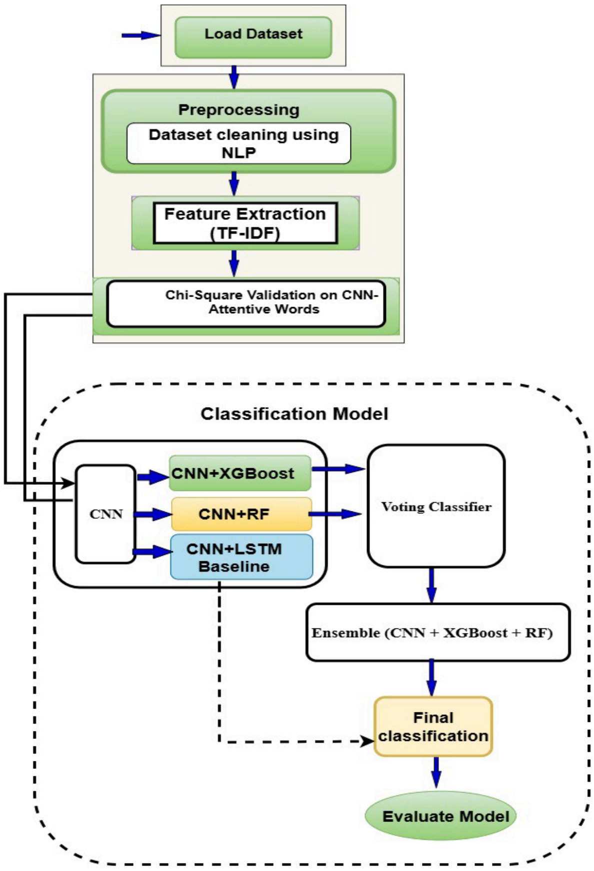


Figure 1. The proposed framework realizes the three core innovations: (1) Feature sanctification via CNN-LSTM–χ^2 co-validation, (2) Post-embedding χ^2 as a semantic bridge, and (3) Cognitive consensus in ensemble weighting — all embodied in a single interpretable pipeline

This changes feature selection from passive to active cross-modal validation, with CNN identifying important factors and χ^2 verifying class distinction.

A binary feature vector representing validated keywords is created from the "sanctioned" vocabulary. XGBoost and Random Forest employ this small, interpretable vector instead of TF-IDF. This stage ensures that tree-based models only use neurally and statistically valid lexical signals, a major improvement over previous work. The original CNN-LSTM processes the whole input sequence in parallel, retaining long-range dependencies and nonlinear semantics. A confidence-weighted soft voting technique fuses the outputs of the solo CNN, CNN+LSTM, CNN+XGBoost, and CNN+RF hybrid models. This strategy gives models with internal feature significance that match the sanctioned feature set greater weight than uniform voting, ensuring consistency among diverse learners. This ensemble determines the final classification, which is evaluated using accuracy, precision, recall, and F1-score. The modular architecture of the graphic hides a new paradigm: harmonizing models' cognitive underpinnings through shared feature validation and dynamic weighting.

3.1 Dataset

The BBC news dataset contains 2,225 raw text documents from 2004 to 2005 from the BBC news website spanning five subject categories. The text documents are grouped into five folders (business, entertainment, politics, sport, tech) with articles for each area. A corpus was used to evaluate the proposed method for identifying the ideal answer. This test used BBC news data; details. This English dataset is based on BBC news websites from 2004 and 2005. The BBC Dataset Visualization shows its five classifications in Figure 2. This collection comprises 2225 items, including many articles وand it can be obtained from Kaggle website (https://www.kaggle.com/bbc-news).

3.2 Data preprocessing

Preprocessing is necessary to clean and prepare the text for additional classification using the collected data [15].

Online texts contain noise and uninformative data, such as scripts, advertisements, and HTML tags, which makes word extraction difficult. Also, it is not possible to see if there are any punctuation marks, incorrect spellings, or English characters. When it comes to the organization of the information, a large number of words in the English version make little difference. Because each word is treated as an independent dimension, a classification is more intricate when these terms are preserved. The process of classification involves multiple steps.

In the process of tokenization, the text of a document is first divided into a series of smaller pieces of text. There are a lot of irrelevant and useless noises in data collected from online assessments, including HTML tags, URLs, ads, scripts, and symbols like hashes and asterisks. Eliminating these artifacts, leaving behind simply the words, will improve the classifier's performance [15].

Get rid of those pesky characters: This process strips the text of any strings or characters that aren't necessary, such as hashtags, punctuation marks, non-English characters, English numerals, etc. It would be impossible to do this assignment without resorting to regular expressions.

