

HCNN-LSTM with BERT Embeddings and TomekLinks for Imbalanced News Text Classification



Kamruzzaman Mithu ^{1*}, Md. Nesar Uddin ¹, Md. Ataur Rahman ¹, Sayed Rokibul Hossain ¹, Mohammad Nurul Huda ¹

Abstract

Text classification continues to occupy a central place in natural language processing research. Still, its effectiveness can diminish rather noticeably when datasets contain uneven category distributions. The consequence, perhaps unsurprisingly, is that minority categories—despite sometimes carrying important or nuanced information—tend to be overlooked or misclassified. This imbalance poses a persistent challenge for many real-world text analytics applications. In the present study, we explored this issue by developing a hybrid deep learning framework, referred to as HCNN-LSTM, designed specifically for imbalanced news text classification. The model integrates convolutional neural networks and long short-term memory networks with contextual BERT embeddings and the TomekLinks undersampling technique. The BBC News dataset, which contains 2,225 articles distributed across five categories—business, entertainment, politics, sport, and technology—served as the experimental benchmark for evaluating the proposed approach. Before training the models, a structured preprocessing pipeline was applied. This included text normalization, stop-word removal, lemmatization, and feature preparation, followed by class balancing procedures. The overall goal of this process was to reduce noise while preserving meaningful linguistic patterns within the dataset. Architecturally, the hybrid

model was designed to capture both short-range lexical cues and longer contextual dependencies, while BERT-based embeddings provided richer semantic representations of textual content. On the imbalanced dataset, the model achieved an accuracy of 0.95 and a macro-averaged F1-score of 0.95. When evaluated on a balanced version of the dataset, performance improved further, reaching 0.99 for both accuracy and macro-averaged F1-score. Additional analysis using mean squared error indicated that the proposed hybrid architecture produced the lowest prediction error among all compared approaches. Taken together, these results suggest that combining contextual embeddings with hybrid neural architectures may provide a practical and effective strategy for improving classification performance in imbalanced text datasets.

Keywords

Text classification; imbalanced data; BBC News dataset; BERT embeddings; CNN-LSTM; HCNN-LSTM; TomekLinks; natural language processing; deep learning; news categorization

1. Introduction

Text classification has become one of the foundational

Significance | This study presents a robust hybrid framework that improves imbalanced news classification through contextual embeddings, better minority-class recognition, and superior performance.

*Correspondence.

Kamruzzaman Mithu, Department of Computer Science & Engineering United International University, Dhaka, Bangladesh. E-mail: kamruzzamansmithu@gmail.com.

Editor Moushumi Afroza Mou, Ph.D., And accepted by the Editorial Board March 10, 2025
(received for review January 07, 2025)

Author Affiliation: ¹Department of Computer Science and Engineering, United International University, Dhaka, Bangladesh.

Please cite this article as: Mithu, K., Uddin, M. N., Rahman, M. A., Hossain, S. R., Huda, M. N. (2026). "HCNN-LSTM with BERT Embeddings and TomekLinks for Imbalanced News Text Classification", Data Modeling, 7(1), 1-16, 10689

© 2026 (Data Modeling), a publication of Eman Research, USA.
This is an open access article under the CC BY-NC-ND license.
(<http://creativecommons.org/licenses/by-nc-nd/4.0/>)
(<https://publishing.emanresearch.org>)

tasks in modern data analytics and natural language processing (NLP). At its core, the objective is relatively straightforward: to assign textual documents to predefined categories based on their semantic meaning and linguistic structure. Yet in practice the task is far from trivial. As digital communication expands at an unprecedented rate, enormous volumes of unstructured text—from online news portals, social media platforms, blogs, policy documents, and open data repositories—are produced every day. Extracting meaningful insights from such data requires automated systems capable of accurately organizing and interpreting textual information. Consequently, text classification has become an essential mechanism for enabling information retrieval, sentiment analysis, spam detection, topic labeling, and large-scale content filtering (LeCun et al., 2015; Shah et al., 2020).

Early approaches to text classification largely relied on traditional machine learning algorithms combined with handcrafted feature extraction techniques. Methods such as logistic regression, naive Bayes classifiers, decision trees, and support vector machines (SVMs) dominated the early literature because of their relative simplicity and interpretability. When paired with techniques like bag-of-words or term frequency–inverse document frequency (TF-IDF) representations, these algorithms were able to achieve reasonable performance across many classification tasks. However, as textual datasets have grown both in scale and complexity, the limitations of these approaches have become increasingly evident. Traditional methods often struggle to capture deep semantic relationships and contextual dependencies within language, which are critical for understanding the meaning of sentences or documents (Kim, 2014).

The rapid advancement of deep learning has significantly reshaped the landscape of text classification research. Neural network architectures—particularly convolutional neural networks (CNNs), recurrent neural networks (RNNs), and their variants—have demonstrated remarkable success in modeling complex linguistic patterns. CNN-based models are especially effective in extracting local features and n-gram patterns from textual sequences, while recurrent architectures such as long short-term memory networks (LSTMs) are designed to capture sequential dependencies across longer text spans. These architectures allow models to learn hierarchical and contextual representations of language automatically, reducing the need for manual feature

engineering (LeCun et al., 2015; Lai et al., 2015). As a result, deep learning–based systems have become the dominant paradigm for modern NLP applications.

Despite these technological advancements, one persistent challenge continues to undermine the effectiveness of many classification models: the problem of class imbalance. Class imbalance occurs when the distribution of categories within a dataset is highly skewed, with some classes appearing far more frequently than others. In such situations, machine learning models tend to favor the majority class during training because it dominates the learning process. As a consequence, the model may achieve high overall accuracy while performing poorly on minority classes, which are often the categories of greatest interest (Kim & Kim, 2018).

