Precise Sentiment Analysis For COVID-19 Vaccination Responses Using BILSTM-MCNN Model

Md. Abdur Rahman Al-Mahadie [1], Rubaiyat Yasmin[1], Mousumi Haque[1,*], and Mirza A. F. M. Rashidul Hasan[1]

[1]Department of Information and Communication Engineering, University of Rajshahi, Rajshahi-6205, Bangladesh

mahadieice@gmail.com

sheba@ru.ac.bd

mouice@ru.ac.bd

mirza_ice@ru.ac.bd

[*]Corresponding Author

Abstract

In the past three years, the COVID-19 pandemic has become the main threat to mankind diversely affecting people all over the world. To mitigate the risk, the COVID-19 vaccination is now continuing throughout the world. COVID-19 vaccination has been an important topic of discussion throughout the world since the start of vaccination campaigns. In this paper, we have worked through a system methodology to evaluate public sentiment toward COVID-19 vaccination using social media platform such as Twitter dataset. This system methodology has analyzed the public sentiment using a lexicon-based approach named Valence Aware Dictionary for sEntiment Reasoner (VADER). To evaluate public sentiment more perfectly, we partitioned the sentiments into five classes named Very Negative Impression, Negative Impression, Neutral Impression, Positive Impression, and Very Positive Impression rather than the three sentiments used in general named positive, negative, and neutral. To classify the sentiment more perfectly, we have proposed the BILSTM-MCNN model which is a combination of Bidirectional Long Short-Term Memory (BILSTM) and Multiple Layers Convolutional Neural Network (MCNN). The Proposed BILSTM-MCNN model exhibits superior performance with a 96.82% accuracy for the Twitter dataset as well as outstrips other Machine Learning (ML) and Deep Learning (DL) models. This research provides a good projection of the thoughts of the mass population about COVID-19 vaccination and supports the aim of vaccinating most of the people worldwide.

Keywords

COVID-19, COVID-19 Vaccination, Sentiment Analysis, CNN, LSTM, BILSTM, CNN-LSTM

1. Introduction

The COVID-19 pandemic has been one of the most dangerous threats, and it is still a significant hazard. It has altered our perceptions of safety in many aspects of our life. Countries were required to implement many preventative measures for avoiding the enhancement of COVID-19, based on advice from international and local health organizations [1]. Home isolation, working remotely, and wearing face masks are just a few examples. But these are not the permanent solution for preventing COVID-19 from the world entirely. Vaccination was discovered to be the only way to effectively combat coronavirus and, in most cases, eliminate it. However, research done by WHO reveals that some people are still unwilling on getting the COVID-19 vaccination. Public faith in vaccination is a vital aspect of achieving an improved and maintained vaccination coverage rate [2]. To be successful, these vaccinations must not only be recognized as safe and efficacious by experts but also broadly accepted by the general population. Social media is a widely utilized forum for debating the spread of COVID-19 and the need for vaccinations [5]. People freely share their feelings on social media concerning different issues, containing the COVID-19 pandemic and its vaccination. Nowadays Twitter, Facebook, YouTube, and Reddit are famous social media platforms across the whole world. We choose the dataset from Twitter for our research on sentiment analysis. Twitter is chosen because it provides a wealth of information that is commonly utilized in sentiment analysis [9].

Many studies of sentiment analysis have evaluated the sentiments of people before and after the introduction of COVID-19 vaccinations [1-10]. The study [1] tried to find out the sentiments of the people of the USA and UK about COVID-19 vaccination before starting the vaccination program. They used the Twitter dataset of 2,43,883 unique tweets between 28 July to 28 August 2020. They found that amount of Negative tweets was 8.4%, Positive tweets were 28.2% and Neutral tweets were 63.4%. The researchers [2] studied 15000 unique COVID-19 vaccine-related tweets from Twitter in January 2021 for Sentiment and opinion analysis. The amounts of Positive, Neutral, and Negative tweets were 49%, 21%, and 30%. They used Support Vector Machine (SVM) and Naïve Bayes (NB). The accuracy of SVM was 78% without part-of-speech (POS tag) but decreased to 74% with POS tag. The accuracy NB was 77% without POS tag and increased to 80% with POS tag. The study [3] proposed a model using LSTM and Bidirectional LSTM (BILSTM) for predicting the sentiment classes. They considered COVID-19 vaccine-typed tweets from Twitter from January 2021 to June 2021. The percentage of Positive, Negative, and Neutral sentiments were 33.96%, 17.55%, and 48.49%, respectively. The prediction accuracy of LSTM and BILSTM was 90.59% and 90.83% respectively. The study [4] analyzed public sentiment on COVID-19 vaccination of people of Bangladesh using a manually prepared dataset. They prepared the dataset of 1647 responses by setting five questions on google form from February 16 to April 05, 2021. They used NB, DNN, LSTM, RNN, and BERT models for their analysis. NB showed 81% accuracy and the BERT model showed 86% accuracy which was the highest. The dimension of the dataset was the limitation of this research because by predicting only 1647 responses, the sentiment of people of the whole of Bangladesh about the COVID-19 vaccine was not predicted properly. The study [11] examined public sentiment on COVID-19 vaccine-related Reddit posts. They collected 2,66,840 Reddit posts from 20 January to 20 December 2020. They found that people started to spread COVID-19 vaccine-related misinformation during the announcement of the Pfizer vaccine and after starting the vaccination of the Johnson & Johnson vaccine, people started to think positively about COVID-19 vaccination.

In this paper, our contribution will be twofold. Firstly, we have proposed a model that determines the sentiment of tweets and defines the choice of the most fitting model to be used for classifying the sentiments of the tweets. This paper proposes the BILSTM-MCNN model which is a combination of BILSTM and Multiple Layers Convolutional Neural Network (MCNN) for sentiment analysis. Secondly, a comparative study based on the performance of our proposed BILSTM-MCNN model with the ML models and DL models has been executed and their appropriateness for the task undertaken in this paper is studied.

The outline of the paper is arranged as follows. Section 2 provides the system methodology of the proposed method. Section 3 discusses the proposed system and architectures. The results of the research have been described in Section 3. Section 4 finally concludes the paper.

