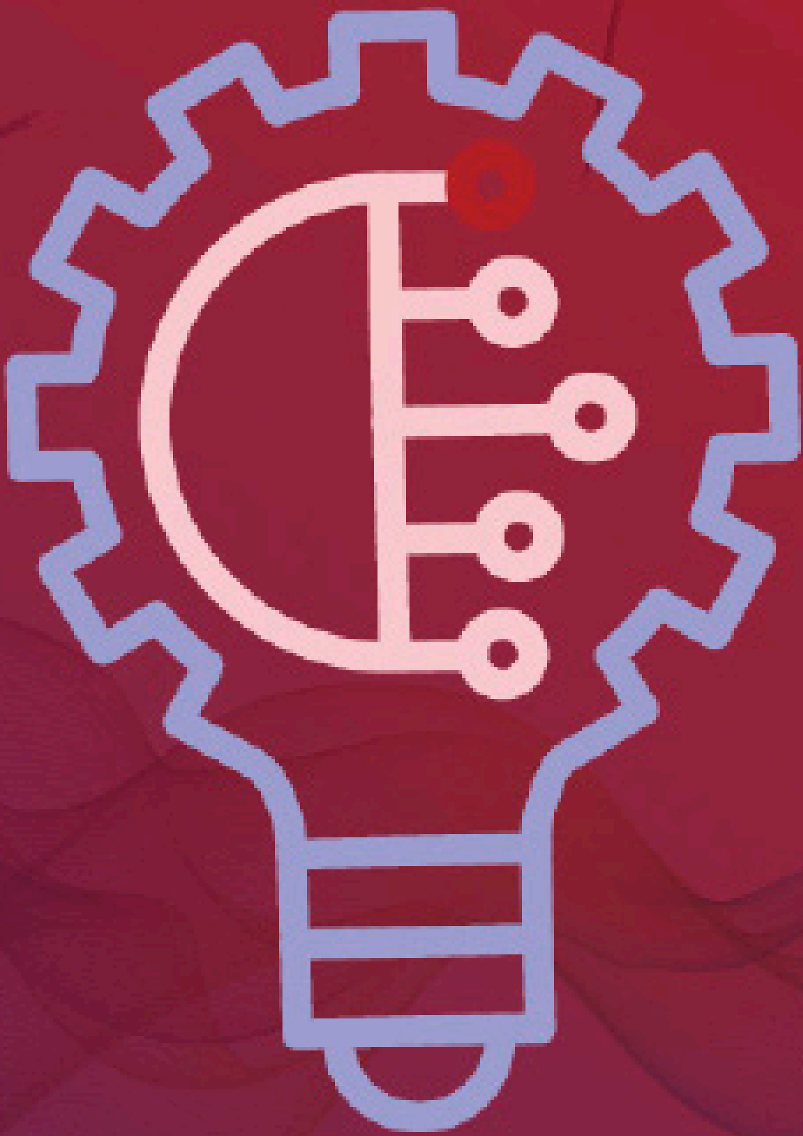
AMERICAN JOURNAL OF

ICT AND INNOVATION (AJICTI)

VOLUME 1 ISSUE 1 (2026)



PUBLISHED BY

E-PALLI PUBLISHERS, DELAWARE, USA

American Journal of
ICT and Innovation (AJICTI)

Volume 1 Issue 1, Year 2026
https://journals.e-palli.com/home/index.php/ajicti

Novel Model for the Sentimental Analysis of Twitter Data Using Deep Learning

Afrasiab Khan[1], Rabbia Mahum[1], Md Hamza[2*]

Article Information

Received: November 05, 2025

Accepted: January 14, 2026

Published: February 23, 2026

Keywords

Kaggle, LSTM, Machine Learning, Naive Bayes, Sentiment Analysis, Twitter

ABSTRACT

Sentiment Analysis is the scientific technique of studying various tweet data from Twitter, hence specifying them either positive, negative, or a neutral one. Numerous algorithms/ techniques exists for sentiment analysis like "KNN", "Naïve Bayes" and "Random Forest" which are based on machine learning or deep learning. Although, such models are employed with an accuracy much less than that required for a better and novel framework model. Therefore, it is very necessary to develop a light-weight, accurate and robust model with high precision and accuracy. Through this study, we have suggested an efficient and novel framework for sentiment analysis of Twitter data named as Improved LSTM Model. We modified/variated the base network of LSTM employing additional layers for better accuracy. We employed an improved LSTM technique for sentiment analysis in our improved model that categorizes the Twitter data better by classifying it into positive, negative, and neutral sentiments. Moreover, the proposed model is more compact due to its novel architecture than the base network. Furthermore, the performance of the traditional/old machine learning models named Random Forest, KNN, and Naïve Bayes based on Twitter data have been analyzed, however, the performance and results of our proposed DL model are much better than ML algorithms. We employed Kaggle dataset for the training/testing of the proposed model, which contains almost 1 million tweets. To evaluate the efficiency of the improved model, we utilized extensive experimentation which demonstrates that our algorithm beats existing sentiment analysis approaches, with an average accuracy of 90.

INTRODUCTION

In recent years, text-based document production has been increasing very vastly. There exist almost 120 microblogging sites on the internet and Twitter is one of the most famous in all. There exists a huge increase in the social media content on the internet and people express their thoughts and views in the form of messages and comments on social media websites like Facebook, YouTube, Twitter, etc. (Go, Bhayani, & Huang., 2009) considering Twitter a very famous and well-known social media platform among others, people can share and tweet their views about a particular subject, area, or current affairs. Moreover, their views or opinions comprise just 2 to 5 lines or maybe less than that, and that short view is known as a tweet. Mostly, up to 140 characters makes a complete tweet at maximum. These opinions may be concerning arts, entertainment, and fashion industry or any political affair as well as natural disaster emotions. It is the foremost step to pre-process and categorize the opinions of the tweets. If we try to understand the unstructured data or views, it can be very time-consuming as the tweets contain misspelled words, abbreviations, repeated words, and phrases that require to be filtered.

Thus, it is an era of science and technology, and today, around 8000 tweets are tweeted about a particular scenario every second. The scientific study of such tweets to specify and categorize them as positive one, negative, or neutral is called sentiment analysis (Mahum *et al.*, 2021). So, by definition, sentiment analysis is a natural language processing technique to systematically recognize, abstract, and measure affecting states and particular information. It is a procedure of analyzing and labeling the emotion or sentiment over any given review, tweet, or text piece to identify the sense of the opinion. Here, the main challenge is that we do not know if the expressed view or opinion in a document, or number of sentences is said in a positive, negative, or neutral way due to huge data (Mahum *et al.*, 2021). The visual representation of a sample of tweets and their sentiments is given in Fig. 1. In addition, some examples for the same are given in Table I. According to research, 90% of global data is unarranged and unstructured. Therefore, it is difficult to evaluate all the data in a timely and effective manner. So, the basic objective of our study is to review the views of people about different events and categorize them as positive, negative, or neutral reviews. So, we can say sentiment analysis is the method of analyzing information that responds to a text and expresses any opinion. It is a process by which data is extracted from people's perceptions, assessments, and feelings about organizations and events. In making decisions, others' opinions have a profound effect on customer satisfaction. The most important benefit of sentiment analysis is that it makes you understand the customers' feelings about the concerned product and it can help improve your product and services.

In Machine learning, sentiment analysis utilizes two major forms, which are supervised and unsupervised learning.

[1] UET Taxila, Punjab, Pakistan
[2] HITEC University Taxila, Pakistan
[*] Corresponding author's email: hamza67661@gmail.com

We don't provide training data in unsupervised learning while it is present in supervised learning. Sentiment analysis was introduced by Turney (Munir *et al.*, 2022; Mahum *et al.*, 2022). Traditionally, it was a classification task based on supervised learning and SVM is the most popular classifier among the algorithms which have been used for sentiment analysis. Most events for instance natural disasters like earthquakes, floods, accidental events, or political transitions such as terrorist attacks disturb people all over the world badly and in such events, it is very complicated to analyze the emotions of people during such crises. For example, during the September 11 attack, positive opinions were emerging first, and then negative reviews became part of social media platforms. With the implication of sentiment analysis, the users can face criticism about the services of a particular company or the company might be able to understand the feedback of the users as what they feel about the services of the company.

