



Article

Analysis and Prediction of User Sentiment on COVID-19 Pandemic Using Tweets

Nilufa Yeasmin^{1,†}, Nosin Ibna Mahbub^{1,†} , Mrinal Kanti Baowaly^{2,†} , Bikash Chandra Singh^{1,*,†} ,
Zulfikar Alom^{3,†} , Zeyar Aung⁴ and Mohammad Abdul Azim^{3,†}

¹ Department of Information and Communication Technology, Islamic University, Kushtia 7003, Bangladesh; nilufayeasmin299@gmail.com (N.Y.); mahbub.nosin@gmail.com (N.I.M.)

² Department of Computer Science and Engineering, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj 8100, Bangladesh; baowaly@gmail.com

³ Department of Computer Science, Asian University for Women (AUW), Chattogram 4000, Bangladesh; zulfikar.alom@auw.edu.bd (Z.A.); azim@ieee.org (M.A.A.)

⁴ Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi 127788, United Arab Emirates; zeyar.aung@ku.ac.ae

* Correspondence: bikash.singh@ice.iu.ac.bd; Tel.: +880-1648583273

† These authors contributed equally to this work.

Abstract: The novel coronavirus disease (COVID-19) has dramatically affected people's daily lives worldwide. More specifically, since there is still insufficient access to vaccines and no straightforward, reliable treatment for COVID-19, every country has taken the appropriate precautions (such as physical separation, masking, and lockdown) to combat this extremely infectious disease. As a result, people invest much time on online social networking platforms (e.g., Facebook, Reddit, LinkedIn, and Twitter) and express their feelings and thoughts regarding COVID-19. Twitter is a popular social networking platform, and it enables anyone to use tweets. This research used Twitter datasets to explore user sentiment from the COVID-19 perspective. We used a dataset of COVID-19 Twitter posts from nine states in the United States for fifteen days (from 1 April 2020, to 15 April 2020) to analyze user sentiment. We focus on exploiting machine learning (ML), and deep learning (DL) approaches to classify user sentiments regarding COVID-19. First, we labeled the dataset into three groups based on the sentiment values, namely positive, negative, and neutral, to train some popular ML algorithms and DL models to predict the user concern label on COVID-19. Additionally, we have compared traditional bag-of-words and term frequency-inverse document frequency (TF-IDF) for representing the text to numeric vectors in ML techniques. Furthermore, we have contrasted the encoding methodology and various word embedding schemes, such as the word to vector (Word2Vec) and global vectors for word representation (GloVe) versions, with three sets of dimensions (100, 200, and 300) for representing the text to numeric vectors for DL approaches. Finally, we compared COVID-19 infection cases and COVID-19-related tweets during the COVID-19 pandemic.

Keywords: COVID-19; tweets; sentiment analysis; machine learning; neural network; natural language processing



Citation: Yeasmin, N.; Mahbub, N.I.; Baowaly, M.K.; Singh, B.C.; Alom, Z.; Aung, Z.; Azim, M.A. Analysis and Prediction of User Sentiment on COVID-19 Pandemic Using Tweets. *Big Data Cogn. Comput.* **2022**, *6*, 65. <https://doi.org/10.3390/bdcc6020065>

Academic Editor: Domenico Talia

Received: 26 April 2022

Accepted: 5 June 2022

Published: 10 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Coronavirus disease, also known as COVID-19, is a recent virus disease that emerged in 2019 [1]. Many patients with unexplained origin pneumonia appeared in Wuhan, China, in December 2019. These patients have been traced back to the Wuhan seafood, and wet animal wholesale market through contact tracing [2]. Chinese authorities performed a deep sequence analysis of the symptoms providing ample evidence that the novel coronavirus was the disease's causative agent of coronavirus disease (COVID-19). Since then, COVID-19 has spread rapidly in China and other countries worldwide. The World Health Organization has declared COVID-19 a public health emergency of international concern

and an epidemic. WHO has proposed two essential strategies to prevent this pandemic from spreading: isolation and self-quarantine. China implemented one of the most comprehensive lockdowns, shutting down 20 provinces and regions.

