IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 3 August 2023, accepted 24 August 2023, date of publication 29 August 2023, date of current version 5 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3309697

RESEARCH ARTICLE

MCNN-LSTM: Combining CNN and LSTM to Classify Multi-Class Text in Imbalanced News Data

KHAN MD HASIB[1], **(Member, IEEE), SAMI AZAM**[2], **(Member, IEEE), ASIF KARIM**[2], **(Member, IEEE), AHMED AL MAROUF**[3], **(Member, IEEE), F M JAVED MEHEDI SHAMRAT**[4], **(Member, IEEE), SIDRATUL MONTAHA**[5], **KHENG CHER YEO**[2], **MIRJAM JONKMAN**[2], **(Member, IEEE), REDA ALHAJJ**[3,6,7], **(Senior Member, IEEE), AND JON G. ROKNE**[3], **(Life Fellow, IEEE)**

[1]Department of Computer Science and Engineering, Bangladesh University of Business and Technology, Dhaka 1216, Bangladesh
[2]Faculty of Science and Technology, Charles Darwin University, Casuarina, NT 0810, Australia
[3]Department of Computer Science, University of Calgary, Calgary, AB T2N 1N4, Canada
[4]Department of Computer System and Technology, University of Malaya, Kuala Lumpur 50603, Malaysia
[5]Department of Computer Science and Engineering, Daffodil International University, Dhaka 1207, Bangladesh
[6]Department of Computer Engineering, Istanbul Medipol University, 34810 İstanbul, Turkey
[7]Department of Heath Informatics, University of Southern Denmark, 5230 Odense, Denmark

Corresponding author: Asif Karim (asif.karim@cdu.edu.au)

ABSTRACT Searching, retrieving, and arranging text in ever-larger document collections necessitate more efficient information processing algorithms. Document categorization is a crucial component of various information processing systems for supervised learning. As the quantity of documents grows, the performance of classic supervised classifiers has deteriorated because of the number of document categories. Assigning documents to a predetermined set of classes is called text classification. It is utilized extensively in a wide range of data-intensive applications. However, the fact that real-world implementations of these models are plagued with shortcomings begs for more investigation. Imbalanced datasets hinder the most prevalent high-performance algorithms. In this paper, we propose an approach name multi-class Convolutional Neural Network (MCNN)-Long Short-Time Memory (LSTM), which combines two deep learning techniques, Convolutional Neural Network (CNN) and Long Short-Time Memory, for text classification in news data. CNN's are used as feature extractors for the LSTMs on text input data and have the spatial structure of words in a sentence, paragraph, or document. The dataset is also imbalanced, and we use the Tomek-Link algorithm to balance the dataset and then apply our model, which shows better performance in terms of F1-score (98%) and Accuracy (99.71%) than the existing works. The combination of deep learning techniques used in our approach is ideal for the classification of imbalanced datasets with underrepresented categories. Hence, our method outperformed other machine learning algorithms in text classification by a large margin. We also compare our results with traditional machine learning algorithms in terms of imbalanced and balanced datasets.

INDEX TERMS Big data, text classification, imbalanced data, machine learning, MCNN-LSTM.

I. INTRODUCTION

Researchers in the scientific community create an enormous volume of work each year. Peer-reviewed English-language scholarly journals produced 2.5 million articles in 2014, according to a recent study [1]. The number of journals and publications continues to expand at an increasing rate. The sheer amount of these papers necessitates the use of automated categorization and organizing systems. The use of artificial intelligence in monitoring and evaluating large-scale

The associate editor coordinating the review of this manuscript and approving it for publication was Filbert Juwono.

data for sustainable development goals has been revolutionized. Data scientists play a critical role in the actual implementation of sustainable development initiatives by automatically collecting and organizing large amounts of data. Again, big data for research and development is a stark contrast to big data for the general public, which has prompted us to reconsider our approach to automated text categorization using Artificial Intelligence (AI). After data is acquired, it is extremely difficult to delete an instance because it is underrepresented in the dataset, rather it may demonstrate insights into human well-being and development [2]. As a result, while dealing with a large amount of data, there is a risk of jumping to conclusions without a thorough knowledge of the complex dynamics. It is necessary to keep in account the digital gap between nations and their way of news representation when categorizing text data for development purposes. A common classification problem is like this: When the set of features is given a set of categorized samples from two or more classes (called a ''training set''), it puts each category in the class with the most similarities. Categorization trees, Naïve Bayes, Support Vector Machines (SVM), Neural Nets, and Ensemble approaches have been utilized recently in automatic document classification. In terms of interpretability, classification trees and Naïve Bayes approaches are excellent and yet they are usually not as precise as other methods.

Therefore, since sample dimensions and the number of features and sub-fields have grown in recent years, this is no longer feasible in terms of automated document categorization which has become increasingly difficult. Research fields that were barely recognized only five years ago are now seeing rapid development and attention. This expansion of subfields has happened in a variety of areas, including biology (e.g., CRISPR-CA9 [3]), material science (e.g., chemical programming [4]), and health sciences (e.g., precision medicine [5]). It is essential to designate a paper by its particular area, but also to organize it within its general field and its connected sub-fields, because of the increasing number of sub-fields. The hierarchical document categorization approach [6] has become useful to solve this problem. This may be done using many of the current ways for categorizing documents due to the main subject area, but few of them can effectively categorize documents into specific subfields or areas of specialty. There are a growing number of class labels that existing document classification methods are unable to manage because of the combination of top-level fields and all subfields. Many researchers work with only CNN or only RNN model to classify text data whether the performance is not up to the mark [7]. Compared to correctly portrayed, balanced data, we've discovered, data that's unbalanced is more difficult to categorize effectively. It's time to use computational intelligence, notably neural network methods, in order to bridge this difference [8]. In this area, our suggested deep neural network model offers considerable benefits. Text classification of unbalanced data is the subject of our study, and we are looking for ways to make it more accurate and find groups that aren't getting enough attention. Normal machine learning algorithms give the impression that they are good at classifying, but when we look more closely, we see that categories that aren't represented well are classified badly and don't show up in the general results. A combining method of MCNN-LSTM has been suggested that aims to address this issue while retaining acceptable overall accuracy.

Rest of the segment of the paper is designed as follows; part II defines the research scope and aim of the work and part III shows a short preview of researchers that they have done in the past in that field. The background study is shown in part III. In section IV, our proposed methodology has been shown. The experimental berry has been described in section V and section VI, VII puts an end to the paper with the limitations and challenges of the work.

II. RESEARCH AIM AND SCOPE OF THE PAPER

This research study will focus on the text categorization of imbalanced data, specifically on how to enhance accuracy while identifying underrepresented groups using a hybrid framework called MCNN-LSTM. The goal of this research work is to concentrate on the text classification of unbalanced data. Necessary procedures may be summarized as follows:

1. Provided a data preprocessing step of two hundred thousand dataset.
2. Balanced the imbalanced data using an under-sampling technique Tomek-Link algorithm.
3. Word Embedding technique is used to extract the most relevant feature of the dataset.
4. Proposed the hybrid framework MCNN-LSTM model how to improve upon the accuracy while comparing with traditional machine learning model in terms of the balanced and imbalanced dataset.

III. LITERATURE REVIEW

The difficulty of classifying with unbalanced data has attracted the attention of researchers, who have suggested and created a number of potential solutions and applications. Classifying using ML algorithms may be challenging when dealing with unbalanced data. The authors of this study [9] experimented with a dataset of unstructured data acquired from the Human Resource field comprises of documentation of job descriptions which was highly imbalanced for different classes. To convert into structured data, all the documents are segmented to extract meaningful information. Moreover, several widely used methods are implemented to transform text documents into numerical representations. Experimental studies were conducted on several demonstrations of text using various classification algorithms under a well-balanced scheme to deal with the issue of imbalance text data of numerous degrees. They proposed a cost sensitive tactic where the costs are computed using a Differential Evolution algorithm [10]. The implementation of the approach was carried out in two steps namely resampling methods and cost-sensitive learning. Firstly, after adjusting the costs at the class level, they are developed at the data instance level. They employed Logistic Regression, SVM and Decision Tree

Classifier as classification algorithms. They found that Logistic regression and SVM are more biased in terms of performance in the context of high imbalance data compared to decision trees. Modified version of CNN and LSTM are proven to work satisfactorily to classify brain tumor on 3D MRI scans [11]. A hybrid approach combining CNN and LSTM, performing ablation study, was proposed in [11], which sets up a basis for our proposed approach in this paper.

A novel framework named BSIL was introduced by [12] which was constructed on brainstorm optimization (BSO), to magnify the competence of NN in imbalanced text data classification. This framework had the capability of the BSO processes by sampling imbalanced datasets precisely. Their first approach involves employing global random sampling and scrambling segmentation methods to create several relatively composed subsets of an imbalanced dataset. Afterwards, a parallel technique is applied to train a classifier on a subset proficiently. Finally, to accomplish a precise prediction outcome, they proposed a decision-making layer. Recorded results showed that BSIL connected with CNN [12], RNN and Self-attention model, performed with the highest accuracy. BSIL was formed based on the concept of distributed ML that could process large-scale data in a particular node. Five text classification datasets in English and Chinese were experimented to analysis. Their experimental results show that LSTM and SaNN models performed yielding the lowest F1_Score of 37.5% and 24.4% respectively. The best performance was achieved from BSIL-CNN model on the CR dataset, evaluated based on the achieved highest F1_score of 76.4%.

The authors of this study [13] represented a novel classifier to solve the task of text-mining, as well as interactive information access. Their proposed approach was able to extract hierarchical relations between topics, conducting unsupervised clustering of text documents and keywords. They extracted keywords and weight using the graph-of-words method and used a bipartite co-clustered graph on a two-dimensional plane. Using three classifiers: KNN, SVM and graph, on the BBC sport corpus dataset, they got the F1_score of 97.77%, 95.09% and 95.98% respectively. While experimenting on BBC news corpus, for the similar classifiers, F1_score of 95.56%, 97.19% and 92.06% were achieved respectively.

