A novel ensemble approach for Twitter sentiment classification with ML and LSTM algorithms for real-time tweets analysis

Thotakura Venkata Sai Krishna[1], Tummalapalli Siva Rama Krishna[2], Srinivas Kalime[3], Chinta Venkata Murali Krishna[4], Sadineni Neelima[5], Raja Rao PBV[6]

[1]Department of Computer Science and Engineering-Data Science, QIS College of Engineering and Technology (Autonomous), Ongole, India
[2]Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Kakinada, Kakinada, India
[3]Department of Computer Science and Engineering, Sreyas Institute of Engineering and Technology, Bandlaguda, Hyderabad
[4]Department of Computer Science and Engineering-Data Science, NRI Institute of Technology, Pothavarappadu, India
[5]Department of IT, Aditya Engineering College, Surampalem, India
[6]Department of Computer Science and Engineering, Shri Vishnu Engineering College for Women(A), Bhimavaram, India

Article Info

Article history:

Received Feb 12, 2024
Revised Feb 26, 2024
Accepted Mar 4, 2024

Keywords:

Social media
Twitter API
Machine learning
NLP
Deep learning
VADER

ABSTRACT

Social media sentiment classification was an essential consideration in natural language processing (NLP) for evaluating normal people's perspectives on a given topic. With Twitter's massive rise in popularity in recent years, the capacity to extract information about public sentiment from tweets became a major focus. This paper not only analyzed public sentiment through data from Twitter but introduced a novel ensemble approach in the methods employed for Twitter sentiment classification. Real-time tweets on various topics, including "covid," "crime," "spam," "flipkart," "migraine," and "airlines," were extracted and thoroughly examined to gain insight into public opinions. Leveraging the Twitter API for real-time tweet extraction, natural language processing techniques were applied to clean the tweet data. Subsequently, we applied several machine learning (ML) algorithms Naïve Bayes, decision tree (DT), random forest (RF), logistic regression (LGR), and deep learning (DL) algorithms recurrent neural network (RNN), LSTM, and GRU individually. Later, we proposed a novel ensemble of ML and DL algorithms for sentiment classification, with a novel emphasis on ensemble techniques and enhanced the accuracy with a significance compared to individual ML or DL model applied. The experimental results demonstrated that our novel ensemble approach achieved high accuracy when compared to existing work.

Corresponding Author:

Sadineni Neelima
Department of Information Technology, Aditya Engineering College
Surampalem, India
Email: neelima.sadineni@gmail.com

1. INTRODUCTION

As a new medium for exchanging ideas and perspectives, social media has been recognized as an essential aspect of today's world. There are a number of popular social media platforms currently in use, including Facebook, Twitter, Instagram, linked in, and so on. Twitter is a social networking platform whose major goal is to connect individuals and allow them to share their ideas with a massive group. Twitter comprises mainly opinions regarding companies, services, celebrities, events, or anything else that users are interested in. The success of Twitter has subsequently piqued the curiosity of many analysts, who have evaluated Twitter data for a number of purposes. A rising number of businesses are centering their business strategies on online comments, sentiments, and opinions about products from customers and prospects, and some are even attempting to anticipate the acceptance and rejection of various brands using this data. In 2024, Twitter's global viewership was estimated to be about 335 million monthly active users [1]. Because of this, organizations and people who want to know how the public feels about their products, announcements, and other events are more interested in collecting and analyzing this collective sentiment. However, it is not a straightforward approach as it may appear. The enormous number of tweets makes manual emotion identification impossible. A popular approach is the "lexicon-based method," which analyzes tweets by keywords. It uses a positive and negative lexicon to determine key phrase frequency weights. The proportion forecasts the tweet's positive or negative rating. Twitter linguistic shift may make lexicon-based mood evaluation ineffective. Thus, many Twitter sentiment analysis approaches employ deep learning (DL) and machine learning (ML). In this paper, ML and DL algorithms categorized Twitter sentiments. First, we utilized natural language processing (NLP) to prepare text data for ML or DL, then numerous ML and DL classifiers for twitter sentiment classification.

Sindhuja *et al.* [2] explored Twitter sentiment analysis, a popular public and private opinion-sharing platform. Due to emotional impact on daily life, sentiment analysis measured opinion polarity. The research categorizes attitudes using Naïve Bayes and text preprocessing. Avoid stop words, normalize case, and stem or lemmatize during preprocessing. Twitter sentiment analysis showed user sentiment. Polaritypositive, negative, or unbiased provided vital public opinion data. The Naïve Bayes method was used in [3] to evaluate polarity and subjectivity in TextBlob. This included "ineffective police," "chief of police," and "law enforcement officer." Results showed a negative sentiment trend based on keywords. Parveen *et al.* [4] tested the gated attention recurrent network (GARN) for sentiment categorization using recurrent neural network (RNN) and attention processes and achieved goos results. Abad *et al.* [5] identified URLs using support vector machines (SVMs), random forests (RFs), decision trees (DTs), k-nearest neighbors (KNNs), and Bayesian optimization. Random instance selection, data reduction based on locality-sensitive hashing (DRLSH), and border point extraction based on locality-sensitive hashing (BPLSH) improved computational efficiency. RFs had high accuracy, recall, and F1-scores, but SVMs competed with longer training periods. Bokolo *et al.* [6] used ML models to analyze criminal intent sentiment. It compared Twitter developer account and Kaggle sentiment 140 datasets. Lakshmanarao *et al.* [7] applied DL RNN variants for twitter sentiment classification and reported good results.