Since December 2019, when the world has been struggling with COVID-19 and most people have been placed on lockdown, at that time, people have used social media heavily to exchange their thoughts, opinions, and feedback on COVID-19. Various modes of social networking (for example, LinkedIn, Twitter, Facebook, and YouTube) have developed into different content categories. Twitter is a well-known microblogging social networking platform where users can send messages of up to 280 characters in duration, referred to as “tweets”. According to estimates, Twitter has over 330 million daily active users as of 2019. Every day, they send out over 500 million tweets [3]. Text content containing people’s opinions, thoughts, and feedback has appeared due to Twitter’s rapid growth. Moreover, people have been using Twitter to chat, share their emotions, and disseminate details about the crisis, whether it is cyclones [4], Ebola [5], floods [6], or Zika [7]. Twitter has grown in popularity as a platform for people to express their opinions on various topics.

The complexity in identifying the appropriate approach for mining and decoding these data is vital for natural language processing (NLP) research, also known as text sentiment analysis [8]. Text sentiment analysis primarily consists of text classification, knowledge retrieval, and text creation techniques [9]. Traditional text processing methods produce the vector representation of the text using the bag-of-words model [10]. Nonetheless, the standard bag-of-words paradigm suffers from grammatical and semantic loss. We have used a low-dimensional vector representation called Word Embedding to increase performance.

This paper examined the Twitter opinion of nine states in the United States (US) on COVID-19 from 1 April to 5 April 2020. In addition, we have developed popular machine learning (ML) and deep learning (DL) models to predict user feelings towards COVID-19 based on tweets. In ML techniques, we contrasted bag-of-words and term frequency-inverse document frequency (TF-IDF) to describe text to numeric vectors. We compared encoding techniques and two separate word embedding systems (word to vector (Word2vec) and global vectors for word representation (GloVe)) for calculating the numeric vectors from the text (i.e., tweets) in DL techniques. And finally, we performed a comparison between COVID-19 infection cases and COVID-19 related tweets during the COVID-19 pandemic.

In short, the following are the major contributions of our work:

1. We examined people’s emotions with COVID-19 by considering neutral, positive, and negative labels.
2. We used ML models to calculate the accuracy of various ML approaches to classify the user’s feelings about COVID-19 and show that the random forest provides a better result than other ML models.
3. We have expanded our focus on exploring DL models to classify the user’s sentiment about COVID-19, compute the DL models’ predictive performance, compare the ML models’ results, and show that Maximum times DL models provide a better result than ML models.
4. We try to relate COVID-19 outbreak cases and COVID-19-related tweets among the nine states in the USA.

The remaining paper is arranged as follows. Section 1 begins with a brief introduction. Section 2 gives a quick rundown of the related literature. A precise explanation of the whole methodology is illustrated in Section 3. Section 4 discusses the experimental findings. Finally, Section 5 outlines the conclusion and possible future work.

2. Related Works

Data from social networks are used in analytics to understand human behaviors [11–17]. During the COVID-19 pandemic, general people have faced a significant psychological burden because of long-term financial and social crises. It is essential to analyze public opinion to understand people’s sentiments and feelings facing the pandemic.

Sentiment analysis is an efficient approach for text analysis that automatically mines unstructured information for a sentiment like social media, emails, and customer service tickets. However, machine learning (ML) approaches can use various kinds of data to mine information automatically [18–23]. For example, Jain et al. [18] explore different measures for Twitter sentiment analysis using ML algorithms. A comprehensive methodology is specified for sentiment analysis. The multinomial Naïve Bayes and the decision tree models are employed as tools for analysis. The decision tree obtains the best results as evaluations showing 100% accuracy, precision, recall, and F_1 -score. From various countries, there are a large number of researchers trying to converge and distribute COVID-19 [19,24] Twitter datasets. Based on COVID-19 specific tweets, the authors of [11] use three different Twitter datasets to perform sentiment analysis. After collecting the datasets, data is preprocessed, TF-IDF is used for vector representation, and several ML models are used to predict sentiments. After evaluating, the decision tree provides the best accuracy compared to others and is 93%. The authors in [12] extract opinions from Twitter based on particular keywords and then use the Naïve Bayes classifier (NBC) algorithm to identify tweet sentiments.

