International Journal of Engineering in Computer Science

E-ISSN: 2663-3590
P-ISSN: 2663-3582
www.computersciencejournals.com/ijecs
IJECS 2025; 7(1): 32-39
Received: 17-11-2024
Accepted: 25-12-2024

N John Kuotsu
Department of Computer Science, Fazl Ali College, Mokokchung, Nagaland, India

Corresponding Author:
N John Kuotsu
Department of Computer Science, Fazl Ali College, Mokokchung, Nagaland, India

A hybrid deep learning and tree-based model for enhanced sentiment analysis of IMDB movie reviews

N John Kuotsu

DOI: https://doi.org/10.33545/26633582.2025.v7.i1a.153

Abstract
Sentiment analysis (or opinion mining), is a natural language processing (NLP) technique that harnesses Information from text data that is naturally subjective. Opinions, emotions, and attitudes are things that this information can contain. In recent years, there has been an increasing focus on sentiment analysis because of the explosive growth of unstructured data on the internet. Sentiment analysis tasks have seen state of the art performance using deep learning models. However, these models can be computationally intensive and high dimensional can produce problems of overfitting when working with relatively limited data. In order to overcome these limitations, a new hybrid model is proposed using a combination of LSTM network for feature extraction and Random Forest classifier for the final sentiment classification improving the accuracy of the sentiment analysis, while reducing computational cost. In this paper, the proposed hybrid model is designed and implemented, and evaluated on a publicly available dataset of movie reviews.

Keywords: NLP, sentiment analysis, feature extraction, opinion mining, IMDB reviews dataset LSTM, random forest

Introduction: Sentiment analysis: Sentiment analysis aka opinion mining is studying the people's views, feelings, emotions, assessments, and attitudes towards entities (products, services, organization, individual, issues, events, themes, their properties) in the computer world [1]. In the field's genesis and quick expansion, we also have an unprecedented expanse of opinioned material in a digital format-that is if we consider reviews, blogs, forum debates, Twitter, microblogs, reviews and social networks as the epitome of what social media could be. Since its creation in the early 2000s (NLP), sentiment analysis has become one of the most active fields of study.
This is investigated in the areas of data mining, text mining, web mining and information retrieval. Due to its significances in business and society as a whole, it has shifted from computer science to social sciences and management sciences, like finance, communications, marketing, political science, health science and even history. Opinions are fundamental to practically all human actions and are key influencers of our behavior, which explains their prevalence. The way we understand and interact with the world, and the choices we make, can be significantly shaped by the perspectives and interpretations of others. It is common for individuals and businesses to seek advice from others when making decisions. One way to understand the emotional tone of text is through sentiment analysis, a process that involves identifying positive or negative sentiments. This technique is frequently employed by businesses to analyze social data, assess brand perception, and understand customer behavior.

Types of Sentiment Analysis
The scope of sentiment analysis encompasses the polarity (positive, negative, or neutral) of a text, as well as the identification of specific emotions and moods (such as anger, joy, or sadness), the level of urgency (urgent or not urgent), and even the intentions (interested versus not interested) of the writer. The accompanying figure illustrates the diverse categories of sentiment analysis [2].

a) **Emotion detection:** Sentiment analysis which detects emotion is way beyond the polarity analysis in identifying emotion such Happy, frustrated, rage, sad and more.
b) **Aspect-based Sentiment Analysis:** When you're analyzing text sentiments,

you'll most likely want to understand the features or qualities that people are referring to in a positive, neutral or negative way.

c) **Fine Grained Sentiment Analysis:** For higher accuracy in polarity classification for your business this may be helpful to expand the polarity categories to include additional levels of positive and negative sentiment, perhaps Very Positive, Positive, Neutral, Negative, and Very Negative.

d) **Multilingual sentiment analysis:** It's hard to do sentiment analysis in multiple languages. A significant amount of post processing and resources are required. Many resources of sentiment lexicons are available online, while resources such as translated corpora may require some noise detection techniques work. These resources are, however, only useful when you have proficiency in coding.