Abbas *et al.* [8] applied voting classifier for twitter sentiment analysis. For sentiment analysis, they applied combination of four algorithms logistic regression (LGR), the nave Bayes classifier, the DT, and MLP. The authors used with a dataset size of 2,000 tweets and successfully classified tweets into negative and positive with their proposed method. Rodrigues *et al.* [9] proposed several ML and DL algorithms for real time twitter sentiment classification. They extracted a spam/ham sentiment dataset with 5,570 tweets. Several ML models applied and later DL techniques like RNN, long short-term memory (LSTM), CNN (1D) also applied for twitter sentiment classification. After applying all algorithms, the authors concluded that they achieved best accuracy with LSTM model. Bhatnagar and Choubey [10]. proposed sentiment analysis on various topics such as corona, political tweets, e-shopping. To derive an overall sentiment of connected topics, the authors presented an infrastructure that integrates sentiment analysis and community detection. Jumhare *et al.* [11] proposed twitter sentiment analysis on violence data. The authors mainly concentrated on tweets related to "URI attack". More than 4,500 tweets are collected from twitter and applied unsupervised learning algorithms for sentiment analysis. AlBadani *et al.* [12] applied SVM for tweet sentiment classification and achieved good results. The authors used "Airline Tweets" dataset from Kaggle and applied preprocessing techniques. Later, they applied ULMFiT (a transfer learning approach) in combination with SVM for sentiment analysis. Gil-Ramírez and Guilleumas-García [13] applied ML algorithms for sentiment-analysis. The proposed method extracted 27,660 tweets from twitter based on the keywords 'mobile' and 'learning.' Later, based on the tweet text, they assigned polarity to the tweet as 'negative' or 'positive.'

Priya and Rani [14] proposed telugu tweet sentiment analysis framework. The proposed method uses word embedding techniques followed by deep leaning RNN for tweet classification and achieved 80% accuracy. Li and Fleyeh [15] retried twitter tweets related to the topic "IKEA stores" and build a model for sentiment analysis. For English tweets, lexicon approach used for assigning polarities. But there is no predefined dictionary for 'swedish' tweets ML oriented approach used for analysis of Swedish ikon messages. Kaneria and Patel [16] proposed naïve bayes and SVM classifiers for twitter sentiment classification. Savita and Gore [17] applied SVM classifier for twitter sentiment classification. The model was implemented in two different datasets with tweets of "movie-reviews" and "product-based reviews". Amolik *et al.* [18] applied SVM and naïve bayes classifier for classification of movie reviews as "positive", "negative" or "neutral". A total of 1,950 tweet related movie reviews collected and applied ML classifiers NB and SVM and achieved 65% and 75% accuracies respectively for sentiment classification. Faizan [19] applied various feature extraction techniques for Twitter sentiment classification. A total of 3,000 tweets related to "US Airlines" were collected and achieved good accuracy with the KNN classifier.

Pandey *et al.* [20] proposed ML techniques for sentiment classification. After the extraction of tweets, SVM was applied and successfully classified tweets with respect to their correct class labels. Srinivas *et al.* [21] created a deep neural network (DNN) model to identify fake job listings. KNN, Naïve Bayes, DT, SVM, and RF classifiers were compared and evaluated for text classification and achieved good results. Maithili *et al.* [22] used a multi-layered extreme learning machine to identify Twitter spam. Additionally, the Word2Vec model converted dataset words into multi-dimensional vectors. Modified extreme learning machine (MELM) improved on ELM by adding additional hidden layers and adaptively initializing weights between input, first hidden layer, and bias. The network output weights were calculated using least squares. Three spam datasets were extensively tested to determine the MELM model's spam detection ability. Nistor *et al.* [23] a dataset with 1,578,627 tweets is analyzed for sentiment analysis with 20 different RNN architectures. Out of 20 architectures, the gated recurrent unit (GRU) architecture achieved 80% accuracy for sentiment classification. Arun *et al.* [24] proposed twitter sentiment classifier for demonetization messages from twitter platform. The author extracted 13,000 tweets from twitter and analyzed the sentiments.

The majority of previous studies focused on ML or DL models for static Kaggle datasets containing only one sentiment content. In contrast, this paper introduces ML and DL algorithms tailored for real-time Twitter data, extending sentiment analysis to six categories. The paper is structured as follows: Section 2 details the proposed method, section 3 presents results and discussion, and section 4 offers conclusions.

2. METHOD

The proposed method is shown Figure 1. Real time tweets from twitter are retrieved. Twitter API with tweety [25] used for extracting real time tweets. Several genres of tweets extracted. The keywords used for extracting tweets are are "covid," "crime," "spam," "flipkart," "migraine," and "airlines." All the extracted tweets are created as separate datasets. Later, NLP approaches such as "stop-word removal," "stemming" applied for cleaning tweet text. The extracted tweets have no labels (positive or negative).

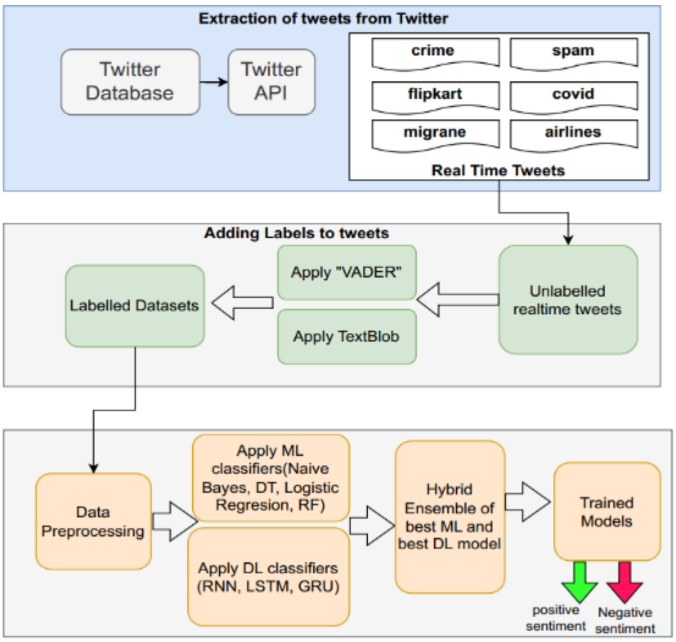

Figure 1. Proposed methodology

For assigning tweet labels, we used two sentiment analysis approaches "TextBlob" and "VADER" valence aware dictionary for sentiment reasoning (VADER). Using these two approaches tweet labels are assigned. Now all the datasets with class labels are available for applying ML algorithms. Next, we applied four ML algorithms for sentiment classification. Five ML algorithms Naïve Bayes, RF, DT, and LGR are used in this work. Next, we also applied RNN and LSTM algorithms for training the labelled tweets for sentiment classification. After comparing all the algorithms, the best model which gives more accuracy is used for analyzing the sentiments of new dataset.