Pokharel et al. [25] describe the Nepalese citizens' sentiments about the coronavirus outbreak. They collect tweets using keywords specified as CORONAVIRUS and COVID-19. The sentiment analysis is performed based on those tweets shared in Nepal from 21 May to 31 May 2020. In [26,27], authors have developed a mediative fuzzy correlation technique that can show the relationship between the increments of COVID-19-positive patients in terms of the increments concerning the passage of time.

In 2006, G.E. Hinton first proposed DL. Now DL is a subset of the ML process that refers to Deep Neural Network [28]. At present, the performance of DL algorithms yields effective natural language processing in sentiment analysis over multiple datasets. For instance, the authors of [29] proposed a model that combines a convolutional neural network (CNN) and long short-term memory (LSTM) to predict the sentiment of Arabic tweets. They gain F_1 -score is about 64.46% compare to the state-of-the-art DL model's about 53.6% F_1 -score. Goularas et al. [30] propose models that involve combinations of CNN and long short-term memory (LSTM) networks. Also, they compare two popular word embedding systems for vector representation, such as the Word2Vec and GloVe models. Their main contribution is sentiment analysis with the same dataset to analyze the performances and evaluate the process under a single testing framework. Ain et al. [31] critique and review several papers using DL techniques like convolutional neural networks, recursive neural networks, and recurrent neural networks/LSTM for analyzing user sentiment.

Cliche [32] develops two DL models(named CNN and LSTM) to predict binary classification for sentiment analysis using the pre-trained model and gets less than 73% accuracy. Chen et al. [33] propose a combined advanced LSTM-CNN model based on the model proposed by Sosa P M. and compared it with other combined LSTM-CNN models, achieving 78.6% accuracy. Ali et al. [34] apply sentiment analysis over a dataset of English movie reviews (IMDb dataset [35]) using DL techniques to classify these dataset files into positive and negative reviews.

Sosa P M [36] combine two deep learning models named long-short term memory (LSTMs) and convolutional neural networks (CNNs) to perform sentiment analysis based on Twitter data and finally compare their accuracy against regular CNN and LSTM networks. Moreover, their combined model CNN-LSTM gained (3%) better and 3.2% worse accuracy than standard CNN and LSTM models. Another proposed LSTM-CNN model gains 8.5% and 2.7% better accuracy than regular CNN and LSTM models.

3. Methodology

This section describes the comprehension involved in the study and all materials and processes that are used. We have pre-processed the raw data after collecting it to eliminate any irregularities. We used sentiment analysis to evaluate the sentiment of each document. Then we have extracted features using various techniques. Finally, we used

machine learning and deep learning models to classify user sentiments. Our approach is depicted in Figure 1. As previously mentioned, our whole procedure is outlined below.

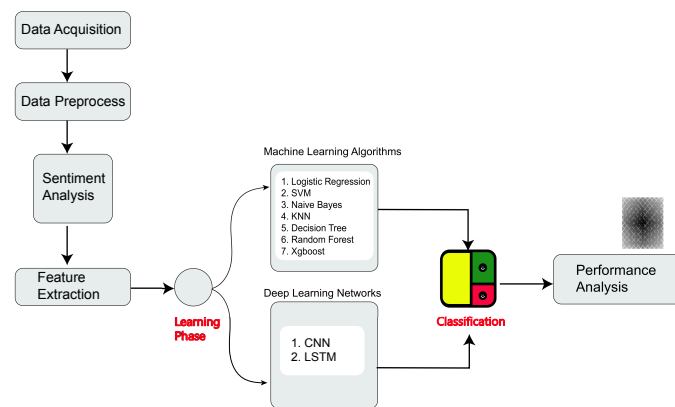


Figure 1. Proposed methodology and the workflow of our work.

3.1. Data Acquisition

In this work, we look at two distinct datasets. These two datasets are as follows:

Dataset-I. Dataset-I was obtained from Kaggle [37] and contains a large number of tweet texts on Covid19 that include the keywords “Corona”, “Covid19”, and “Coronavirus” (case ignored). We took 15 days of data from this dataset from 1 April to 15 April 2020, belonging to nine states in the United States, including Arizona, Washington, Florida, Georgia, Nevada, California, New York, Texas, and Illinois, for our research purposes. In this study, we considered the nine states of the United States since Twitter is the most popular social media site in the United States, with the greatest amount of tweets posted by users from these nine states. Due to computer resource constraints, we only used 15 days of data (1 April to 5 April) to conduct our research. Table 1 shows the number of tweets gathered from these nine states.