2. System Methodology

In this paper, we have worked through a system methodology to assess the public sentiment on COVID-19 vaccination responses on Twitter. The outline of the system methodology is given in Figure 1. We consider the Twitter dataset from Kaggle containing tweets on the COVID-19 vaccine from people all over the world. For training the model, the first step is to prepare the dataset. Firstly, we preprocess the unprocessed tweets. Our primary goal is to classify the tweets into five sentiment classes. So, we have applied the preprocessed tweets to the sentiment analysis algorithm named Valence Aware Dictionary for sEntiment Reasoner (VADER) lexicon. From the VADER, tweets are classified into five different sentiments named Very Negative Impression, Negative Impression, Neutral Impression, Positive Impression, and Very Positive Impression. The data is then applied to the proposed BILSTM-MCNN model to justify the performance of the lexicon-based model.

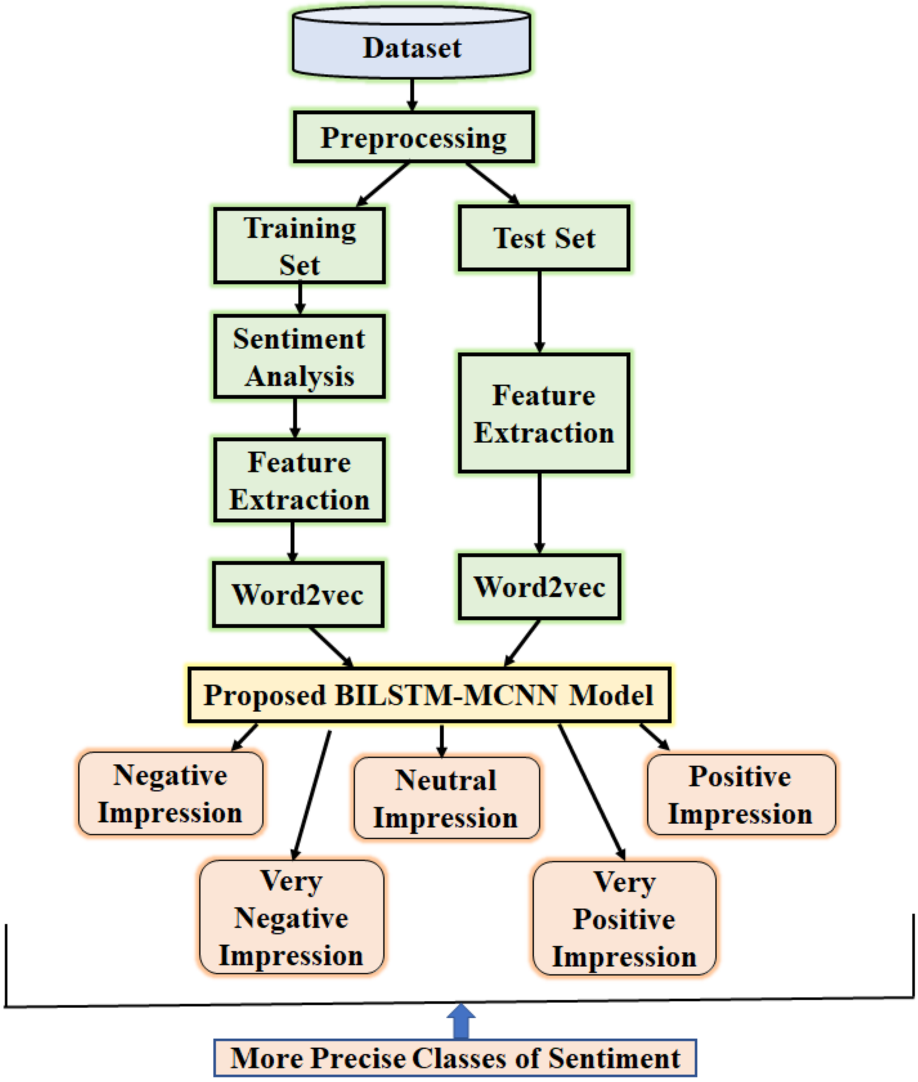

Figure 1. Full outline of the System Methodology

2.1. Dataset

In this research, we used Kaggle to collect the dataset "All COVID-19 Vaccines Tweets," which contains data from practically all notable vaccines. Retweet, user verified, date, source, retweets, hashtags, id, and text are all columns of this dataset. The tweets are from December 10, 2020, to November 23, 2021. The size of the dataset is 2,28,207 by 16.

2.2. Preprocessing

Data preprocessing aims to clean raw datasets by eliminating noise and uninformative sections, making them acceptable for data analysis and producing accurate and trustworthy findings [10]. In preprocessing the training datasets, we need to do the followings.

- Removal of Unicode strings from the tweets.
- Conversion of the upper case to lower case.
- Removal of the URL address.
- Removal of the hashtag in front of a word.
- Removal of integers.
- Removal of HTML special entities (e.g. &).
- Removal of tickers.
- Removal of hyperlinks.

- Removal of Punctuation.
- Removal of whitespaces.
- Removal of single space remaining at the front of the tweets.
- Removal of usernames.
- Removal of special characters.
- Removal of stop words.
- Tokenization
- Stemming
- Lemmatization

The tweets are being processed and ready for sentiment analysis.

2.3. Modified Sentiment Analysis Approach Using Vader Algorithm

In most of the cases using the VADER approach, Negative, Neutral, and Positive sentiment classes are classified [9]. In this research, we want to categorize the dataset into five sentiment classes named Very Negative Impression, Negative Impression, Neutral Impression, Positive Impression, and Very Positive Impression. The steps for classifying sentiment are shown in Figure 2.

When a sentence is run through Vader Lexicon, it detects sentimental terms and their intensity. The VADER defines the sentiment compound score by evaluating sentimental terms and the intensity of the text with the built-in algorithm. The sentiment compound score is the most often used metric in sentiment analysis; a sentiment compound score is a float value in the interval [-1, +1]. We then classify five sentiment classes namely Very Negative Impression, Negative Impression, Neutral Impression, Positive Impression, and Very Positive Impression by using the compound score.