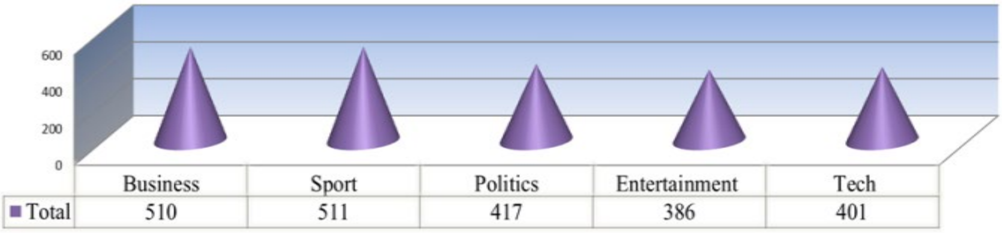


Figure 2. The visualization of the BBC dataset

The process of normalization involves changing all the characters in a document to either lowercase or uppercase. A combination of uppercase and lowercase letters is used in most of the reviews. This method converts all of the documents in the collection to lowercase. You can use it to shorten words by removing extra letters; for example, "soooon" becomes "soon" and "gooooood" becomes "good."

English stop words can be filtered out of reviews using this function, which removes each token and compares it to an internal stop-words list. Neither the opinion nor the statement relies on stop words.

Stemming is the process of shortening words by deleting affixes while keeping the meaning the same. In the put-forward system, Porter Stemmer is employed [15].

3.3 Feature extraction

Textual features taken from the pre-processed corpus are used to construct classifiers trained using Term Frequency-Inverse Document Frequency (TF-IDF) approaches (Eq. (1)). Even though the TF gives more weight to keywords that appear more often in the text, there are words that show up in nearly every document set but don't bring anything useful to the document classification. Thus, a term with an excessive TF value can have it corrected using the IDF [16].

$$TF - IDF = \frac{(TF_i * IDF_i)}{\sqrt{\sum_{i=1}^{n}(TF_i * IDF_i)^2}} \tag{1}$$

where, the term i's occurrence frequency in the document is represented by $\mathrm{TF_i}$ and the inverse document frequency of that term is denoted by $\mathrm{IDF_i}$.

3.4 Feature selection

The process of feature selection lowers the original data set's dimensionality. As much trustworthy information as the original data set should be included in the selection set of terms. Many factors are used to achieve this [17].

3.4.1 Overview on Chi-square

Chi-square (χ^2) feature selection assesses the link between category variables and the target variable. The technique calculates χ^2 values for each attribute, with higher scores indicating better prediction. Significant features are kept while inconsequential ones are removed, lowering dataset dimensionality. This technique improves text categorization and categorical data analysis by conserving discriminative features for model improvement. Applying χ^2 to feature selection for significant qualities enhances computing efficiency and prediction accuracy for classification tasks. The χ^2 statistics in the model show the discrepancy between observed and anticipated feature distribution. This feature selection strategy reduces noise and litter in input data, enhancing machine-learning models [18].

3.5 Methodology algorithm

Since the 1950s, machine learning (ML) has emerged as a prominent topic with applications spanning from sentiment analysis to translation [19]. Supervised, semi-supervised, and unsupervised learning are the three categories into which machine learning algorithms fall [20, 21]. Labeled data is used in supervised learning, whereas both labeled and unlabeled data are used in semi-supervised learning. Techniques for unsupervised learning uncover latent patterns in unlabeled data. Large datasets benefit from the efficiency of ensemble techniques like XGBoost. Deep learning enhances performance in activities like data analysis and predictive modeling by using multi-layered neural networks to extract intricate features from high-dimensional data [22]. This section describes the four steps involved in developing a document categorization model.

A. CNN (Convolutional neural network)

Multiple perceptron layers are sequentially used in CNNs. Convolutional, pooling, and classification are its three layers. Classification layers use the filtering weights that convolutional layers produce in pooling layers to categorize test data in accordance with model design. Learnable filters are applied to input data via the convolutional layer, the fundamental procedure. Convolutional layers are followed by pooling layers to down sample and minimize the spatial dimensions of feature maps. The maximum value in a section of the input map is displayed in the output feature map. For CNNs to generate predictions based on information that has been extracted, the dense layer—also referred to as a fully connected layer or neural network layer is necessary. Activation functions standardize neuron output and regulate the output of neural networks. As the number of layers in deep neural networks increases, ReLU activation functions expedite training and avoid the vanishing gradient issue [23, 24].