Table 1. DATASET-I.

Name of the State	Number of Tweets
Arizona	34,588
California	195,602
Florida	105,525
Georgia	51,098
Illinois	63,110
Nevada	24,926
New York	104,874
Texas	158,319
Washington	84,960

Dataset-II. To explore the association between the number of tweets and the number of covid cases, we obtained another dataset from Kaggle that contained the number of covid cases in each state of the United States, as shown in Table 2. To conduct the investigation, we used the same nine states as in Dataset-I: Arizona, Washington, Florida, Georgia, Nevada, California, New York, Texas, and Illinois. To perform a comprehensive analysis, we also looked at the number of covid cases identified in 15 days between 1 April and 5 April 2020.

Table 2. DATASET-II.

Name of the State	Number of Covid Cases
Arizona	238
California	791
Florida	977
Georgia	2302
Illinois	1164
Nevada	140
New York	874
Texas	2471
Washington	569

3.2. Data Processing

Twitter’s language model has its own set of properties. Raw tweets typically include much noise, misspelled words, and many abbreviations and slang phrases that limit our model accuracy. To improve accuracy and remove noisy features, we pre-processed the data. The following steps are performed to pre-process the dataset:

1. Firstly, we removed all forms of symbols such as #,@,!,\$,%,&, HTML tags, and numbers included in the whole dataset. We used a regular expression module from the Python language to perform these steps.
2. Our collected dataset contains both lower case and upper case words. We convert all words into lower case words.
3. Then, we performed tokenization on our whole text data. Tokenization means the division of smaller units of a comprehensive text document, such as individual terms or phrases [38].
4. Finally, we utilized stemming on our whole text dataset to get clean tweet text. Stemming is an approach for obtaining the root shape of terms by eliminating their affixes [39]. We utilized the NLTK library from Python to perform tokenization and stemming.

3.3. Sentiment Analysis

Analyzing a text and evaluating its sentiment is known as sentiment analysis. The aim is to assess whether or not user text conveys positive, negative, or neutral sentiments. We use the *TextBlob* library, which can process these three types of classification [40].

To get classification, *textblob* provides polarity(P) and subjectivity(S) value. When the polarity value is greater than 0 ($p > 0$), it is positive, and it is neutral when the polarity value is equal to 0 ($p = 0$). Otherwise, it is negative. The subjectivity is a floating-point integer of [0.0, 1.0], with 0.0 being highly objective and 1.0 being highly subjective. Each tweet is labeled with sentiments after these measures are completed.

For the sentiment label results, we can take some real-life examples of COVID-19. Let us consider the tweets *tweet1* and *tweet2* in Example 1 and Example 2 to deal with which label they belong to. More precisely, the probability of positive, negative, and neutral statements is shown below.

Example 1.

1. *tweet1* = *TextBlob*("The older people and others who have serious health problems are at higher risk of getting very sick from COVID-19");
2. *print*(*format*(*tweet1*.*sentiment*))
3. *Sentiment*(*polarity* = -0.2113, *subjectivity* = 0.625)

Generally, it is difficult to label, but *textblob* makes it easy to label. As we can see, the polarity value is -0.2113, and the subjectivity value is 0.625. Since the polarity is -0.2113, it indicates that the tweet is negative, and a subjective score of 0.625 suggests that it is subjective.

Example 2.

1. `tweet2= TextBlob("COVID-19 is bringing people closer during lockdown period.");`
2. `print(format(tweet2.sentiment))`
3. `Sentiment(Polarity = 0.0, Subjectivity = 0.0)`

The above sentiment has a polarity score of 0.0 and a subjectivity score of 0.0, indicating the statement is neutral and highly objective. However, in the manual approach, it is too hard to say, "is it positive or neutral?". For this reason, we use the `textblob` library function to get the labels in our dataset.

3.4. Feature Extraction

Feature extraction enhances the performance of trained models by extracting features from input data. We have performed several feature extraction techniques that convert text data into numeric vectors.