This study [14] conducted their research to improve classification performance on unbalanced and noisy text data using different techniques and three NN based approaches: a one-layer perceptron, a two-layer Doc2Vec network and finally a multilayer perceptron. The dataset used for the study was collected from Kaggle, and consists of 57,647 English song texts with their corresponding artist name and song titles. In preprocessing phase, various meaningful features were extracted from the text data. The proposed model: extended multi-layer perceptron performed with the highest F1_score of 79%. They came to a conclusion that to deal with imbalanced data oversampling or under sampling methods can be applied.

This work [15] provided a simple yet efficient technique to deal with sentiment classification challenges given by domain-sensitivity and data imbalance. One lexicon graph was constructed using a NN-based architecture to learn word representations from large-scale, out-of-domain corpora, and the other was built using word vectors trained on a small-scale, domain-specific corpus by means of a count-based Singular Value Decomposition (SVD) model. A framework built on domain-adaptive feature generation is refined such that both generic and domain-specific information may be extracted. To address the issue of biased results from sentiment studies based on a minority social class, they developed a new oversampling approach. They played around with datasets from the book, dvd, elec, ktichen, and SemEval-2013(Test) categories. Accuracy levels of 78.5, 75.5, 78.5, and 81.1 percent were attained on the Books, DVD, Electronics, and Kitchen datasets, respectively. The model achieved an F1_score of 81.5% on the SemEval-2013(Test) dataset.

In this research [16], a text data classification algorithm based on improved KNN is introduced using clustering center method to address the issue of calculation complexity. They eradicated the highest effect from the training text data by compressing the dataset and removing samples adjacent to the border. Afterwards, k-means clustering algorithm is used to cluster the training samples belong to each type. Finally, after introducing a weight value, the reformed data samples are employed to achieve KNN based text classification. Preprocessing step carried out in this research includes sentence segmentation, eliminating stop words, extracting features and calculating the weight. Their proposed model outperformed the traditional KNN algorithm with an average Precision, Recall and F1 score of 91.33%, 91.15% and 91.23% reducing the calculation complexity.

A hybrid model combining CNN and Support Vector Machine (SVM) was proposed in this research [17], to detect handwritten digits on MNIST dataset which contains 70,000 data elements altogether. In this method, CNN is employed to extract features [18], where SVM was used as a binary classifier substituting the softmax layer from CNN. The MNIST dataset contains images of assorted and extremely biased handwritten digits. Experimental result shows the efficiency of adopting the framework by attaining a detection accuracy of 99.28%. To classify sentence level, CNN is employed in this study [19] to conduct a series of experiments, training on the uppermost of pre-trained word vectors. A simple CNN architecture can be hyperparameter tuned to upgrade performance [20]. Authors showed that, using fine-tuning approach, task-specific vectors can aid to improve classification accuracy. Moreover, the proposed architecture was modified to employ both task-specific and static vectors. The proposed CNN-multichannel model outperformed several previous studies by recording an accuracy of 95%.

As CNN and RNN are regarded as two mainstream networks of processing natural languages, the authors of this paper [21] combined these two architectures to represent and

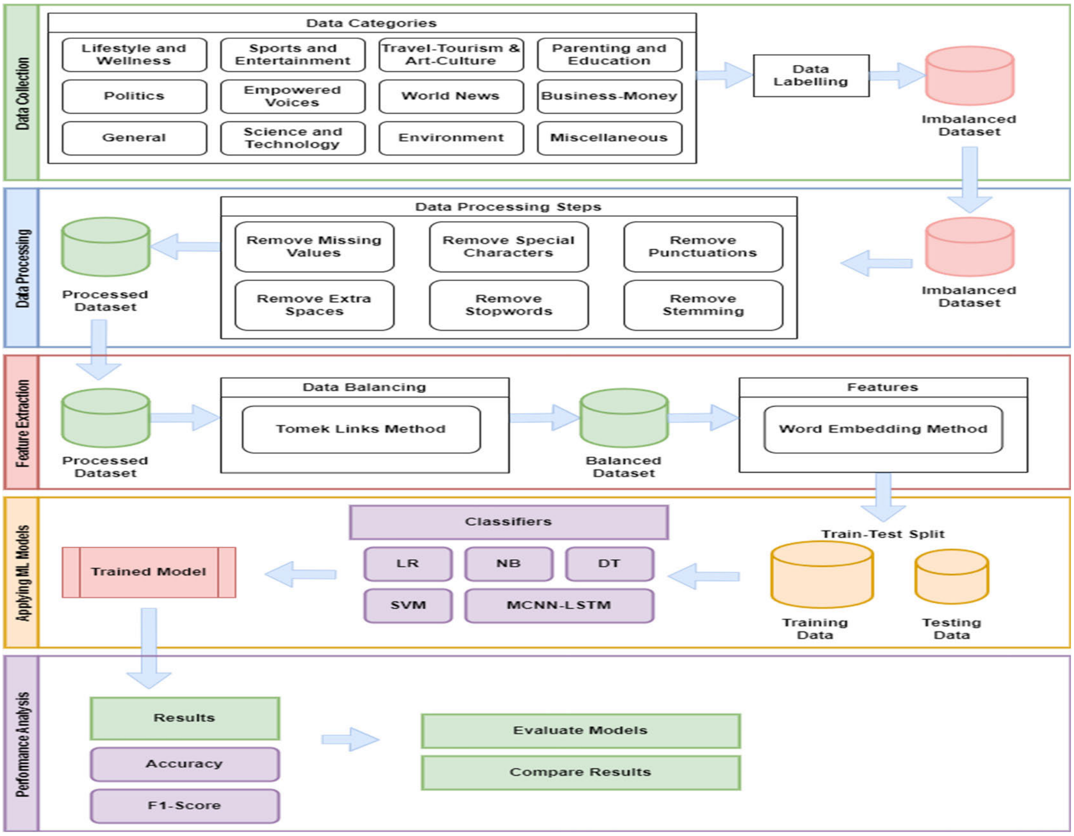

FIGURE 1. Workflow of the proposed method.

classify text data and named the model C-LSTM. In this regard, to extract a series of upper-level phrase representations, CNN is employed. Afterwards, these features are passed into LSTM network to acquire the sentence representation [22]. Their model could capture both local features and global along with temporal sentence semantics. Stanford Sentiment Treebank (SST) benchmark dataset comprising of 11855 movie reviews is utilized for the experiment. The highest accuracy of 94.6% was acquired from their proposed model that proves the effectiveness of their adopted approach.

IV. METHODOLOGY

Fig. 1 displays the structure of the suggested technique and the stages involved. Initially, data preparation and standardization procedures were used for the dataset to clean and prepare it for future processing as shown in below pseudocode 1. We have developed our study on 200k news headlines datasets by applying Deep Learning techniques. Firstly, we have focused on data preprocessing where we deal with the missing values of the data [23], [24]. Secondly, remove the special character of the data and remove punctuation. Thirdly, we cope with the problem of removing extra space. Finally, we work on removing stop words and stemming. In the data extraction part, handling skewed values is always challenging to handle, from our dataset we have used Tomek-Link to handle the imbalanced data. Though Tomek-Link algorithm can be used as undersampling method, it cannot balance the majority and minority class data [25]. After data extraction, we performed some feature extraction using unigram, bigram, and trigram. In the classification part, we have applied multiple machine learning algorithms such as: LR, SVM, NV, and DT and lastly work on our proposed MCNN-LSTM. For result performance, we used the accuracy, F1-score, recall and precision in the training model and lastly compare the results with our suggested model.

To develop our study, we have followed the process,

1. Dataset Collection
2. Dataset Preprocessing and Customized Dataset Preparation
3. Proposed MCNN-LSTM model

A. DATASET COLLECTION

We have collected a dataset from HuffPost where the total instance is about 202,372 and the number of total features is 40. This dataset contains 6 years of news headlines and short description data from different newspapers. To assess the model's performance, the text employs 41 large-scale text categorization datasets. These datasets are grouped into 12 different sorts of categories. The dataset is partitioned by 70:30 ratio for training and testing.

1) DATA LENGTH

The following Fig. 2 depicts the data length of the dataset where we can see that the highest percentage of the category of the dataset is "Lifestyle and Wellness" and the number is 20.2% while the lowest number of data is "Environment" with the number of 2.0%. From this chart, we can visualize that in our work which type of category of the dataset plays a vital role in terms of producing commendable outcomes. The "Politics", "Sports and Entertainment" are used 16.3% and 15.1% respectively.

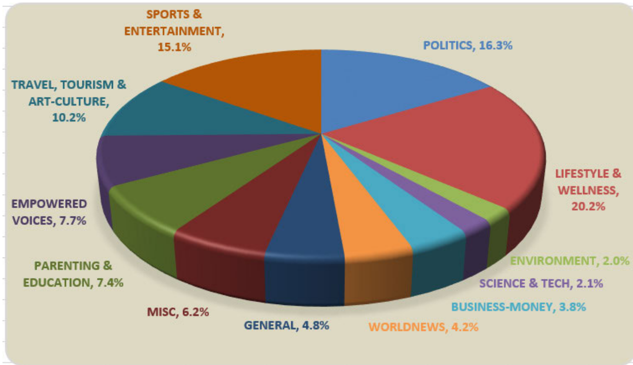

FIGURE 2. Data Length of the dataset.

B. DATASET PREPROCESSING AND CUSTOMIZED DATASET PREPARATION

1) DATA BALANCING

After the data collection process, the raw data may have different issues, such as noise, imbalance, or bias. These issues may affect the accuracy of categorization process. Data pre-processing is a technique which includes data cleaning mining, balancing etc.