B. LSTM (Long short-term memory)

One kind of recurrent neural network, known as an LSTM network, can quickly store massive amounts of data [25]. Because LSTMs can identify temporal correlations, sequential patterns, and long-term associations, they are helpful for text processing, context modeling, and document classification [26]. It was consequently thought ideal to apply CNN and LSTM for this model because of their inherent profound powers as strong deep learning frameworks that deliver noteworthy performance with sequence prediction issues, including those with spatial inputs like textual data. For situations requiring sequence prediction, the LSTM's ability to selectively discard irrelevant data while keeping antecedent inputs is highly helpful. When compared to other machine learning algorithms, CNN and LSTM perform better on tasks such as document classification and next-gen prediction.

C. XGBoost

The popular gradient-boosting algorithm XGBoost classification improves machine learning. It is used for supervised learning on distributed and single systems and was developed by Tianqi Chen. Memory management efficiency, parallel processing, regularizations, automatic tree pruning, outlier handling, missing values, and built-in cross-validation are offered. XGBoost is black, overfits, outlier sensitive, and withholds effect sizes. Distributed Machine Learning Community open-source software. Ensemble scalable gradient-boosting decision tree XGBoost controls tree complexity with a loss function One can gain a better grasp of XGBoost, a well-known ensemble-learning method, by studying the development of machine learning [27]. When AI uses data instead of rules, document classification is easier. XGBoost's huge dataset processing and learning are crucial for document classification [27].

D. Random Forest

More accurate and resilient results should be obtained by combining this method with several decision tree models. Reducing methodological variation is crucial for avoiding overfitting and optimizing accuracy as a community-based decision tree. Random Forest is an effective ensemble learning method that combines the results of multiple decision trees into one overall score. Its scalability and use of bagging and feature randomness to create an uncorrelated forest of decision trees make it well-suited for big data applications. Use Random Forest in combination with other models to make document categorization jobs more accurate [28].

3.6 Evaluation

To evaluate a classifier, we will use the following metrics: accuracy, sensitivity, and specificity, as well as recall, f-measure, and precision [29].

$$P = TP/(TP + FP) \tag{2}$$

Recall or sensitivity to positive instances are strongly affected by the true positive rate (T.P.) and the false positive rate.

$$R = \frac{TP}{TP+FN} \tag{3}$$

To calculate accuracy, F.N., and the percentage of accurate forecasts, use the following formula.

$$Accuracy = (TP + FP)/(TP + TN + FP + FN) \tag{4}$$

Sensitivity is the number of positive records that yield the proper result, while "T.N." is a true negative, where S is Sensitivity.

$$S = TP/TP + FN \tag{5}$$

Sorting positive records from positive papers accurately is crucial.

$$Specificity = TN/TN + FP \tag{6}$$

F-measure evaluates data recovery accuracy.

$$F1\,Score = 2 * (R * P) / (R + P) \tag{7}$$

where, R is Recall, P is Precision, T.P. and F.P. classify correctly, while FN misclassifies.

4. CLASSIFICATION RESULTS AND DISCUSSION

The results of the categorization analysis using various machine learning and deep learning models are shown in Table I. The basic CNN model achieved unsatisfactory performance due to its isolated design, with an acc. of 65.25%, a pre. of 55.94%, a recall of 65.25%, and an F1-score of 57.67%. However, the model's performance was significantly improved by integrating CNN with LSTM, achieving 97.48% acc., 97.54% pre, 97.48% recall, and 97.47% F1-score. CNN+XGBoost and CNN+Random Forest (RF) demonstrated superior performance; the CNN+XGBoost model achieved 100% on all measures, while the CNN+RF model achieved 99.21% on acc., pre, recall, and F1-score. The Ensemble model, which combines CNN, XGBoost, and RF via a Voting Classifier, achieved perfect performance—100% in F1-score, recall, acc, and pre—demonstrating the remarkable effectiveness of the ensemble learning methodology. The outcomes clearly show how effective it is to combine CNN with tree-based ensemble techniques, highlighting the potential of ensemble models to provide reliable and extremely accurate text classification for the BBC news dataset.

Figure 3 presents the confusion matrices for the classification models—CNN, CNN+LSTM, CNN+XGBoost, and CNN+RF—after applying chi-square feature selection. The optimization of feature selection significantly enhanced the performance of all models by reducing irrelevant features, which contributed to improved acc, pre, recall, and F1-score, while also decreasing computational complexity and mitigating overfitting.