Traditional Bag-of-words (BoW): BoW model is essential for encoding and retrieving information (IR) in natural language. This model, which ignores grammar and word order while preserving multiplicity, identifies a text as a bag of its terms [41].

Term Frequency-Inverse Document Frequency (TF-IDF): The TF-IDF is a scoring metric used in information retrieval (IR) or overview [42]. The primary objective of TF-IDF is to emphasize the significance of a word in a given text. The definition is computed by combining the following metrics: (i) the number of instances in a text that a word appears and (ii) the word's inverse document frequency over a set of documents.

Word Embeddings: Word embedding is an art of vector representation. The background reason is to seize as much applicable, semantical, and syntactical information. Every word is represented as a numerical-valued vector in a predefined vector space. The most popular word embedding methods used to convert words into numerical vectors are BoW, TF-IDF, Word2Vec, GloVe, fastText, etc. We used two-word embedding methods for our conveniences, like Word2vec and GloVe embedding.

Word to Vector (Word2Vec): Mikolov et al. [43] proposed a well-known word embedding technique named Word2vec, and this method maps those kinds of words that have similar meanings and are close to each other. This technique uses two types of methods, and the first one is the skip-gram model, which accepts the center word as an input, sends them to an embedding layer, and then predicts the context words in a small dataset. The continuous Bag of Words (CBOW) model is the second one that uses context words as input, sends them to an embedding layer, and finally predicts the original or center word. CBOW works very fast and provides better representations for the most frequent words.

Global Vectors for Word Representation (GloVe): Pennington et al. [44] proposed a model very similar to word2vec and can be used to gain dense word vectors. However, the working methodology of GloVe embedding is slightly different from word2vec and is trained on an aggregated word-word co-occurrence matrix. A given corpus depicts the frequency of words co-occurring with each other. The basic system of methods of the GloVe embedding model is to create substantial word-context co-occurrence matrix pairs as every material in this matrix represents a word.

3.5. Classifier Models

Several classification methods have already been used to analyze user sentiment in online social networks. It is worth noting that the classifiers are primarily associated with (i) ML and (ii) DL techniques. We use eight classification models in this study, including six ML and two DL classifiers as described below.

3.5.1. Machine Learning (ML) Techniques

We used several ML algorithms in our study description as below.

Logistic Regression (LR): LR is a statistical method that employs a logistic equation in its simplest form that describes a binary dependent variable. However, there are many

more complicated variations [45]. In regression analysis, logistic regression is utilized to calculate the variables of a logistic method.

Support Vector Machine (SVM): The SVM is a plane-based classification algorithm that constructs a discrete hyperplane in the descriptive space of training data and components [46]. The instances or cases are categorized according to which side of the hyperplane they are on. SVM divides the hyperplane that moves through the center of the two groups in a linearly separable dataset, separating the two. SVM's main objective is to reveal the best hyperplane in training data between two data groups. By solving an optimization problem, SVM finds the hyperplane using the following equation [47]:

$$\max Q(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j d_i d_j X_i^T X_j \quad (1)$$

where, $0 \leq \alpha \leq C$ for $i = 1, 2, \dots, n$.

k-Nearest Neighbour (k-NN): The k-NN method is one of the most straightforward machine learning algorithms available. It is based on supervised learning techniques. It stores all the data available and, depending on the similarities, classifies a new data point. This ensures that if new data is acquired, it can be easily classified using the k-NN method [48].

Multinomial Naïve Bayes: The Naïve Bayes method is a technique that uses the Bayes theorem to handle classification issues. It is a probabilistic classifier, which means it predicts based on the probability of an object [49]. The probability of X observation belonging to class Y_k (for example, with X being a vector of word occurrences or word counts) is calculated using the following equation [50]:

$$P(Y_k|X) = \frac{P(Y_k)P(X|Y_k)}{P(X)} \quad (2)$$

The multinomial Naïve Bayes classifier is an improved version of the Naive Bayes classifier that is primarily used for text [51].

Decision Tree (DT): A DT is a tree structure resembling a flowchart, with core nodes marked by rectangles and ovals indicating the leaf nodes [52]. The Decision Tree method pertains to supervised learning methods.