2.1. Extracting live tweets from Twitter

There are twitter datasets available in ML repositories. But we extracted live tweets from twitter. In twitter daily vast number of tweets data generated. One of the Twitter API named Tweepy [25] allows users to retrieve tweets based on topics of interest. To do this, the first user must have a Twitter account with developer access. The user should send a request mail to twitter to get developer access. After getting developer access through reply mail, user can get keys and tokens. These secret keys are helpful for extracting live tweets. After getting keys and tokens, a python script with several predefined functions used for extracting tweets from twitter.

2.2. Sentiment analysis techniques

The Lexicon-based technique assesses a message by summing the sentiment ratings of all the terms in the content using a pre-defined sentiment lexicon. With the use of a valence dictionary, terms in text are rated as positive or negative. We can calculate an overall emotion score by counting the amount of positive and negative words in the text and adding these values analytically. For adding labels to the tweet, we applied two libraries “TextBlob” and “VADER”. Vader is a lexicon-based approach. The reason for applying both techniques are that VADER tends to do better with slang, emoji, and other non-traditional languages, whereas TextBlob excels with more formal languages. So, we used both the libraries for getting maximum performance of the proposed model.

2.3. ML and DL algorithms

After assigning labels, we applied four ML classifiers NB, LGR, DT, and RF. Later we applied RNN and LSTM also for sentiment classification. All the applied models are evaluation for accuracy.

2.3.1. ML classifiers

Naïve Bayes is a high-bias, low-variance classifier even with a limited data set. It is a probability-oriented approach. Bayes theorem is the heart of this algorithm. It's easy to use and doesn’t require a lot of computing power. Text categorization is easily done with this algorithm. LGR is also known as “max-entropy” classification or the log-linear classifier. A logistic function is used to model the probability of the probable outcomes of a single attempt in this model. DT is a tree algorithm gets its name from the fact that it presents decisions and decision-making processes in a tree-like structure. It assists us in making the best decision in a difficult circumstance with multiple options, where we must choose the best method to achieve the best result. RF algorithm generates a forest with a large number of trees, as the name implies. It is a supervised classifier useful for solving issues involving both classification and regression.

2.3.2. RNN/LSTM/GRU

RNNs model their weights using historical time series data, and they’re used to solve a variety of fascinating ML tasks including text processing. As the amount of the time-series input rises, a RNN using classical neurons becomes slower. This problem can be solved in LSTM. LSTM contains cells. LSTM cells are meant to hold information as it passes through a chain of neurons, ensuring that just the relevant data passes through the system with no need to store the entire data sequence. GRU is another variant of RNN which has less gates than LSTM.

3. RESULTS AND DISCUSSION

3.1. Creation of dataset

For all the selected six topics live tweets are extracted from twitter using tweepy package. But at a time, twitter allows users to retrieve only 3,200 tweets. It only allows to extract tweets of past seven days only. So, we extracted seven days tweets with six topics “covid”, “crime”, “spam”, “flipkart”, “migraine”, “airlines”. We extracted 28,800 live tweets from the topics “covid”, “crime”, “spam”, “migraine”, “airlines” and 28,100 tweets from the topic “flipkart”. All the tweets are unlabeled. Later, TextBlob and vader libraries used for assigning tweets labels.

3.2. Applying TextBlob to assign labels

‘TextBlob’ is a Python package [26]. It offers a basic API for performing standard NLP tasks like part of speech (POS) tagging, noun-phrase extraction, sentiment analysis, classifications etc. A tuple of the type sentiment is returned by the sentiment attribute with “polarity” and “subjectivity”. The polarity value is in between -1.0 and 1.0. Here -1.0 (or <0) means negative sentiment, +1.0 (or >0) means positive sentiment and 0.0 means neutral sentiment subjectivity is a value in the range of 0.0 and 1.0. with 0.0 being objective and 1.0 being subjective. Using ‘TextBlob’ python package, polarity sore is calculated for each tweet in a dataset and tweet labels are assigned.

3.3. Applying VADER to assign labels

VADER is a rule based and lexicon-oriented sentiment analysis approach [27]. It calculates text sentiment using a collection of lexical features that are divided as positive or negative depending on the sentiment polarity score. It gives probability values to declare sentiment as "positive", "neutral" or "negative". Based on the obtained probabilities, the tweet labels are assigned. After applying TextBlob and VADER sentiment analysis techniques, each tweet is labelled with "positive", "negative" or "neutral" sentiment. Algorithm 1 for extracting tweets and assigned sentiments through "TextBlob" and "VADER" is shown.

Algorithm 1. for extracting tweets and assigning sentiments:

```
Input: empty lists -Tweets, Sentiment1, Sentiment2
Output: Three lists, list of tweets (Tweets), list of sentiments assigned with tetblob
(Sentiment1), list of sentiments assigned with vader (Sentiment2)
Step 1. import required python packages (tweety, TextBlob, vedar etc.)
Step 2. Provide information about tokens and keys (with twitter development access)
Step 3. Use Tweety. Cursor to extract tweets with required details (like count and keyword
to search)
Step 3.1. for each tweet calculate sentiment score using TextBlob
          Step 3.1.1.  if score>0, Sentiment1 is "Positive"
                     if score<0, Sentiment1 is "Negative"
                           if score=0, Sentiment1 is "Neutral"
 Step 3.2. for each tweet caculate sentiment score using vader
         Step 3.2.1.  if score>0.05, Sentiment2 is "Positive"
                    if score<-0.05, Sentiment2 is "Negative"
                          otherwise, Sentiment2 is "Neutral"
```

After applying algorithm, six datasets created each for one topic out of "covid," "crime," "spam," "migraine," "airlines," and "flipkart" topics. Table 1 shows details of dataset. The number of tweets is more than 28,000 for all contents. All these datasets used in further experiments.