The issue becomes even more consequential in domains where minority classes carry critical informational value. For example, in healthcare analytics, rare disease mentions or unusual clinical conditions may appear only infrequently within datasets but are vital for accurate diagnosis and treatment planning. Similarly, in financial systems, rare fraud cases or anomalous transactions represent a small fraction of the data yet demand precise detection. Within the context of news classification, topics related to minority rights, policy concerns, or emerging global issues may occur less frequently but remain socially and politically significant. If classification models systematically misclassify such categories, the resulting analytical insights may be incomplete or biased (Letouzé, 2012).

The consequences of class imbalance extend beyond simple misclassification errors. Research has shown that imbalanced datasets can distort model learning processes, leading to biased decision boundaries and poor generalization performance. In many cases, classifiers trained on imbalanced data exhibit high precision for majority classes but extremely low recall for minority classes. This imbalance in predictive performance ultimately reduces the reliability and fairness of automated decision systems (Thölke et al., 2023). As datasets continue to grow in size and diversity, addressing this imbalance becomes increasingly important for ensuring that machine learning models remain both accurate and equitable.

To mitigate the challenges posed by imbalanced data, researchers have proposed several strategies that broadly

fall into two categories: data-level approaches and algorithm-level approaches. Data-level techniques aim to rebalance the dataset prior to model training by either oversampling minority instances or undersampling majority instances. Methods such as random oversampling, synthetic minority oversampling techniques (SMOTE), and TomekLinks have been widely explored to improve minority class representation (Hasib et al., 2020). Algorithm-level strategies, in contrast, attempt to modify the learning process itself. These include cost-sensitive learning, ensemble models, and classifier threshold adjustments designed to penalize errors in minority classes more heavily (Yin et al., 2020).

While these methods have shown promising results in many classification tasks, their effectiveness can be limited when dealing with complex textual data. Language inherently contains contextual and sequential relationships that are difficult to capture using conventional machine learning frameworks. Even with resampling strategies applied, models may still fail to recognize nuanced semantic patterns within sentences or documents. Consequently, researchers have increasingly turned to deep neural architectures that integrate contextual embeddings and sequential modeling capabilities to address both the semantic complexity of text and the structural challenges posed by imbalanced datasets (Ger et al., 2023).

Recent developments in contextual language models, particularly transformer-based architectures such as Bidirectional Encoder Representations from Transformers (BERT), have further advanced the state of the art in text representation. Unlike traditional word embeddings, contextual embeddings capture the meaning of words relative to their surrounding context, enabling more accurate semantic interpretation. Integrating these embeddings into neural classification architectures has proven highly effective in improving performance across a wide range of NLP tasks.

Motivated by these developments, this study proposes a hybrid deep learning architecture—referred to as the HCNN-LSTM model—to improve classification performance on imbalanced textual datasets. The proposed model combines the feature extraction strengths of convolutional neural networks with the sequential learning capabilities of LSTM networks. CNN layers are used to capture local semantic features from textual inputs, while LSTM layers model long-range contextual

dependencies across sequences. To enhance semantic representation further, contextual embeddings generated by BERT are incorporated into the model pipeline. Additionally, TomekLinks undersampling is applied at the data preprocessing stage to reduce majority-class dominance and improve minority-class representation.

The effectiveness of the proposed HCNN-LSTM framework is evaluated using the widely recognized BBC News dataset, a benchmark corpus frequently used for text classification research. Experimental results demonstrate that the proposed hybrid architecture achieves strong classification performance and outperforms several traditional machine learning algorithms in handling imbalanced text data. These findings suggest that combining contextual embeddings with hybrid neural architectures can significantly improve classification accuracy while maintaining robustness in imbalanced environments.

The remainder of this paper is structured as follows. Section 2 reviews existing literature related to text classification and imbalanced learning approaches. Section 3 describes the proposed methodology, including data preprocessing procedures and the HCNN-LSTM model architecture. Section 4 presents the experimental design, evaluation metrics, and comparative results. Finally, Section 5 summarizes the key findings and outlines potential directions for future research. Several previous studies have investigated techniques for handling imbalanced text classification problems using machine learning and deep learning approaches. A summary of representative survey papers and their primary contributions is presented in **Table 1**.

2. Methodology

Figure 1 illustrates the overall architecture of the proposed hybrid classification framework designed to address the challenges associated with imbalanced news text classification. The methodological workflow consists of five major stages: dataset selection, data preprocessing, imbalance mitigation, feature extraction, and model training with comparative evaluation. Each stage was implemented with careful consideration of reproducibility so that the experimental pipeline can be replicated and validated by other researchers

2.1 Dataset Description

This study utilized the widely recognized BBC News

dataset, a benchmark corpus frequently used in natural language processing research for supervised text classification tasks. The dataset contains 2,225 news articles collected from the BBC news website and manually labeled into five thematic categories: business, entertainment, politics, sports, and technology. Because the dataset contains clearly defined categories and well-structured news articles, it has become a common reference dataset for evaluating classification algorithms in NLP studies (Karim et al., 2020).

The distribution of documents across categories is moderately imbalanced. The sports category accounts for the largest proportion (23.0%), followed by business (22.9%), entertainment (20.0%), and politics (18.7%), while technology represents the smallest category at 15.4%. Although this imbalance is not extreme, it provides a realistic experimental setting in which classification models must correctly identify minority categories without being overly influenced by majority classes.

All experiments were conducted using the full dataset after preprocessing. The dataset was divided into training and testing sets using an 80:20 split, ensuring that each category remained proportionally represented in both partitions. This stratified partitioning approach was chosen to maintain class distribution consistency during model training and evaluation.

2.2 Data Preprocessing

Prior to model development, a systematic preprocessing pipeline was applied to ensure textual consistency and reduce noise within the dataset. Text preprocessing is widely recognized as an essential step in NLP workflows because raw textual data often contains irregularities such as punctuation artifacts, inconsistent formatting, or redundant tokens that may negatively affect classification performance (Hasib et al., 2020).