Pseudocode 1 Handling Imbalanced Dataset With kNN (Dataset = (X, Y))

1. q = number of instances
2. k= number of boosting iterations
3. (misclassified_weight$^+$. classified_weight$^-$) = weight_Assignment(dataset,k)
4. initialize weight $x_i \epsilon$ Q to 1/q.
5. **for** i = 1 to k **do**
6. create balanced dataset Q_i with distribution Q using Under-sampling(dataset)
7. Decision Tree T_i from Q_i employing
8. los_sum $\leftarrow \Sigma_{\, T_i(x_i) \neq y_i} Q_i^t$ misclassified_weight$^+$ (i)
9. acc_sum $\leftarrow \Sigma_{\, T_i(x_i) = y_i} Q_i^t$ classified_weight$^-$(i)
10. accumulate the error rate of T_i, error (T_i)
11. **if** error (T_i) $\geq$ 0.5 **then**
12. go back to step 5 and try again
13. **end if**
14. **for** each $x_i \epsilon Q_i$ that correctly classified **do**
15. accumulate weight of x_i by $\alpha_t \leftarrow$ (error (T_i)/1-error(T_i)) //update weights
16. **end for**
17. normalize Q
18. **end for**

return the class with largest weight

Here, hyper-parameter optimization determines the parameter k. Then, random under-sampling is conducted on each of the newly established clusters by picking 50 percent of the instances at random and eliminating the remaining instances. As clustering is performed prior to sampling, this approach should theoretically perform best when the dataset tends to be utilized for building a large number of clusters. Each instance is first assigned a weight of 1/q, where q is the total number of training instances. The weights of occurrences are modified based on their classification.

To solve the data imbalance problem, there are different techniques adopted to apply on the skewed or imbalanced data when both majority and minority class occurrences overlap. For this study, we have applied the Tomek-Links technique [26], which examines these pairings of instances while removing the majority class's data from each pair. The purpose of this algorithm is to clean up the boundary between the minority and majority classes. Named after Ivan Tomek himself, Tomek-Links [26] is an undersampling approach which is a modification of condensed nearest neighbors; however, it cannot balance data from the majority and minority classes. Tomek-links is employed as a data cleaning approach in our strategy to eliminate noisy data. Pseudocode 1 shows the possible way to handle the data imbalance problem with kNN algorithm.

2) DATASET PARTITION

We divide the dataset into 12 categories, where the name of categories are: "Lifestyle and Wellness", "Politics", "Sports and Entertainment", "Travel_Tourism and Art_Culture", "Empowered Voices", "Parenting and Education", "Misc", "General", "Worldnews", "Business and Money", "Science and Tech", and "Environment". The highest number of sub-categories is 5 for "Lifestyle and Wellness", "Sports and Entertainment", "Sports and Entertainment" category and the lowest number of sub-categories is 1 for "Politics". The dataset is splitted into 70:30 for training and testing part. Table 1 shows the dataset partition of our work. From table 1, we also analyzed that the highest number of test set is 14275 for "Lifestyle and Wellness" categories and the lowest number of test set is 1450 for "Environment" category.

C. PROPOSED MCNN-LSTM MODEL

Initially, we developed a structure utilizing the CNN approach. Second, we installed an LSTM-based model. Lastly, we created a hybrid model that combines the CNN and LSTM techniques. Based on the testing findings that we will describe in the next section, the hybrid model provided the best performance. For this reason, we will focus on describing the CNN-LSTM hybrid model that we refer to as MCNN-LSTM. The suggested CNN-LSTM model is shown in Fig. 3, followed by pseudocode 2.

Here, we discuss the CNN-LSTM model's design and operation concept. As mentioned above, the output of the

TABLE 1. Dataset partition.

Dataset	Number of sub-categories	Training set	Test set
Lifestyle and wellness	5	33306	14275
Politics	1	26974	11560
Sports and entertainment	5	24842	10647
Travel-tourism & art-culture	5	17044	7805
Empowered voices	4	12888	5524
Parenting and education	4	17598	5279
Misc.	4	10310	4419
General	4	8055	3452
World news	3	7068	3029
Business-money	2	6460	2769
Science and tech	2	3653	1565
Environment	2	3382	1450

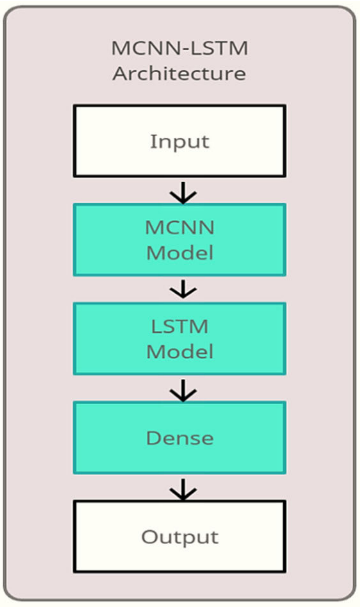

FIGURE 3. Proposed MCNN-LSTM Model.

Embedding layer corresponds to a CNN layer of type "Convolution 1D." In order to lower the dimensionality of the CNN output [19], a MaxPooling layer is then implemented. Next, an LSTM layer is linked. In the last step, a dense output layer with a single neuron and a sigmoid activation function is used to determine class decisions. In the following pseudocode 2, we analyzed the proposed MCNN-LSTM method. Here, we take the dense layer activation function is "ReLU" and the input is the total number of features. The number of outputs is 12 and the dropout layer, we used a dropout rate of 0.5.

A typical method for classifying text data is the matching of news categories to the substance of news stories. The news category dataset has around 200k unique articles divided into 42 categories, making it an appropriate dataset for our study. We attempted to arrange these 12 categories into 12 groups while also attempting to balance the dataset. To prepare our text data for machine learning, we must modify its structure to be more conducive to analysis by our classification model and extraction of pertinent information.

The proposed MCNN utilizes sentences as regions, splitting an input text into several regions so that the useful affective information in various regions can be extracted and weighted according to their contribution to the multi class prediction, as opposed to the standard CNN which considers the entire text as input [54]. On other hand, the LSTMs process sequential data, and as the number of layers is expanded, the input data is gradually abstracted at higher and higher levels. The number of memory cells in a layer is less crucial than the network depth. Stacked LSTM is used to refer to LSTMs that have numerous layers. The sequential output is always provided by these upstream levels rather than a single output result.

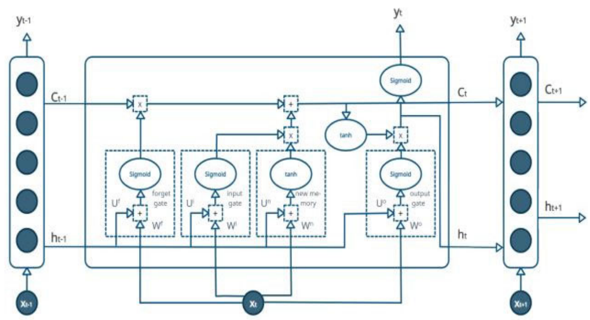

FIGURE 4. Working procedure of LSTM in our proposed method.

A sigmoid function and applied filter may eliminate unnecessary cell state information from forget gates to improve LSTM network performance [27]. Fig. 4 depicts the structure of LSTM in our proposed model. The sigmoid and tanh functions are used here so that we can be sure that the material that is being contributed to the unit condition is not very important. There are a few ways to express in Equation 1:

$$\begin{aligned} u_r &= \varphi\left(E_i \cdot [x_{n-1}, y_n] + j_i\right) \\ \tilde{D}_p &= \alpha\left(E_u \cdot [x_{t-1}, y_n] + j_u\right) \\ D_p &= q_c \circ D_{n-1} + u_r \circ \tilde{D}_p \end{aligned} \tag{1}$$

where it refers to the input gate and E_i and J_i are weights and bias of the input gate. Approximately D_p refers to the potential update vector, D_p is the new cell state and D_{n-1} is the previous cell state. In addition to that, it makes a vector by the application of the tanh function, which works in a manner analogous to the input gate. Equation 2 depicts the work:

$$\begin{aligned} a_q &= \sigma\left(E_o \cdot [x_{n-1}, y_n] + j_o\right) \\ f_q &= o_q * tanh\left(D_p\right) \end{aligned} \tag{2}$$

where a_q refers to the output gate, f_q refers to the new hidden state, and E_o and j_o denotes to the weights and biases of the output gate are discussed here. Because of its usefulness in assessing and forecasting temporal sequence data, we've opted to employ a traditional LSTM network. When training regular RNNs, the vanishing gradient issue may arise [55]. Because of this need, LSTMs were developed. When compared to other sequence learning algorithms like RNNs and hidden Markov models, LSTMs perform much better. Effective preprocessors effectively represent the document in terms of both space and time needs, while preserving retrieval efficiency. A basic extraction technique collects all files from their respective directories and combines them into a single JSON file. We referred to the technique of cleaning a text and deleting any extraneous material as noise removal [28]. This includes text cleaning (to remove punctuation marks and extra space), stopword removal (removal of all language-specific functional words, such as pronouns, prepositions, and conjunctions), and lemmatization (grouping together the various inflected forms of words so they can be analyzed as a single term using the Natural Language Toolkit (NLTK) [29].

Pseudocode 2 MCNN-LSTM (Dataset = (X, Y))

1. Input: N (A set of n text data, $N = N_1, N_2, \ldots, N_n$)
2. Output: Y (Text label: 0 or 1)
3. foreach text N_i in N do
4. V_i = Word_Embedding (N_i)
5. end
6. foreach V_i do
7. $C_i = CNN(V_i)$
8. end
9. foreach C_i do
10. $O_i = LSTM(C_i)$
11. end
12. foreach Oi do
13. Y_i = sigmoid (O_i) // Dense layer with sigmoid function
14. end

1) FEATURE SELECTION

Feature selection approaches are crucial for machine learning algorithms, since they are required to extract the most informative features for categorization. The primary objective of feature selection is to eliminate input characteristics that are irrelevant to categorization [56]. This may reduce the cost of training and increase precision. Feature selection enables the machine learning model to exclude irrelevant and redundant predictors and build a new strategy for achieving more accuracy with less text input. Redundant features may hinder the efficiency of machine learning systems. We utilized the word embedding approach for choosing features [30]. Word embedding is a term used in natural language processing (NLP) to describe the representation of words for text analysis, often in the form of a real-valued vector that encodes the word's meaning so that words that are are presumed to have comparable meanings.

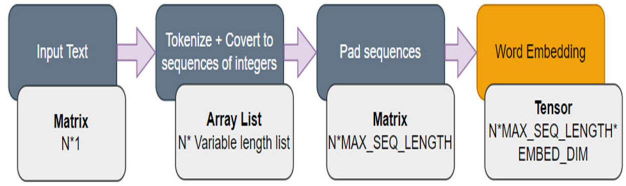

FIGURE 5. Word embedding technique of our work.