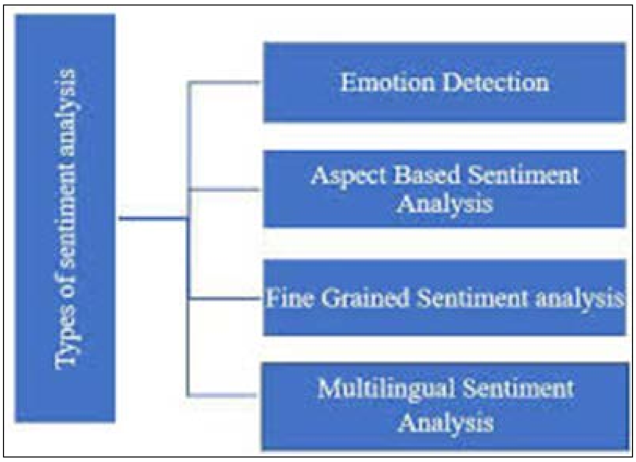

Fig 1: Types of Sentiment Analysis

Applications of sentiment analysis
There are so many ways to use sentiment analysis in just so many different industries: finance, retail, hospitality, technology. A list of common ways sentiment analysis is utilized in business is given below:

Social Media Monitoring: For example, sentiment analysis is used in social media monitoring to gain insights on how customers react to certain, as well as urgent topics, especially in real time.

Brand Monitoring: For marketers, there is ample information at their fingertips not only on social media but in the form of news sites, blogs, forums, product reviews, and so much more. In addition to looking at the total number of mentions, they could review the quality of each single mention, as well as the quality of the whole.

Voice of Customer: while social media and brand monitoring goldmines give us real time unfiltered data about customer sentiment, we can also apply this analysis to surveys as well as our customer support conversations.

Market Research
Sentiment analysis helps in all forms of market research and competition analysis. It helps whether someone is trying to enter a niche market, predict the future, or gaining an edge, sentiment analysis can help.

Problem Statement
Social network websites like Facebook, WhatsApp, Twitter, Hike messenger and LinkedIn provides the users to share their opinions and feelings by post photos and videos. These reviews must be summarized or must be confirmed whether these are good or bad. For knowledge extraction, analyzing, and decision making, machine learning algorithms must be used to build the model to understand. To find the scores of the opinions and derive conclusions, framework is proposed. Opinion mining is the classification of opinions, whereas sentiment analysis derives the scores for the opinions.

There are several approaches to sentiment analysis, including lexicon-based methods, machine learning-based methods, and hybrid methods. Methods such as Lexicon-based rely on predefined dictionaries of words and their associated sentiment values. While simple and efficient, these methods are not capable of handling intricacies of natural language and are restricted to simple language constructs.

However, Machine learning based methods tend to require training a model on a labelled text and a sentiment label dataset. Sentiment of unseen text data is then predicted through the use of the model.

Sentiment analysis tasks have already reached state of the art by using deep learning models, particularly Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). Although these models are computationally expensive and potential overfitting arises when working with small data.

Proposed System
A sentiment analysis model is proposed for movie reviews, which is based on a deep learning model and a tree-based model. Based on a dataset of 25,000 labeled movie reviews, it is trained and evaluated on 25,000 labeled review testing set. This is then used as input to the learning model to generate the feature, which is then trained on the dataset with labels to train a Random Forest model. The trained model is used to predict sentiment labels for each review in the testing set and the accuracy of the model is computed with respect to these predicted labels.

Here, the approach using deep learning and tree-based models for such system has the potential to have better accuracy in sentiment analysis than traditional methods.

Objectives
The goal of this work is to offer a hybrid model of deep learning and tree-based models, which can contribute to the accuracy of sentiment analysis while reducing computational cost. The purpose is to attain the following goals:

- Build a Sentiment Analysis hybrid model which is the combination of deep learning and tree-based models.
- Evaluate the performance of the hybrid model on a publicly available dataset of movie reviews

Background
Sentiment analysis approaches
Machine Learning Model: Machine learning is a subset field of artificial intelligence involved in designing systems to learn from data, theorizing results on the basis of properties known a priori and by learning from training data. Labeled training data is required to train these models and then apply to some generic classification tasks. Few of

the machine learning models are:

- Linear Regression
- Logistic Regression
- Naive Bayes
- Support Vector Machines (SVM)
- Decision Trees
- Gradient Boosting Machines

Deep Learning Model

State of the art performance of different natural language processing (NLP) tasks, including sentiment analysis have been achieved using deep learning models. These models learn complex representations of the input data by layers, stacking many nonlinear transformations. Deep learning models are used in the sentiment analysis where the deep learning models may automatically pick out features from text data and therefore can capture complex relationships between words and phrases that are difficult to model with traditional methods.