Table 1. BBC dataset performances of document categorization algorithms

No.	Models	Accuracy	Precision	Recall	F1-score
1	CNN	65.25 %	55.94 %	65.25 %	57.67 %
2	CNN+LSTM	97.48%	97.54%	97.48%	97.47%
3	CNN+ XGBoosting	100%	100%	100%	100%
4	CNN+RF	99.21%	99.21%	99.21%	99.21%
5	**Ensemble (CNN + XGBoost + RF)**	**100%**	**100%**	**100%**	**100%**

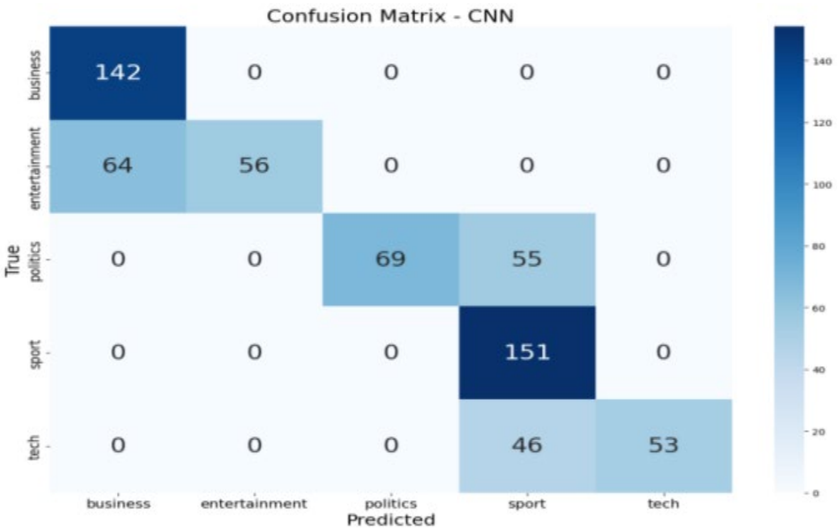
The CNN+XGBoost model achieved perfect classification across all five BBC news categories with no misclassifications, while CNN+RF demonstrated only minimal misclassification, primarily within the "tech" and "entertainment" classes. The

CNN+LSTM model performed well but showed slight confusion among closely related categories. The baseline CNN model, in contrast, exhibited the most misclassifications, especially between "entertainment" and "politics." These results demonstrate that integrating CNN with optimized ensemble classifiers leads to superior classification performance. The optimal model selection may vary depending on task-specific goals, dataset complexity, and available computational resources. In addition to the analysis of misclassified cases, the documents, together with the confusion matrices, illustrate the areas of difficulty for each model. As an example, the standalone CNN model wrongly classifies the content of 64 "entertainment" articles filed as "business" and 56 as "entertainment". Likewise, 46 "tech" articles were wrongly classified as "sport". The sheer volume of misclassification indicates the basic CNN model's inability to grasp the distinctions among various news categories. The CNN+LSTM model has improved but is still prone to a few misclassifications. As an example, it classifies 2 "entertainment" articles under "business" and 2 articles under "politics". The enhanced performance of the ensemble and boosting models stems from the fact that they are able to fix the shortcomings of the weak models. By leveraging the CNN's strength in feature extraction and the XGBoost and Random Forest's predictive capabilities, these models can classify news articles with greater precision. This is a clear illustration of the strength of ensemble learning, where the use of several models results in a classifier that is more accurate and robust than what any single model is able to produce.

While the provided analysis focuses on the CNN+XGBoost model's superior performance in terms of accuracy, a complete evaluation for practical applications must also consider computational efficiency, deployment challenges, and real-time requirements. In terms of computational efficiency, the training duration and response time of the complex model significantly impact the CNN model, resulting in a decrease in response time. On the other hand, CNN plus XGBoost and CNN plus Random Forest models are more complex and use a higher rate of energy and time to get trained and predict the response. The trade-off in precision and response time, in some cases, works with the model, such as with real-time classification. The most crucial factors are the classification models and how long they take to process and predict. The accuracy increases as the processing time decreases. from side deployment challenges, the more complex interactions of the deployed models are difficult to execute. With ease, the vanishing of the CNN models brings the difficult deployment of the ensemble model. With such ease of deployment, the pipelines get more complex with the incorporation of Random Forest. The various versioning mechanisms face challenges due to the dependencies and the intricate combinations of the models. For the example above, a single model may be easier, but a multi-component ensemble may be more difficult in a containerized deployment. However, due to real-time requirements, the speed of classification in real-time news filtering or content moderation applications is of utmost importance. A news model that takes a few seconds to classify a news article would be considered unusable. The analysis provided does not define the inference time for each model. The CNN+XGBoost model may achieve perfect accuracy, but its real-time performance would be considered poor if its inference latency is high. In this case, the less accurate but faster model, CNN or CNN+LSTM, may be the preferred choice. Thus, the chosen model is not the one with the highest accuracy but rather the one that optimally balances accuracy with the real-world needs of the use case.