Table 1. Details of created datasets

Topic	Dataset name	Tweets
Crime	crime_dataset	28,800
Covid	covid_dataset	28,800
Spam	spam_dataset	28,800
Migraine	migraine_dataset	28,800
Airlines	airline_dataset	28,800
Flipkart	flipkart_dataset	28,100

3.4. Applying data preprocessing techniques

After creating six datasets with labels using TextBlob and VADER, we applied several text processing techniques for tweets. First, we removed useless symbols from the text. Later, we removed stop words. Stop words (like is, was, an) are not useful for classifying tweet as positive or negative. Later Stemming of words applied. Stemming is a process of removing prefixes to words. Now, cleaned datasets obtained after applying all these techniques to six datasets. These datasets are ready for applying ML, DL algorithms.

3.5. Applying ML techniques

After creating six datasets with labels using TextBlob and VADER, we applied four ML algorithms on all datasets. However, there are large number of ML classifiers available, these four classifiers given more accuracy values in several experiments. For all experiments, we consider positive and negative tweets in the dataset and it is split into train and test set in the ratio of 80% and 20%. Naïve bayes, LGR, DT, and RF algorithms applied on six datasets. The results of the experiments are shown in Table 2 and Figure 2. From Table 2, it is observed that NB given good accuracy for 'TextBlob' labelled datasets in most of the cases. For 'airline' and 'migraine' datasets, NB given good accuracy.

3.6. Applying Naïve Bayes, LGR, DT, and RF

The results with four ML classifiers are shown in Table 2 and Figure 2. The LGR model, especially with TextBlob, demonstrated superior performance across various datasets. In crime and migrane datasets, LGR with TextBlob achieved accuracies of 90% and 88%, respectively. For the covid dataset, LGR with VADER led with 90% accuracy. In spam and airline datasets, both LGR with TextBlob and LGR with

VADER achieved 90% accuracy. In flipkart dataset also LGR performed well with 94% accuracy. The RF model consistently demonstrated robust performance across datasets, with notable accuracies. In the crime dataset, RF with VADER achieved 90%, while in the flipkart dataset, RF with TextBlob reached 90% accuracy. For covid, spam, and airline datasets, both RF with TextBlob and RF with VADER showcased reliable accuracies, ranging from 87% to 90%. The DT model also performed well, achieving accuracies in the range of 85% to 88% across various.

Table 2. Results with ML techniques

Dataset name	NB-Acc (TextBlob)	NB-Acc (VADER)	LGR-Acc (TextBlob)	LGR-Acc(VADER)	DT-Acc (TextBlob)	DT-Acc (VADER)	RF-Acc (TextBlob)	RF-Acc(VADER)
crime_dataset	80%	59%	90%	91%	88%	90%	88%	90%
covid_dataset	70%	81%	89%	90%	87%	87%	87%	87%
spam_dataset	65%	74%	90%	89%	86%	85%	86%	85%
migraine_dataset	64%	78%	88%	87%	85%	83%	85%	83%
airline_dataset	74%	84%	91%	91%	88%	87%	88%	87%
flipkart_dataset	81%	78%	94%	94%	91%	90%	91%	90%

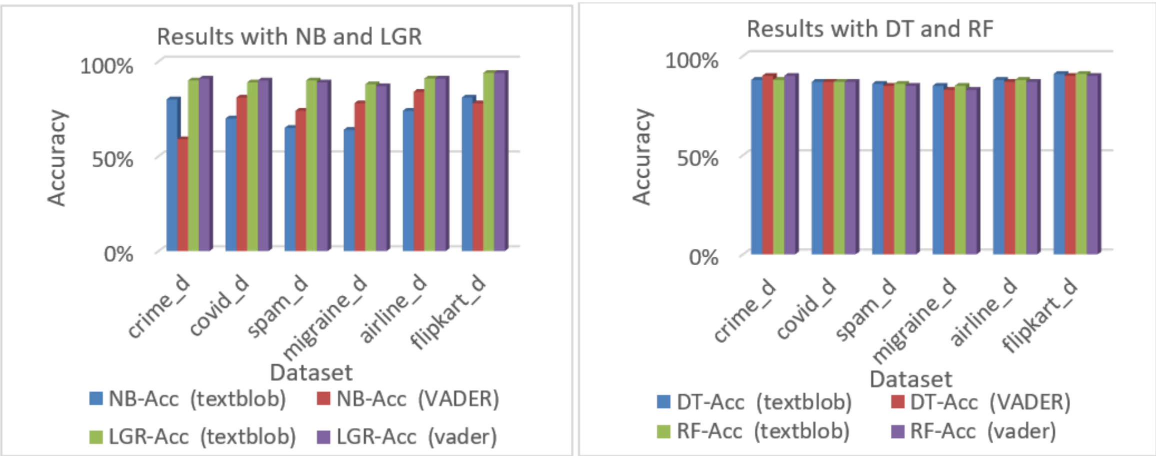

Figure 2. Results with ML models

3.7. Applying DL algorithms

Three DL models RNN, LSTM, and GRU applied for twitter sentiment classification. Keras offers an embedding layer that is frequently used for text data analysis through neural networks. Because it's vital to encode words into integers in ML solutions and services, each word is assigned a unique integer. We first created a vocabulary and then used the Tokenizer API to format our input data suitable for classification task. This embedding layer starts with random weights and subsequently learns to generate word vectors from tweet text. The embedding that is created can be kept and utilized for future modelling.