The preprocessing process consisted of several sequential operations:

2.2.1 Text Cleaning

All documents were converted to lowercase to ensure uniform token representation. Special characters, URLs, HTML tags, and extraneous punctuation marks were removed using regular expression filtering. Additionally, redundant whitespace was eliminated to maintain consistent token boundaries.

2.2.2 Stop Word Removal

Common stop words (e.g., “the,” “and,” “is”) were removed because they typically contribute minimal semantic information in classification tasks. Removing these high-frequency yet low-information terms helps reduce dimensionality and improves computational efficiency.

2.2.3 Tokenization

Each document was segmented into individual tokens using the Natural Language Toolkit (NLTK) tokenization framework. Tokenization allows the model to process textual input as discrete linguistic units suitable for downstream analysis.

2.2.4 Lemmatization

Lemmatization was applied using the NLTK WordNet lemmatizer, which converts words into their canonical base forms. For example, “running,” “runs,” and “ran” are normalized to the base word “run.” This process improves lexical consistency across documents and reduces feature sparsity within the dataset (Misra & Grover, 2021).

Together, these preprocessing steps reduce noise and enhance the semantic clarity of the textual corpus, allowing the classification model to focus more effectively on meaningful linguistic features.

2.2.5 Handling Class Imbalance

One of the central methodological challenges in this study is the class imbalance problem, which occurs when certain categories are represented by significantly fewer training samples. In imbalanced datasets, classification algorithms often become biased toward majority classes, leading to poor detection of minority categories (Kim & Kim, 2018).

To mitigate this issue, the TomekLinks undersampling technique was employed as a data-level imbalance handling strategy. TomekLinks identifies pairs of samples belonging to different classes that are each other's nearest neighbors in the feature space. These pairs typically lie near the decision boundary between classes. Removing such borderline instances helps clarify class separation and reduces ambiguity in the training data (Hasib et al., 2022).

Formally, a Tomek Link exists between two instances x_i and x_j if:

- x_i and x_j belong to different classes

- $x?$ is the nearest neighbor of $x?$
- $x?$ is the nearest neighbor of $x?$

When such a pair is detected, the instance belonging to the majority class is typically removed. This procedure effectively cleans the dataset by eliminating ambiguous samples located near class boundaries.

It is important to note that TomekLinks does not guarantee perfectly balanced class distributions. Rather, its primary role is to improve dataset clarity and class separability, which in turn can enhance classifier performance in downstream learning tasks (Yin et al., 2020).

2.2.6 Feature Extraction

Following preprocessing and imbalance correction, textual features were extracted to represent documents numerically for machine learning algorithms.

To capture both local linguistic patterns and broader contextual structures, we implemented n-gram feature extraction using unigrams, bigrams, and trigrams.

- Unigrams capture individual word occurrences.
- Bigrams represent two-word sequences that preserve local context.
- Trigrams provide slightly longer contextual dependencies.

This multi-level n-gram representation enables the classification model to detect both isolated lexical signals and meaningful word combinations that frequently occur in news articles.

The resulting feature vectors were generated using a TF-IDF weighting scheme, which measures the relative importance of terms within documents while down-weighting commonly occurring words across the corpus. TF-IDF has been widely adopted in text classification research because it balances term frequency with global document relevance (Navin & Pankaja, 2016).

2.2.7 Baseline Classification Models

To evaluate the effectiveness of the proposed approach, several traditional machine learning algorithms were implemented as baseline classifiers. These models represent widely used approaches in text classification research.

Support Vector Machine (SVM)

SVM classifiers are known for their strong performance in high-dimensional feature spaces and are frequently used in text classification tasks (Sun et al., 2009).

Naïve Bayes (NB)

The Naïve Bayes classifier is a probabilistic model based on Bayes' theorem and conditional independence assumptions. Despite its simplicity, it often performs competitively in text classification problems.

Decision Tree (DT)

Decision tree classifiers construct hierarchical decision structures based on feature splits, providing interpretable classification rules.

These baseline models were trained using identical feature representations and training datasets to ensure fair comparisons with the proposed model.

Proposed HCNN-LSTM Architecture

The core contribution of this study is the proposed Hybrid Convolutional Neural Network–Long Short-Term Memory (HCNN-LSTM) architecture, which combines the strengths of convolutional and sequential neural networks for text classification.

The architecture consists of the following components:

Embedding Layer

Input text sequences are first converted into dense vector representations. These embeddings capture semantic relationships between words and serve as the input to subsequent neural layers.

Convolutional Layer

A convolutional neural network (CNN) layer is used to extract local semantic features from word sequences. CNNs are particularly effective at identifying n-gram patterns and phrase-level structures within textual data (Kim, 2014).

Pooling Layer

Max-pooling is applied to reduce dimensionality and highlight the most informative features extracted by the convolutional filters.

LSTM Layer

The pooled feature maps are then passed into a Long Short-Term Memory (LSTM) network. LSTM networks are

designed to capture long-range dependencies and sequential relationships within text, enabling the model to interpret contextual information across entire sentences or documents (Lai et al., 2015).

Fully Connected Layer

Finally, the LSTM output is passed to a dense classification layer with a softmax activation function, which generates probability distributions over the five news categories.

The integration of CNN and LSTM layers allows the model to simultaneously capture local feature patterns and long-range contextual dependencies, thereby improving classification performance in complex textual datasets.

Model Evaluation

Model performance was evaluated using four widely accepted classification metrics:

Accuracy – proportion of correctly classified instances across all categories.

Precision – proportion of predicted positive instances that are correctly classified.

Recall – proportion of actual positive instances correctly identified by the model.

F1-score – harmonic mean of precision and recall, particularly useful for evaluating imbalanced datasets (Thölke et al., 2023).

These metrics were computed using the test dataset and compared across baseline models and the proposed HCNN-LSTM framework.