Convolutional Neural Network (CNN). Originally CNNS were designed for image processing applications but have been repurposed as tools for NLP tasks such as sentiment analysis. CNNs are usually used to learn features from the text data based on the convolutional filters of CNNs applied to the input [3].

Recurrent Neural Network (RNN). The inherent nature of language is sequential and so RNNs are ideally suited to model sequential data, in particular natural languages.

Long Short-Term Memory (LSTM). Sentiment analysis tasks rely on the capturing of the long-term dependencies in sequential data, for which LSTM network is a type of RNN but highly effective. Amongst the main benefits of LSTM [4] is its capacity to survive from the vanishing gradient issue in traditional RNNs, which tends to make the system of learning long time dependencies complex.

Tree-Based Model

A large class of machine learning algorithms used in wide variety of applications, most particularly sentiment analysis is tree-based models. Based on decision trees, these models follow hierarchical scheme of the decisions and consequent outcomes through their branching scheme.

The working principle of tree-based models is to systematically partition the dataset according to the rules discovered in training phase. The complex interrelationship of dataset features and target variables lead to these rules, which comprise an overall decision framework that can be used to classify novel instances with high accuracy.

The Random Forest, introduced by Leo Breiman in 2001 [5], is a versatile method that has been applied to bioinformatics, financial analysis, computer vision and many other fields to date. Random Forest is based on ensemble learning, where a single algorithm creates a collection of decision trees to improve accuracy. In this methodology, a novel training paradigm utilizes individual decision trees trained on randomly selected subsets of the training data to generate final predictions by combining the outputs of all constituent trees.

It has a certain special ability of the Random Forest algorithm's capability in terms of it being robust in handling high dimensional data with lots of features and immune against overfitting. To do this, the feature selection process at each decision tree node is sophisticated in that only a random subset of features is used to make splitting decisions. Due to this strategic randomization, inter tree correlations are kept low, while decreasing overfitting tendency improvements are thereby obtained, allowing the model to generalize better.

Hybrid Model

In Sentiment Analysis the idea of combining a number of models to enhance prediction accuracy has been extensively studied. There have been promising results in the hybrid models between deep learning and machine learning models. A long tradition is Sentiment Analysis is use of multiple models hoping that a combination will provide better prediction. In the area of sentiment analysis, deep learning and tree-based models have so far been shown to be promising. The hybrid model approach has presented good results in terms of enhancing a sentiment analysis label accuracy and at the same time overcoming the disadvantages of deep learning models alone in terms of high computational cost and overfitting.

Related Works

In the last few years, the use of deep learning models like Convolutional Neural Networks (CNNs), Long Short Term Memory (LSTM) networks, and transcending the popularity of RoBERTa among others has helped in sentiment analysis taking it to a different level [6]. Local features are captured by CNNs, sequential dependencies by LSTMs [7]. However, transformer models, especially RoBERTa, have achieved higher accuracy but computational demand on account of large number of model parameters [8]. The hybrid CNN-LSTM architecture has been previously explored examining how both model's strengths can be combined. Jang *et al.* presented the CNN-BiLSTM hybrid with attention mechanism that combines the local and global dependencies [9]. In a similar fashion, Umer *et al.* also showed the efficacy of CNN-LSTM hybrids on different datasets such as Twitter US with an accuracy of 82% [10]. Nevertheless, these studies primarily reported the study of the accuracy improvement while no comparative evaluations to the transformer based models have been provided. RoBERTa was integrated with LSTM by Tan *et al.* with promising results but without solving issues of models complexity vs computational efficiency [11].