Above the Fig. 5 represents the workflow of the word embedding technique to select the feature of the model. The input text is the number of Matrix where the number is considered the numerical value. Then we tokenize the sentence after converting it from numerical to text data then convert the sequence of integers. Tokenization removes important data from your environment and replaces it with these tokens. Again, to get the numeric value of the sentence, we use the pad sequences and lastly for embedded words we use the max_sequence_length 500.

Reducing the dataset's dimension and deleting extraneous characteristics helps provide a thorough classification model. The primary difficulty of the feature reduction technique is to identify the optimal subset of characteristics in order to get the most accurate categorization. In Fig. 6, which depicts the confusion matrix between the model's features, the dataset's categories are represented as properly and incorrectly categorized.

The "Lifestyle and Wellness" category are correctly classified with 376 class. The highest number of classified values is 3258 which denotes the "Empowered Voices" and the lowest number of values is "Lifestyle and Wellness" category. The confusion matrix is a popular metric for assessing machine learning algorithms. Consequently, selecting proper evaluation criteria is crucial for addressing the data imbalance issue [31], [32], [33], [34]. Again, the "Politics" category data are classified correctly with the value of 677 which is lower compared to "Parenting and Education" category data of 816.

V. EXPERIMENTAL RESULTS AND DISCUSSION

We demonstrate the applicability of our proposed approach by using a dataset including news category information. Each word is encoded as a continuous, low-dimensional real-value vector by the embedding layer. To emphasize the benefits of the network structure we created, we manipulated the word embedding factor relative to other network models. All deep learning models used in the experiment trained their word vectors using word2vec. We retain the model parameters with the greatest validation precision and use them to analyze the test set. Local features may be successfully retrieved from a sequence using the CNN layer [35]. In addition, we use a Dropout layer to prevent overfitting because doing so takes significantly less time. Dropout has the effect of reducing the aggregate adaptability of neuron modes while simultaneously

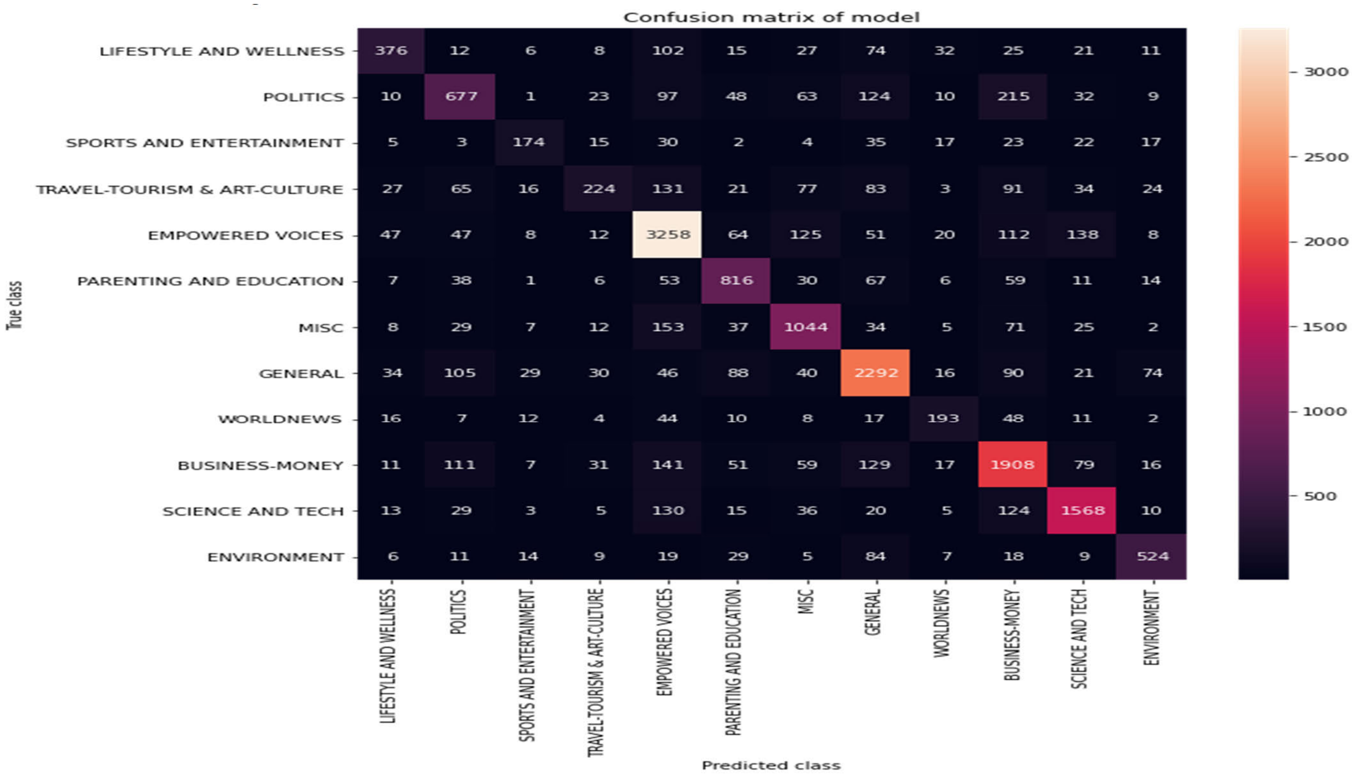

FIGURE 6. Confusion matrix of the model.

increasing their generalizability [36]. After each of the CNN and CNN-LSTM layers comes the dropout, which has a value of 0.5 and is applied there. Our model's parameters are initialized in a random fashion, with the exception of the maximum sequence length, which we set to 500 characters, and the embedding, which we set to 50. The Dropout layer is a mask that, when applied, prevents certain neurons from contributing information to the layer below it while leaving all other neurons unaltered [37]. We applied a Dropout layer to the input vector, in which case it will nullify some of the characteristics of the input vector. Alternatively, we applied it to a hidden layer, in which case it will nullify some of the neurons that are buried behind the hidden layer [38]. To evaluate the performance of the machine learning models, we used mainly two criteria: accuracy score and F1 score.

Accuracy: The percentage of correct predictions made in relation to the total amount of data collected is the definition of accuracy. Training data are used in order to perform accuracy evaluations.

$$\text{Accuracy} = \frac{Number\ of\ Correct\ Prediction}{Total\ Number\ of\ predictions\ made}$$

F1-Score: The F1 Score may vary anywhere from 0 to 1, and it represents the harmonic mean of the accuracy and recall scores. It is a measurement of how precise the classifier is. Our model is considered to be more accurate when it has a higher F1 Score [39]. It may be expressed numerically as follows:

$$\text{F1-Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Precision: The term "precision" refers to the accuracy of the classifiers. It falls somewhere in the middle of the range [0 to 1].

Recall: The ability to recall information provides insight into the comprehensiveness of a classifier. It falls somewhere in the middle of the range [0 to 1].

Macro Averaged F1 Score Versus Accuracy: The macro averaged F1 score is the average of the F1 scores from each category; the number of samples in each class is ignored while calculating this average [40]. We intend to obtain comparable accuracy and a macro averaged F1 score while classifying unbalanced datasets, such that the accuracy represents the categorization for all classes.

A. DATASET MEASUREMENT

We measure the length analysis and sentiment analysis for the measurement of the dataset. To demonstrate the type and context of the application areas such as entertainment, politics, technologies etc. we have performed the length and sentiment analysis on the dataset. The lengths of each text data are not same and that gives a diversity, which portraits the real-world situations [41]. On the other hand, it is evident that sentiment analysis gives an insight of the data before the main exploratory analysis. Based on the histogram and density measurements the analysis are shown in the next sections.

1) LENGTH ANALYSIS

It is vital to consider the length of the text since a simple calculation may provide a wealth of information [42]. Perhaps we are fortunate enough to learn, for instance, that one category is consistently longer than another, in which case the length would be the only attribute required to create the model. Unfortunately, this won't be the case due to the identical lengths of news headlines, but it's worth a go. There are many length measurements for textual data. The most prevalent length measurements for text data are shown in Fig. 7.

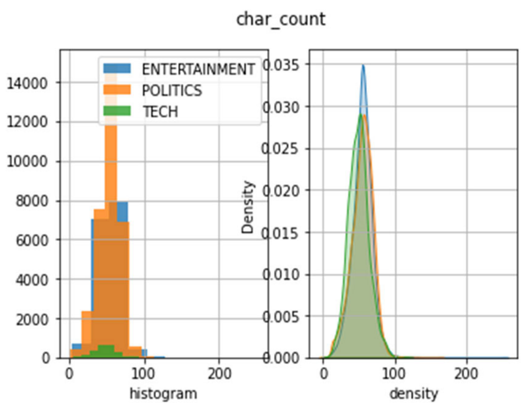

FIGURE 7. **Length analysis of the dataset.**

2) SENTIMENT ANALYSIS

Sentiment analysis is the numerical or categorical representation of the subjective feelings of text data. Due to the ambiguity of natural language, calculating sentiment is one of the most difficult NLP jobs. For instance, the statement "This is so horrible that it's wonderful" has several meanings. A neural sentiment might come from a model assigning a positive signal to the word "good" and a negative signal to the term "bad." This occurs as the context is uncertain.

The optimal strategy would be to train a sentiment model that is tailored to your data. When there is little time or data, one may use pre-trained models, such as Textblob and Vader. Textblob, which is developed on top of NLTK, is one of the most popular; it can assign polarity to words and calculate the average of the whole text. Vader (Valence aware dictionary and sentiment reasoned) is a rule-based model that performs especially effectively with data from the news media. The majority of the headlines are impartial, with the exception of the Politics news, which is slanted negatively and Tech news that has a spike on the positive tail. The following Fig. 8 depicts the sentiment analysis of the dataset.

3) COMPARISON OF THE RESULT

In our work, we consider the both imbalanced and balanced dataset then we compare the result along with our proposed model MCNN-LSTM.