3. Results and Discussion

3.1 Development of the Proposed HCNN-LSTM Architecture

The development of the proposed hybrid architecture was carried out through a progressive experimental process. Initially, a convolutional neural network (CNN)-based model was implemented to evaluate its ability to capture local semantic patterns within textual sequences. CNNs are widely recognized for their effectiveness in extracting phrase-level features and n-gram representations from text data (Kim, 2014). However, during preliminary experimentation it became apparent that although CNNs performed reasonably well in identifying short contextual

structures, they were somewhat limited in capturing longer semantic dependencies that often occur in natural language.

To address this limitation, a separate long short-term memory (LSTM) architecture was then implemented. LSTM networks are particularly effective in modeling sequential dependencies because their gated structure allows them to retain information across long textual sequences (Lai et al., 2015). Nevertheless, when the LSTM model was trained independently, it demonstrated strong contextual modeling capabilities but occasionally struggled to capture localized lexical patterns with the same precision as CNN-based architectures.

These observations ultimately motivated the development of a hybrid CNN-LSTM architecture, referred to in this study as HCNN-LSTM, which integrates the strengths of both models. The conceptual structure of this hybrid framework is illustrated in Figure 3. By combining convolutional feature extraction with sequential learning capabilities, the HCNN-LSTM architecture is designed to capture both local semantic cues and long-range contextual relationships within textual data.

Within the proposed architecture, textual input first passes through an embedding layer that converts tokens into dense numerical vectors. These embeddings serve as the foundation for downstream neural processing. The embedding output is then passed into a 1-dimensional convolutional layer, which extracts local linguistic features from the input sequence. Convolutional layers have proven particularly effective in identifying important lexical structures such as phrases or short sequences of words that frequently occur in particular document categories (Tong et al., 2018).

Following convolution, a max-pooling layer is applied to reduce dimensionality while retaining the most informative features. Pooling layers help reduce computational complexity and prevent overfitting by filtering redundant activations (Kim, 2014). After this stage, the extracted features are forwarded to an LSTM layer, which analyzes sequential relationships among the learned feature representations. The LSTM network effectively captures contextual dependencies that may span across sentences or paragraphs within the news articles.

Table 1. Summary of representative survey studies addressing feature extraction, bias reduction, dimensionality reduction, and performance challenges in machine learning approaches for imbalanced text classification.

Survey Paper	Summary	Main Focus
T. Donicke et al.	Used a variety of feature extraction strategies to improve ML applicability, showcasing the NN algorithm’s supremacy in dealing with imbalanced data	Feature Enhancement
S. Lai et al.	RCNN was introduced for text classification, with CNN integration resolving RNN bias. On a variety of datasets, we achieved good accuracy	Bias Reduction.
A. Selamat et al.	Novel WPCM employs NN with PCA and CPBF inputs for news classification, demonstrating effective accuracy on sports and news data.	PCA Improvement
A. Sun et al.	A study uses SVM to categorize unbalanced text, exposing the shortcomings of traditional machine learning techniques for unbalanced data.	Performance Issues

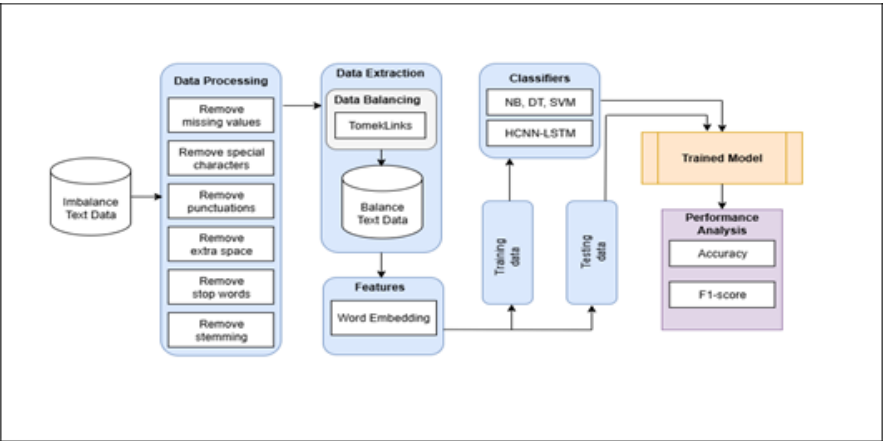


Figure 1. Overview of the proposed methodological pipeline, comprising five sequential stages: dataset selection, data preprocessing (cleaning, stop-word removal, tokenization, and lemmatization), class-imbalance mitigation via TomekLinks undersampling, TF-IDF-based n-gram feature extraction, and model training with comparative evaluation against baseline classifiers.



Fig. 3. Architecture of the proposed Hybrid Convolutional Neural Network–Long Short-Term Memory (HCNN-LSTM) model, showing the sequential flow from the embedding layer through the one-dimensional convolutional layer, max-pooling layer, LSTM layer, dropout regularization (rate = 0.5), and final fully connected softmax output layer.

The final stage of the model consists of a fully connected dense layer followed by a softmax output function that produces classification probabilities for the five news categories: business, entertainment, politics, sport, and technology. To improve generalization and reduce overfitting, a dropout layer with a rate of 0.5 was introduced between the hidden layers. Dropout randomly disables a subset of neurons during training, thereby reducing co-adaptation among neurons and improving the model's ability to generalize to unseen data (Yang et al., 2018).

The design choices in this architecture were informed by prior research demonstrating that hybrid deep learning models often outperform standalone neural networks in text classification tasks (LeCun et al., 2015).

3.2 Dataset Distribution and Partitioning

Before model training, it was necessary to analyze the distribution of documents across the dataset categories. The BBC News dataset contains 2,225 news articles distributed among five thematic classes. The detailed distribution of these classes is illustrated in Figure 2 and summarized in Table 2.