Sentiment analysis at different levels can be applied to sentences, documents, aspects, or user levels. But in our approach, we study the sentence-level sentiment analysis of Twitter data through the technique of machine / deep learning. Analyzing emotions or feelings is much more difficult in the processing of natural language as machines are to be trained just like the human brain to analyze and understand emotions. Some of the key barriers to sentiment analysis are tone, sarcasm, polarity, emojis, idioms, negations, comparative sentences, and audio-video tone.

The major contributions of the study are:

- To propose a novel and efficient algorithm that can analyze a Twitter dataset of any category and organize the tweets as positive, negative, and neutral as per perception. As per our knowledge, our proposed algorithm is the improved and novel successful implementation of the sentiment analysis with multiple layers, however, most of the existing techniques only work with a single layer of the network.
- The algorithm/technique is the modified form of a pre-defined deep learning framework named Long Short-Term Memory (LSTM). We have added two extra layers of the network that would increase the compactness and firmness of the network.
- To train and analyze the performance of our suggested system, the dataset comprising millions of tweets has been dispersed into training, authentication, and testing sets. The algorithm conquered 90% accuracy for the sentiment analysis and outperforms the traditional/existing sentiment analysis algorithms or deep learning models.

Figure 1: Diagram Showing Positive, Negative & Neutral Sentiments

Table 1: Tweet examples as positive, negative, and neutral emotions

Sentiment	Tweets
Positive	Good morning, Ali!!! I hope u have an amazing day :)
	That's overwhelming :) I'm super intense to hear/see it all.
Negative	I too much hate times square personally as it's too busy and boring.
	It is very good when you sat in a restaurant waiting for your favorite food and there is a line forming to criticize a manager.
Neutral	I have learned from quarantine days that folks like to do home activities and exercises.
	I have maintained a list of all the people I know in my life and the places I've visited.

LITERATURE REVIEW

There exist various methods and techniques based on sentiment analysis by many researchers.

(Mahum, R., *et al.*,2022) used CBoW (stands for Continuous bag-of-words) to analyze Amazon user reviews data to extract sentiments achieving an accuracy of only 2% and also tried and word vector model CDSAWE. The review data was classified either as positive or negative. However, this model was quite slow in learning the sentiment word embedding. The results showed that the suggested model can be integrated into multiple other domains. (Akgül *et al.*, 2016) used aspect level sentiment analysis using 4 datasets giving out better accuracy than the previous ones. They combined different natural language processing techniques with neural models to implement the target. Different embedding techniques were used for this purpose. Implementation was done using Python Jupyter Notebook. The proposed methodology was quite effective enough as compared to the previous sentiment analysis techniques.

(Diamantini *et al.*,2019) has projected an Attention-Based Bi-directional CNN-RNN Data Model (ABCDM) and the effectiveness was examined through sentiment polarity detection. The experiment was conducted on review data as well as Twitter data attaining considerable accuracy. All work was done in Keras Library in Python. Firstly, the ABCDM model reviewed long reviews of the users, and secondly short tweets. This paper aimed to overcome all the drawbacks coming in the existing models like RNN and CNN. ABCDM was developed for the English language, however, it may also be used for supplementary languages.

(Téllez *et al.*, 2017) proposed a Syntactic Action-Rule-based Decision Regression algorithm shortly named SARDR, which is corpus-based sentiment analysis and gives 88% accuracy using Twitter-related data. The proposed method consisted of steps such as collecting data, removing unwanted data in pre-processing, checking missing values, features extraction (parts of speech, synonyms, idioms, etc.), tweets weightage assignment, and finally cataloging the tweets using SARDR algorithm. (Denilson Alves Singh *et al.*, 2007) surveyed different works on sentiment analysis for the Portuguese language. He examined different deep learning techniques to ensure accurate sentiment analysis. The Portuguese language uses four sentiment lexicons such as OpLexicon, LIWN Dictionary, SentiLex, and synsets. The paper mainly focused on opinion mining by considering different datasets. As it was a review paper, it analyzed a very flexible way of sentiment analysis, however, the main conclusion was that translating the text into the English language could be more effective than in the Portuguese language. (Lukas Steppen Turney. ,2002) analyzed various videos using SenticNet and classified video segments. They used a lexicon knowledge-based extraction approach to implement the target. SenticNet was used to extract the natural language concepts. A large multi-model dataset known as Muse-CAR was utilized and the video segments were classified into arousal and valence. (Pang *et al.*,2002) presented a Multi-model Sentiment analysis i.e. MuSe-Car for different car reviews, utilizing the datasets in terms of compilation and annotations. To collect data, different videos from YouTube were selected in a semi-automatic process having the keywords "car brand" and "review". The videos were analyzed by viewing around 10% of each video. The model consisted of around 300 audio-visual materials.

(Savage and Torgler, 2013) worked on unsupervised information and improved a semi-supervised tweet sentiment classification algorithm. Because of the popularity and novelty of deep learning approaches, authors in (Comito *et al.*, 2019) investigated the main reasons behind the rumors regarding Twitter data. LSTM network was presented for rumor stance detection and in the second stage, the supervised learning approach was supported using the classifiers like SVM, NB, DT, etc. Multimodal sentiment analysis based on text and video is examined using different architectures employing the bc-LSTM model acquiring greater accuracy in the results (Sindhu and Vadivu, 2021). Most of the recent work done by researchers is focused on crisis events, the authors explored an algorithm to look into tweets tweeted after an earthquake. In another research, the authors launched a hybrid approach to find out the sentiments from tweets related to the crisis in the time, when people were imprisoned and fraught for their lives during difficult times of crisis.

(Basiri *et al.*, 2021)performed excellent sentiment analysis of tweets during the crisis and dangerous events by using the Bayesian Network with greater accuracy. (Kavitha., 2021)analysis of opinions made from tweets is analyzed using Natural Language Processing Toolkit (NLTK). Some authors intended to perform the same on Twitter data i.e., Turkish and English Twitter data using the Doc2Vec algorithm. (Pereira., 2021)sentiment analysis during the most crisis moments all over the world i.e., COVID-19 Tweets is performed through the use of a deep learning-based model with a median accuracy and precision of 79%. The Naïve Bayes algorithm was trainned for categorizing the sentiments of tweets as positive, negative, or neutra. (Stappen *et al.*, 2021)three basic classifiers have been introduced to thoroughly examine the sentiments of Twitter data out of which two attained good accuracy. As it is an era of big data, many authors tried to apply a big data approach to the tweet's sentiments using the Hadoop framework as in (Stappen *et al.*, 2021).

Using numerous approaches ML like Max Entropy, Naive Bayes, and SVM, the authors tried to find sentiments behind tweets in which opinion mining was involved Carley *et al.* (2016). (Araque *et al.*, 2017) have used the K-mean Clustering approach using big data on Twitter to find out positive or negative opinions with an accuracy of 87%. Another effort done using various algorithms for sentiment analysis of Twitter text and messages. A new method known as HL-NBC has been used in an improved way for sentiment classification with the best accuracy of 82%.

(da Silva *et al.*, 2016)tried to perform sentiment analysis on the opinions using Text Mining approaches on two levels, first by searching the polarity words from the pool of words and secondly by training the machine learning algorithm. (Akhtar *et al.*, 2018) proposed model is based on the famous KNN algorithm. N-gram modeling method has been deployed for feature extraction and it was concluded by the authors that the proposed system performs than any other. (Poria *et al.*, 2018) there is a comparison between different techniques of deep learning like CNN, RNN, and LSTM, and a single framework has been evaluated. Different twitter datasets have been extracted directly from Twitter API containing reviews about KFC and McDonald's, and a model has been trained (Ragini *et al.*, 2018). Moreover, sentiment analysis has been performed on a dataset containing tweets about COVID-19 (Ruz *et al.*, 2020). (Huq *et al.*, 2017) different movie reviews are being pre-processed and trained using a model using a CNN and LSTM combination of to produce accurate.precise results. Last, but not least (Setik *et al.*, 2021), the DBN algorithm of DL has been applied to train the model and the tweets are based on COVID-19, which gives the best result. However, in the latest years works are concluded much more about sentiment analysis and we will be comparing our proposed system with the latest works at the end of the study. Here, in Table 2, we are presenting some of the latest works along with their full information about datasets and algorithms.