The modified approach of sentiment analysis for the system is shown as follows:

Begin

Put the processed text into the VADER lexicon

Get compound score from VADER lexicon

Condition:

If (compound score< -0.5) **then**

Sentiment = Very Negative Impression

else If (compound score< 0 and compound score>= 0.5) **then**

Sentiment = Negative Impression

else If(compound score > 0 and compound score<= 0.5) **then**

Sentiment = Positive Impression

else If (compound score > 0.5) **then**

Sentiment = Very Positive Impression

else

Sentiment = Neutral Impression

condition end

End

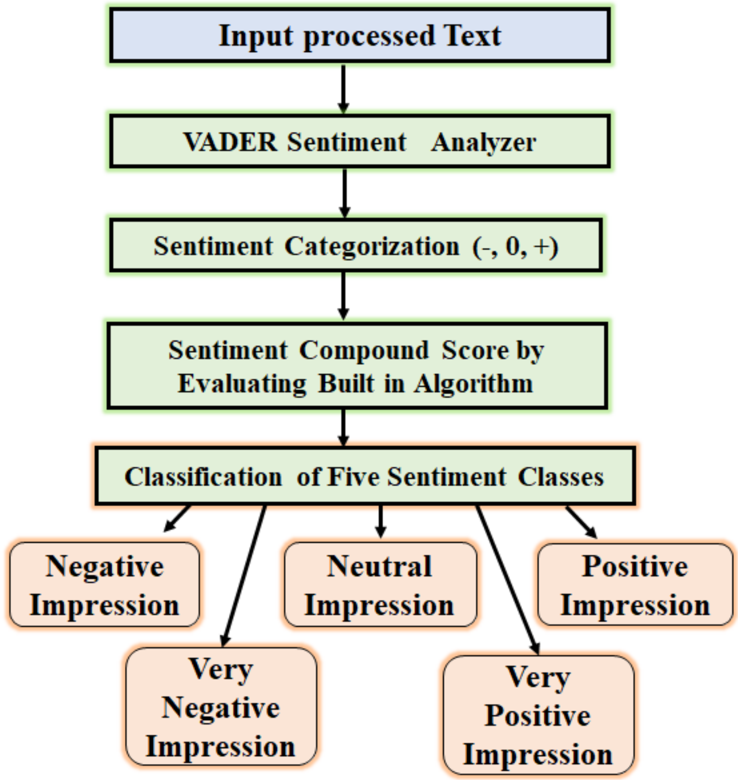

Figure 2. Block diagram of Modified Sentiment Analysis procedure

2.4. ML Models

Traditional ML classifiers such as SVM, Naïve Bayes (NB), K-nearest Neighbor (KNN), Random Forest (RF), Extreme Gradient Boosting (XGBoost), and Logistic Regression (LR) have used to get a benchmark result for our model.

2.4.1. SVM

SVM introduced in 1995, is a popular ML algorithm [12]. SVMs are originally used to address binary classification issues, and their fundamental job is to create a hyperplane. The function of a hyperplane is to optimize the space between two regions that defines the greatest spectral separation. SVM can handle both linear and complicated problems, as well as simple and sophisticated classification tasks. It can also handle separable and non-separable problems in both linear and nonlinear situations [13].

2.4.2. NB

The NB is a text mining classification algorithm used in sentiment analysis [14]. The major characteristic of the NB classification is that it generates a strong hypothesis for each condition or occurrence [14]. The presence of a character in a class is considered to be independent of the presence of any other characteristics by NB. The NB algorithm matches the data with the list of words to categorize the documents to their exact category [15].

2.4.3. KNN

In many instances, KNN is a non-parametric supervised classification approach that is simple yet successful. The distance between the test information and all of the preparation tests is utilized to ascertain it. The exemplary KNN utilizes Euclidean distance to figure out which of the "K" examples is nearest to another one. Experiencing the same thing, "K" alludes to the number of neighbors considered for deciding the class [16]. If the distance between the training samples and the query is less than or equal to the Kth shortest distance, the K closest neighbor may be used. Then pick the K texts that are nearest to the preparation set of archives and the text sets to be ordered.

2.4.4. RF

A RF algorithm combines numerous decision trees to provide better predictive performance than a single decision tree algorithm working on its own [17]. Every decision tree will vote for a specific category label for a given piece of data, and the category label with the most votes will be assigned to that data point. It starts with the foundation node and works its way down through an end node [18]. The data is subsequently placed in the class which corresponds to the end node.

2.4.5. XGBoost

XGBoost is a gradient boosted decision tree solution that improves rapidity and performance. XGBoost is notable for its ability to manage data that is uneven. Trees are shaped progressively in support, with each resulting tree expecting to limit botches from the earlier one [19]. Each tree builds on the knowledge of its ancestors and corrects residual faults [19]. XGBoost utilizes a packed segment-based structure in which the information is kept pre-arranged to prevent sorting the data again in each node[19]. Each characteristic just has to be sorted once this way.

2.4.6. LR

LR is a well-known method for resolving binary classification issues. Once the problem is changed to a binary classification type, LR may be used in a variety of domains, including text mining. The basic premise method is to compute the likelihood of a positive tweet based on rules found from a vast quantity of data. LR always predicts between 0 and 1. LR fits an S-shaped curve instead of a regression line which helps to predict between two maximum values (0 or 1).

2.5 DL Models

Traditional DL models like Convolutional Neural Networks (CNN), LSTM, BILSTM and CNN-LSTM were used to find the best sentiment classification.

2.5.1. CNN

CNN are DL models that use many layers of arrays to handle data. The fundamental difference between a CNN and a regular neural network is that a CNN receives input in the form of a two-dimensional array and analyses it rather than extracting features. The three main principles of a CNN are local respective fields, convolution, and pooling [20]. The input neurons are linked to the concurrent layers of the CNN. The weight of the hidden neuron is learned through layer connections, which are related to movement from one layer to the next. The shared weights are used to map these relationships between the input layer and the hidden layer [21]. The CNN is capable of capturing tight semantic relationships in text areas.