Several alternative models and approaches have also been explored by several recent studies. A model with attention to Bi-LSTM was proposed by Wu *et al.* (2021) that was significantly improved on sentiment classification tasks [12]. Cambria and Trueman (2021) presented a convolutional stacked Bi-LSTM with an additive attention model that would lead to improved sentiment analysis performance [13]. Wang *et al.* (2023) also further refined an answer selection method based on Bi-LSTM model with an attentive scheme and showed the effectiveness of the scheme in intelligent medical service applications [14]. Almars (2022) also developed an attention based Bi-LSTM Model for Arabic depression classification, proving that the model's capabilities pass through different languages and domains [15]. The experiments conducted by (Omarov & Zhumanov, 2023) on benchmark datasets like the Kaggle Emotion Detection dataset show that the results achieved by the Bi-LSTM models with attention mechanisms outperform the results of the existing state of the art models [16].

The Convolutional Neural Networks (CNN) integration with Random Forest works well in the study by Shruti *et al.* as it improves the classification accuracy of mango leaf diseases. This hybrid approach combines the strength of both these approaches for better performance of the disease class identification [17].

Chen *et al.* conclude that the integration of LSTM and GRU model into deep random forest algorithm enhances the prediction accuracy in discrete process and contributes to the field of predictive modeling [18].

Methodology

The proposed system is a hybrid model of sentiment analysis which consists of a deep learning model plus a tree-based model. This is trained and evaluated on 50,000 movie reviews dataset. LSTM, a deep learning model, generates features for a random forest model that is trained on the labeled training set. The model which was trained was used to predict sentiment labels for each review in the testing set, and the accuracy of model was calculated based on predicted labels.

In the methodology, deep learning and tree-based models are used to classify the text data.

- First step is preparing the dataset for the implementation of the hybrid model for sentiment analysis.
- Preprocessing is the second step to convert the text data to lowercase, to remove special characters, and then to tokenize it.
- The tokenized data was then used to train LSTM Deep Learning model.
- The generated output from LSTM was used as an additional feature for a Random Forest Model.
- The trained Random Forest model was used to classify the test data.
- Finally, the evaluation metrics such as accuracy, precision, recall, and f1-score were calculated.

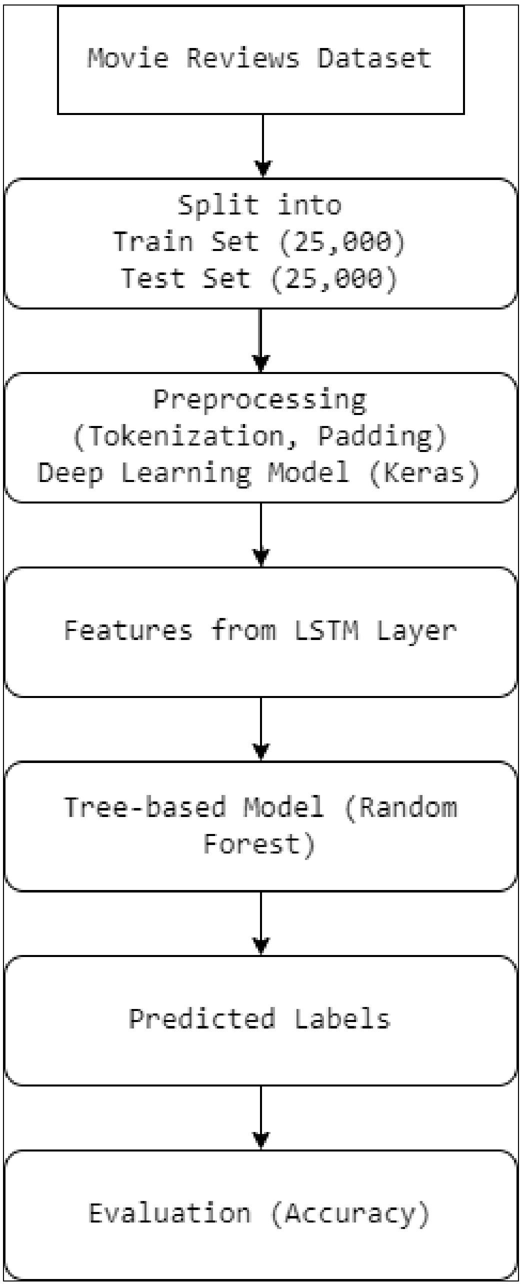

Fig 2: Flow diagram of the Methodology

Implementation

This approach is implemented on Google Collab and Python language.