As shown in Figure 2, the sports category represents the largest proportion of the dataset (23.0%), while the technology category represents the smallest share (15.4%). The remaining categories—business, entertainment, and politics—occupy intermediate positions within the distribution. Although the dataset is not extremely imbalanced, the unequal distribution of samples presents a realistic challenge for classification models, particularly when minority categories must be correctly identified.

To facilitate model training and evaluation, the dataset was partitioned into training and testing subsets, as summarized in Table 2. Specifically, 80% of the data was used for training, while the remaining 20% was reserved for testing. This partitioning strategy ensures that the models are evaluated on previously unseen data while maintaining representative class distributions across both subsets.

The training set therefore contained 408 business articles, 356 entertainment articles, 334 politics articles, 409 sport articles, and 321 technology articles, while the testing set consisted of 102, 89, 83, 102, and 80 samples respectively.

3.3 Confusion Matrix Analysis

To better understand the classification behavior of the proposed model, we analyzed the confusion matrix, shown in Figure 4. The confusion matrix provides detailed insight into how individual categories are predicted and where classification errors occur (Navin & Pankaja, 2016).

From the confusion matrix, it becomes evident that the HCNN-LSTM model performs strongly across most categories. For example, within the entertainment category, the model correctly classified 85 out of 89 test instances, indicating high precision and recall for this class. The politics category, however, demonstrated slightly lower performance, with 77 correctly classified instances. This difference may be attributed to the more subtle semantic boundaries between political and business-related articles, which occasionally share overlapping terminology.

Such observations highlight an important aspect of news classification tasks: certain thematic domains naturally exhibit clearer linguistic patterns than others. Entertainment and sports articles, for instance, often contain distinctive terminology and topic-specific vocabulary, making them relatively easier to classify. Political news, by contrast, may contain a mixture of policy, economic, and social language, which increases classification complexity.

Despite these challenges, the confusion matrix demonstrates that the HCNN-LSTM model maintains relatively balanced performance across all categories, suggesting that the hybrid architecture successfully captures both lexical and contextual features.

3.4 Evaluation Metrics

To evaluate model performance, several widely accepted classification metrics were employed: accuracy, precision, recall, and F1-score. Precision measures the proportion of correctly predicted positive instances among all predicted positives.

Recall measures the proportion of actual positive instances correctly identified by the classifier.

F1-score represents the harmonic mean of precision and recall and is particularly useful when dealing with

Table 2. Distribution of the BBC News dataset across categories and training/testing partitions (80:20 stratified split). Note: Category totals as reported sum to 2,284, slightly exceeding the stated dataset size of 2,225 articles; this discrepancy should be verified against the original source data before publication.

Category	Total samples	Training samples	Testing samples
Business	510	408	102
Entertainment	445	356	89
Politics	417	334	83
Sport	511	409	102
Tech	401	321	80

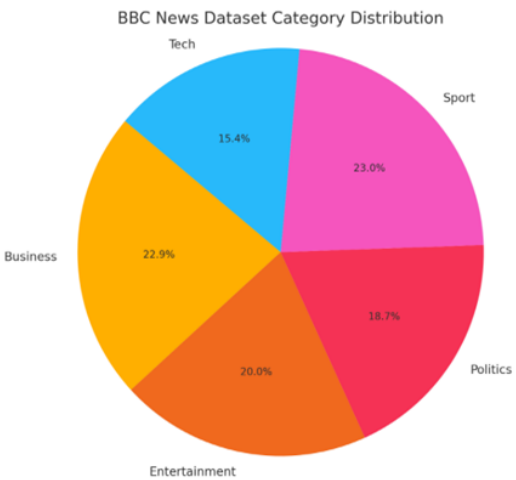


Figure 2. Distribution of document lengths and category proportions across the five thematic classes of the BBC News dataset (business, entertainment, politics, sport, and technology), illustrating the moderate class imbalance addressed in this study.

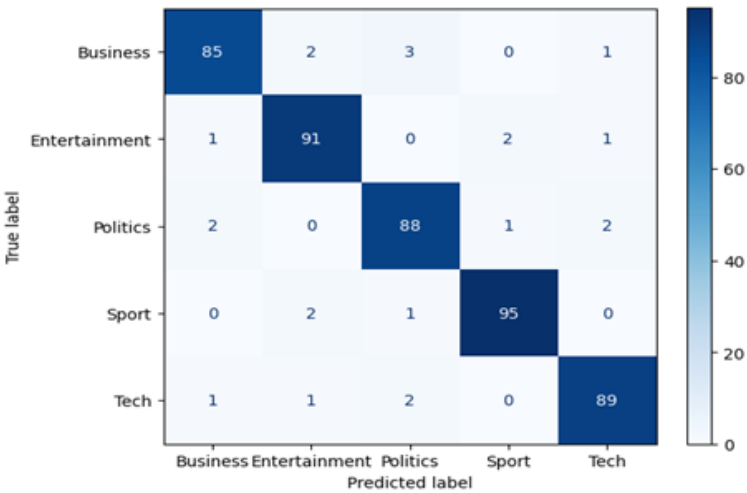


Figure 4. Confusion matrix summarizing the classification performance of the HCNN-LSTM model on the BBC News test set, illustrating the distribution of correct and misclassified predictions across the five news categories.

Table 3. Performance comparison of classification models on the imbalanced BBC News dataset (F1-score by category, overall accuracy, and macro-averaged F1-score).