2.5.2. LSTM

LSTM network is used to recall the history of positive values over a short period when undertaking any sort of text analysis. The benefit of employing LSTMs is that the network will remember what it has read earlier, allowing it to have a greater comprehension of the input [22]. It could be able to handle phrases with shifting emotions in our instance. It has three gates: an input gate that reads the input, an output gate that writes the output to the following layers, and a forget gate that determines which data should be remembered and which should be forgotten. LSTM provides fine-grained control over memory by allowing us to manage how the current input influences the formation of the new memory, how memories influence the design of the new memory, and which elements of the memory are necessary for creating the output [22].

2.5.3. BILSTM

BILSTMs are based on bidirectional RNNs, which use two hidden layers to interpret sequence inputs in both forward and backward directions [3]. BILSTMs are a kind of LSTM in which two hidden layers are combined into a single output layer. A forward and backward LSTM layer is constructed in an unfolded BILSTM layer. The forward layer produces one output. The backward layer also produces another output. Two outputs produced are concatenated into one final output which becomes the final BILSTM layer output.

2.5.4. CNN-LSTM

CNN-LSTM model is the combination of CNN and LSTM model [23]. The architecture of the CNN-LSTM model is shown in Figure 3. An initial convolution layer in a CNN-LSTM architecture is responsible for accepting word embedding input. So, the embedded words are sent to CNN as input [24]. The output is made to a smaller dimension before being passed to an LSTM layer. The flatten layer takes the output of the LSTM layers and flattens them and turns them into a single vector and applies weights to predict the final classification decision [25]. The single yield value from the fully connected layer is passed through the activation layer Softmax to determine the text into any of five sentiment classes. The dropout layer is responsible for disregarding neurons at random during training [25].

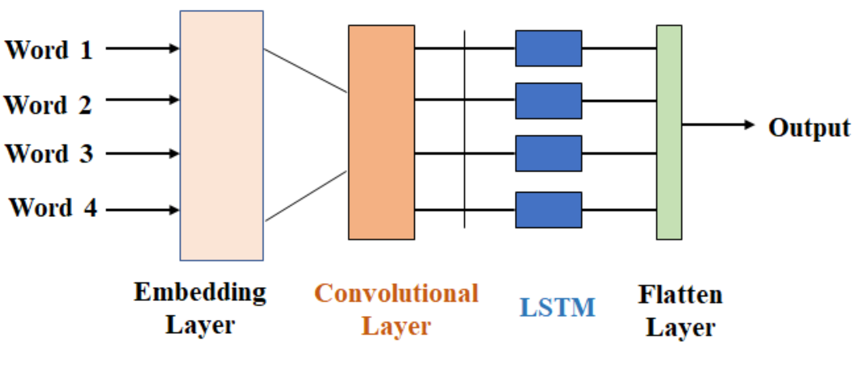

Figure 3. Architecture of CNN-LSTM model

3. Proposed Method

In this paper, we have proposed a deep learning model named BILSTM-MCNN model which is the combination of BILSTM and MCNN. The CNN can capture close semantic relationships in local regions of text, but BILSTM can capture long-term semantic dependence between the words and sequences of words. The architecture of the BILSTM-MCNN model is as shown in Figure 4.

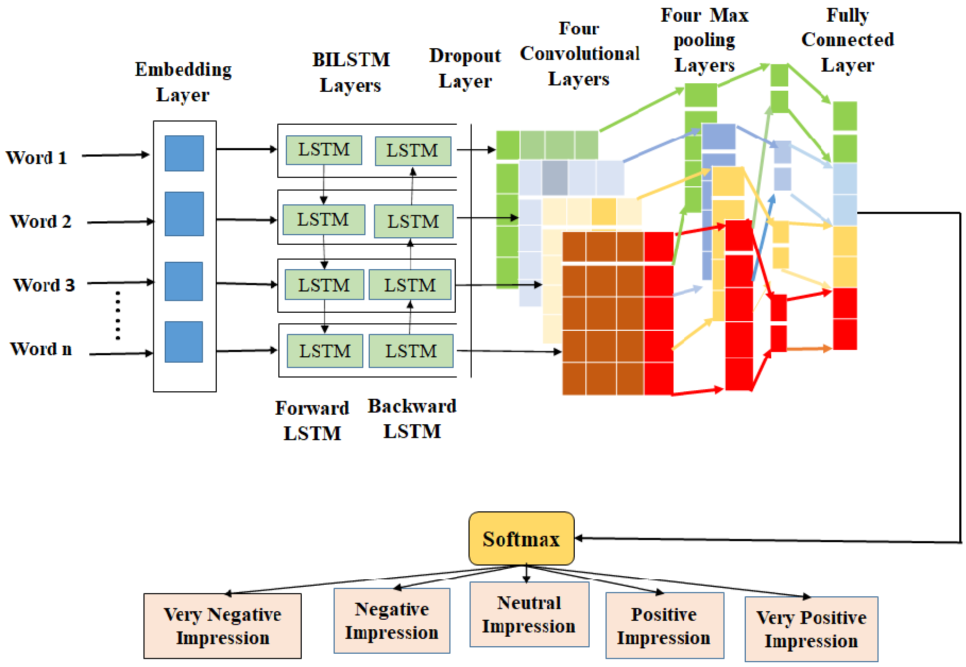

Figure 4. Architecture of proposed BILSTM-MCNN model

Firstly, the preprocessed text is fed into an embedding layer. The output of the word embedding is the input of BILSTM layers. BILSTM consists of one forward LSTM and one backward LSTM layer. BILSTM layer is responsible for the extraction of temporal features. After the BILSTM layer, four other convolutional layers are used for extracting spatial features. And between the four convolutional layers, there are three max-pooling layers. At the end of the fourth convolutional layer, there is a global max-pooling layer. Then there is a fully connected layer to flatten all outputs from previous max-pooling layers to turn the outputs into a single vector. Finally, the output of the model is obtained from a fully connected layer followed by a Softmax activation function resulting in any of five sentiment classes. The BILSTM processes words from word embedding. It analyses whole tweets and finds the similarity and long-term relationships between words starting from the beginning to the end of tweets whereas the CNN model identifies the short-term relationships between words from the beginning to the end of tweets. The description of all the layers of the proposed BILSTM-MCNN model is given below.