3.8. Applying RNN, LSTM, stacked-LSTM, and GRU

The results with DL models are shown in Table 3 and Figure 3. The RNN, LSTM, and Stacked LSTM (SLSTM) models exhibited commendable performance across various datasets. In the flipkart dataset, LSTM with TextBlob achieved an impressive accuracy of 95.20%, surpassing other approaches.

Table 3. Results with DL models

Dataset Name	RNN-Acc (textblob)	RNN-Acc (VADER)	LSTM-Acc (textblob)	LSTM-Acc (vader)	SLSTM-Acc (textblob)	SLSTM-Acc (VADER)	GRU-Acc (textblob)	GRU-Acc (vader)
crime_dataset	89.6%	90%	91.1%	91%	90.8%	92%	91.3%	91.9%
covid_dataset	87.8%	86.7%	90.7%	88.7%	90.7%	88.9%	90.7%	88.2%
spam_dataset	87.1%	87.2%	90.3%	88.2%	89.4%	88.2%	90.3%	88.5%
migraine_dataset	86.9%	85.9%	87.9%	87.3%	87.1%	87.5%	88.2%	86.7%
airline_dataset	90%	89%	90%	91.5%	91.5%	90.7%	91.7%	90.8%
flipkart_dataset	89.6%	90%	95.2%	91.5%	95%	93.1%	94.8%	93.2%

The crime dataset demonstrated robust performance, with RNN reaching 89.60% accuracy with TextBlob, while LSTM achieved 91.10% accuracy with TextBlob. Notably, the airline dataset showcased consistent high accuracy for both RNN and LSTM, ranging from 89% to 91.50%. The SLSTM and GRU models consistently delivered strong sentiment analysis results, with notable performances on datasets like flipkart (SLSTM achieved 95% accuracy with TextBlob) and crime (SLSTM achieved 90.80% accuracy with TextBlob). The airline dataset displayed reliable accuracy ranging from 90.70% to 91.70% for both SLSTM and GRU.

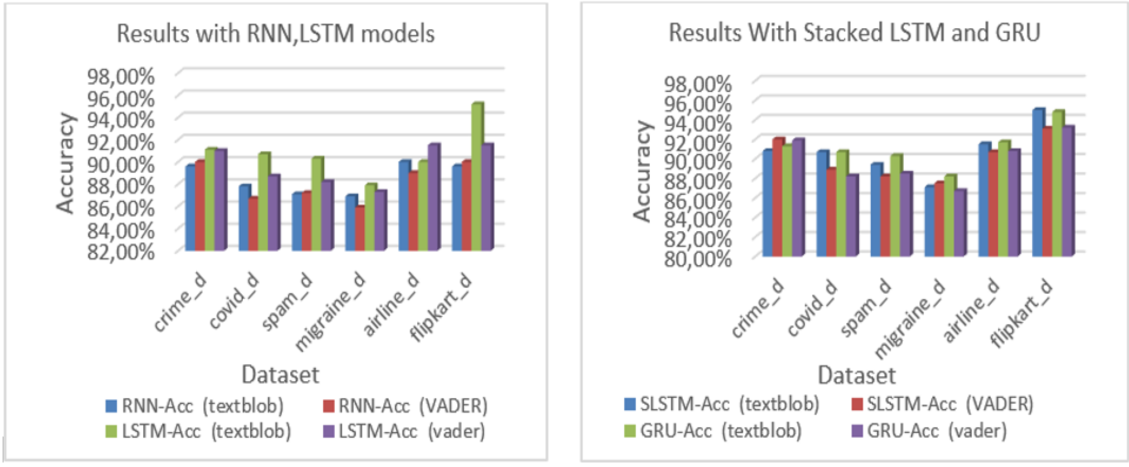

Figure 3. Results with RNN, LSTM models and results with SLSTM, GRU

3.9. Novel ensemble approach

After implementing ML and DL models, we proposed a novel fusion of best ML and best DL algorithms to enhance the accuracy of each dataset. In developing a hybrid sentiment analysis model, a systematic process was undertaken, evaluating ML and DL models across diverse datasets. The stacking of best ML and best DL model is taken and stacked. The algorithms used in hybrid integration is shown in Table 4.

Table 4. Performance evaluation

Dataset name	Best ML model accuracy	Best DL model accuracy	Proposed hybrid ensemble merge	Accuracy achieved with hybrid ensemble model
crime_dataset	LGR(VADER):91%	GRU(VADER):91.9%	LGR+GRU (VADER)	93.4%
covid_dataset	LGR(textblob):89%	LSTM(textblob):90.7%	LGR+LSTM (TextBlob)	91%
spam_dataset	LGR(textblob):90.3%	LSTM(textblob):90%	LGR+LSTM (TextBlob)	90.7%
migraine_dataset	LGR(textblob):88%	LSTM(textblob):87.9%	LGR+LSTM (TextBlob)	91%
airline_dataset	LGR(textblob):91%	GRU(textblob):91.7%	LGR+GRU (TextBlob)	92.8%
flipkart_dataset	LGR(textblob):94%	LSTM(textblob):95.2%	LGR+LSTM (TextBlob)	96.8%

For the crime_dataset, the ML model LGR with VADER sentiment analysis demonstrated a robust accuracy of 91%, while the DL counterpart, GRU with VADER, excelled with a slightly higher accuracy of 91.9%. The hybrid integration of these models, denoted as LGR+GRU (VADER), signifies a collaborative approach, capitalizing on the interpretability of LGR and the nuanced context understanding of GRU. This fusion aims to provide an even more comprehensive and accurate sentiment analysis for crime-related content. Moving to the covid_dataset, the ML model LGR with TextBlob achieved a commendable accuracy of 89%, while the DL model LSTM with TextBlob outperformed with an accuracy of 90.7%. The proposed hybrid model, LGR+LSTM (TextBlob), merges these models, leveraging the strengths of LGR in capturing general patterns and the nuanced understanding capabilities of LSTM. This hybrid approach aims to enhance sentiment analysis performance in the context of COVID-related content.