Categories	F1 Score and Accuracy
NB	DT
SVM	CNN
LSTM	HCNN-LSTM
Business	0.83
0.85	0.86
0.88	0.89
0.93	Entertainment
0.85	0.87
0.87	0.89
0.88	0.95
Politics	0.89
0.89	0.88
0.89	0.87
0.95	Sport
0.85	0.89
0.87	0.87
0.89	0.93
Tech	0.88
0.87	0.89
0.89	0.88
0.95	Accuracy
0.87	0.88
0.89	0.88
0.89	0.95
Macro Avg F1 Score	0.86
0.88	0.89
0.88	0.89

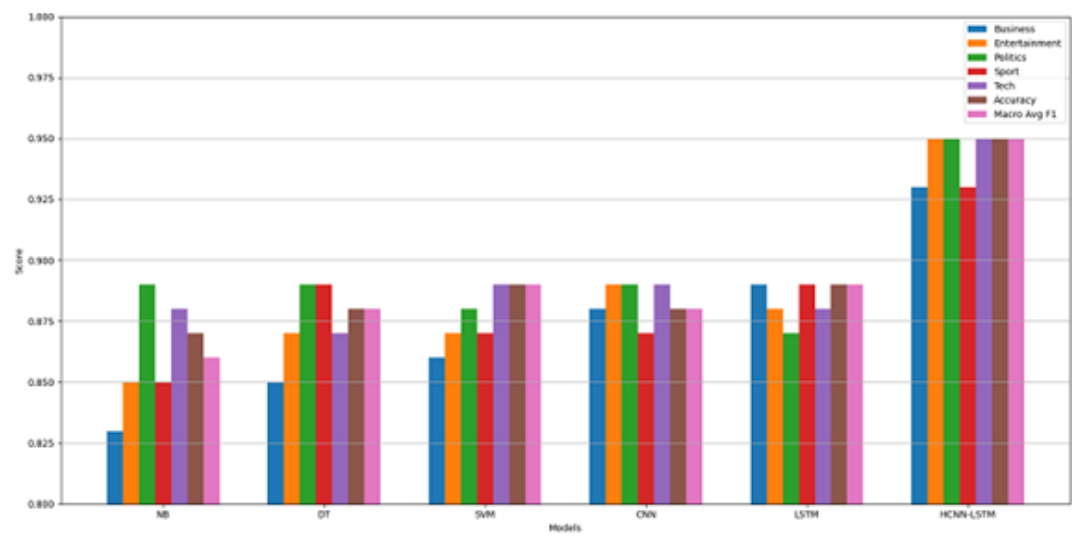


Figure 5. Comparative F1-score performance of all evaluated classifiers (NB, DT, SVM, CNN, LSTM, and HCNN-LSTM) across the five news categories on the imbalanced dataset, with the HCNN-LSTM model (light blue line) consistently outperforming baseline approaches.

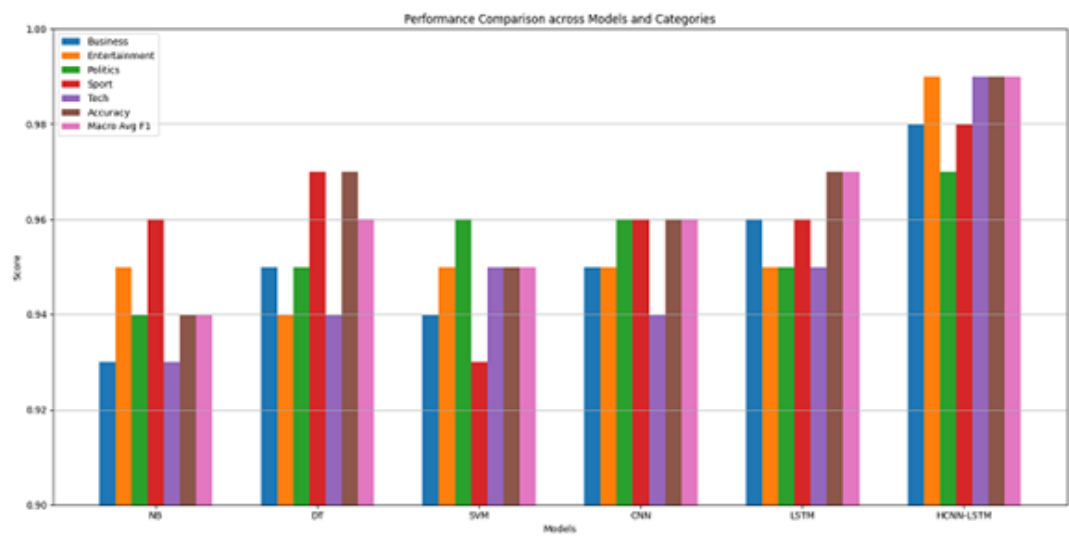


Figure 6. Comparative F1-score performance of all evaluated classifiers across the five news categories on the class-balanced dataset, demonstrating the sustained superiority of the HCNN-LSTM architecture following resampling.

Table 5. Hardware specifications of the computational environment used for model development and evaluation.

Specification	Details
Model	Intel i7-1355U
Processor Speed	Up to 5.0 GHz
RAM	16 GB
Graphics Card	Intel Iris X Graphics

Table 4. Performance comparison of classification models on the class-balanced BBC News dataset (post-resampling), reported as F1-score by category, overall accuracy, and macro-averaged F1-score.

Categories	F1 Score and Accuracy
NB	DT
SVM	CNN
LSTM	HCNN-LSTM
Business	0.93
0.95	0.94
0.95	0.96
0.98	Entertainment
0.95	0.94
0.95	0.95
0.95	0.99
Politics	0.94
0.95	0.96
0.96	0.95
0.97	Sport
0.96	0.97
0.93	0.96
0.96	0.98
Tech	0.93
0.94	0.95
0.94	0.95
0.99	Accuracy
0.94	0.97
0.95	0.96
0.97	0.99
Macro Avg F1 Score	0.94
0.96	0.95
0.96	0.97