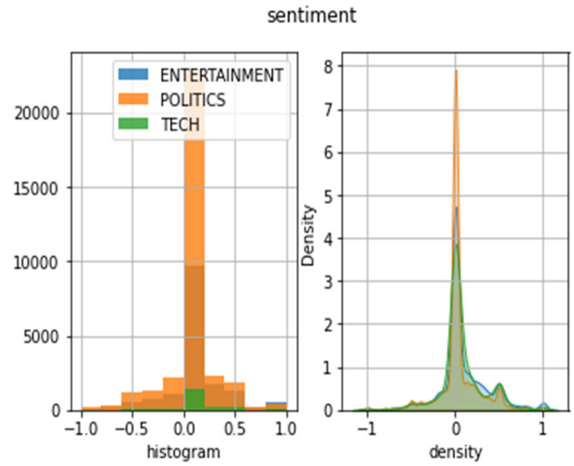

FIGURE 8. **Sentiment Analysis of the dataset.**

TABLE 2. **Results of LR, NB, DT, SVM, CNN, LSTM, MCNN-LSTM applied on imbalanced dataset.**

Categories	F1 Score and Accuracy						
	LR	NB	DT	SVM	CNN	LSTM	MCNN-LSTM
LIFESTYLE AND WELLNESS	76%	81%	87%	86%	85%	87%	**88%**
POLITICS	85%	91%	90%	88%	91%	91%	**94%**
SPORTS AND ENTERTAINMENT	87%	89%	85%	92%	94%	94%	**95%**
TOURISM & ART-CULTURE	88%	85%	86%	91%	88%	89%	**92%**
EMPOWERED VOICES	79%	81%	80%	85%	86%	85%	**93%**
PARENTING AND EDUCATION	83%	76%	83%	88%	88%	**91%**	89%
MISC	85%	84%	87%	89%	86%	92%	**94%**
GENERAL	79%	77%	79%	85%	**92%**	89%	88%
WORLDNEWS	82%	83%	86%	90%	92%	92%	**94%**
BUSINESS-MONEY	86%	84%	81%	86%	**95%**	88%	92%
SCIENCE AND TECH	91%	89%	93%	92%	94%	89%	**97%**
ENVIRONMENT	88%	89%	92%	94%	92%	**98%**	96%
ACCURACY	86.63%	88.68%	88.91%	89.75%	94%	92%	**95.4%**
MACRO AVG F1 SCORE	87%	86%	87%	90%	92%	90%	**95%**

In the above table 2, we compare the result at first with the imbalanced dataset in terms of F1 score result. From the table, we analyzed that our proposed model MCNN-LSTM outperformed the traditional machine learning: Logistic Regression (LR), Naïve Bayes (NB), Decision Tree (DT) and Support Vector Machine (SVM) in terms of all the categories of the dataset for imbalanced dataset.

Table 2 reveals that our model MCNN-LSTM (95.4%) outperforms current methods for finding balanced text data. The dataset's F1 scores show how effectively each category is recognized. The table below illustrates that both algorithms categorize an unbalanced sample effectively, and all categories have similar F1 scores. The macro averaged F1 scores 95% and accuracies do not fluctuate, showing that each class is correctly categorized.

We can see that our suggested MCNN-LSTM model outperformed the other standard machine learning model in terms of f1-score and accuracy, as shown in Fig. 7. The MCNN-LSTM model, shown by the light blue line, performs better on the skewed dataset than the other models.

TABLE 3. Results of LR, NB, DT, SVM, CNN, LSTM, MCNN-LSTM applied on balanced dataset.

Categories	F1 Score and Accuarcy						
	LR	NB	DT	SVM	CNN	LSTM	MCNN-LSTM
LIFESTYLE AND WELLNESS	92%	88%	92%	90%	94%	95%	**96%**
POLITICS	96%	97%	96%	94%	97%	97%	**98%**
SPORTS AND ENTERTAINMENT	95%	96%	94%	96%	94%	97%	**97%**
TOURISM & ART-CULTURE	93%	95%	96%	96%	96%	**98%**	96%
EMPOWERED VOICES	94%	88%	89%	91%	95%	97%	**95%**
PARENTING AND EDUCATION	96%	98%	99%	94%	95%	97%	**96.65 %**
MISC	92%	89%	90%	94%	**96%**	95%	95%
GENERAL	91%	96%	95%	97%	94%	97%	**97%**
WORLDNEWS	93%	97%	98%	97%	96%	98%	**99.14 %**
BUSINESS-MONEY	93%	95%	93%	96%	94%	98%	**97%**
SCIENCE AND TECH	93%	**98%**	**96%**	97%	**98%**	96%	95%
ENVIRONMENT	95%	95%	95%	94%	96%	97%	**97%**
Accuracy	94.70' %	97.40 %	96.60%	97.60 %	97%	96%	**99.71 %**
Macro Avg F1 Score	98%	97%	97%	98%	97%	98%	**98%**

In the above table 3, we compare the result at first with the balanced dataset in terms of F1 score and accuracy result. From the table, we analyzed that our proposed model MCNN-LSTM outperformed the traditional machine learning: Logistic Regression (LR), Naïve Bayes (NB), Decision Tree (DT) and Support Vector Machine (SVM) in terms of all the categories of the dataset for imbalanced dataset. Table 3 shows the results of our study, which show that when it comes to identifying balanced text data, standard techniques are beaten by our proposed model MCNN-LSTM (98.65%). When we compare the F1 scores of every sample in the dataset, we can see how well each category is identified. Both algorithms do well, as shown in the table below when categorizing a balanced sample, and all categories have comparable F1 scores. For comparing multiple algorithms or situations, the F1-scores and accuracy has been used evidently in many researches including personality prediction [43], eye disease prediction [44], predicting the sentiment trends and emotion patterns [45], etc.

From the following Fig. 9 and Fig. 10, we visualize that our proposed model MCNN-LSTM mostly outperformed the other traditional machine learning model in terms of f1-score and accuracy. The light blue color line denotes the MCNN-LSTM model which outperforms the other model in different color in terms of balanced dataset.

TABLE 4. Results of LR, NB, DT, SVM, CNN, LSTM, MCNN-LSTM applied on imbalanced dataset.

Categories	F1 Score/Accuracy	
	Score archived by proposed MCNN-LSTM is Higher than LR, NB, DT, SVM, CNN and LSTM	
	YES	NO
Lifestyle and wellness	✓ (88.00 %)	
Politics	✓ (94.00%)	
Sports and entertainment	✓ (95.00 %)	
Tourism & art-culture	✓ (92.00%)	
Empowered voices	✓ (93.00%)	
Parenting and education		✓
Misc	✓ (94.00%)	
General		✓
Worldnews	✓ (94.00%)	
Business-money		✓
Science and tech	✓ (97.00%)	
Environment		✓

TABLE 5. Results of LR, NB, DT, SVM, CNN, LSTM, MCNN-LSTM applied on balanced dataset.

Categories	F1 Score/Accuracy	
	Score achieved by proposed MCNN-LSTM is Higher than LR, NB, DT, SVM, CNN and LSTM	
	YES	NO
Lifestyle and wellness	✓ (96.00%)	
Politics	✓ (98.00%)	
Sports and entertainment	✓ (97.00%)	
Tourism & art-culture		✓
Empowered voices	✓ (95.00%)	
Parenting and education	✓ (96.65%)	
Misc		✓
General	✓ (97.00%)	
Worldnews	✓ (99.14%)	
Business-money	✓ (97.00%)	
Science and tech		✓
Environment	✓ (97.00%)	

In the above table 4 and 5, we visualize that in case of balanced dataset our proposed MCNN-LSTM model has

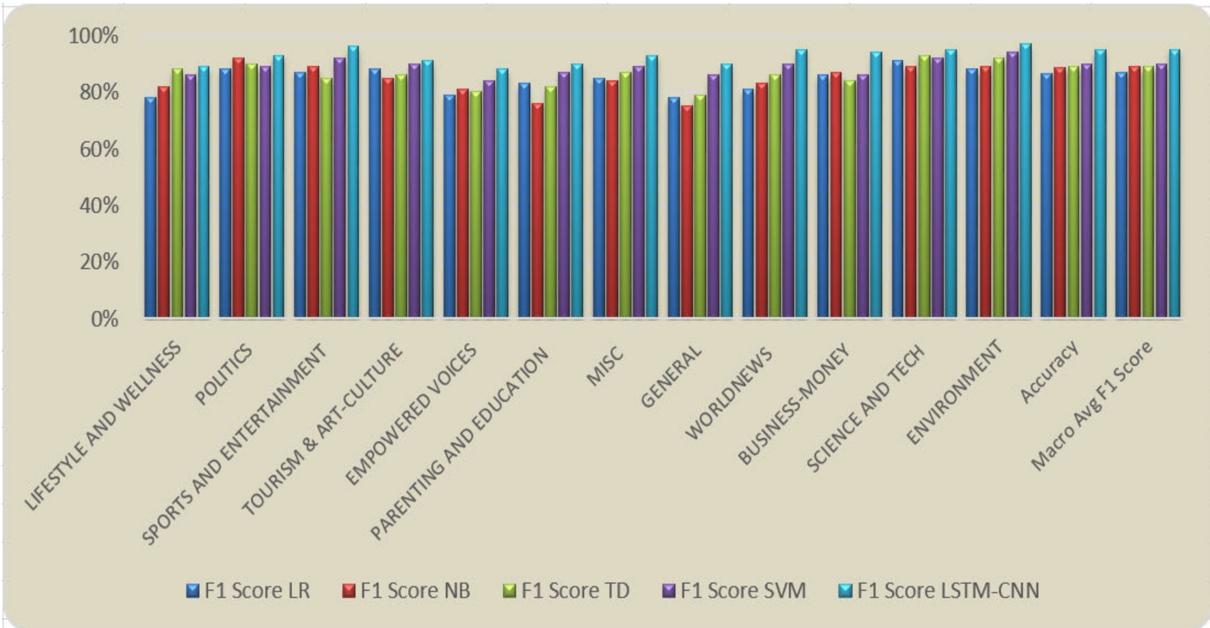

FIGURE 9. F1 Score comparison for the imbalanced dataset.