3.1 Embedding Layer

The input of our proposed BILSTM-MCNN model is a set of preprocessed tweets to predict its sentiment classes. The embedding layer is the input layer of the BILSTM-MCNN model. So, we have to pass the preprocessed input text to the embedding layer. The role of the embedding layer is to convert words to real values where each passed input becomes a matrix of real numbers. This conversion is based on a dictionary, named word embedding, where each word is represented by a vector of real numbers. In general, it is generated from techniques that take a set of textual documents to find a numerical representation of words where the distances between them are semantic ones. As Deep Learning needs numbers as input, we use Keras to generate word embedding that converts the text data into vector values. The embedding assumes that two words sharing similar contexts also have a similar meaning and consequently have a similar vector representation. The embedding layer receives three arguments, input dimension, output dimension, and input length. The size of vocabulary in the dataset is the input dimension. Output dimension defines the size of the output vector for respective words from this layer. Input length assigns the length of input sequences. We used the word embedding technique with BILSTM- CNN models because representing words into vectors provides better performance for natural language processing (NLP) problems.

3.2 BILSTM Layer

The proposed BILSTM-MCNN model contains an initial BILSTM layer which will get word embeddings for each token in the tweet as inputs. The output of the embedding layer is fed to the BILSTM layer to obtain BILSTM feature vectors. As in the BILSTM model, the memory cell is introduced to preserve information for a long period, it is expected that the feature vector created by the BILSTM layer carries the overall dependencies of the whole sentence. BILSTM uses two hidden layers to parse sequence inputs in both forward and backward paths. BILSTM blocs two hidden layers into one output layer. The final output of the BILSTM layer is obtained by concatenating both the forward and backward layer outputs. The final output of the BILSTM layer can be represented as a vector in the same way that the final output of an LSTM layer. The perception behind using the BILSTM layer is that its output tokens will store information not only of the initial token but also any previous tokens; In other words, the BILSTM layer is generating a new encoding for the original input.

3.3 Convolutional Layer

The output of the BILSTM layer is then fed into a convolutional layer. The convolutional layer is the most important unit in our model, which uses convolution kernels to convolve the inputs. Our BILSTM-MCNN model consists of four convolutional layers. The convolutional layers' employ rectified linear units (ReLU) to calculate the feature map. We have used Keras open-source python library for numerical computation. Convolutional layers in the convolutional neural networks play the role of a feature extractor that extracts the local features. The main function of the convolutional layer is to recommend the valuable features from the BILSTM layer based on the used activation function and create them in the feature map. Each filter slides over the BILSTM layer to extract features from the represented vectors for each text. The convolutional filter starts sliding from the beginning of the convolutional layer to the end. In each sliding step. In each sliding step, the filter generates the best features from the tokens using the activation function ReLU. Then, the output features were concatenated in feature maps as vectors.

3.4 Max Pooling Layer

In convolutional neural networks, the Pooling layer is one of the most commonly used layers. Dimension reduction for abstract representation is one of the uses of the pooling layer, which reduces the number of parameters required and, as a result, the calculation time of models. Max pooling is one of the most prevalent pooling structure models. The pooling layer minimizes the feature map dimension, which helps with cost computation in subsequent layers and captures the feature of tweet text efficiently. Between the four convolutional layers in our proposed BILSM-MCNN model, four max-pooling layers execute the downsampling process. It has two functions. The first is to minimize the parameters while keeping the dominating characteristics, while the second is to filter out the interference noise created by unconscious jitter. Max pooling layer minimizes the spatial size of the representation, reducing the number of parameters and training time. It aids in the reduction of overfitting as well as computing expenses. We employ one-dimensional pooling for CNN in this paper.

3.5 Fully Connected Layer

Fully connected layers convert multi-D feature maps into a 1D feature vector. Each node of the fully connected layer is connected with the nodes of the upper layer, thus the weight parameters of the fully connected layer may occupy the most. The convolution and max-pooling layers in BILSTM-MCNN model break down the text into features and analyses them independently. So, the fully connected layer takes the output of the previous max-pooling layers and flattens them, and turns them into a single vector that can be an input for the next stage takes the inputs from the feature analysis, and applies weights to predict the final classification decision.

3.6 Dropout Layer

The dropout layer is used to prevent the deep learning model from overfitting. We use dropout to prevent the network from overfitting. The dropout layer randomly sets the outgoing edges of a hidden unit to zero at each update of the training phase of the model. After the BILSTM layers, each BILSTM feature map vector passes through the dropout layer to prevent the overfitting of the neural network. It provides a technique to regularize this deep learning model. Also, it improves the generalization techniques for the network to equally consider all the inputs in the convolutional layers without focusing on a specific one.

3.7 Activation Function

We use the Softmax classifier as an activation function in our proposed BILSTM-MCNN model. The Softmax classifier converts the output of the upper layer into a probability vector whose value represents the probability of classes to which the current text belongs. The single yield value from the fully connected layer is passed through the activation layer Softmax to determine the text into any of five sentiment classes namely Very Negative Impression, Negative Impression, Neutral Impression, Positive Impression, and Very Positive Impression.

4. Results and Discussion

The results of our research work have been discussed in this section. Firstly, Sentiment analysis is done. Then we validate our proposed model by comparing with existing ML and DL models.

4.1 Sentiment Classification

In this subsection, we have analyzed the public sentiment on COVID-19 vaccination based on public responses on Twitter platform. Here we have classified the sentiment into five classes rather than three classifieds in general. Our partitioned five sentiments are Very Negative Impression, Negative Impression, Neutral Impression, Positive Impression, and Very Positive Impression. The sentiment analysis of the tweets is shown in Figure 5.