Table 2: Related work for Sentiment Analysis of Twitter Data

Paper	Description of Dataset	Machine Learning Techniques	Used Languages	Stated Accuracy (%)
Bilgin and Şentürk (2017)	Dataset # 1 1000 +ve and 1000 -ve reviews. Dataset # 2 700 +ve and 700 -ve reviews	Naïve Bayes, Support Vector Machine	English	81.14
Hintalapudi *et al.* (2021)	1 Dataset containing 100 positive and 100 negative reviews	Support Vector Machine	Chinese	62.90
López-Chau *et al.* (2020)	1 twitter dataset containing 11399 feedback items.	Support Vector Machine	English	77
Ali *et al.* (2021)	1 dataset containing movie reviews.	Support Vector Machine, regression	English	66.3
Kharde and Sonawane (2016)	Movie reviews dataset. 1000 +ve and 1000 -ve reviews	Support Vector Machine, Naïve Bayes	English	77.5
Khan *et al.* (2020)	1021 documents containing different domains' information	Support Vector Machine, KNN, Naïve Bayes	Chinese	82
Alsaeedi and Khan (2019)	Collection of 10 000 tweets from the Catalan referendum of 2017.	RVFL	Spanish	82.9
Rodrigues and Chiplunkar (2019)	A dataset consisting of 300 positive, 300 negatives, and 300 neutral tweets.	Support Vector Machine	Arabic	86.8
Mittal *et al.* (2015)	The Internet Movie Database dataset contains 1000 positive and 10000 negative tweets.	Naïve Bayes	English	81.42
Rathi *et al.* (2018)	1390 movie reviews, among which half are positive, and half are negative, approximately.	Hybrid Support Vector Machine	English	89

MATERIALS AND METHODS

In our study, a strong and novel framework for the sentimental analysis of Twitter data is proposed. As we mentioned earlier, sentiment analysis actually means checking whether the emotions, attitude, and state of mind of the particular message are positive, negative, or neutral. Therefore, each approach will involve the following steps after getting the Twitter data such as cleaning of the data using pre-processing techniques, training of the algorithm, testing of the model, and results in validation.

To accomplish the specified purpose, we have trained three distinct machine learning models for the same dataset taken from Kaggle. Then, to achieve our goal of best accuracy, we compared the accuracy of our improved deep learning model with these ML models by following these steps Data exploration, Data visualization, Training and Testing the models, and extracting and comparing the results with our improved model. The flow illustration of projected system is given in Figure 2.

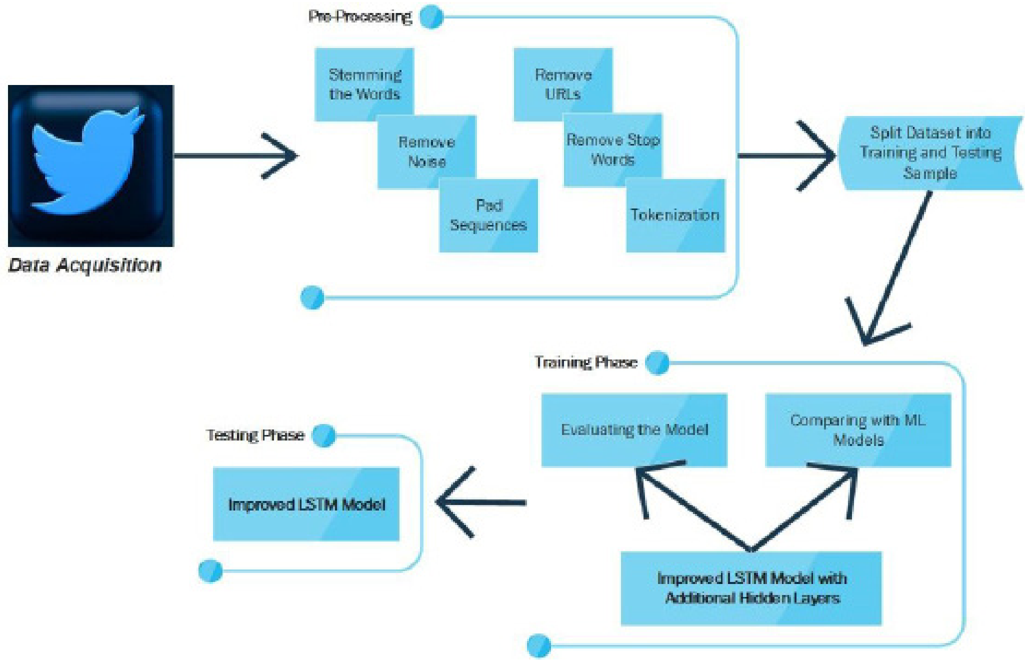


Figure 2: Flow diagram for the proposed model

Environment
The project is done in Anaconda Software using Python language. Table 3 represents the basic environment of our system. However, the various packages used in it are listed below with a minor description of each:

NumPy
Numerical Python using which all mathematical and logical calculations can be performed.

Pandas
A library used to work with relational and labeled data fields. It is used to read the CSV file.

Matplotlib
A graph drawing plotting library, used to visualize the data.
Seaborn
A data visualization library.

TextBlob
A package to process textual data in python.

NLTK
National language toolkit used to implement actions using human language data.

WordCloud
A data visualization technique used to represent data in an arrangement and to create a tag cloud.

Keras
Used in python language to implement neural networks easily.

Table 3: Environment of Proposed Model

System	**Lenovo**
RAM	4 GB
Torch	1.3.1
iNLTK library	$Default$
NLTK library	3.2.6
Pandas	1.4.2
Numpy	1.32.5
Scikit-learn	0.24.1
Keras	5.3.3
Tensorflow	2.4.4

Data Acquisition
The very foremost step in sentiment analysis of Twitter data is data acquisition. Data acquisition means collecting the data from a reliable source so that our task of achieving good accuracy can be performed. In the proposed system, the dataset is taken from (Goularas and Kamis., 2019) in a CSV file. Moreover, it is a subordinate of Google LLC as well as an online group of data scientists and experts. It permits many users to find and distribute their datasets and lets them train their models for various projects. We have imported all the important libraries in which the one pandas is imported to read the .csv file of a dataset and get information about the dataset. The exact value of the number of tweets is 1.04876 million Tweets. After reading the dataset and checking out the column labels of a dataset, we have to switch to pre-processing step.

Pre-Processing
Pre-processing is done to formulate samples and test data for the purpose of training. The tweets dataset

contains a lot of tweets and a wide variety of opinions and views given by people for a particular subject. The overall dataset is consisting 5 column labels, which are sentiment (like positive, or negative integer value), tweet id and date, flag, user information, and tweet text. As we have no work with Id, Flag, and User information, so we have simply dropped those columns to get a clean dataset. Additionally, the tweets may contain unwanted data which is of no use like URLs, hashtags, irregular random words, extra punctuation marks, short words, etc. Therefore, they are required to be removed to process further. Tweets consist of URLs, user names like @John, re-tweets, and sometimes duplicate words and smileys with tweeted text, therefore their removal is necessary to improve the performance of tweets. Another obstacle is the erratic casting like "HeLLo", in such a case, the text is converted to a proper case sentence. Moreover, some Twitter posts use prolonged words like "It is veryyyyy baddddd", such words are replaced by other suitable words. Slang words like "g8" and "w8" are removed or replaced. Stop words are the articles that appear with a high frequency in the dataset like "the", "a", and "an", such words are removed to compress the selected dataset. So, the tweets have been pre-processed for removing extra elements and getting a clean dataset, which may contain only the information essential for training/testing. In addition, we have divided pre-processed dataset tweets into training and testing sets keeping in view that there should not be overlapping between both sets. The training dataset is used to train the model and the testing data is used for performance evaluation and results of the novel model. In addition to this, the tokenization, lemmatizing and stemming phase is completed, and then the proposed model is employed.

Sentiment Analysis Steps

When all the requirements are done with a pre-processed dataset, then each tweet is then characterized by any of the emotions like positive, negative, or neutral.

Mathematical Expressions and Symbols

$$C = \begin{cases} \text{(Positive tweet) if sentiment score of the tweet} \geq 0.2 \\ \text{(Negative tweet) if sentiment score of tweet} \leq 0 \\ \text{(Neutral tweet) if sentiment score of the tweet is between 0 and 0.2} \end{cases}$$

Word Embedding

Word embedding is another name for the representation of text, most probably through vector representation. Now, words or phrases from the vocabulary are shown or converted as vectors of real numbers in a low-dimensional space. We have used a word embedding layer in our (LSTM) network. This idea can even be used for making the vectors of sentences or even whole documents. The recent inventions in this regard are ELMo and BERT. The major limitation of using word embedding is that words or phrases with multiple meanings are mixed into a single representation. So, polysemy and homonymy are not handled properly.