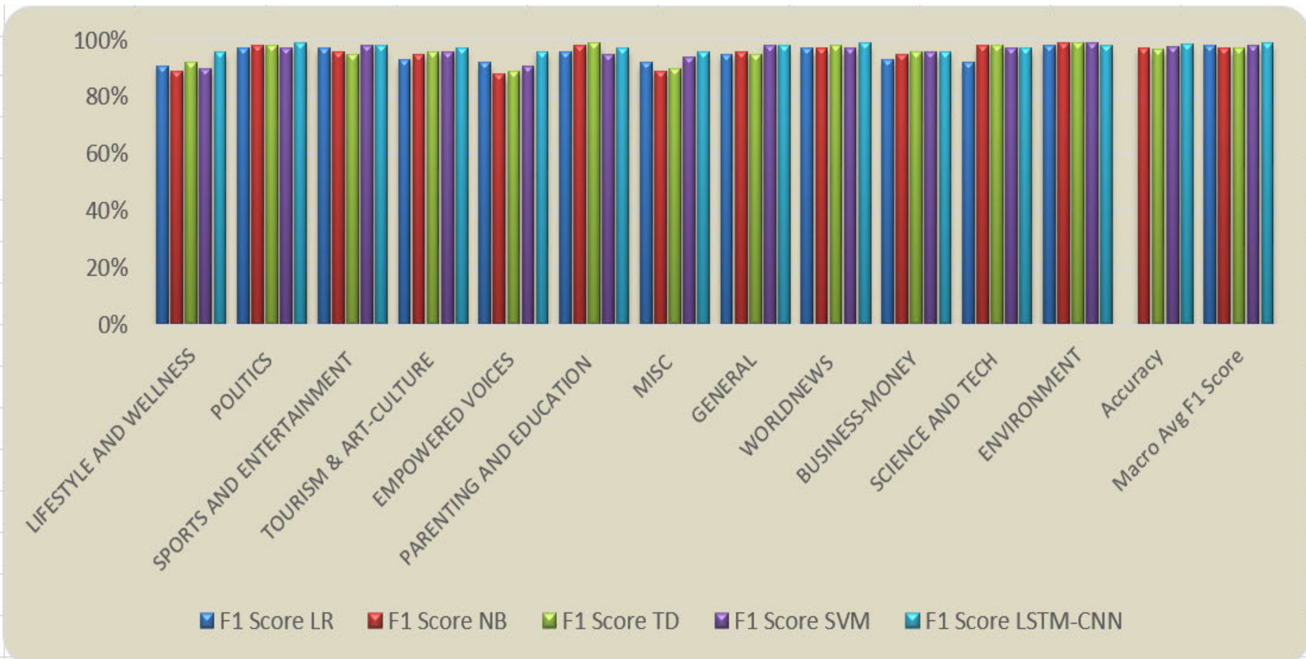

FIGURE 10. F1 Score comparison for the balanced dataset.

achieved better result in 9 categories while for the imbalanced dataset we get better result in 7 categories comparing with traditional machine learning algorithms which denote that our proposed model performs better. Fig. 11 depicts the accuracy comparison for the balanced dataset.

We applied Mean squared error (MSE), Log loss (LL) and Cohen's kappa Score in terms of get the robustness of our proposed model.

a: MEAN SQUARED ERROR (MSE)

Mean squared error (MSE) [39] is typically regarded as the most essential metric for evaluating various machine learning and deep learning algorithms, including our proposed model MCNN-LSTM. The lowest error which was approximately 0.5% is observed with the help of our proposed MCNN-LSTM approach, while the highest error rate has come from the SVM algorithm (just over 4%). Surprisingly, CNN and

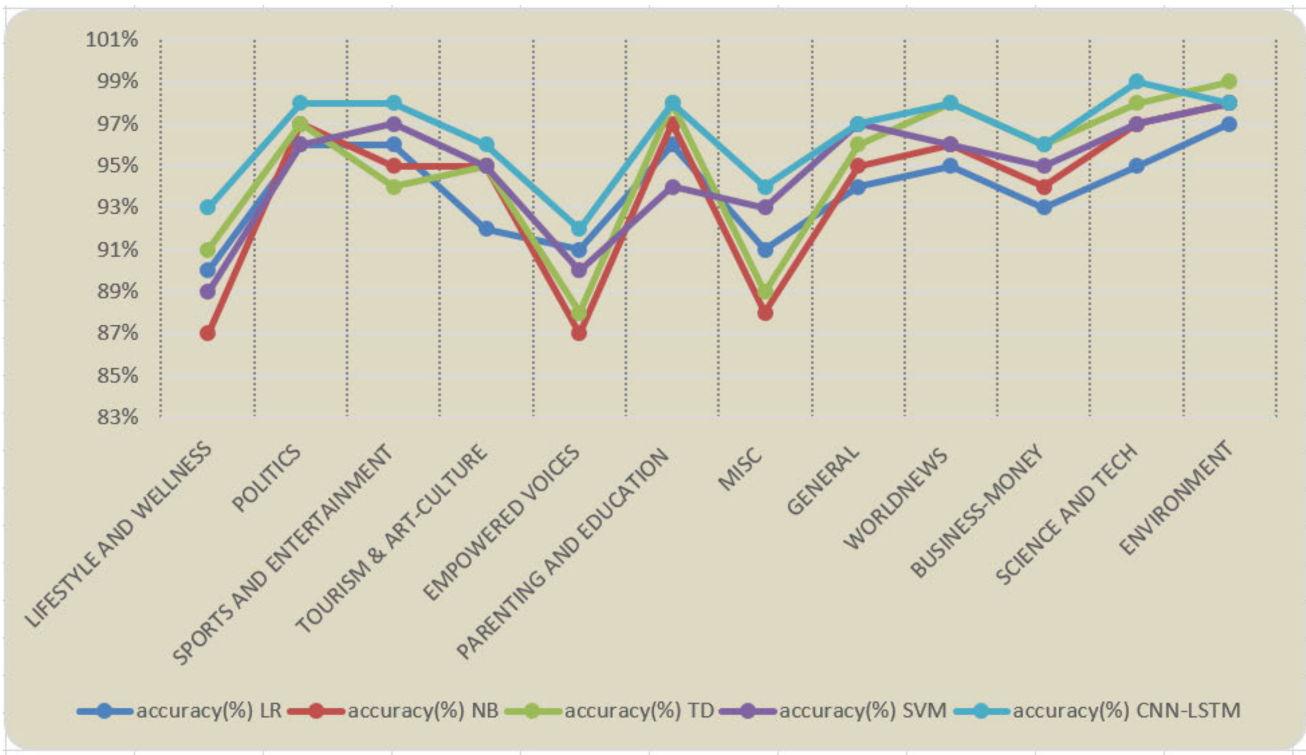

FIGURE 11. Accuracy comparison for the balanced dataset.

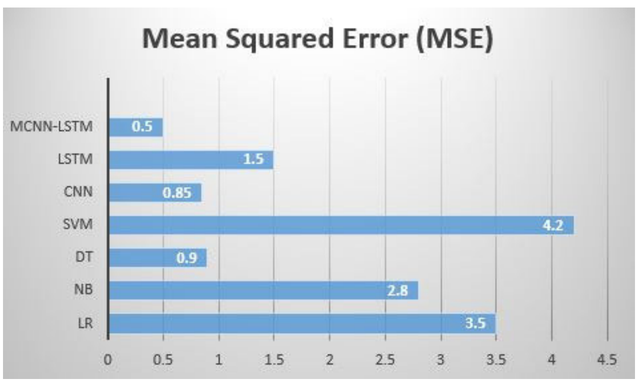

FIGURE 12. Means Squared Error (MSE).

FIGURE 13. Log Loss (LL).

DT (just under 1%) generate similar MSE values. Though, LSTM scores 1.5% and NB gives 2.8% MSE value accordingly. Furthermore, LR is the only method which displays a higher MSE value similar to the SVM technique (Fig. 12).

b: LOG LOSS

The lower the log-loss, the more closely the predicted value matches the genuine one (0 or 1 in case of binary classification) [40]. The log-loss value increases as the degree of discrepancy between the expected and observed probabilities grows. Fig. 13 displays the log loss (LL) for the models we applied along with our proposed one. The proposed MCNN-LSTM model produces the lowest LL value of 0.5 along with CNN with the same value, while the highest LL value is 4.5 of NB. While LSTM produces 1.5 and SVM, DT and LR gives the output of 2.5, 2.3 and 3.2 accordingly.

c: COHEN'S KAPPA SCORE

Cohen's Kappa is a statistical metric for determining the level of agreement between two raters of the same quantity [41]. From Fig. 14, we can assure that our proposed model MCNN-LSTM achieves the highest value of 1 while the lowest one come from SVM with 0.81. Again, CNN and LSTM provides 0.94 and 0.92 accordingly. In terms of traditional machine learning algorithms, we analysis that

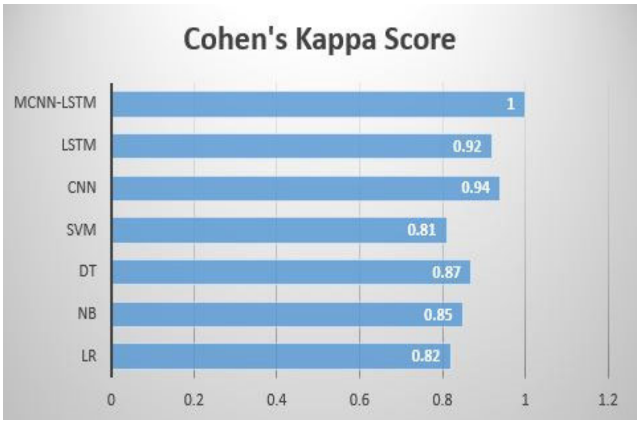

FIGURE 14. Cohen's kappa score.

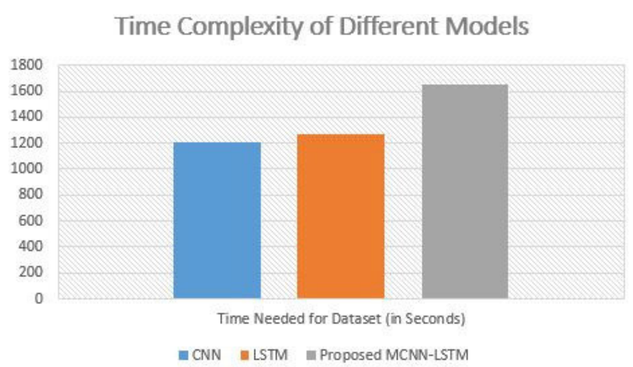

FIGURE 15. Time complexity of each model based on chosen parameters.