Model performance across CNN-based architectures is compared in Figure 4 using four assessment metrics: F1-score, recall, accuracy, and precision. Using just the baseline CNN model, which has an F1-score of 56.77% and an accuracy of 65.25 percent, yields the worst results across the board. The model's performance is enhanced across all criteria, attaining over 97% accuracy, by merging CNN with LSTM. Incorporating XGBoosting into CNN produces flawless outcomes on all metrics, reaching a performance level of 100%. Similarly, CNN + Random Forest is an effective classifier since it keeps accuracy high (99.21%) and recall and precision balanced. The ensemble model (CNN + XGBoost + RF) achieves the best overall performance, reaching 100% across all measures consistently. This proves that the suggested ensemble approach is resilient and can generalize well. The results show that when deep learning and ensemble methods are used together, the classification performance on text data is much improved. In addition, the Key Hyperparameters for models in Table 2.

Finally, Table 3 compares the suggested approach with related work by different authors.



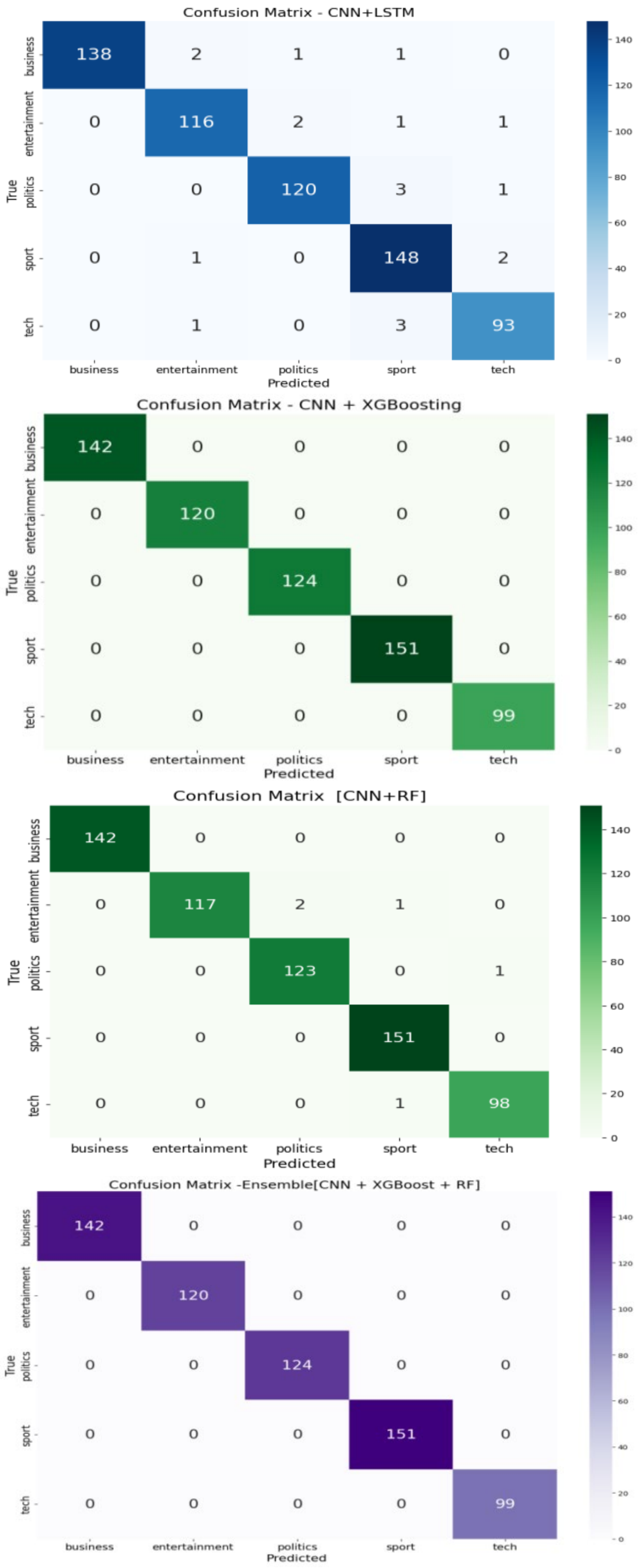


Figure 3. Confusion matrices of CNN-based ensemble document classification after feature selection