In the spam_dataset, the ML model LGR with TextBlob yielded an accuracy of 90.3%, closely followed by the DL model LSTM with TextBlob at 90%. The hybrid integration, LGR+LSTM (TextBlob), combines these models to create a comprehensive sentiment analysis solution, blending the interpretability of

LGR with the sequential learning capabilities of LSTM. For the migraine_dataset, LGR with TextBlob achieved an accuracy of 88%, while the DL model LSTM with TextBlob secured 87.9%. The hybrid model, LGR+LSTM (TextBlob), strives to merge these models for a more nuanced sentiment analysis approach, aiming to capture subtleties in sentiment expression related to migraines. In the airline_dataset, LGR with TextBlob achieved an accuracy of 91%, and the DL model GRU with TextBlob surpassed with an accuracy of 91.7%. The hybrid model, LGR+GRU (TextBlob), combines these models, aiming to leverage the strengths of LGR and GRU for a robust sentiment analysis solution tailored for airline-related content.

Lastly, for the flipkart_dataset, LGR with TextBlob achieved an impressive accuracy of 94%, while the DL model LSTM with TextBlob excelled further at 95.2%. The hybrid integration, LGR+LSTM (TextBlob), signifies a collaborative approach, combining the interpretability of LGR with the sequential learning capabilities of LSTM to enhance sentiment analysis performance in the context of Flipkart-related content. Later, the proposed hybrid models applied on six datasets. The results are shown in Table 4 and Figure 4.

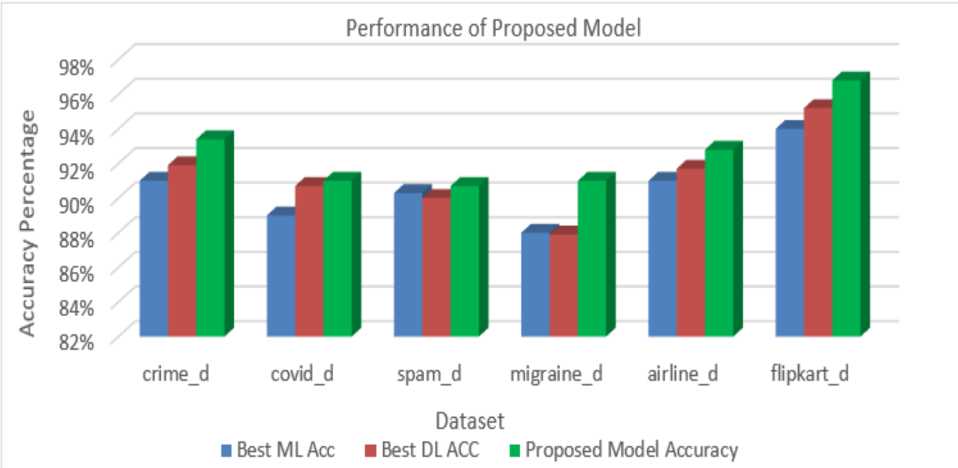

Figure 4. Results with proposed hybrid ensemble

For the crime_dataset, the best ML model achieved an accuracy of 91%, while the optimal DL model reached 91.90%. The integration of these models in the proposed hybrid approach resulted in a further enhancement, yielding an accuracy of 93.40%. In the context of the covid_dataset, the best ML and DL models achieved accuracies of 89% and 90.70% respectively. The proposed hybrid model, combining LGR with TextBlob and LSTM with TextBlob, surpassed both individual models, achieving an accuracy of 91%. For the spam_dataset, the best ML and DL models attained accuracies of 90.30% and 90% respectively. The proposed hybrid model, integrating LGR with TextBlob and LSTM with TextBlob, resulted in an accuracy of 90.70%. In the migraine_dataset, the best ML and DL models achieved accuracies of 88% and 87.90% respectively. The proposed hybrid model, combining LGR with TextBlob and LSTM with TextBlob, given accuracy of 91%. For airline_dataset, the best ML and DL models achieved accuracies of 91% and 91.70% respectively. The proposed hybrid model, integrating LGR with TextBlob and GRU with TextBlob, exhibited an accuracy boost to 92.80%. Finally, for the flipkart_dataset, the best ML and DL models achieved accuracies of 94% and 95.20% respectively. The proposed hybrid model, combining LGR with TextBlob and LSTM with TextBlob, showcased the highest accuracy increase to 96.80%. This underscores the efficacy of the hybrid approach in delivering superior sentiment analysis performance in the six contexts used.

3.10. Comparison with existing work

Table 5 see in Figure 3 shows accuracy comparison with previous works. Sidharta *et al.* [3], the authors gain 82.6% accuracy with voting classifier. Rodrigues *et al.* [9], the authors achieved 80.4% accuracy with RNN. Kaneria and Patel [16], the authors achieved 86% accuracy with MNB. In this paper, we applied proposed models to six different datasets for sentiment classification and achieved good accuracy for all datasets. Figure 3 shows accuracies obtained in proposed work when comparison with previous works.

Table 5. Comparison with previous work

Model	Accuracy
Voting classifier [8]	82.6%
RNN [14]	80.4%
Multinomial Naïve Bayes [21]	86%
Proposed method	93.4% (crime_dataset)
	96.8% (flipkart_dataset)