Tokenization, Lemmatization, and Stemming

In Python tokenization means the division of text into small lines called tokens, words, even the creation of non-English language words. Tokens are words, phrases, integers, or punctuation marks. In this step, small units are made by finding word boundaries, which are the endings of the current word and the beginning of the next word. Afterward, those tokens are considered the first step in stemming and lemmatization. There are many ways to make tokens using Python language such as using the split () function, using common expressions, using NLTK library, using Spacy library, using Keras library, or using Gensim. In our case, we used the split () function to make tokens. In natural language processing, we come across a situation where two or more words have the same root. For example, three words - agreed, agree, and agreeing with the roots of the same word agree. Searches involving any of these terms should be considered as the same root word. It is therefore important to combine all words with their roots. The NLTK library has ways to make this link and provide output that reflects the root word. Lemmatization is similar to stemming but brings context to words. So it goes further by linking words that have the same meaning in one word. For example, if a category has names like cars, trains, and cars, we will link everything to the car. Stemming is the process of minimizing word rotation in its roots such as arranging a group of words in a single stem even if the stem itself is not a functional word in the Language. There are various stemmers available with python Languages such as Porter Stammer or Lancaster Stammer, Porter Stemmer is the oldest one founded in early 1979. Lancaster Stemmer was founded in 1990 and uses a more aggressive approach than the Porter Stemming Algorithm. After tokenization, stemming, and lemmatizing, the dataset containing train and test data is ready for training and testing respectively.

Sentiment Analysis through ML Algorithms

Various methods/techniques have been introduced for sentimental analysis of Twitter data containing millions of tweets but here, in our study, we are only dealing with three of the algorithms in detail, which are described in detail below:

Random Forest Classifier

Random Forest (RF) algorithm is a supervised machine learning algorithm, that is mainly based on ensemble learning. By definition, Ensemble learning is a collection of different algorithms together to make it a powerful prediction model. In comparison to a single tree decision solution, it grows multiple trees and is based on a tree-structured classifier. It is an innovative version of bagging. After importing the desired dataset, exploratory data analysis is performed to check if there exist some common trends in the dataset or not. To make the classification model more accurate as well as to acquire a clean dataset, the pre-processing step is performed.

To remove slang words, punctuation, hashtag (#), @ username, emails, etc., cleansing of data is employed. It reduces noise in the dataset. To change the uppercase letters into lowercase, case folding is utilized. In addition to this, tokenizing is done to separate each word (Goularas and Kamis., 2019). Furthermore, in the stemming step, the aim is to convert all the affixes into simple words. After pre-processing technique, TF-IDF is implemented to further clean the dataset. By definition, TF means how many times a particular word comes in a single document. Therefore, for our concern, the TF value of the term I is as follows:

TFi = tfik (1)

Where tfi determines the frequency (how many times it appears) of term i in the document. As in the case of a very large dataset, the frequency can be large, so the formula takes the shape.

TFi = log2 (tfij) (2)

For the definition of IDF, many formulae have been proposed, however, the following formula given by Sparck Jones and Roberson is meaningful in our case.

IDFi=log2 (N/nj)+1=log2(N)- log2(nj)+1 (3)

Where N represents the total number of documents in the group while nj denotes the total number of documents that consists of at least the one-time existence of the term i.

The major steps involved in the random forest algorithm are such as bootstrap trials drawn from the original dataset. Moreover, for each bootstrap trial, unpruned classification is grown. Then by aggregating the predictions of ntree trees, new data is predicted by giving it positive, negative, or neutral sentiment. The data was separated into 6 groups, each one of which was labeled polarity wise i.e. positive sentiment, negative and neutral. Therefore, for public tweets, the supply of sentiments across all the tweets was like in the following graph. It is evident from the given chart that most of the tweets are negative.

KNN Algorithm

A supervised learning approach to find the emotions in a Twitter dataset has been deployed. In this approach, we used a variety of features for the classification like firstly word feature, secondly n-gram feature, thirdly pattern feature, and fourthly but last punctuation feature.

Firstly, in the Word feature, every tweet data is considered to be a binary feature El

(Rahman *et al.*, 2019). By using a dictionary, we found out which of the tweets are containing stop words and at last, we removed them as stop words do not need to be considered while sentiment analysis. Furthermore, if there exist two or more two punctuation symbols in a text, then they are considered to be word features El (Rahman *et al.*, 2019). To calculate feature weight, the following formula has been used:

ws= Nf/(coun (f)) (4)

Where Nf signifies the number of features existing in the tweet and count (f) represents total features in the whole dataset.

Secondly, in the N-gram feature, a queue of 3-6 words in a sentence is considered to be a binary n-gram feature, and then are they further processed.

Furthermore, thirdly, in the pattern feature, the words are classified into three types in pattern features such as high-frequency words (HFWs), content words (CWs), and regular words (RWs). So, to consider it a word frequency to be a high-frequency word if and only if frf > FH. In addition to this, if frf < FC then the word is considered to be a content word. However, the rest of the words are defined to be regular words. If frf ∈(FH,FH+ FC/2, then the word is categorized as HFW, and if frf ∈(FH+ FC/2, FC the word is categorized as CW.

Fourthly, in punctuation features, the data is separated and sorted into five general features like tweet length, question marks, exclamation marks, quotes, and capitalized words. The weight wp of the punctuation feature is calculated as below:

wp=(3*Np)/(Mp*(Mw+Mng+Mpa) (5)

Where Mw represents the word feature, Mng represents the n-gram feature and Mpa represents the maximum value for the pattern feature. In addition to these 4 features, we also used a feature named key-based feature, which gave us a better result. Therefore, after the pre-processing, the KNN algorithm has been applied and a better result of 88% accuracy is attained.

Naïve Bayes Classifier

Naïve Bayes Classifier is from a group of algorithms based on Bayes' Theorem in which each pair of features classified is independent of others. As, tweets are extracted and defined in the form of unstructured data, therefore these are required to be converted into a structured form, then features are extracted from them. Naïve Bayes is one of the best classifiers for feature classification while neglecting the grammar. First, data streaming is performed to acquire the tweets. Twitter has two types of APIS for the extraction of tweets. Firstly, for dumping old tweets, Search API is used while secondly, for dumping live tweets Streaming API is used. In our approach, the pre-processing has been done using Natural Language Toolkit. The sentiments are estimated based on the feature extraction of scored words. In pre-processing, filtering, tokenization, and removal of stop words are being done by which we get a clear dataset. Moreover, the Chi-square [44] method has been used to test the word scores of the features. This method lists all the positive and negative words in the form of a diagram. After that, the frequency distribution table of all the words is built. Suppose, the total number of tweets is 'n', the conditional probability or chances of class 'i' for tweets that contain 'w' is p (w), the percentage of tweets containing the class i be 'Pi', and the percentage of tweets which contain the word 'w' be 'F(w)'. γi is the procedure of measurement of the correlation amongst conditions and classes. Therefore, the γi2 of the word among word 'w' and class 'i' is defined as:

$$\gamma_i{}^2 = \frac{n, F(w)^2, (p1(w)=P_1)^2}{F(w), (1-F(w)), P_t, (1-P_1)} \quad (6)$$

Further, the Naïve Bayes classifier is used to predict. The Bayes theorem is given as follows:

P (H/X)= (P(H/X)P(H))/(P(X)) (7)

Where X represents the Tuples, H denotes the Hypothesis, P(H|X) represents the Posterior probability of H which is conditioned on X which means that it is the Probability that the Hypothesis holds given the value of X. Furthermore, P (H) denotes Prior probability of H i.e., the Probability that H holds irrespective of the tuple values, P (X|H) signifies the posterior probability of X which is conditioned on H i.e. the Probability that X will have certain values for a given Hypothesis, P(X) characterizes Prior probability of X i.e. the Probability that X will have some values.

The features which contain the highest score are considered to be the highest feature of the tweets. At last, one-third of the features are nominated for training and the rest of the tweets are selected for testing. Lastly, the Naïve Bayes classifier technique is trained with experimentation which gives better accuracy.

Improved LSTM Model

LSTM [45] abbreviated as Long-Short Term Memory is a deep learning-based method/procedure that has been implemented for sentiment analysis, linguistics, as well as text prediction. Somehow, it is also used for speech analysis. Being a special artificial neural network, it mainly works for omitting long-term dependencies in the concerned dataset. Taking a wide view, LSTM is an updated version of RNN (Recurrent Neural Network) to eradicate the most prominent problem of RNN which is the vanishing gradient problem. In any neural network in deep learning, the weights are updated and tested in the training phase through calculation of the error and the through back-propagation in the network. As far as our case is concerned, it is quite complex and difficult because we need to propagate by time to these neurons for training. In the architecture of LSTM, at the top, LSTM architecture contains a memory cell, whose job is to transfer the information from a particular time instance to the destination time instance efficiently. So, as compared to RNN, it can remember much more information efficiently and conveniently. By taking the help of values, we can add or remove any information.