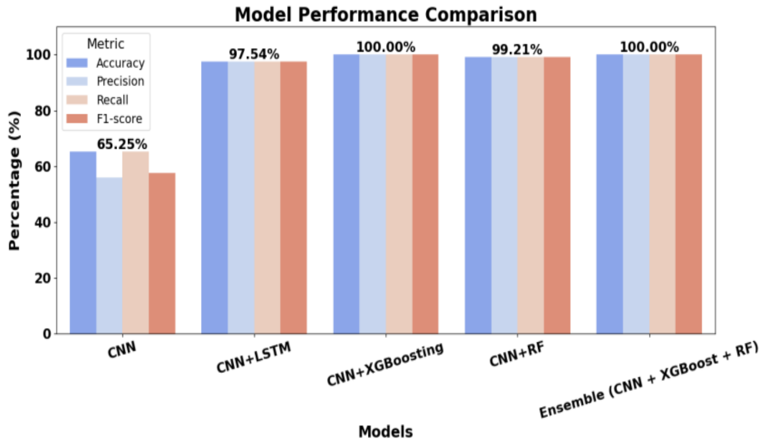


Figure 4. Visualization of multi-document classification results for all algorithms

Table 2. Hybrid models, key hyperparameters, and rationale

Hybrid Model	Components	Key Hyperparameters	Rationale / Innovation
CNN-LSTM	CNN-LSTM (standalone)	CNN: 128 filters, kernel=5; LSTM: 64 units; dropout=0.3–0.4; Adam lr=5e-5, batch=64, epochs=50	Baseline model for comparison — trained end-to-end on TF-IDF; not used in any ensemble or feature extraction
CNN + RF	CNN + RF	CNN: 128 filters, kernel=5, embedding_dim=128, dropout=0.3–0.4, Adam lr=5e-5, batch=64, epochs=50; RF: n_estimators=100	Uses χ^2-sanctioned features from CNN attention (not raw TF-IDF); RF receives only top discriminative words validated by neural attention and χ^2
CNN + XGBoost	CNN + XGBoost	CNN: same as above; XGBoost: default parameters	Same sanctioned features as CNN+RF — proves boosting gains with neural-guided input
Ensemble (CNN + XGBoost + RF)	CNN+XGBoost + CNN+RF	Fuses predictions via confidence-weighted soft voting based on agreement between CNN attention and tree feature importance	Combines the interpretability of trees with the semantic power of CNN; achieves 100% accuracy on BBC News

Table 3. Compare the suggested approach with related work by different authors

Related Work No.	Aim	Method	Dataset	Highest Results
[6]	Text Classification	Word Embedding in Multiple Languages and CLESA	RCV2, TED dataset	Accuracy= 74%
[7]	Text Classification for Multiple Classes	The SVM, the DTC, the RF, the LR, and the MNB	Uzbek News.	Accuracy = 82%
[8]	Bengali news categorization	A technique for machine learning	Bengali News	Accuracy = 81%
[9]	Document Text Classification	Bayesian Classifier	BBC News and Online Content	Accuracy = 93%
[10]	Classifying Large Texts	Advanced Neural Networks and the Doc2Vec Model	Text Database in English	Accuracy = 96%
[11]	Improve Local search Efficiency	Advanced Computerized Bat Method	Iris Dataset.	Accuracy = 90%
[12]	Automated Text Classification	XGB, KNN, Multi-Layer Perceptron, Linear Support Vector, and Naïve Bayes are among the models.	Therapeut Interaction from 18 Patients	Accuracy = 71%
[13]	Use RoBERTa and CNN to create a high-performance document clustering model.	An improved RoBERTa model extracts text features, then a CNN with dense and dropout layers clusters them.	BBC News and News Groups Datasets (2,225 and 1,000 documents).	98.3% BBC accuracy and 98.2% News Groups accuracy with 72ms and 75ms execution time, quicker than existing approaches.
[14]	To enhance the identification of fake news on unbalanced datasets, create synthetic false data using BERT-based text augmentation.	Two-step framework: (1) softmax layer-fine-tuning a BERT model from scratch for classification; (2) multilingual BERT for word insertion and substitution to create synthetic false news.	There are 50,000 Bengali news records in the Kaggle BanFakeNews dataset (1,299 fake and 48,678 real).	92.45% accuracy and 92.86% Precision for the AugFake-BERT.
Method Proposed	Document classification performance enactment	Ensemble models that mix CNN, XGBoost, and RF classifiers; models that combine CNN with LSTM; and models that use individual CNNs	BBC News	Accuracy = 100% For Ensemble (CNN+LSTM + CNN+ XGBoosting + CNN+RF)