DT produces the highest one with 0.87 while NB and LR gives 0.85 and 0.82 respectively.

From the above statistical analysis, we assure that our proposed model MCNN-LSTM produces the best outcomes in terms of justification.

In this table 6, we compare our model result with the other model. In [42], the author used an imbalanced data of Indonesian News Title where their best method RoBERTa gave 99.52% accuracy while in their dataset. Also in terms of [57] and [58] works our model outperformed them with an accuracy of 99.65% which prove the robustness of our proposed model.

To be able to reproduce the same outcomes, we have shown the machine learning parameters and hyper parameters needed in this proposed approach in table 7. All the machine learning models used in this paper are developed in python using sklearn or keras packages. The necessary default parameters and hyper parameter could be utilized from the python packages. Therefore, the information of the hyper parameters used are provided.

Randomly selecting hyper parameter values instead of grid searching may provide better models faster. The specifics of the deep learning model, together with the suggested MCNN-LSTM, are shown in table 8. When it came to optimizers, we utilized Adam [50] for all of the models. As for activation functions, we used relu [51] for CNN and LSTM, and we used LeakyRelu [52] for our suggested technique. The table allowed us to do further analysis, which revealed that CNN and LSTM each had a total of two hidden levels, but the solution that we have developed has four hidden layers. The learning rate determines how big a number ought to be [53], and for our suggested model, we picked 0.0005 as the learning rate. Once again, the batch size for CNN and LSTM is 32, while the batch size for the suggested model is 50. Epoch's size for our suggested technique is 80, whereas the corresponding values for CNN and LSTM are 50 and

TABLE 6. Comparison of accuracy with some existing literature.

Paper	Dataset for Classification	Best Method Accuracy
Muliono et. al. [42]	Indonesian News Title	RoBERTa – (Accuracy: 99.52%)
Surolia et al. [57]	PoliEMO Indian News Dataset	Bi-LSTM (Accuracy: 68%)
Ge et al. [58]	Chinese news text	CNN (Accuracy: 97.84%)
Proposed method	Collected from HuffPost	MCNN – LSTM (Accuracy: 99.65%)

TABLE 7. Hyper parameters for the machine learning approaches.

Machine Learning Approach	Instrument/tool used	Information of the Hyperparameters
LR	Python sklearn	sklearn.linear_model.LogisticRegression [46]
NB	Python sklearn	sklearn.naive_bayes [47]
DT	Python sklearn	sklearn.tree.DecisionTreeClassifier [48]
SVM	Python sklearn	sklearn.svm.SVC [49]

TABLE 8. Hyperparameters of the deep learning models along with proposed MCNN-LSTM.

Model	No. of Hidden Layers	Optimizer	Activation Function	No. of Epochs	Learning Rate
CNN	2	Adam	Relu	50	0.001
LSTM	2	Adam	Relu	60	0.002
Porposed MCNN - LSTM	**4**	**Adam**	**LeakyRelu**	**80**	**0.0005**

TABLE 9. Environmental setup.

Model	intel i7-9750H
Processor	@3.20GHz processor
Ram	16 GB
Graphics Card	GeForce GTX 1650 NVIDIA GPU

TABLE 10. Time taken by the deep learning models along with proposed MCNN-LSTM on Dataset.

Model	No. of convolutional & fully connected layers	Time taken on Dataset
CNN	2C+1F	1206
LSTM	2C+2F	1273
Proposed MCNN-LSTM	**3C+2F**	**1652**

60, respectively. The setup has been completed and the implementation has been carried out on the system shown in table 9.

Table 10 displays the actual amount of time, in seconds, that various models took to process the HuffPost dataset on the computer system that was constructed according to table 10. Each of the models is run for a total of 20 epochs to get the real time again.

Table 9 shows that the model's temporal complexity grows in tandem with the number of convolution layers and fully linked ones. The time it takes to execute a model depends on a variety of factors, including the complexity of the model and the number and size of filters used. Therefore, it shown that the temporal complexity of the model is dependent not only on the number of convolution layers but also on the number of filters and the size of the filters. Running the LSTM model took a little longer than the previous CNN model, but only by about 5% to 10%, demonstrating that the increasing number of fully connected layers affects only the overall time complexity. The LSTM model has the same number of convolution layers as previous models, but 2 fully connected layers. Our suggested model is the most time-consuming since it has 3 convolution and 3 fully linked layers.

The amount of time (in seconds) spent by each model is shown in Fig. 15, and it is dependent on the parameters.

VI. LIMITATIONS AND CHALLENGES

The research is carried out with the use of the huge dataset for the first time that are accessible to the public. The deep learning model that was suggested is put through its paces by being applied to this datasets in order to assess its effectiveness. It can be seen from the recorded performance metrics that the new model has a consistent level of accuracy and a high level of accuracy when it is applied to this large number of dataset. In addition to this, the model is able to generate the results in the shortest period of time possible.

The limitation of the proposed system are as follows:

In this work, the transfer learning is not applied along with the machine learning and deep learning model.

- Different newspaper datasets are not compared here. Only Indonesian newspaper dataset is compared.
- In terms of balancing dataset with the help of Tomek-Link algorithm, the result gives very good performances though there is a slight number of imbalance data.
- The per epoch time for a dense layer is greater than the per epoch time of the convolution layer, fully connected layer. To get the good result, we vary the epoch size in terms of the deep learning model along with proposed model.
- There is a number of different features and optimization techniques available which is not applied here in the model.

The challenges of the proposed system are as follows:

- Dealing with highly imbalanced and noisy dataset.
- Finding the best approach to design a model that can analyze imbalanced dataset.
- Finding the appropriate preprocessing step convenient to our work.
- Finding the appropriate preprocessing step convenient to our work.
- Selecting the correct model to classify our data.

VII. CONCLUSION

When it comes to supervised learning, content categorization is a crucial component of several information processing approaches. The efficiency of traditional supervised algorithms has decreased as a result of an increase in the number of records to be processed. The majority of high-performance algorithms are slowed down when presented with datasets that are not balanced. We have come up with a one-of-a-kind model that we call the multi-class MCNN-LSTM. This model combines a convolutional neural network (CNN) with an extended short-term memory network (LSTM) and has the ability to learn several different classes. The MCNN-LSTM is able to learn phrase-level features thanks to a convolutional layer, and these characteristics are then fed into an LSTM so that it can learn long-term dependencies. The dataset is first brought into balance using the Tomek-Link method before the suggested model, which is superior to the work that came before it, is used. According to the findings of this research, it performs far better than other machine learning algorithms when it comes to text classification. Investigating the impact that deep neural networks have on datasets acquired from the actual world with the intention of achieving sustainable development goals is likely to be the next step in the future. Utilizing transfer learning, the many feature extraction methods that are compatible with deep neural networks (such as principal component analysis, or PCA), and other variants of neural network models are part of the ongoing research in this area.

REFERENCES

[1] H. Z. Mauni, T. Hossain, and R. Rab, "Classification of underrepresented text data in an imbalanced dataset using deep neural network," in *Proc. IEEE Region Symp. (TENSYMP)*, Jun. 2020, pp. 997–1000.

[2] E. Letouzé, "Big data for development: Challenges and opportunities," UN, New York, NY, USA, Tech. Rep., May 2012.

[3] G. Song, M. Jia, K. Chen, X. Kong, B. Khattak, C. Xie, A. Li, and L. Mao, "CRISPR/cas9: A powerful tool for crop genome editing," *Crop J.*, vol. 4, no. 2, pp. 75–82, Apr. 2016.

[4] J. P. Banâtre, P. Fradet, and Y. Radenac, "Principles of chemical programming," *Electron. Notes Theor. Comput. Sci.*, vol. 124, no. 1, pp. 133–147, 2005.

[5] E. A. Ashley, "Towards precision medicine," *Nature Rev. Genet.*, vol. 17, no. 9, pp. 507–522, Sep. 2016.

[6] K. Md. Hasib, F. Rahman, R. Hasnat, and Md. G. R. Alam, "A machine learning and explainable AI approach for predicting secondary school student performance," in *Proc. IEEE 12th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2022, pp. 0399–0405.

[7] I. K. Sastrawan, I. P. A. Bayupati, and D. M. S. Arsa, "Detection of fake news using deep learning CNN–RNN based methods," *ICT Exp.*, vol. 8, no. 3, pp. 396–408, Sep. 2022.

[8] A. Borrego, D. Dessì, I. Hernández, F. Osborne, D. R. Recupero, D. Ruiz, D. Buscaldi, and E. Motta, "Completing scientific facts in knowledge graphs of research concepts," *IEEE Access*, vol. 10, pp. 125867–125880, 2022.

[9] C. Padurariu and M. E. Breaban, "Dealing with data imbalance in text classification," *Proc. Comput. Sci.*, vol. 159, pp. 736–745, Jan. 2019.

[10] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[11] S. Montaha, S. Azam, A. K. M. R. H. Rafid, Md. Z. Hasan, A. Karim, and A. Islam, "TimeDistributed-CNN-LSTM: A hybrid approach combining CNN and LSTM to classify brain tumor on 3D MRI scans performing ablation study," *IEEE Access*, vol. 10, pp. 60039–60059, 2022.

[12] J. Tian, S. Chen, X. Zhang, and Z. Feng, "BSIL: A brain storm-based framework for imbalanced text classification," in *Proc. CCF Int. Conf. Natural Lang. Process. Chin. Comput.* Cham, Switzerland: Springer, 2019, pp. 53–64.

[13] S. Redmond and E. Rozaki, "Using bipartite graphs projected onto two dimensions for text classification," in *Proc. 5th Int. Conf. Adv. Comput., Commun. Inf. Technol. (CCIT)*, Zürich, Switzerland, Sep. 2017, pp. 55–59.

[14] M. Damaschk, T. Donicke, and F. Lux, "Multiclass text classification on unbalanced, sparse and noisy data," in *Proc. 1st NLPL Workshop Deep Learn. Natural Lang. Process.*, 2019, pp. 58–65.