4. CONCLUSION

This work explored sentiment analysis on Twitter data, employing a multifaceted approach to extract, process, and analyze tweets across diverse domains. The initial phase involved utilizing the Twitter API to extract real-time tweets on topics such as “covid,” “crime,” “spam,” “flipkart,” “migraine,” and “airlines.” By establishing a streamlined process involving a Twitter Developer account, a Twitter App, and the Tweepy library, the study achieved efficient access and retrieval based on specific search criteria. For sentiment classification, individual ML and DL models were applied to each dataset. In the crime_dataset, LGR with VADER achieved 91% accuracy, while GRU with VADER excelled slightly higher at 91.9%. The hybrid model, denoted as LGR+GRU (VADER), achieved a collaborative accuracy of 93.40%. Moving to the covid_dataset, LGR with TextBlob achieved 89%, and DL model LSTM with TextBlob outperformed with 90.7%. The proposed hybrid model, LGR+LSTM (TextBlob), achieved a superior accuracy of 91%. In the spam_dataset, LGR with TextBlob yielded 90.3%, and DL model LSTM with TextBlob closely followed at 90%. The hybrid model, LGR+LSTM (TextBlob), resulted in a notable accuracy improvement to 90.70%. Similarly, in the migraine_dataset, LGR with TextBlob achieved 88%, and DL model LSTM with TextBlob secured 87.9%. The hybrid model, LGR+LSTM given accuracy of 91%. In the airline_dataset, LGR with TextBlob achieved 91%, and DL model GRU with TextBlob surpassed with 91.7%. The hybrid model, LGR+GRU (TextBlob), exhibited an accuracy boost to 92.80%. Lastly, for the flipkart_dataset, LGR with TextBlob achieved 94%, while DL model LSTM with TextBlob excelled further at 95.2%. The hybrid integration, LGR+LSTM showcased the highest accuracy increase to 96.8%. The hybrid models achieved accuracy enhancements ranging from 1.4% to 4.8%. For future work, the exploration of ensemble models, incorporating the strengths of various classifiers, could further enhance accuracy and robustness. Investigating the impact of fine-tuning hyperparameters and incorporating domain-specific lexicons may contribute to even more nuanced sentiment analysis results.

REFERENCES

[1] Statista, Number of Twitter users worldwide from 2010 to 2021 (in millions). [Online]. Available: https://www.statista.com/statistics/303681/twitter-users-worldwide/. Accessed on November 12, 2023.

[2] M. Sindhuja, K. S. Nitin, and K. S. Devi, “Twitter sentiment analysis using enhanced TF-DIF Naïve Bayes classifier approach,” in *Proceedings - 7th International Conference on Computing Methodologies and Communication, ICCMC 2023*, IEEE, Feb. 2023, pp. 547–551. doi: 10.1109/ICCMC56507.2023.10084106.

[3] S. Sidharta, Y. Heryadi, L. Lukas, W. Suparta, and Y. Arifin, “Sentiment analysis of hashtag activism on social media Twitter,” in *ICCoSITE 2023 - International Conference on Computer Science, Information Technology and Engineering: Digital Transformation Strategy in Facing the VUCA and TUNA Era*, IEEE, Feb. 2023, pp. 73–77. doi: 10.1109/ICCoSITE57641.2023.10127833.

[4] N. Parveen, P. Chakrabarti, B. T. Hung, and A. Shaik, “Twitter sentiment analysis using hybrid gated attention recurrent network,” *Journal of Big Data*, vol. 10, no. 1, p. 50, Apr. 2023, doi: 10.1186/s40537-023-00726-3.

[5] S. Abad, H. Gholamy, and M. Aslani, “Classification of malicious URLs using machine learning,” *Sensors*, vol. 23, no. 18, p. 7760, Sep. 2023, doi: 10.3390/s23187760.

[6] B. G. Bokolo, E. Ogegbene-Ise, L. Chen, and Q. Liu, “Crime-intent sentiment detection on Twitter data using machine learning,” in *Proceedings - 2023 8th International Conference on Automation, Control and Robotics Engineering, CACRE 2023*, IEEE, Jul. 2023, pp. 79–83. doi: 10.1109/CACRE58689.2023.10208384.

[7] A. Lakshmanarao, A. Srisaila, and T. S. R. Kiran, “Twitter sentiment classification with deep learning LSTM for airline tweets,” in *8th International Conference on Advanced Computing and Communication Systems, ICACCS 2022*, IEEE, Mar. 2022, pp. 520–524. doi: 10.1109/ICACCS54159.2022.9785208.

[8] A. K. Abbas, A. K. Salih, H. A. Hussein, Q. M. Hussein, and S. A. Abdulwahhab, “Twitter sentiment analysis using an ensemble majority vote classifier,” *Journal of Southwest Jiaotong University*, vol. 55, no. 1, 2020, doi: 10.35741/issn.0258-2724.55.1.9.

[9] A. P. Rodrigues *et al.*, “Real-time twitter spam detection and sentiment analysis using machine learning and deep learning techniques,” *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–14, Apr. 2022, doi: 10.1155/2022/5211949.

[10] S. Bhatnagar and N. Choubey, “Making sense of tweets using sentiment analysis on closely related topics,” *Social Network Analysis and Mining*, vol. 11, no. 1, p. 44, Dec. 2021, doi: 10.1007/s13278-021-00752-0.

[11] N. Jumhare, G. R. Rajeswari, and B. Jayakrishnan, “Sentiment analysis based on Twitter data on violence,” *Asian Journal of Pharmaceutical and Clinical Research*, vol. 10, no. 13, pp. 474–475, Apr. 2017, doi: 10.22159/ajpcr.2017.v10s1.20521.

[12] B. AlBadani, R. Shi, and J. Dong, “A novel machine learning approach for sentiment analysis on Twitter incorporating the universal language model fine-tuning and SVM,” *Applied System Innovation*, vol. 5, no. 1, 2022, doi: 10.3390/asi5010013.

[13] H. Gil-Ramírez and R. M. Guilleumas-García, “Methodology for sentiment analysis in twitter posts about mobile learning,” *South Florida Journal of Development*, vol. 2, no. 2, pp. 2718–2728, Jun. 2021, doi: 10.46932/sfjdv2n2-125.