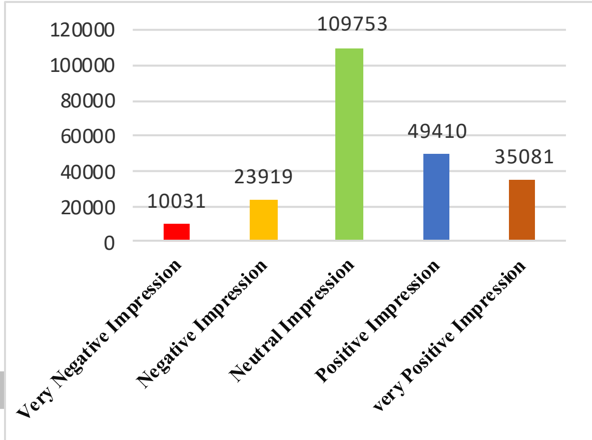

Figure 5. Sentiment analysis of tweets

The dataset has 2,28,227 unique tweets. Among which, 1,09,753 tweets have Neutral Impression which is the highest. The number of tweets having Very Negative Impression, Negative Impression, Positive Impression, and Very Positive Impression are respectively 10,031, 23,919, 49,410, and 35,081. From Figure 5, we can say that the greatest number of the people have neutral impression about COVID-19 vaccine. We have classified the sentiment of the Twitter dataset using a Pie diagram in Figure 6.

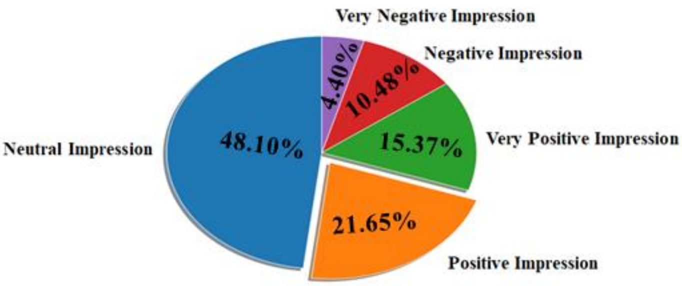

Figure 6. Pie diagram of Sentiment analysis of tweets.

From Figure 6, we have seen that the percentage of tweets which has a Neutral Impression is 48.10%. The percentage of tweets having Very Negative Impression, Negative Impression, Positive Impression, and Very Positive Impression are respectively 4.40%, 10.48%, 21.65%, and 15.37%. From the Figures 5 and 6, we can say that only 4.40 % people have very negative impression about COVID-19 vaccine whereas 21.65% people have positive impression and 15.37% people have very positive impression about COVID-19 vaccine and people with neutral impression is 48.10%. So, most of the people have neutral and positive sentiment about COVID-19 vaccine. So, COVID-19 vaccine is not a matter of fear for most of the people.

4.2 Performance of the Proposed BILSTM-MCNN Model

Here we have used the confusion matrix to evaluate the performance of our proposed model. The confusion matrix of BILSTM-MCNN model is given in Figure 7.

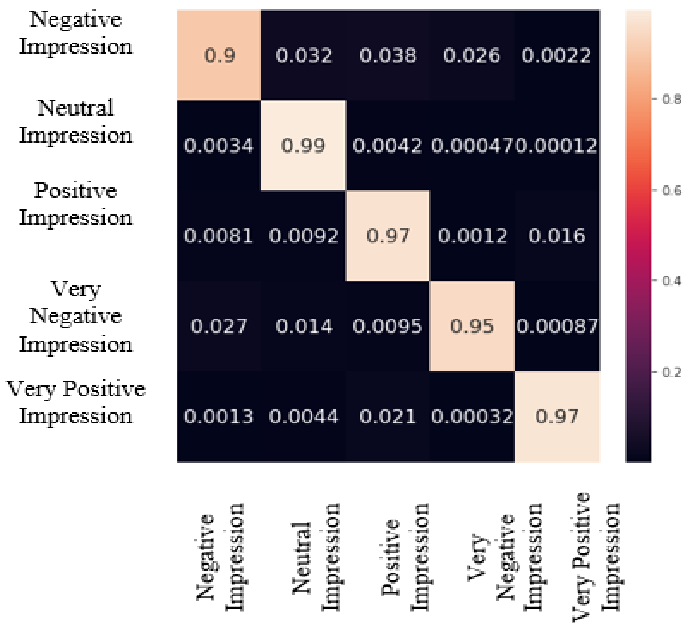

Figure 7. Confusion matrix of Proposed BILSTM-MCNN Model

From the confusion matrix of Figure 7, we have seen that the Proposed BILSTM-MCNN model has predicted the five sentiment classes with high accuracy. It has correctly classified 99% Neutral Impression, 97% Positive Impression, 97% Very Positive Impression, 90% Negative Impression, and 95% Very Negative Impression whereas 3.2%, 3.8%, and 2.6% Negative Impression are incorrectly classified as Neutral Impression, Positive Impression, and Very Negative Impression, respectively. 2.1% of Very Positive Impression are incorrectly classified as Positive Impression by the Proposed BILSTM-MCNN model whereas 2.7% of very Negative Impression are incorrectly classified as Negative Impression by the proposed BILSTM-MCNN model. From this discussion, we can say that our proposed BILSTM-MCNN model can correctly classify all the sentiment classes with high accuracy.

4.3 Comparison of the Proposed BILSTM-MCNN Model with Other Models

Here we have compared the proposed BILSTM-MCNN model with ML and DL models. The comparison performance of the Proposed BILSTM-MCNN model with ML and DL models has been evaluated based on accuracy, precision, recall, and f1-score. Here we have studied how accurately the proposed BILSTM-MCNN model could classify the sentiment classes than the other models. The comparison of models is shown in Table 1.

Table 1. Comparison of our proposed BILSTM-MCNN model with ML and DL models

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
SVM	83.23	78	74	76
Random Forest	80.21	80	80	79
Naïve Bayes	64.68	67	64	59
KNN	60.40	61	60	57
LR	80.29	80	80	79
XGB	62.53	67	63	57
CNN	95.22	95	95	95
LSTM	95.30	95	95	95

BILSTM	95.60	96	96	96
CNN-LSTM	96.01	96	96	96
Proposed BILSTM-MCNN	96.82	97	97	97

The comparison of the proposed BILSTM-MCNN model with other state-of-art models based on accuracy, precision, recall and f1-score metrics are shown in Table 1. According to the Table 1, our proposed BILSTM-MCNN model has achieved 96.82% accuracy for the Twitter dataset. The precision, recall and f1-score of the BILSTM-MCNN model are respectively 97%. So, our proposed BILSTM-MCNN model has showed highest accuracy, precision, recall and f1-score performance than other models.