[15] Y. Li, H. Guo, Q. Zhang, M. Gu, and J. Yang, "Imbalanced text sentiment classification using universal and domain-specific knowledge," *Knowl.-Based Syst.*, vol. 160, pp. 1–15, Nov. 2018.

[16] Y. Zhou, Y. Li, and S. Xia, "An improved KNN text classification algorithm based on clustering," *J. Comput.*, vol. 4, no. 3, pp. 230–237, Mar. 2009.

[17] S. Ahlawat and A. Choudhary, "Hybrid CNN-SVM classifier for handwritten digit recognition," *Proc. Comput. Sci.*, vol. 167, pp. 2554–2560, Jan. 2020.

[18] S. Montaha, S. Azam, A. K. M. R. H. Rafid, P. Ghosh, M. Z. Hasan, M. Jonkman, and F. De Boer, "BreastNet18: A high accuracy fine-tuned VGG16 model evaluated using ablation study for diagnosing breast cancer from enhanced mammography images," *Biology*, vol. 10, no. 12, p. 1347, Dec. 2021.

[19] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*.

[20] S. Zobaed, F. Rabby, I. Hossain, E. Hossain, S. Hasan, A. Karim, and K. M. Hasib, "DeepFakes: Detecting forged and synthetic media content using machine learning," in *Artificial Intelligence in Cyber Security: Impact and Implications*. Cham, Switzerland: Springer, 2021, pp. 177–201.

[21] M. Shi, K. Wang, and C. Li, "A C-LSTM with word embedding model for news text classification," in *Proc. IEEE/ACIS 18th Int. Conf. Comput. Inf. Sci. (ICIS)*, Jun. 2019, pp. 253–257.

[22] K. M. Hasib, N. A. Towhid, and M. R. Islam, "HSDLM: A hybrid sampling with deep learning method for imbalanced data classification," *Int. J. Cloud Appl. Comput.*, vol. 11, no. 4, pp. 1–13, Oct. 2021.

[23] A. Karim, S. Azam, B. Shanmugam, and K. Kannoorpatti, "Efficient clustering of emails into spam and ham: The foundational study of a comprehensive unsupervised framework," *IEEE Access*, vol. 8, pp. 154759–154788, 2020.

[24] K. M. Hasib, M. S. Iqbal, F. M. Shah, J. Al Mahmud, M. H. Popel, M. I. H. Showrov, S. Ahmed, and O. Rahman, "A survey of methods for managing the classification and solution of data imbalance problem," 2020, *arXiv:2012.11870*.

[25] J. Grover and R. Misra, "Sculpting data for ML: The first act of machine learning," in *Sculpting Data for ML: The First Act of Machine Learning*, 1st ed. Grover & Misra, Jan. 2021.

[26] I. Tomek, "Two modifications of CNN," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 11, pp. 769–772, Nov. 1976.

[27] S. Kumar, D. Kumar, P. K. Donta, and T. Amgoth, "Land subsidence prediction using recurrent neural networks," *Stochastic Environ. Res. Risk Assessment*, vol. 36, no. 2, pp. 373–388, Feb. 2022.

[28] M. Beckmann, N. F. F. Ebecken, and B. S. L. Pires de Lima, "A KNN undersampling approach for data balancing," *J. Intell. Learn. Syst. Appl.*, vol. 7, no. 4, pp. 104–116, 2015.

[29] K. M. Hasib, M. I. H. Showrov, J. Al Mahmud, and K. Mithu, "Imbalanced data classification using hybrid under-sampling with cost-sensitive learning method," in *Edge Analytics*. Singapore: Springer, 2022, pp. 423–435.

[30] S. Bird, "NLTK: The natural language toolkit," in *Proc. COLING/ACL Interact. Presentation Sessions*, 2006, pp. 69–72.

[31] L. Ge and T.-S. Moh, "Improving text classification with word embedding," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 1796–1805.

[32] S. Moriya and C. Shibata, "Transfer learning method for very deep CNN for text classification and methods for its evaluation," in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 2, Jul. 2018, pp. 153–158.

[33] G. Mujtaba, L. Shuib, R. G. Raj, and R. Gunalan, "Detection of suspicious terrorist Emails using text classification: A review," *Malaysian J. Comput. Sci.*, vol. 31, no. 4, pp. 271–299, Oct. 2018.

[34] J. R. M. Navin and R. Pankaja, "Performance analysis of text classification algorithms using confusion matrix," *Int. J. Eng. Tech. Res. (IJETR)*, vol. 6, no. 4, pp. 75–78, 2016.

[35] X. Tong, B. Wu, S. Wang, and J. Lv, "A complaint text classification model based on character-level convolutional network," in *Proc. IEEE 9th Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Nov. 2018, pp. 507–511.

[36] L. Cai and T. Hofmann, "Hierarchical document categorization with support vector machines," in *Proc. 13th ACM Int. Conf. Inf. Knowl. Manag.*, Nov. 2004, pp. 78–87.

[37] Y. Yang, L. Zheng, J. Zhang, Q. Cui, Z. Li, and P. S. Yu, "TI-CNN: Convolutional neural networks for fake news detection," 2018, *arXiv:1806.00749*.

[38] K. Md. Hasib, A. Tanzim, J. Shin, K. O. Faruk, J. A. Mahmud, and M. F. Mridha, "BMNet-5: A novel approach of neural network to classify the genre of Bengali music based on audio features," *IEEE Access*, vol. 10, pp. 108545–108563, 2022.

[39] K. M. Hasib, N. A. Towhid, K. O. Faruk, J. Al Mahmud, and M. F. Mridha, "Strategies for enhancing the performance of news article classification in Bangla: Handling imbalance and interpretation," *Eng. Appl. Artif. Intell.*, vol. 125, Oct. 2023, Art. no. 106688.

[40] M. Z. Naser and V. K. Kodur, "Explainable machine learning using real, synthetic and augmented fire tests to predict fire resistance and spalling of RC columns," *Eng. Struct.*, vol. 253, Feb. 2022, Art. no. 113824.

[41] S. Erdoğan and S. S. Döven, "Evaluation of platelet indices in children with renal scarring based on diagnostic accuracy criteria and Cohen's Kappa," *Pediatrics Int.*, vol. 64, no. 1, Jan. 2022, Art. no. e15055.

[42] Y. Muliono, F. L. Gaol, B. Soewito, and H. L. H. S. Warnars, "Hoax classification in imbalanced datasets based on Indonesian news title using RoBERTa," in *Proc. 3rd Int. Conf. Artif. Intell. Data Sci. (AiDAS)*, Sep. 2022, pp. 264–268, doi: 10.1109/AiDAS56890.2022.9918747.

[43] A. A. Marouf, Md. K. Hasan, and H. Mahmud, "Comparative analysis of feature selection algorithms for computational personality prediction from social media," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 3, pp. 587–599, Jun. 2020, doi: 10.1109/TCSS.2020.2966910.

[44] A. A. Marouf, M. M. Mottalib, R. Alhajj, J. Rokne, and O. Jafarullah, "An efficient approach to predict eye diseases from symptoms using machine learning and ranker-based feature selection methods," *Bioengineering*, vol. 10, no. 1, p. 25, Dec. 2022, doi: 10.3390/bioengineering10010025.

[45] A. A. Marouf, J. G. Rokne, and R. Alhajj, "Detecting and understanding sentiment trends and emotion patterns of Twitter users—A study on the demise of a bollywood celebrity," *Big Data Cognit. Comput.*, vol. 6, no. 4, p. 129, Oct. 2022, doi: 10.3390/bdcc6040129.

[46] *Python Scikit Learn Logistic Regression Parameters*. Accessed: Apr. 2, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[47] *Python Scikit Learn Naïve Bayes Parameters*. Accessed: Mar. 21, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html

[48] *Python Scikit Learn Decision Tree Parameters*. Accessed: Apr. 6, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[49] *Python Scikit Learn Support Vector Machine Parameters*. Accessed: Apr. 6, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.SVM.SVC.html

[50] M. Liu, D. Yao, Z. Liu, J. Guo, and J. Chen, "An improved Adam optimization algorithm combining adaptive coefficients and composite gradients based on randomized block coordinate descent," *Comput. Intell. Neurosci.*, vol. 2023, pp. 1–14, Jan. 2023.

[51] C. Chen, F. Min, Y. Zhang, and H. Bao, "ReLU-type Hopfield neural network with analog hardware implementation," *Chaos, Solitons Fractals*, vol. 167, Feb. 2023, Art. no. 113068.

[52] D. V. Dung, N. D. Song, P. S. Palar, and L. R. Zuhal, "On the choice of activation functions in physics-informed neural network for solving incompressible fluid flows," in *Proc. AIAA SCITECH Forum*, Jan. 2023, p. 2023.

[53] J. Lu, "AdaSmooth: An adaptive learning rate method based on effective ratio," in *Sentiment Analysis and Deep Learning*. Singapore: Springer, 2023, pp. 273–293.

[54] K. Abbas, M. K. Hasan, A. Abbasi, U. A. Mokhtar, A. Khan, S. N. H. S. Abdullah, S. Dong, S. Islam, D. Alboaneen, and F. R. A. Ahmed, "Predicting the future popularity of academic publications using deep learning by considering it as temporal citation networks," *IEEE Access*, vol. 11, pp. 83052–83068, 2023.

[55] A. Khan, J. P. Li, N. Ahmad, S. Sethi, A. U. Haq, S. H. Patel, and S. Rahim, "Predicting emerging trends on social media by modeling it as temporal bipartite networks," *IEEE Access*, vol. 8, pp. 39635–39646, 2020.

[56] A. Khan, J. P. Li, and M. A. Husain, "Power grid stability analysis using pipeline machine," *Multimedia Tool. Appl.*, vol. 82, pp. 25651–25675, Feb. 2023.

[57] A. Surolia, S. Mehta, and P. Kumaraguru, "Understanding emotions: A PoliEMO datasetand multi-label classification in Indian elections," Res. Square, 2023, doi: 10.21203/rs.3.rs-2694319/v1.

[58] B. Ge, C. He, H. Xu, J. Wu, and J. Tang, "Chinese news text classification method via key feature enhancement," *Appl. Sci.*, vol. 13, no. 9, p. 5399, Apr. 2023.