Dataset

The study used a publicly available dataset (Large Movie Review Dataset (often referred to as the IMDB dataset), which can be accessed at the following link: With the data of 50,000 movie reviews available from the (https://ai.stanford.edu/~amaas/data/sentiment/), a constant work. The reviews were divided into two separate sets: a training set with 25k reviews and their sentiment labels (positive or negative), and a testing set with 25k reviews.

Then processing of the dataset was done by splitting it into two sets, a training set and a testing set having each set of 25k reviews. To train the deep learning and tree-based models, the training set was used; to evaluate the performance of the models, the testing set was used. Sentiment polarity of the reviews in the training and testing sets were prelabeled as '1' for positive review and '0' for negative review.

Fig 3: Word cloud visualization of the positive sentiment tweets from the training dataset

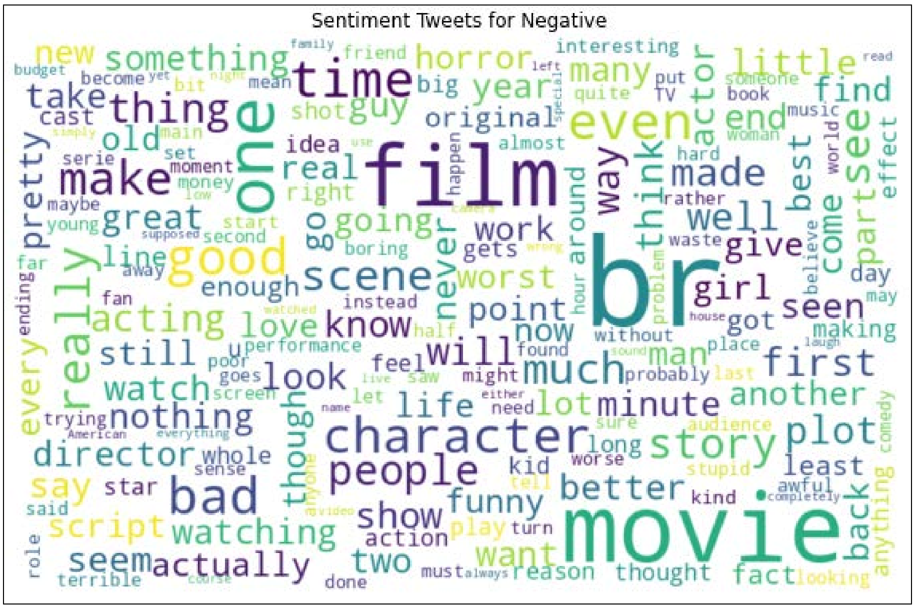

Fig 4: Word cloud visualization of the negative sentiment tweets from the training dataset

Text preprocessing

Before being used for training the model some steps are preprocessed using the text data. Firstly, all the text in the document was converted to lowercase letters so the text is the same across. Second step is removal of all the non-alphanumeric characters (punctuation etc) from the text. The text is made clean and uniform and only that relevant information is present to be done so. Finally, the text was strip of any whitespace characters in the front and back.

After the text is preprocessed, the Tokenizer class from Keras library was used to tokenise them. The main job of the tokenizer is to split the given text into many independent words, and to give each word its corresponding integer value normally. It’s done to have a vocabulary of all the words in the text data. On the training data, the tokenizer is fitted to a vocabulary of the most common word in the text.

So, the vocabulary size is set to 1000, i.e. only the top 1000 most frequent words was considered. After that, tokenizer's texts_to_sequences() method is used to convert texts into sequences of integers. The last, since all sequences have to have a same length, is padded with zero using the pad_sequences() method finally. That guarantees that the input data has the same shape and size for the model.