In comparison to its predecessor RNN, LSTM has a special type of units as well as ordinary units. Three gates combine to make up LSTM, which are named as Input Gate as the first one, Forget gate as the second, and Output Gate as the third. These gates are mathematically represented as:

it = σ(WI[ht-1,xt] + bi) (8)

ft = σ(Wf[ht-1,xt] + bf) (9)

ot = σ(Wo[ht-1,xt] + bo) (10)

Where "σ" shows sigmoid function. Furthermore, "wi" shows the weight for the input gate,"wf" denotes the weight for forget gate, and "wo" represents the weight for the output neuron gate. Moreover, "wt" denotes the input at timestamp't'. "bi" represents the bias for the input gate, "bf" denotes the bias for forget gate and "bo" represents the bias for output gate. Figure 3 shows the basic building architecture of the block of LSTM.

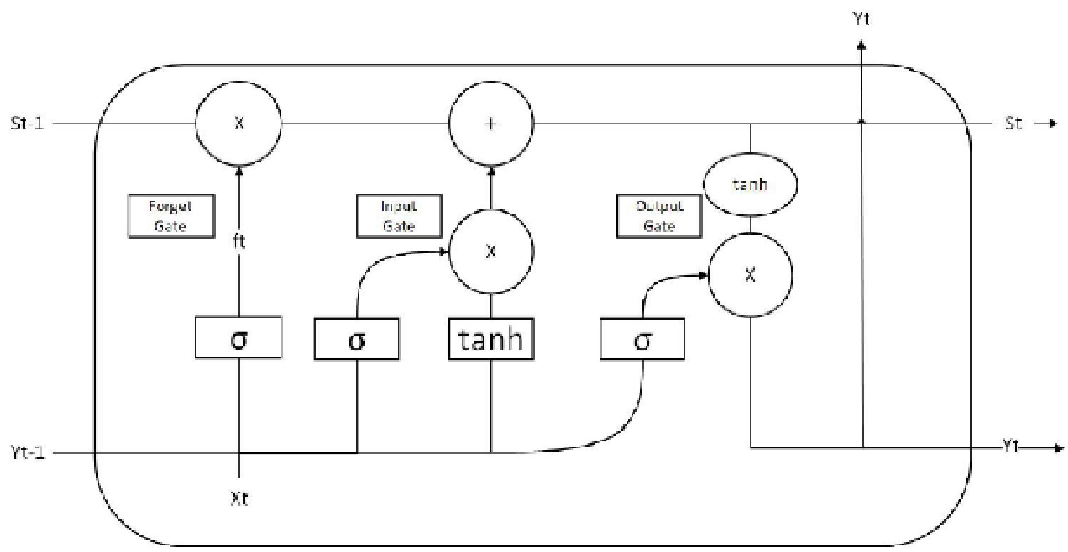


Figure 3: Basic Architecture of a Block of LSTM

In the figure, St-1 represents the memory cell, St Shows that information is carried to the next state through the memory pipeline. In addition to this, Yt-1 denotes output from the previous state whereas Yt denotes output from the previous state. Xt is the basic input given to the block of LSTM. As far as the working of the block of LSTM is concerned, it contains four layers that are used to interact with each other and their purpose is to produce an output of that cell as well as the cell state. Moreover, the two items are then passed on to the next upcoming hidden layer. As in RNNs, they have only a single neural network layer of tanh, LSTMs consist of three extra logistic sigmoid gates and one tanh layer. From a wide perspective, gates determine how the part of the information needed by the next upcoming cell is and also which part of the information needs to be discarded. Therefore, the output is usually in the range of 0-1. '0' denotes 'reject all' and '1' denotes 'include all'. The architecture summary of the layers of the LSTM model is given in the following Table 4:

Table 4: Layers of LSTM

Type of Layer	Shape of Output	Parameter #
lstm (LSTM)	(None, 70, 64)	38144
lstm_1 (LSTM)	(None, 70, 64)	33024
lstm_2 (LSTM)	(None, 70, 64)	33024
lstm_3 (LSTM)	(None, 70, 64)	33024
lstm_4 (LSTM)	(None, 70, 64)	33024
lstm_5 (LSTM)	(None, 70, 64)	33024
lstm_6 (LSTM)	(None, 64)	33024
Dense (Dense)	(None, 20)	1300

The original LSTM Model has a single hidden layer, which is further followed by a standard feedforward output layer. Our improved LSTM model is the extension of that original model that contains multiple hidden layers, and each hidden layer is composed of multiple memory cells. Adding extra hidden layers makes the model deeper and complex enough to accurately bring the result. So, in other words, adding extra layers make the model deep, and increasing the network depth provides an alternate way and this solution will require fewer neurons and will train faster. The addition of layers adds levels of abstraction of input observations over time. Each LSTMs memory cell requires a 3D input. When an LSTM processes one input sequence of time steps, each memory cell will output a single value for the whole sequence as a 2D array. To stack LSTM layers, we need to change the configuration of the prior LSTM layer to output a 3D array as input for the subsequent layer. We can continue to add hidden LSTM layers as long as the prior LSTM layer provides a 3D output as input for the subsequent layer.
The algorithm for the advanced suggested system is given below.

Algorithm 1. Improved LSTM Model
Input
A Twitter dataset from a standard website

Processing
Remove waste data from tweets

Output
A clean data of tweets without unwanted data

START

1. Take the dataset of tweets (in our case, taken from the Kaggle Website)
2. Read the full information of the dataset from the CSV file.
3. Remove extra columns from the dataset.
4. Remove user handles starting with @ and hashtags (#)
5. Remove numbers and special characters
6. Remove single characters
7. Remove all the null strings
8. Obtain the pre-processed term set
9. Use Lemmatizer and make a token for each word, which means to split letters (Tokenization)
10. Shows Stop words like a, an, and the and remove them
11. Using Word cloud, show the commonly used words in the dataset
12. Stemming the words
13. Combining words back to tweets Tokenizing and padding data
14. Creating the LSTM model
15. Training the model

END

Model Training
To train the proposed model with sentiments of positive, negative, and neutral, an efficient algorithm of LSTM for learning is employed, however, while building the model, we have added additional two layers of LSTM to make sure that the attained results are more accurate than the previous ones. The two layers are being added as hidden layers between input and output layers. Input layer is the embedding layer, which converts the tweets into vectors, and the output layer is the dense layer, whose job is to concatenate all the previous layers to give better accuracy of the model. More than 572, 81 parameters are being used for model training. After model training with proper epoch value, batch size, and validation split, testing is done to check the accuracy and results of our improved model. In addition, to perform sentiment analysis with proper results and comparison with our improved model, we have employed supervised learning approaches for the training of the model. The most common classifiers we used for our training model with machine learning techniques are the following:

1. KNN Algorithm
2. Naïve Bayes
3. Random Forest

These three algorithms' accuracy is then compared with our efficient and improved LSTM model. The major steps involved in the model are the same as normal sentiment analysis but additional layers of the network have been added to the model, which resulted in the improvement of the accuracy of the overall model.

RESULTS AND DISCUSSION
This section shows the details of the evaluation of the experimentation and the results of our suggested model. In section 4.1 dataset detail is discussed. In section 4.2 employed evaluation setup along with metrics is presented while in section 4.3 the overall performance of over proposed model is described. In section 4.4, we presented

the performance and accuracies of our employed ML algorithms.

Dataset

The dataset has been attained from a reliable data source Kaggle (Rathi *et al.*, 2018) which contains a total of 1 million tweets for certain situations. Here, in this section, we will be looking at the complete analysis of the dataset utilized for our proposed model. Therefore we analyzed our dataset to extract a percentage of positive, negative, and neutral tweets from our dataset and as per our analysis, our dataset contained 63% positive tweets, 30% negative tweets, and only 7% neutral tweets, which are graphically shown in Figure 4.

Apart from this, we also used Word loud library to graphically visualize the most frequently used positive and negative words from the tweets. We skipped the neutral, a reason that they were only 7% in all. The positive and negative tweets graphically in the form of a Word Cloud are shown in Figure 5 and 6 respectively.