4.3.1 Comparison Based on Accuracy Metric

The accuracies obtained by the proposed BILSTM-MCNN model using the Twitter dataset are compared with other ML and DL algorithms which is shown in Figure 8.

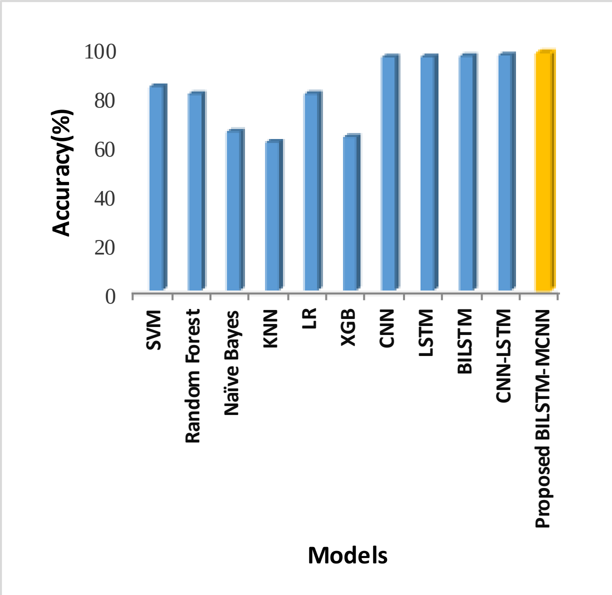

Figure 8. Comparison based on Accuracy Metric

The comparison of our proposed BILSTM-MCNN model with other state-of-art models based on accuracy metrics is shown in Figure 8. According to the Table 1 and Figure 8, we have seen that, our proposed BILSTM-MCNN model has achieved 96.82% testing accuracy. The testing accuracy of ML models: SVM, Naïve Bayes, KNN, Logistic Regression, Random Forest, and XGB are 83.23%, 64.68%, 60.40%, 80.29%, 80.21%, and 62.53% respectively. The accuracy of DL models: CNN, LSTM, BILSTM, and CNN-LSTM are respectively 95.22%, 95.30%, 95.60%, and 96.01%. From this discussion, it is clear that the ML algorithm can not perform well. The DL model has performed well than ML models. Our proposed BILSTM-MCNN model has higher accuracy than other ML and DL models.

4.3.2 Comparison Based on Precision, Recall and F1-score Matrices

The comparison of the BILSTM-MCNN model with other state-of-art models based on Precision, Recall and F1-score metrices is shown in Figure 9.

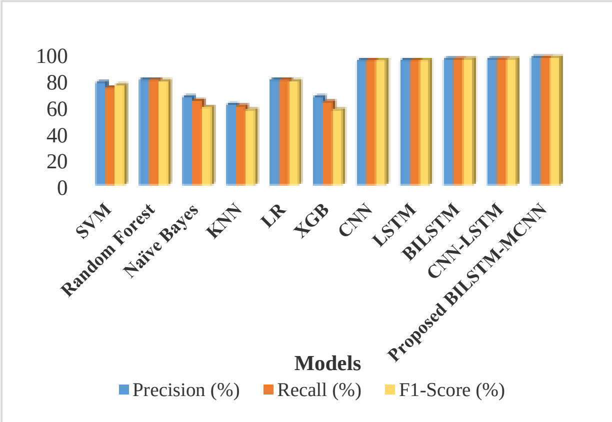

Figure 9. Comparison based on Precision, Recall, and F1-score Metrices

From Table 1 and Figure 9, we have seen that the ML model has less precision, recall, and f1-score percentage than DL models. The Precision of SVM, RF, NB, KNN, LR, XGB, CNN, LSTM, BILSTM, and CNN-LSTM models are 78%, 80%, 67%, 61%, 80%, 67%, 95%, 95%, 96%, and 96% respectively. The recall of SVM, RF, NB, KNN, LR, XGB, CNN, LSTM, BILSTM, CNN-LSTM models are 74%, 80%, 64%, 60%, 80%,63%, 95%,95%, 96%, and 96% respectively. The f1-score of SVM, RF, NB, KNN, LR, XGB, CNN, LSTM, BILSTM, CNN-LSTM models are 76%, 79%, 59%, 57%, 79%,57%, 95%,95%, 96%, and 96% respectively. The precision, recall and f1-score of our proposed BILSTM-MCNN model are 97% which is the highest. So, BILSTM-MCNN model shows better precision, recall and f1-score performance than other ML and DL models.

5. Conclusion

In this paper, we aimed at more precise classification of public sentiment toward COVID-19 vaccination using social media data with effective machine learning and deep learning approaches. VADER, a lexicon-based approach is first applied to understand the sentiment of people in five different classes. Our findings showed that people who tweeted has 4.40% Very Negative Impression, 10.48% Negative Impression, 48.10% Neutral Impression, 21.65% Positive Impression, 15.37% Very Positive Impression. To justify the performance of lexicon-based model, alongside our proposed BILSTM-MCNN model, ML models like SVM, KNN, Naïve Bayes, RF, LR and XGB as well as DL models like CNN, LSTM, BILSTM, and CNN-LSTM model have also been considered. From the experiment of the confusion matrix, it is seen that our proposed BILSTM-MCNN model has correctly predicted the five sentiment classes with high accuracy. Our proposed BILSTM-MCNN model has exhibited accuracy of 96.82% which is better than other ML and DL models. The precision, recall, and f1 score of BILSTM-MCNN model are 97% respectively. The BILSTM-MCNN model has showed higher accuracy of around 0.81% higher than CNN-LSTM model and around 1% higher precision, recall, and f1-score than the CNN-LSTM model. Our research work shows that the proposed BILSTM-MCNN model outperforms other ML and DL models. Hence, this paper provides a precise projection of thoughts of the mass population about COVID-19 vaccination and supports the aim of vaccinating most of the people around the world.