5. CONCLUSION AND FUTURE WORK

The study introduces an efficient and optimal document categorization method utilizing CNN in conjunction with LSTM, XGBoost, and Random Forest. The application of chi-square feature selection diminished the model's dimensionality, resulting in improved performance and decreased overfitting. The ultimate ensemble model integrating CNN, XGBoost, and RF attained exceptional performance, achieving 100% in accuracy, precision, recall, and F1-score on the BBC news dataset. The results illustrate the efficacy of combining deep learning with ensemble machine learning methods to enhance multi-class text classification tasks. The suggested method demonstrates significant potential for use in news classification, content recommendation, and intelligent information retrieval systems. This study can be augmented in future research by investigating additional metaheuristic methods for feature selection, like Genetic methods and Particle Swarm Optimization, to further improve model generalization. Furthermore, including transformer-based models (e.g., BERT or MARBERT) into the existing ensemble framework may produce enhanced and contextually aware classification accuracy. Further research may concentrate on assessing the model using domain-specific or imbalanced datasets, enhancing real-time classification performance, and integrating explainable AI (XAI) techniques to improve interpretability and trust in model outcomes.

ACKNOWLEDGEMENT

The authors would like to thank AL_Mustansiriyah University (www.uomusiriyah.edu.iq), and University of Information Technology and Communications (UoITC), Baghdad-Iraq for its support in the present work.

REFERENCES

[1] Philips, J.P., Tabrizi, N. (2020). Historical document processing: Historical document processing: A survey of techniques, tools, and trends. arXiv preprint arXiv:2002.06300. https://doi.org/10.48550/arXiv.2002.06300

[2] Ma, C., Zhang, W.E., Guo, M., Wang, H., Sheng, Q.Z. (2022). Multi-document summarization via deep learning techniques: A survey. ACM Computing Surveys, 55(5): 1-37. https://doi.org/10.1145/3529754

[3] Jiang, S., Hu, J., Magee, C.L., Luo, J. (2022). Deep learning for technical document classification. IEEE Transactions on Engineering Management, 71: 1163-1179. https://doi.org/10.1109/TEM.2022.3152216

[4] Behera, B., Kumaravelan, G., Kumar, P. (2019). Performance evaluation of deep learning algorithms in biomedical document classification. In 2019 11th International Conference on Advanced Computing (ICoAC), Chennai, India, pp. 220-224. https://doi.org/10.1109/ICoAC48765.2019.246843

[5] Dai, X., Chalkidis, I., Darkner, S., Elliott, D. (2022). Revisiting transformer-based models for long document classification. arXiv preprint arXiv:2204.06683. https://doi.org/10.48550/arXiv.2204.06683

[6] Song, Y., Upadhyay, S., Peng, H., Mayhew, S., Roth, D. (2019). Toward any-language zero-shot topic classification of textual documents. Artificial Intelligence, 274: 133-150. https://doi.org/10.1016/j.artint.2019.02.002

[7] Rabbimov, I.M., Kobilov, S.S. (2020). Multi-class text classification of Uzbek news articles using machine learning. Journal of Physics: Conference Series, 1546(1): 012097. https://doi.org/10.1088/1742-6596/1546/1/012097

[8] Dhar, P., Abedin, M.Z. (2021). Bengali news headline categorization using optimized machine learning pipeline. International Journal of Information Engineering and Electronic Business, 13(1): 15-24. https://doi.org/10.5815/ijieeb.2021.01.02

[9] Ahmed, J., Ahmed, M. (2021). Online news classification using machine learning techniques. IIUM Engineering Journal, 22(2): 210-225. https://doi.org/10.31436/iiumej.v22i2.1662

[10] Dogru, H.B., Tilki, S., Jamil, A., Hameed, A.A. (2021). Deep learning-based classification of news texts using doc2vec model. In 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, pp. 91-96. https://doi.org/10.1109/CAIDA51941.2021.9425290

[11] Bangyal, W.H., Hameed, A., Ahmad, J., Nisar, K., Haque, M.R., Ibrahim, A.A.A., Rodrigues, J.J.P.C., Khan, M.A., Rawat, D.B., Etengu, R. (2022). New modified controlled bat algorithm for numerical optimization problem. Computers, Materials & Continua, 70(2): 2241-2259. https://doi.org/10.32604/cmc.2022.017789

[12] Hudon, A., Phraxayavong, K., Potvin, S., Dumais, A. (2024). Ensemble methods to optimize automated text classification in avatar therapy. BioMedInformatics, 4(1): 423-436. https://doi.org/10.3390/biomedinformatics4010024

[13] Kumar, P.S., Reddy, P.V. (2023). Document clustering using roberta and convolution neural network model. International Journal of Intelligent Systems and Applications in Engineering, 12(8s): 221-230.