Deep learning model

With the Keras library we define the architecture a Deep Learning LSTM model. The model input is text data preprocessed, tokenized and padded and has vocabulary size of 1000. There is an embedding layer that embeds the input sequence of the words to fixed size dense vectors. It is then followed by a SpatialDropout1D layer that randomly drops out entire 1D feature maps in the input. Whereas the output goes into an LSTM of 196 units with 0.2 dropout rate. Then, we have a very compact layer with a sigmoid activation function and one that outputs a probability value for a binary classification.

Binary cross entropy loss, Adam optimizer & accuracy as the metric is used to compile the model. The model was trained was trained was trained with a batch size of 256 and verbose=1.

Once the model is trained, the model was utilized to predict on the test dataset. Binary predictions are obtained from rounding the predicted probabilities. An additional feature 'lstm_output' is saved which stores these binary predictions.

The hyperparameters used are:

Embed_dim: The embedding dimension for the input layer. It is set to 128 in this case.

Lstm_out: The number of output nodes for the LSTM layer. It is set to 196 in this case.

Max_features: The maximum number of words to keep in the vocabulary, based on word frequency. It is set to 1000 in the tokenization step.

Input_length: The length of each input sequence. It is set to 20 in the embedding layer.

Dropout: The dropout rate for the LSTM layer. It is set to 0.2 in this case.

Recurrent_dropout: The dropout rate for the recurrent connections in the LSTM layer. It is set to 0.3 in this case.

Epochs: The number of times the model is trained. It is set to 5 in this case.

Batch_size: This refers to the number of samples processed per update of the gradient during training.. It is set to 256 in this case.

Feature generation

Features for the tree-based model were generated using a trained deep learning model. For this specifically, the output of the LSTM layer was used as features for the random forest model.

The output of LSTM layer is the feature generation stored inside the variable `lstm_output`. It is added as another feature to the model. The input sequence of words is processed by the LSTM layer and returns a sequence of hidden states, serving as the features for classification. For this case, the output of the LSTM layer will be used to predict whether the text data is positive or negative. That outputs a numpy array of binary values for the predicted probability of the LSTM model. The output feature from this array is used as input feature for the random forest model.

Random Forest model

The features from the machine learning deep learning model was used on a training set to train a random forest model. Features were then generated via a trained LSTM model and trained a Random Forest classifier on LSTM output as an additional feature. The Random Forest model was instantiated with 20 estimators and random state of 42. The model was then fit using the concatenated training set, consisting of the original feature set (X_train) and the LSTM output feature set (lstm_output).

Testing set was used to generate the LSTM output too, using trained LSTM model for it. The test feature set was generated by concatenating the test set with LSTM output feature set. The Random Forest model was then used to make predictions on the test feature set.

Testing

Random forest model was trained to predict the labels of the reviews in testing set, and calculate the accuracy defined as how good the model predicts the reviews labels.

Results and Analysis

Evaluation Metrics

Performance of the proposed approach (Hybrid Model LSTM + Random Forest), with LSTM output feature, was evaluated using the accuracy score, precision, recall, and F1 score.

Accuracy score: It is the proportion of correct predictions out of all the predictions that the model makes. But if the classes in the dataset are unbalanced (the number of instances of each class are not close to equal), it's a bad metric.

Precision: This metric is just a count of how many true positives (the instances the model marked as positive that were indeed positive) out of all the positive instances the model marked as positive (including false positives). When the focus is on minimizing false positives, this is a useful metric.

Recall: It’s this metric which counts true positives as all correct positives and false negatives as total number of actual positives in the dataset (total possible positives), i.e. this is the fraction of true positives. When the focus is on minimizing false negatives, it’s a useful metric.

F1 score

The latter is a performance metric that’s the harmonic mean of precision and recall. It provides a fair comparison between precision and recall. If both false positives and false negatives are important, it’s a useful metric.

An evaluation of the Random Forest model with LSTM output is printed to give an overview the model’s performance as shown in Figure 5.