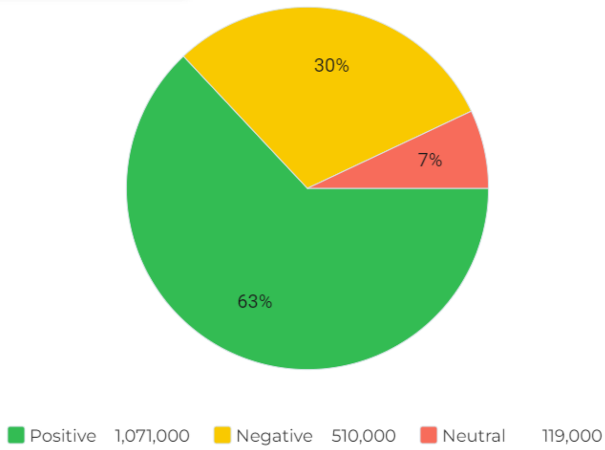


Figure 4: Pie Chart for Dataset Sentiments

Figure 5: Positive Tweets Word Cloud

Figure 6: Negative Tweets Word Cloud

Evaluation Setup

For the evaluation of the performance of our improved model, we separated the Twitter data into training and testing sets. The experiment was performed using Anaconda software with Jupyter Notebook IDE. Moreover, we have used Python language for the execution of the model. We have used 10 epochs value for training the model to get better accuracy. However, we have added two additional layers for the network to increase the accuracy up to 90%. All experiments were performed on Anaconda software with Python framework using Keras v0.1.1 library and many others mentioned in the environment section. We investigated the performance of the suggested model using the pre-processed cleaned dataset. However, it is noted that results are high and efficient in accuracy for two extra layers we added to the network of the model. It is noticed while experimenting that accuracy is rising as we increase the number of epoch values. Here we describe the accuracy, performance, time complexity and overall dataset result in the form of Table 5.

Table 5: Model Hypermeter

Parameters	Values
Dataset	Twitter Dataset (CSV)
Software Tool	Anaconda
Coding Language	Python
Epochs	40
Metrics	Accuracy

Metrics

The metrics employed for the evaluation of the anticipated model are True Positive (TP), True Negative (TNg), False Positive (FP), False Negative (FNg), True Neutral (TNt), and False Neutral (FNt). TP (True Positive) means the percentage of prediction that a tweet is taken from a concerned class and it fits into that class in real. TNg (True Negative) means the percentage of prediction of an observation that a tweet doesn't belong to a concerned class and it doesn't fit that class in reality. FP (False Positive) means the percentage of prediction of an observation that a tweet is taken from a negative class, however, it is predicted as it doesn't fit that class. FNg (False Negative) means the percentage of prediction of observation that if it is not taken from a negative class but it is predicted as it does fit to that class in real. TNt (True Neutral) means the percentage of prediction of an observation that belongs to a class but is not certain whether it is positive or negative. FNt (False Neutral) means the percentage of prediction of an observation that doesn't belong to a specific class, however also, not certain whether it is positive or negative.These four metrics have been thoroughly examined and deployed to build a confusion matrix for the result of the testing dataset. A confusion matrix represents the performance of a classification algorithm in the form of a table. It visualizes the summary of the classification algorithm. It shows the actual (by row) and predicted (by column) distribution of classes over the classes, which are, in our case, negative, positive, and neutral. Also, it helps in finding the correctness of our classification model as well as errors in it. To evaluate the performance of classifiers, we need to draw a confusion matrix. It is the summary of all the predictions made during a classification problem. The order of the confusion matrix is based on the number of classes it has (like positive, negative or neutral, or positive, negative, or neutral all). So, it means that if classes are N, the confusion matrix will have N x N blocks. Suppose x shows the actual class and y shows the predicted class. Confusion metrics equations for the individual class are shown below, where M is representing the matrix.

$$TPx=Mxx, \quad (11)$$

$$FPx=\sum^{n}_{(i=1)} Mix - TPx \quad (12)$$

$$FNx=\sum^{n}_{(i=1)} Mxi - TPx, \quad (13)$$

$$TNx=\sum^{n}_{(i=1)} \sum^{n}_{(j=1)} M_{ij} - TPx - FPx - FNx, \quad (14)$$

Table 6 shows a confusion matric for three classes, here, in our case, positive, negative and neutral.

Table 6: General Form of Confusion Matrix

Actual	Prediction		
	Negative	**Neutral**	**Positive**
Negative	True Negative (TNg)	False Neutral 2 (FNt2)	False Positive 1 (FP1)
Neutral	False Negative (FNg)	True Neutral (TNt)	False Positive 2 (FP2)
Positive	False Negative1 (FNg1)	False Neutral 1 (FNt1)	True Positive (TP)

In addition to the confusion matrix, the four metrics mentioned in this section are used for the evaluation of performance, which is based on values of the confusion matrix.

Accuracy signifies how many correct predictions have been made using anticipated model. The equation of accuracy is given below:

$$Accuracy= (TP+TN)/(TP+TN+FP+FN) \quad (15)$$

Recall also plays an important role in performance evaluation as it is a measure of the algorithm's completeness. A low recall is a sign of many false negatives. To calculate recall, the number of true positive values is divided by the number of false negatives values. Its mathematical form is given below:

$$Recall= TP/(TP+FN) \quad (16)$$

While evaluating a model, we also look at precision because precision is a measure of algorithm exactness. A low precision value is a sign of a large number of false positives. To calculate precision, the number of true positive values is divided by the total number of true positives values and false positives values. The equation for precision is written below:

$$Precision= TP/(TP+FP) \quad (17)$$

The F1 score shows the overall accuracy of the proposed model. As far as F-score is concerned, it is calculated from the values of precision and recall score. The mathematical form of the F1 Score is given below:

F1-Score=2* (precision*recall)/(precision+recall) (18)

Through proper calculations and model training, it is shown in Table VI that our efficient model attained 90% accuracy to categorize Twitter tweets as positive, negative, and neutral.

Performance of Our Proposed Method

Here, we are taking a look at the performance of the improved LSTM model after model training. Table 7 shows the overall training result of our proposed LSTM model and it contains the column labels such as No. of tweets and four metrics such as accuracy, precision, recall, and finally f1-score. Also, our proposed model has been compared with the one, on which work has been done recently and we noticed that our improved version of LSTM works better than every other model researched before.

Table 7: Performance of Improved LSTM

Method	No. of Tweets	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
LSTM [46](2021)	4035	79	80	79	79
Improved LSTM (2022)	1 Million	90	99	91	89

Table 7 clearly depicts that the proposed system attained 90% accuracy to categorize the tweets of Twitter data as positive, negative, and neutral. The precision of the system is 99% and the F1 score is 89%. Furthermore, in Figure 7, the confusion matrix for the testing sample of our dataset of improved LSTM model has been represented.

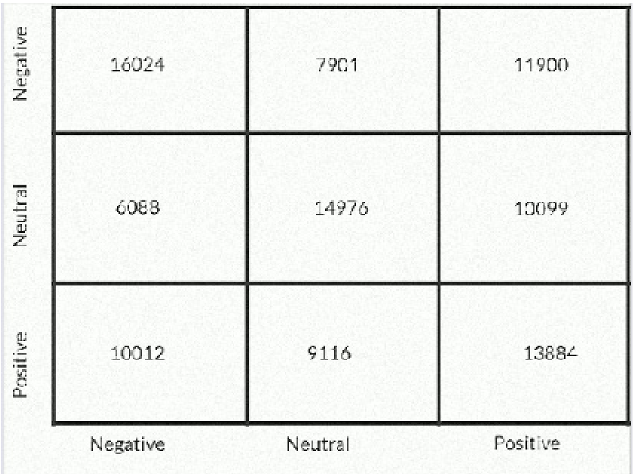


Figure 7: Confusion Matrix of Testing Sample of Dataset

Performance of our ML Algorithms

We employed three of the famous algorithms named Random Forest, Naïve Bayes, and KNN on the same dataset containing 1 Million tweets. The reason for doing this is to compare the accuracies of ML algorithms with the one improved version of our proposed LSTM Model. So, here, in Table 8 we are updating the accuracy of all machine learning algorithms we employed so far in our study.