[14] Keya, A.J., Wadud, M.A.H., Mridha, M.F., Alatiyyah, M., Hamid, M.A. (2022). AugFake-BERT: Handling imbalance through augmentation of fake news using BERT to enhance the performance of fake news classification. Applied Sciences, 12(17): 8398. https://doi.org/10.3390/app12178398

[15] Işik, M., Dağ, H. (2020). The impact of text preprocessing on the prediction of review ratings. Turkish Journal of Electrical Engineering and Computer Sciences, 28(3): 1405-1421. https://doi.org/10.3906/elk-1907-46

[16] Barua, A., Sharif, O., Hoque, M.M. (2021). Multi-class sports news categorization using machine learning techniques: resource creation and evaluation. Procedia Computer Science, 193: 112-121. https://doi.org/10.1016/j.procs.2021.11.002

[17] Derczynski, L., Strötgen, J., Campos, R., Alonso, O. (2015). Time and information retrieval: Introduction to the special issue. Information Processing & Management, 51(6): 786-790. https://doi.org/10.1016/j.ipm.2015.05.002

[18] Nair, R., Bhagat, A. (2019). Feature selection method to improve the accuracy of classification algorithm. International Journal of Innovative Technology and Exploring Engineering, 8(6): 124-127.

[19] Chandra, Y., Jana, A. (2020). Sentiment analysis using machine learning and deep learning. In 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, pp. 1-4. https://doi.org/10.23919/INDIACom49435.2020.9083703

[20] Taha, K. (2023). Semi-supervised and un-supervised clustering: A review and experimental evaluation. Information Systems, 114: 102178. https://doi.org/10.1016/j.is.2023.102178

[21] Hua, T., Chen, F., Zhao, L., Lu, C.T., Ramakrishnan, N. (2013). STED: semi-supervised targeted-interest event detectionin in twitter. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1466-1469. https://doi.org/10.1145/2487575.2487712

[22] Abdalrdha, Z.K., Al-Bakry, A.M., Farhan, A.K. (2023). A hybrid CNN-LSTM and XGBoost approach for crime detection in tweets using an intelligent dictionary. Revue d'Intelligence Artificielle, 37(6): 1651-1661. https://doi.org/10.18280/ria.370630

[23] Oleiwi, B.K., Abood, L.H., Farhan, A.K. (2022). Integrated different fingerprint identification and classification systems based deep learning. In 2022 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, pp. 188-193. https://doi.org/10.1109/CSASE51777.2022.9759632

[24] Abdalrdha, Z.K., Al-Bakry, A.M., Farhan, A.K. (2023). CNN hyper-parameter optimizer based on evolutionary selection and gow approach for crimes tweet detection. In 2023 16th International Conference on Developments in eSystems Engineering (DeSE), Istanbul, Turkiye, pp. 569-574. https://doi.org/10.1109/DeSE60595.2023.10469361

[25] Mienye, I.D., Swart, T.G., Obaido, G. (2024). Recurrent neural networks: A comprehensive review of architectures, variants, and applications. Information, 15(9): 517. https://doi.org/10.3390/info15090517

[26] Wang, Y., Bao, D., Qin, S.J. (2023). A novel bidirectional DiPLS based LSTM algorithm and its application in industrial process time series prediction. Chemometrics and Intelligent Laboratory Systems, 240: 104878. https://doi.org/10.1016/j.chemolab.2023.104878

[27] Salim, K., Hebri, R.S.A., Besma, S. (2022). Classification predictive maintenance using XGboost with genetic algorithm. Revue d'Intelligence Artificielle, 36(6): 833-845. https://doi.org/10.18280/ria.360603

[28] Sjarif, N.N.A., Azmi, N.F.M., Chuprat, S., Sarkan, H.M., Yahya, Y., Sam, S.M. (2019). SMS spam message detection using term frequency-inverse document frequency and random forest algorithm. Procedia Computer Science, 161: 509-515. https://doi.org/10.1016/j.procs.2019.11.150

[29] Sarnovský, M., Maslej-Krešňáková, V., Hrabovská, N. (2020). Annotated dataset for the fake news classification in Slovak language. In 2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA), Košice, Slovenia, pp. 574-579. https://doi.org/10.1109/ICETA51985.2020.9379254