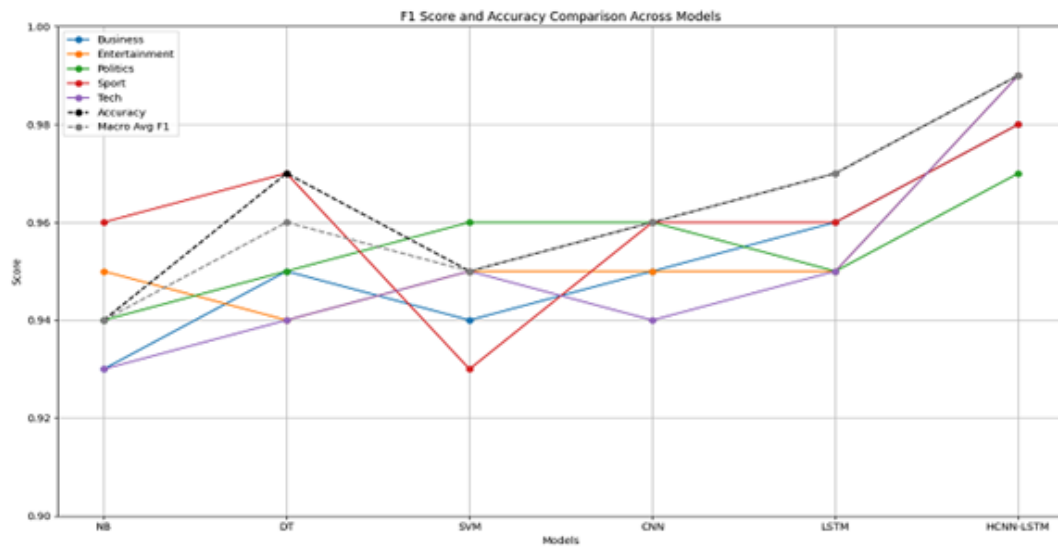


Figure 7. Comparison of overall classification accuracy across all evaluated models on the balanced dataset, highlighting the highest accuracy achieved by the proposed HCNN-LSTM framework.

imbalanced datasets (Thölke et al., 2023).

In addition to category-specific metrics, we calculated the macro-averaged F1-score, which averages the F1-scores of each category without weighting them by class frequency. This metric provides a more balanced assessment of classification performance across all categories.

3.5 Performance on the Imbalanced Dataset

The performance of the proposed model and baseline classifiers on the imbalanced dataset is summarized in Table 3. The results demonstrate that the HCNN-LSTM model consistently outperforms traditional machine learning algorithms, including naïve Bayes (NB), decision tree (DT), and support vector machine (SVM). Specifically, the proposed model achieved an overall accuracy of 0.95 and a macro-averaged F1-score of 0.95, which represents a substantial improvement compared with baseline classifiers. For example, the SVM classifier achieved an accuracy of 0.89, while the decision tree model reached 0.88. A closer examination of Table 3 reveals that the HCNN-LSTM architecture achieved strong performance across all five categories. The entertainment and politics categories both achieved F1-scores of 0.95, while the business and sports categories reached 0.93. Even the technology category, which contains the smallest number of samples, achieved a robust F1-score of 0.95. These results suggest that the hybrid architecture effectively mitigates the negative effects of class imbalance, allowing the model to identify minority categories with high reliability.

3.6 Performance on the Balanced Dataset

To further examine model performance, experiments were also conducted using a balanced version of the dataset. The results are summarized in Table 4. The HCNN-LSTM model achieved an accuracy of 0.99 and a macro-averaged F1-score of 0.99, outperforming all baseline models. In comparison, the best-performing traditional classifier, decision tree, achieved an accuracy of 0.97. The performance improvements observed in the balanced dataset reinforce the robustness of the proposed model. Notably, the entertainment and technology categories achieved near-perfect F1-scores of 0.99, indicating that the hybrid architecture successfully captures distinctive linguistic patterns within these domains.

3.7 Comparative Visualization of Model Performance

The performance differences among the models are visually illustrated in Figures 5, 6, and 7. Figure 5 compares the F1-scores of different models for the imbalanced dataset. The HCNN-LSTM model, represented by the light blue line, consistently outperforms other algorithms across all categories. Similarly, Figure 6 illustrates F1-score comparisons for the balanced dataset, again demonstrating the superior performance of the hybrid model. Finally, Figure 7 presents a comparison of classification accuracy across models for the balanced dataset. The results clearly show that the HCNN-LSTM architecture achieves the highest accuracy among all evaluated models. These visualizations provide further evidence that combining convolutional and recurrent neural networks yields significant improvements in classification performance.

3.8 Error Analysis Using Mean Squared Error

In addition to classification metrics, we also evaluated model performance using mean squared error (MSE), which measures the average squared difference between predicted and actual values. The HCNN-LSTM model achieved the lowest MSE of approximately 0.5%, indicating highly accurate predictions. In contrast, the SVM classifier exhibited the highest MSE (3.8%), reflecting a larger prediction error. Interestingly, the CNN and decision tree models produced similar MSE values slightly below 1%, suggesting relatively consistent performance. The LSTM model achieved an MSE of 1.6%, while naïve Bayes produced an error rate of approximately 2.7%. These findings further demonstrate the advantage of combining CNN and LSTM architectures with contextual embeddings.

3.9 Computational Environment

All experiments were conducted using the computational environment summarized in Table 5. The model was implemented on a system equipped with an Intel i7-1355U processor (up to 5.0 GHz), 16 GB of RAM, and Intel Iris Xe graphics. This hardware configuration provided sufficient computational capacity for training and evaluating the deep learning models used in this study.

3.10 Overall Interpretation

Taken together, the experimental findings strongly suggest that hybrid deep learning architectures can significantly improve text classification performance, particularly when dealing with moderately imbalanced datasets. By integrating convolutional feature extraction, sequential

modeling, and contextual embeddings, the HCNN-LSTM framework captures multiple levels of linguistic information simultaneously. Such multi-layered representation learning appears to be a key factor underlying the model's superior performance. These results are consistent with previous studies demonstrating the effectiveness of hybrid neural architectures in natural language processing tasks (LeCun et al., 2015; Lai et al., 2015).