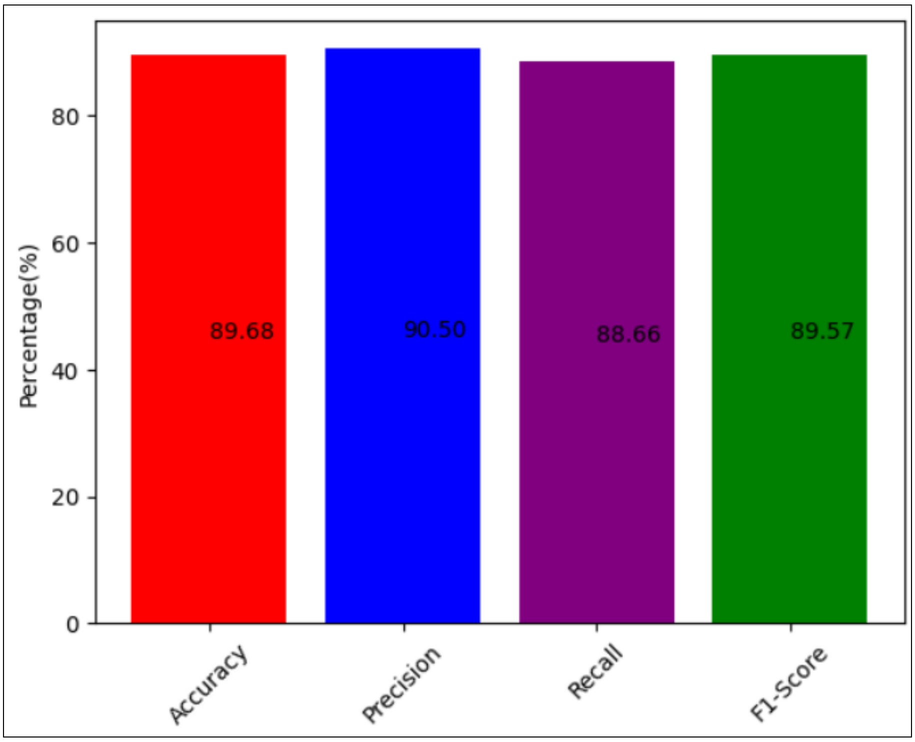

Fig 5: An evaluation of the proposed hybrid model Random Forest model with LSTM output

Result Analysis

In this paper, a hybrid between ML and DL algorithms has been used to analyze sentiments over IMDB movie reviews dataset.

The aim is to detect positive and negative emotions of reviewers through the use of emotional keywords that are commonly used in the reviews. The emotional keywords help to determine the overall sentiment of the reviewer.

In order to accomplish this, a selection of machine learning algorithms including Naives Bayes MNB, LSTM, Random Forest, and a proposed hybrid model of LSTM + Random Forest was used. All models have been tested by accuracy, precision, recall and F1 score as shown in Table 1.

Table: 1 Metrics of Hybrid Model with respect to different existing algorithms

Sl. No	Classifier	Accuracy	Precision	Recall	F1
	Naives Bayes MNB	52.00	52.03	51.36	52.69
	LSTM	72.45	72.39	72.57	72.48
	Random Forest	82.78	86.62	77.54	81.83
	Hybrid Model	89.68	90.50	88.66	89.57

With an accuracy of 89.68%, precision of 90.50%, recall of 88.66% and F1 score of 89.57%, the hybrid model performed the best. Secondly, this demonstrates the usefulness of sourcing both traditional machine learning and deep learning to handle sentiment analysis tasks.

In summary, this method is a good choice, as a way of sentiment analysis tasks in the genre of movie reviews for organizations and business to understand the customers' views and opinions about their products and services.

Conclusion

For this study, a movie reviews dataset from the IMDB dataset using 50,000 reviews were used and the dataset was divided into 25,000 reviews in the training dataset and 25,000 reviews in the testing dataset. In training set, text data were preprocessed by tokenizing and padding sequences to fixed length. An embedding layer, a LSTM layer, SpatialDropout1D layer and an output layer finished with a sigmoid activation function were implemented as components of a deep learning model LSTM. A random forest model was trained using labels from the training set using features generated from the trained deep learning model. First, a random forest model was trained on the reviews in the training set, and this model was used to predict the labels of the reviews in the testing set, and calculate the accuracy of how well the model predicted the labels. Finally, a methodology was developed consisting of training a deep learning model to produce features for a tree-based model, which was then applied to classify sentiment in movie reviews in the test set.