[14] G. B. Priya and M. U. Rani, "A framework for sentiment analysis of Telugu tweets," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 6, pp. 523–525, Aug. 2020, doi: 10.35940/ijeat.f1602.089620.
[15] Y. Li and H. Fleyeh, "Twitter sentiment analysis of new IKEA stores using machine learning," in *2018 International Conference on Computer and Applications, ICCA 2018*, IEEE, Aug. 2018, pp. 4–11. doi: 10.1109/COMAPP.2018.8460277.
[16] D. Kaneria and B. Patel, "Sentiment analysis for Twitter data," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 7S, pp. 98–101, 2020, doi: 10.35940/ijitee.g1019.0597s20.
[17] D. B. Savita and D. Gore, "Sentiment analysis on Twitter data using support vector machine," *International Journal of Computer Science Trends and Technology (IJCST)* –, vol. 4, no. 3, pp. 831–837, 2016.
[18] A. Amolik, N. Jivane, M. Bhandari, and M. Venkatesan, "Twitter sentiment analysis of movie reviews using machine learning technique," *International Journal of Engineering and Technology*, vol. 7, no. 6, pp. 2038–2044, 2016.
[19] Faizan, "Twitter sentiment analysis, " *International Journal of Innovative Science and Research Technology*, vol. 4, no. 2, Feb. 2019.
[20] P. Pandey, M. S. Mishra, and M. D. Rewarikar, "Real-time Twitter sentiment analysis using machine learning using different classification algorithm," *International Research Journal of Engineering and Technology*, pp. 1605–1609, 2020, [Online]. Available: www.irjet.net
[21] G. Srinivas, A. Lakshmanarao, S. Sushma, M. V. Krishna, and S. Neelima, "Fake news detection using ML and DL approaches," in *2023 International Conference on Circuit Power and Computing Technologies (ICCPCT)*, IEEE, Aug. 2023, pp. 1322–1325. doi: 10.1109/ICCPCT58313.2023.10245398.
[22] K. Maithili *et al.*, "An effective Twitter spam detection model using multiple hidden layers extreme learning machine," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 1s, pp. 1–9, 2024.
[23] S. C. Nistor, M. Moca, D. Moldovan, D. B. Oprean, and R. L. Nistor, "Building a Twitter sentiment analysis system with recurrent neural networks," *Sensors*, vol. 21, no. 7, p. 2266, Mar. 2021, doi: 10.3390/s21072266.
[24] K. Arun, A. Srinagesh, and M. Ramesh, "Twitter sentiment analysis on demonetization tweets in India using R language," *International Journal of Computer Engineering In Research Trends*, vol. 4, no. 6, pp. 252–258, 2017, [Online]. Available: www.ijcert.org
[25] Tweepy. (n.d.). Tweepy: Twitter for Python! [Online]. Available: https://www.tweepy.org. Accessed on November 21, 2023.
[26] TextBlob. (n.d.). TextBlob documentation. [Online]. Available: https://textblob.readthedocs.io/en/dev/. Accessed on November 21, 2023.
[27] C. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, no. 1, pp. 216–225, May 2014, doi: 10.1609/icwsm.v8i1.14550.

BIOGRAPHIES OF AUTHORS

Dr. Thotakura Venkata Sai Krishna received Ph.D. degree from Jawaharlal Nehru Technological University, Kakinada in 2018. He completed B. Tech. (CSE) and M.Tech. (Computer Science) from JNTUH in the year 2004 and 2007 respectively. His research areas include Data Mining and Bio informatics. He has several publications in national and International Journals and Conferences. He can be contacted at email: tvsai.kris@gmail.com.

Dr. Tummalapalli Siva Rama Krishna is a Ph.D. in Cloud Forensics, and currently working as Assistant Professor in the Department of Computer Science and Engineering at Jawaharlal Nehru Technological University Kakinada, India. He authored 3 books and obtained 4 patents. He published 23 research papers in reputed journals. He received three awards for his research works and four awards for his contributions in teaching. His areas of research include Cyber Security and Digital Forensics, Artificial Intelligence, and Blockchain. He obtained 10 professional certifications. He is a fellow of India School on Internet Governance and a diploma holder in Internet Governance. He can be contacted at email: srktummalapalli@gmail.com.

Srinivas Kalime currently working as Assistant Professor in the department of CSE at Sreyas Institute of Engineering and Technology, Bandlaguda, Nagole, Hyderabad. He completed his B. Tech and M. Tech in Computer Science and Engineering. He is pursuing Ph.D. in CSE at Osmania University, Hyderabad. His areas of interest are Cyber Security, Internet of Things (IoT). He is a life member of Indian society for Technical Education (ISTE). He can be contacted at email: srinivaskits966@gmail.com.

Chinta Venkata Murali krishna currently working as Associate Professor and HOD in CSE (Data Science) department at NRI Institute of Technology. He is a member of IAENG, IFERP, and INSC. He completed his M.Tech. in CSE in 2009 and is currently pursuing a Ph.D. in Computer Science and Engineering at GITAM (Deemed to be University), Vishakapatnam. He has published research papers in various conferences and journals and has been granted three patents with ten others in the pipeline for the grant. Four of his books have been published by international and national publishing agencies. He was awarded the "Best Researcher Award" from IOSRD in 2018. He can be contacted at email: muralikrishna_chinta2007@yahoo.co.in.

Sadineni Neelima currently working as Associate Professor in the department of Information Technology, Aditya Engineering Colege, Surampalem, Kakinada, A. P, India. She is pursuing Ph.D. from JNTU Kakinada. Her areas of interest are computer networks, machine learning, deep learning, and AI. She has published research papers in various conferences and journals. She can be contacted at email: neelima.sadineni@gmail.com.

Raja Rao PBV received his B. E and M.Tech in Computer Science and Engineering from Madras University and JNTUK Kakinada, respectively. He has obtained his Ph.D. degree in the Ad-Hoc Networks from SSSUTMS, Sehore. He has 18 years of teaching experience. At Present he is working as Associate Professor at Shri Vishnu engineering College for Women(A), Bhimavaram, and Andhra Pradesh. His areas of interest are cyber security, MANET, machine learning, deep learning, cloud computing, and network and system security. He can be contacted at email: rajaraopbv@gmail.com.