Table 8: Performance of our ML Algorithms

Method	No. of Tweets	Precision	Recall	F-Score	Accuracy
Naïve Bayes	1 Million	0.84	0.99	0.91	82%
Random Forest Classifier	1 Million	0.87	0.98	0.92	75%
KNN	1 Million	0.81	0.98	0.88	84.32%

Comparison of Our Improved LSTM Model with our ML Algorithms

Through comparison of the results of our improved LSTM model with the existing ML algorithms that we have worked on. we can get a complete picture of the performance of the ML algorithms with our improved DL algorithm but to better analyze, we need to compare results easily represented by graphs. The overall comparison is shown visually in a graph in Figure 8.

It can be seen that our advanced proposed LSTM-based model trained on full test performs the best as compared to other ML algorithms.

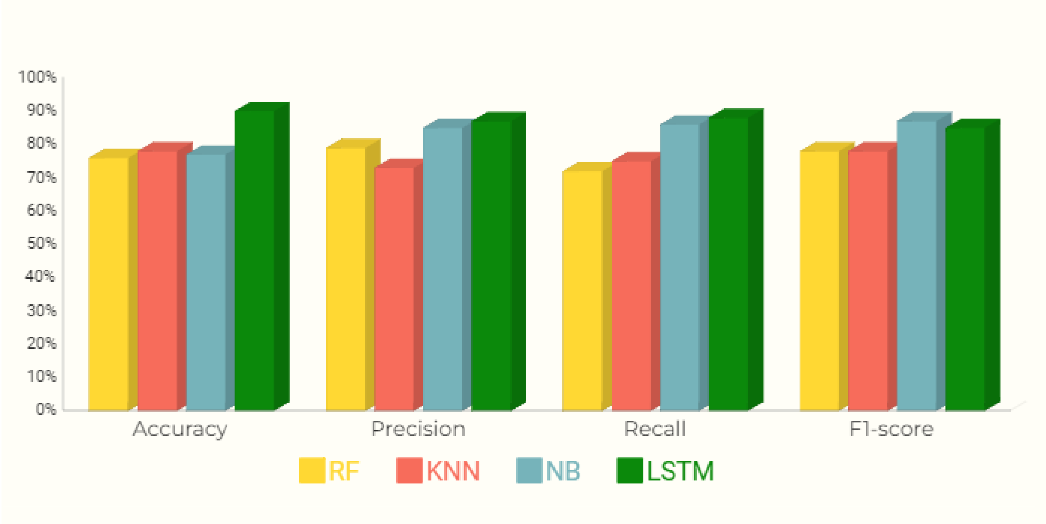


Figure 8: Improved DL model Comparison with ML Algorithms

Model Comparison with existing Sentiment Analysis Models

Here, we look at the evaluation procedure of the improved and novel model with existing methods of sentiment analysis employing binary classification. In comparison to all the algorithms depicted in Table IX, we have noticed that the accuracy of traditional models is not comparable and good enough with our improved LSTM Model. Moreover, in our case, we have employed fine parameters and used a dense architecture to give better accuracy. The improved LSTM Model employs four layers separated through input, output, and forget gates. In addition to the existing layers, we have added extra two hidden layers before the output layer to make our classifier more compact and efficient. Also, our proposed model uses less number of features as compared to traditional algorithms.

Here, in Table 9 we are comparing all the algorithms with our improved efficient LSTM Model.

Table 9: Comparison of Improved LSTM Model with Existing techniques

Reference	Year	Aim	Dataset	Algorithm	ML/DL	Accuracy (%)
[47]	2018	Sentiment Analysis of Twitter Data	1,600,000 Tweets	Hybrid SVM & Decision Tree	ML	82
[24]	2018	Sentiment Analysis of Twitter Data	1 Lakh Tweets	KNN Based Classifier	ML	86
[25]	2019	Sentiment Analysis of Twitter Data	32000 Tweets	Multiple CNNs and Bi-LSTM Model	DL	59
[26]	2019	Sentiment Analysis of Twitter Data	14000 Tweets containing 7000 McDonalds Tweets & 7000 KFC Tweets	Bagging	ML	74
[48]	2020	Sentiment Analysis of Twitter Data	3.5 Million Tweets containing world cup reviews	KNN	ML	87
[27]	2021	Sentiment Analysis of COVID-19 Tweets	1,305,000 Tweets were used to classify them into 8 emotions	Syuzhet	R Programming	70
[49]	2021	Sentiment Analysis of Twitter Data	Almost 25,000 tweets	LSTM-RNN	DL	80
[29]	2022	Sentiment Analysis of COVID-19 Twitter Data	47,000 Tweets of COVID-19 Opinions	DBN	DL	86

CONCLUSIONS

In our study, we presented an improved and robust framework for sentiment analysis and achieved significant performance using Twitter data. This proposed method is based on an improved LSTM network that is a more robust framework and effectively achieves the desired targets and goals. In this regard, we have worked on a dataset containing 27010 tweets for training three distinct machine learning models and another dataset taken containing 1 million tweets for the training of our improved deep learning model. We executed three Machine Learning Algorithms named Random Forest, KNN, and Naïve Bayes. Then after training the main model, we executed an improved deep learning model that gives 90% accuracy as compared to others. Moreover, our proposed technique is efficient and lightweight due to the implication of less number of layers and parameters. In the future, we would be exploring other sentiment datasets to achieve better results from other social media platforms and we will try to improve our training model through fine-tuning. In this research work, we only worked with English tweets and we did not consider the smileys and emoticons tweets. Therefore, we would take the following actions in our future work with some other languages like Chinese and French. To consider emoticons' tweets next time. The dataset contains more than 1 million tweets.

REFERENCES

Akgül, E. S., Ertano, C., & Diri, B. (2016). Twitter verileri ile duygu analizi. *Pamukkale University Journal of Engineering Sciences, 22*(2).

Akhtar, M. S., *et al.* (2018). No, that never happened!! Investigating rumors on Twitter. *IEEE Intelligent Systems, 33*(5), 8–15.

Al-Ayyoub, M., Essa, S. B., & Alsmadi, I. (2015). Lexicon-based sentiment analysis of Arabic tweets. *International Journal of Social Network Mining, 2*(2), 101–114.

Ali, G., *et al.* (2021). Public perceptions of COVID-19 vaccines: Policy implications from US spatiotemporal sentiment analytics. *Healthcare, 9*(8).

Alorini, G., Rawat, D. B., & Alorini, D. (2021). LSTM-RNN based sentiment analysis to monitor COVID-19 opinions using social media data. In *ICC 2021 — IEEE International Conference on Communications*. IEEE.

Alorini, G., Rawat, D. B., & Alorini, D. (2021). LSTM-RNN based sentiment analysis to monitor COVID-19 opinions using social media data. In *ICC 2021 — IEEE International Conference on Communications*. IEEE.

Alsaeedi, A., & Khan, M. Z. (2019). A study on sentiment analysis techniques of Twitter data. *International Journal of Advanced Computer Science and Applications, 10*(2), 361–374.

Annett, M., & Kondrak, G. (2008). A comparison of sentiment analysis techniques: Polarizing movie blogs. In *Conference of the Canadian Society for Computational Studies of Intelligence*. Springer.

Araque, O., *et al.* (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications, 77,* 236–246.

Basiri, M. E., *et al.* (2021). ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. *Future Generation Computer Systems, 115,* 279–294.

Bilgin, M., & Şentürk, İ. F. (2017). Sentiment analysis on Twitter data with semi-supervised Doc2Vec. In *2017 International Conference on Computer Science and Engineering* (UBMK). IEEE.

Carley, K. M., *et al.* (2016). Crowdsourcing disaster management: The complex nature of Twitter usage in Padang, Indonesia. *Safety Science, 90,* 48–61.

Comito, C., Forestiero, A., & Papuzzo, G. (2019). Exploiting social media to enhance clinical decision support. In *IEEE/WIC/ACM International Conference on Web Intelligence Companion* (pp. 244–249).

da Silva, N. F. F., *et al.* (2016). Using unsupervised information to improve semi-supervised tweet sentiment classification. *Information Sciences, 355,* 348–365.

Diamantini, C., *et al.* (2019). Social information discovery enhanced by sentiment analysis techniques. *Future Generation Computer Systems, 95,* 816–828.

El Rahman, S. A., AlOtaibi, F. A., & AlShehri, W. A. (2019). Sentiment analysis of Twitter data. In *2019 International Conference on Computer and Information Sciences (ICCIS)*. IEEE.

Gandhi, U. D., *et al.* (2021). Sentiment analysis on Twitter data by using convolutional neural network (CNN) and long short-term memory (LSTM). *Wireless Personal Communications,* 1–10.

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment analysis. *Entropy, 17,* 252–273.