Future Work

In future work this model could explore some advanced deep learning architectures such as the Transformer class of models to improve the accuracy of the sentiment analysis model. Another area of improvement is investigating the impact of hyperparameter tuning on the hybrid model's performance. Moreover, increasing the labeled dataset's size to train the model on more data can potentially improve its accuracy. Additionally, extending the proposed system to analyze other types of reviews, such as product or restaurant reviews, and investigating the system's potential use in other fields like finance, healthcare, or politics could be explored. Lastly, the use of additional features, such as metadata and user demographics, may be investigated to enhance the model's performance.

References

1. Salah Z, Al-Ghuwairi ARF, Baarah A, Aloqaily A, Qadoumi B, Alhayek M, *et al.* A systematic review on opinion mining and sentiment analysis in social media. Int J Bus Inf Syst. 2019;31(4):530-554.
2. Patil S, Lokesha V. Live Twitter sentiment analysis using Streamlit framework. Proceedings of the

International Conference on Innovative Computing & Communication (ICICC) 2022. 2022 May 25. Available from: https://ssrn.com/abstract=4119949 or http://dx.doi.org/10.2139/ssrn.4119949.
3. Kim Y. Convolutional neural networks for sentence classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014;1746-1751. Doha, Qatar: Association for Computational Linguistics.
4. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735-1780. doi:10.1162/neco.1997.9.8.1735.
5. Breiman L. Random forests. Mach Learn. 2001;45(1):5-32. doi: 10.1023/A:1010933404324.
6. Liu X, Shi T, Zhou G, Liu M, Yin Z, Yin L, *et al.* Emotion classification for short texts: an improved multi-label method. Humanit Soc Sci Commun. 2023;10(1):1-9.
7. Luan Y, Lin S. Research on text classification based on CNN and LSTM. *2019* IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA). IEEE; 2019. p. 352-355.
8. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, *et al.* RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692. 2019.
9. Jang B, Kim M, Harerimana G, Kang S, Kim JW. Bi-LSTM model to increase accuracy in text classification: combining Word2vec CNN and attention mechanism. Appl Sci. 2020;10(17):1-16.
10. Umer MF, Ashraf I, Mehmood A, Kumari S, Ullah S, Choi GS. Sentiment analysis of tweets using a unified convolutional neural network-long short-term memory network model. Comput Intell. 2020;37(2):409-434.
11. Tan KL, Lee CP, Anbananthen KS, Lim KM. RoBERTa-LSTM: A hybrid model for sentiment analysis with transformer and recurrent neural network. IEEE Access. 2022;10:21517-21525.
12. Wu P, Li X, Ling C, Ding S, Shen S. Sentiment classification using attention mechanism and bidirectional long short-term memory network. Appl Soft Comput. 2021;112:107792.
13. Trueman TE, Cambria E. A convolutional stacked bidirectional LSTM with a multiplicative attention mechanism for aspect category and sentiment detection. Cogn Comput. 2021;13:1423-1432.
14. Wang D, Liang Y, Ma H, Xu F. Refined answer selection method with attentive bidirectional long short-term memory network and self-attention mechanism for intelligent medical service robot. Appl Sci. 2023;13(5):3016.
15. Almars AM. Attention-based Bi-LSTM model for Arabic depression classification. CMC-Comput Mater Continua. 2022;71(2):3091-3106.
16. Omarov B, Zhumanov Z. Bidirectional long-short-term memory with attention mechanism for emotion analysis in textual content. Int J Adv Comput Sci Appl. 2023;14(6):129-136. Available from: www.ijacsa.thesai.org.
17. Choudhary S, Choudhary M, Kaur S, Kukreja VK. Integrating CNN and random forest for accurate classification of mango leaf diseases. [Journal Name]. 2024;42(6). doi:10.1109/autocom60220.2024.10486108.
18. Chen J, Wang L, Zheng L, Jin X, Yu J, Bai Y, *et al.* Deep random forest prediction algorithm based on LSTM and GRU improvement and application research. [Journal Name]. 2023;681(6). doi:10.1109/aeeca59734.2023.00126.