References

[1] J. Lappeman, K. Munyai & B. Mugo Kagina, (2021) “Negative sentiment towards COVID-19 vaccines: A comparative study of USA and UK social media posts before vaccination rollout”, *F1000Research*.

[2] N. N. Alabid & Z. D. Katheeth, (2021) “Sentiment analysis of twitter posts related to the covid-19 vaccines”, *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 24, No. 3, pp1727-1734.

[3] K. N. Alam et al., (2021) “Deep learning-based sentiment analysis of COVID-19 vaccination responses from twitter data”, *Computational and Mathematical Methods in Medicine*, Vol. 2021, pp1-15.

[4] A. Ahmed, C. Argho, D. Suben & K. Saha, (2021) “Sentiment analysis of COVID-19 vaccination from survey responses in Bangladesh”, *Research Square*, pp1-15.

[5] K. H. Manguri, R. N. Ramadhan & P. R. M. Amin, (2020) “Twitter sentiment analysis on worldwide COVID-19 outbreaks”, *Kurdistan Journal of Applied Research*, Vol. 5, No. 3, pp54-65.

[6] M. T. J. Ansari & N. A. Khan, (2021) “Worldwide COVID-19 vaccines sentiment analysis through twitter content”, *Electronic Journal of General Medicine*, Vol. 18, No. 6, pp1-10.

[7] R. Marcec & R. Likic, (2021) "Using twitter for sentiment analysis towards AstraZeneca/Oxford, Pfizer/BioNTech and Moderna COVID-19 vaccines", *Postgraduate Medical Journal*, Vol. 98, No. 1161, pp544-550.

[8] Z. B. Nezhad & M. A. Deihimi, (2022) "Twitter sentiment analysis from Iran about COVID 19 vaccine", *Diabetes and Metabolic Syndrome: Clinical Research and Reviews*, Vol. 16, No. 1, pp1-5.

[9] M. A. Mudassir, Y. Mor, R. Munot & R. Shankarmani, (2021) "Sentiment analysis of COVID-19 vaccine perception using NLP", in *Proc. Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp516-521.

[10] R. Adarsh, A. Patil, S. Rayar & K. M. Veena, (2019) "Comparison of vader and lstm for sentiment analysis", *International Journal of Recent Technology and Engineering*, Vol. 7, No. 6, pp540-543.

[11] N. Kumar et al., (2021) "COVID-19 vaccine perceptions: An observational study on Reddit", *medRxiv*. https://www.medrxiv.org/content/10.1101/2021.04.09.21255229v1

[12] F. Rustam, M. Khalid, W. Aslam, V. Rupapara, A. Mehmood & G. S. Choi, (2021) "A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis", *PLOS One*, Vol. 16, No. 2, pp1-23.

[13] K. Mohamed Ridhwan & C. A. Hargreaves, (2021) "Leveraging twitter data to understand public sentiment for the COVID-19 outbreak in Singapore", *International Journal of Information Management Data Insights*, Vol. 1, No. 2, pp1-15.

[14] Pristiyono, M. Ritonga, M. A. Al Ihsan, A. Anjar & F. H. Rambe, (2021) "Sentiment analysis of COVID-19 vaccine in Indonesia using Naïve Bayes algorithm", in *Proc. Annual Conference on Computer Science and Engineering Technology (AC2SET)*, pp1-15.

[15] V. A. Kharde & S. S. Sonawane, (2016) "Sentiment analysis of twitter data: A Survey of Techniques", *International Journal of Computer Application*, Vol. 139, No. 11, pp5-15.

[16] S. Hota & S. Pathak, (2018) "KNN classifier based approach for multi-class sentiment analysis of twitter data", *International Journal of Engineering and Technology*, Vol. 7, No. 3, pp1372-1375.

[17] Y. Al Amrani, M. Lazaar & K. E. El Kadirp, (2018) "Random forest and support vector machine based hybrid approach to sentiment analysis", *Procedia Computer Science*, Vol. 127, pp511-520.

[18] A. Munshi, S. Sapra & M. Arvindhan, (2020) "A novel random forest implementation of sentiment analysis", *International Research Journal of Engineering and Technology*, Vol. 7, No. 6, pp2821-2824.

[19] K. Afifah, I. N. Yulita & I. Sarathan, (2021) "Sentiment analysis on telemedicine app reviews using XGBoost classifier", in *Proc. International Conference on Artificial Intelligence and Big Data Analytics*, pp22-27.

[20] P. K. Jain, V. Saravanan & R. Pamula, (2021) "A hybrid CNN-LSTM: A deep learning approach for consumer sentiment analysis using qualitative user-generated contents", *ACM Transactions on Asian and Low-Resource Language Information Processing*, Vol. 20, No. 5, pp1-15.

[21] Venkatesh, S. U. Hegde, A. S. Zaiba & Y. Nagaraju, (2021) "Hybrid CNN-LSTM model with glove word vector for sentiment analysis on football specific tweets", in *Proc. International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pp1-8.

[22] S. Suriya & K. S. Meena, (2020) "Sentimental analysis of twitter using long short-term memory and gate recurrent unit", *International Journal of Scientific Research in Computer Science and Engineering*, Vol. 8, No. 1, pp1-6.

[23] A. Alabrah, H. M. Alawadh, O. D. Okon, T. Meraj & H. T. Rauf, (2022) "Gulf countries' citizens' acceptance of COVID-19 vaccines-A machine learning approach", *Mathematics*, Vol. 10, No. 3, pp1-20.

[24] I. Priyadarshini & C. Cotton, (2021) "A novel LSTM–CNN–grid search-based deep neural network for sentiment analysis", *The Journal of Supercomputing*, Vol. 77, No. 12, pp13911-13932.

[25] L. Khan, A. Amjad, K. M. Afaq & H. Chang, (2022) "Deep sentiment analysis using CNN-LSTM architecture of English and roman Urdu text shared in social media", *Applied Sciences*, Vol. 12, No. 5, pp1-18.