Goularas, D., & Kamis, S. (2019). Evaluation of deep learning techniques in sentiment analysis from Twitter data. In *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*. IEEE.

Hamza, M., & Arshad, H. (2024). Effectiveness of Scrum Master in Agile projects. *International Journal of Engineering Applied Sciences and Technology, 9*(5), 51–60. https://doi.org/10.33564/IJEAST.2024.v09i05.006

Henríquez, P. A., & Ruz, G. A. (2018). Twitter sentiment classification based on deep random vector functional link. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE.

Hintalapudi, N., Battineni, G., & Amenta, F. (2021). Sentimental analysis of COVID-19 tweets using deep learning models. *Infectious Disease Reports, 13*(2), 329–339.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

Hota, S., & Pathak, S. (2018). KNN classifier-based approach for multi-class sentiment analysis of Twitter data. *International Journal of Engineering and Technology, 7*(3), 1372–1375.

Hota, S., & Pathak, S. (2018). KNN classifier-based approach for multi-class sentiment analysis of Twitter data. *International Journal of Engineering and Technology, 7*(3), 1372–1375.

Huq, M. R., Ali, A., & Rahman, A. (2017). Sentiment analysis on Twitter data using KNN and SVM. *International Journal of Advanced Computer Science and Applications, 8*(6), 19–25.

Kausar, M. A., Soosaimanickam, A., & Nasar, M. (2021). Public sentiment analysis on Twitter data during COVID-19 outbreak. *International Journal of Advanced Computer Science and Applications, 12*(2).

Kausar, M. A., Soosaimanickam, A., & Nasar, M. (2021). Public sentiment analysis on Twitter data during COVID-19 outbreak. *International Journal of Advanced Computer Science and Applications, 12*(2).

Kavitha, P. (2021). Twitter sentiment analysis using syntactic action rule-based decision regression. *Annals of the Romanian Society for Cell Biology,* 6037–6049.

Kazanova, M. M. (2016). *Sentiment140 dataset with 1.6 million tweets.*

Khan, M., Malviya, A., & Yadav, S. K. (2020). Big data approach of sentiment analysis of Twitter data using K-means clustering approach. *International Journal of Mechanical and Production Engineering Research and Development, 10*(3), 6127–6134.

Kharde, V., & Sonawane, P. (2016). Sentiment analysis of Twitter data: A survey of techniques. *arXiv*. https://arxiv.org/abs/1601.06971

López-Chau, A., Valle-Cruz, D., & Sandoval-Almazán, R. (2020). Sentiment analysis of Twitter data through machine learning techniques. In *Software Engineering in the Era of Cloud Computing* (pp. 185–209). Springer.

Luzuriaga, A. I., De Groot, G. D. A., Cortez, J. E., Babanto, N. U., & Ejusa, M. C. (2025). Parking occupant management system using QR code solutions with AES algorithm. *American Journal of Smart Technology and Solutions, 4*(2), 49–56. https://doi.org/10.54536/ajsts.v4i2

Mahum, R., *et al.* (2021). A novel hybrid approach based on deep CNN features to detect knee osteoarthritis. *Sensors, 21*(18), 6189.

Mahum, R., *et al.* (2021). A novel hybrid approach based on deep CNN to detect glaucoma using fundus imaging. *Electronics, 11*(1), 26.

Mahum, R., *et al.* (2022). A novel framework for potato leaf disease detection using an efficient deep learning model. *Human and Ecological Risk Assessment,* 1–24.

Mahum, R., *et al.* (2022). A robust framework to generate surveillance video summaries using combination of Zernike moments, R-transform and deep neural network. *Multimedia Tools and Applications,* 1–25.

Melville, P., Gryc, W., & Lawrence, R. D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Mittal, N., *et al.* (2015). A hybrid approach for Twitter sentiment analysis.

Mullen, T., & Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

Munir, M. H., *et al.* (2022). An automated framework for coronavirus severity detection using combination of AlexNet and Faster R-CNN.

Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv*. https://arxiv.org/abs/cs/0506075

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *arXiv*. https://arxiv.org/abs/cs/0205070

Patel, R., & Passi, K. (2020). Sentiment analysis on Twitter data of world cup soccer tournament using machine learning. *IoT, 1*(2), 14.

Patel, R., & Passi, K. (2020). Sentiment analysis on Twitter data of world cup soccer tournament using machine learning. *IoT, 1*(2), 14.

Pereira, D. A. (2021). A survey of sentiment analysis in the Portuguese language. *Artificial Intelligence Review, 54*(2), 1087–1115.

Poria, S., *et al.* (2018). Multimodal sentiment analysis: Addressing key issues and setting up the baselines. *IEEE Intelligent Systems, 33*(6), 17–25.

Ragini, J. R., Anand, P. R., & Bhaskar, V. (2018). Mining crisis information: A strategic approach for detection of people at risk through social media analysis. *International Journal of Disaster Risk Reduction, 27,* 556–566.

Rathi, M., *et al.* (2018). Sentiment analysis of tweets using machine learning approach. In *2018 Eleventh International Conference on Contemporary Computing (IC3).* IEEE.

Rathi, M., *et al.* (2018). Sentiment analysis of tweets using machine learning approach. In *2018 Eleventh International Conference on Contemporary Computing (IC3).* IEEE.

Reimers, N., & Gurevych, I. (2019). Alternative weighting schemes for ELMo embeddings. *arXiv*. https://arxiv.org/abs/1904.02954

Rodrigues, A. P., & Chiplunkar, N. N. (2019). A new big data approach for topic classification and sentiment analysis of Twitter data. *Evolutionary Intelligence,* 1–11.

Ruz, G. A., Henríquez, P. A., & Mascareño, A. (2020). Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers. *Future Generation Computer Systems, 106,* 92–104.

Saleena, N. (2018). An ensemble classification system for Twitter sentiment analysis. *Procedia Computer Science, 132,* 937–946.

Savage, D. A., & Torgler, B. (2013). The emergence of emotions and religious sentiments during the September 11 disaster. *Motivation and Emotion, 37*(3), 586–599.

Setik, R., Ahmad, R. M. T. R. L., & Marjudi, S. (2021).

Exploring classification for sentiment analysis from halal-based tweets. In *2021 2nd International Conference on Artificial Intelligence and Data Sciences (AiDAS)*. IEEE.

Sheikh, A., Sheikh, M. S., & Rinvee, T. M. (2025). Smart transportation systems with artificial intelligence: Enhancing efficiency, safety, and sustainability. *American Journal of Smart Technology and Solutions, 4*(2), 87–90. https://doi.org/10.54536/ajsts.v4i2

Sindhu, C., & Vadivu, G. (2021). Fine grained sentiment polarity classification using augmented knowledge sequence-attention mechanism. *Microprocessors and Microsystems, 81,* 103365.

Singh, V., *et al.* (2013). Sentiment analysis of textual reviews: Evaluating machine learning, unsupervised and SentiWordNet approaches. In *2013 5th International Conference on Knowledge and Smart Technology (KST)*. IEEE.

Singh, Y., Bhatia, P. K., & Sangwan, O. (2007). A review of studies on machine learning techniques. *International Journal of Computer Science and Security, 1*(1), 70–84.

Srikanth, J., *et al.* (2022). Sentiment analysis on COVID-19 Twitter data streams using deep belief neural networks. *Computational Intelligence and Neuroscience.*

Stappen, L., *et al.* (2021). Sentiment analysis and topic recognition in video transcriptions. *IEEE Intelligent Systems, 36*(2), 88–95.

Stappen, L., *et al.* (2021). The multimodal sentiment analysis in car reviews (MUSE-CAR) dataset: Collection, insights and improvements. *IEEE Transactions on Affective Computing*.

Tan, S., & Zhang, J. (2008). An empirical study of sentiment analysis for Chinese documents. *Expert Systems with Applications, 34*(4), 2622–2629.

Tanaka, H., Shinnou, H., Cao, R., Bai, J., & Ma, W. (2019). Document classification by word embeddings of BERT. In *Computational Linguistics: 16th International Conference of the Pacific Association for Computational Linguistics (PACLING 2019)* (pp. 145–154).

Téllez, E. S., *et al.* (2017). A case study of Spanish text transformations for Twitter sentiment analysis. *Expert Systems with Applications, 81,* 457–471.

Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *arXiv.* https://arxiv.org/abs/cs/0212032

Zhu, S., *et al.* (2013). Chinese microblog sentiment analysis based on semi-supervised learning. In *Semantic Web and Web Science* (pp. 325–331). Springer.