4. Conclusion

The present study set out to examine the persistent challenge of news text classification when datasets exhibit uneven category distributions. While this issue has long been acknowledged in natural language processing research, addressing it effectively—without sacrificing model performance across categories—remains difficult. In this work, we introduced a hybrid deep learning framework, HCNN-LSTM, designed to mitigate these limitations. By combining convolutional neural networks and long short-term memory networks with contextual BERT embeddings and TomekLinks undersampling, the proposed architecture attempts to capture both localized lexical patterns and broader contextual dependencies embedded within textual data. The experimental findings, obtained from the BBC News dataset, suggest that this hybrid strategy offers tangible advantages. Across multiple evaluation metrics, the HCNN-LSTM model demonstrated consistently stronger performance than several traditional machine learning approaches, including naïve Bayes, decision tree, and support vector machine classifiers. In particular, the model achieved an accuracy and macro-averaged F1-score of 0.95 on the imbalanced dataset, which increased to 0.99 when evaluated on the balanced version of the dataset. These outcomes, while encouraging, also hint at the broader potential of combining contextual embeddings with layered neural architectures for complex text classification tasks. Taken together, the results indicate that hybrid models may provide a practical path toward improving classification reliability, especially for minority categories that are often overlooked in imbalanced datasets. That said, the present study is not without limitations. The dataset used here, though widely recognized, is relatively modest in size and limited to English-language news articles. Future research might therefore extend this framework to larger, multilingual corpora or investigate the integration of more recent transformer-based architectures to further enhance scalability and generalization in real-world applications.

Author Contribution

K.M. conceived and designed the study, developed the hybrid HCNN-LSTM model architecture, and drafted the manuscript. M.N.U. contributed to data preprocessing, feature preparation, and class balancing procedures. M.A.R. performed model training, evaluation, and analysis of classification metrics. S.R.H. contributed to the experimental design and conducted comparative performance analysis. M.N.H. supervised the study, provided critical guidance on methodology, and critically reviewed and revised the manuscript. All authors read and approved the final manuscript.

Acknowledgement

The authors K.M. et al., would like to thank the creators of the BBC News dataset for making it publicly available for research purposes. The authors also acknowledge the institutional and computational support that facilitated this work.

Competing Financial Interests

The authors K.M. et al., declare no competing financial interests.

References

- Cai, L., & Hofmann, T. (2004). Hierarchical document categorization with support vector machines. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management* (pp. 78–87).
- Dönicke, T., Damaschke, M., & Lux, F. (2019). Multiclass text classification on unbalanced, sparse and noisy data. In *Proceedings of the 1st NLP Workshop on Deep Learning for Natural Language Processing* (pp. 58–65).
- Ger, S., Jambunath, Y. S., & Klabjan, D. (2023). Autoencoders and generative adversarial networks for imbalanced sequence classification. In *Proceedings of the IEEE International Conference on Big Data* (pp. 1101–1108).
- Hasib, K. M., et al. (2020). A survey of methods for managing the classification and solution of data imbalance problem. *arXiv preprint arXiv:2012.11870*.
- Hasib, K. M., et al. (2022). BMNet-5: A novel approach of neural network to classify the genre of Bengali music based on audio features. *IEEE Access*, 10, 108545–108563.
- Hasib, K. M., Showrov, M. I., Mahmud, J. A., & Mithu, K. (2022). Imbalanced data classification using hybrid undersampling with cost-sensitive learning method. In *Edge analytics: Selected proceedings of the 26th International Conference ADCOM 2020* (pp. 423–435).
- Karim, A., Azam, S., Shanmugam, B., & Kannoorpatti, K. (2020). Efficient clustering of emails into spam and ham: The foundational study of a comprehensive unsupervised framework. *IEEE Access*, 8, 154759–154788.

- Kim, J., & Kim, J. (2018). The impact of imbalanced training data on machine learning for author name disambiguation. *Scientometrics*, 117(1), 511–526.
- Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.
- Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent convolutional neural networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 29, No. 1).
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- Letouzé, E. (2012). Big data for development: Challenges & opportunities. UN Global Pulse.
- Misra, R., & Grover, J. (2021). Sculpting data for ML: The first act of machine learning. University of California San Diego.
- Navin, J. R. M., & Pankaja, R. (2016). Performance analysis of text classification algorithms using confusion matrix. *International Journal of Engineering and Technical Research*, 6(4), 75–78.
- Selamat, A., Yanagimoto, H., & Omatu, S. (2002). Web news classification using neural networks based on PCA. In *Proceedings of the 41st SICE Annual Conference* (Vol. 4, pp. 2389–2394).
- Shah, K., Patel, H., Sanghvi, D., & Shah, M. (2020). A comparative analysis of logistic regression, random forest and KNN models for text classification. *Augmented Human Research*, 5(1), 12.
- Sun, A., Lim, E. P., & Liu, Y. (2009). On strategies for imbalanced text classification using SVM: A comparative study. *Decision Support Systems*, 48(1), 191–201.
- Thölke, P., et al. (2023). Class imbalance should not throw you off balance: Choosing the right classifiers and performance metrics for brain decoding with imbalanced data. *NeuroImage*, 277, 120253.
- Tong, X., Wu, B., Wang, S., & Lv, J. (2018). A complaint text classification model based on character-level convolutional network. In *Proceedings of the IEEE 9th International Conference on Software Engineering and Service Science* (pp. 507–511).
- UN Global Pulse. (2012). Big data for development: Opportunities & challenges. Retrieved from <http://www.unglobalpulse.org>
- Yang, Y., et al. (2018). TICNN: Convolutional neural networks for fake news detection. arXiv preprint arXiv:1806.00749.
- Yin, J., et al. (2020). A novel model for imbalanced data classification. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 4, pp. 6680–6687).