Random Forest (RF): RF is a renowned supervised learning method based on an ensemble learning approach that brings together various classification elements to solve a complex issue and increase the performance of the model [53]. It is a multi-decision tree ensemble classifier that uses a randomly chosen subset of training data and parameters to generate multiple decision trees [54].

Extreme Gradient Boosting (XGBoost): XGBoost is a recent algorithm that has dominated applied ML [55]. XGBoost is a gradient-boosted decision tree implementation optimized for speed and efficiency. XGBoost models necessitate more information and model tuning than techniques such as a random forest to reach optimum performance.

3.5.2. Deep Learning (DL) Techniques

In the previous research, DL techniques had a very high performance compared to traditional ML techniques with automatic feature extraction that successfully executed sentiment analysis. We implement two DL models that are increasingly applied in sentiment analysis: (i) Convolutional neural networks (CNN) and (ii) Long Short Term Memory (LSTM).

Convolutional Neural Networks (CNN): CNN is a particular neural network used in various sections, including natural language processing, speech processing, and computer vision. We used CNN model to analyze user sentiment in the Twitter dataset. Kim [56] proposed CNN first intended 1d-CNN model, suitable with one dimension patterns. A model of this kind is helpful in natural language processing. It takes input sentences with various lengths and provides an output with fixed-length vectors. Severyn et al. [57] proposed a

CNN model with primary elements like sentence matrix, activation, convolutional, pooling, and softmax layers. Our CNN model architecture is as CNN architecture of Kim [56] with minor modifications. Our CNN model also has three layers: convolution (CONV) layer, pooling (POOL) layer, and fully connected (FC) layer. Firstly, the CONV layer receives the input data, then the filter and the input data are calculated by the dot product. We used tweets as the input of the network. The tweets are tokenized into words to map a word vector, i.e., GloVe (embedding with different dimensions (100 dimensions, 200 dimensions, and 300 dimensions)). Then word2vec and encoding techniques ($s \times d$) indicate the entire tweets mapped to a matrix of size, where the symbol s indicates the number of words in the tweet and the dimension of the embedding space is d . For doing the exact dimension of the matrix, we need to follow the strategy of padding $X \in R^{s \times d}$. A single convolution involves a filtering matrix $w \in R^{h \times d}$, where h is the size of the convolution. The operation of convolution can be defined as ([32])

$$c_i = f(\sigma_{j,k}(w_{j,k})(X_{[i:i+h-1]_{j,k}} + b)) \quad (3)$$

where a bias term and a nonlinear function are denoted $b \in R$ and $f(x)$ respectively, we chose a ReLU function as activation function. The output is the concatenation of the convolution operator, and all words in the tweet are defined $c \in R^{s+h+1}$. For each convolution, $c_{max} = \max(c)$ is defined by max-pooling operation. We combine c_{max} into $c_{max} \in R^m$ for every filters into one vector where m is the total number of filters. We used a fully connected layer to pass through a softmax layer and used a dropout layer to reduce the overfitting.

Long Short Term Memory (LSTM): A recurrent neural network (RNN) is a class of artificial neural networks. It has a special type of network named Long Short-Term Memory (LSTM) that can explore and study long-term dependencies. The main applications of LSTM are speech recognition, language modeling, sentiment analysis, and text prediction. Wang et al. [58] first introduce LSTM networks for tweet sentiment analysis. LSTM inaugurates a memory cell that can be conserved state over long periods and overcome this problem of long-distance dependence [59]. A memory cell is the core of LSTM, denoted by c_t as well as connected recurrently to itself. The three multiplication units of LSTM are: (i) an input gate i_t , (ii) a forget gate f_t , and (iii) an output gate o_t . Formally, LSTM can be computed by [60]

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (4)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (5)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (6)$$

$$u_t = \tanh(W_u x_t + U_u h_{t-1} + b_u) \quad (7)$$

$$c_t = i_t \odot u_t + f_t \odot c_{t-1} \quad (8)$$

$$h_t = o_t \odot \tanh(c_t) \quad (9)$$

where hidden unit at time step t denotes h_t , the input at the current time step denotes x_t , bias term stands for b , logistic sigmoid function stands for σ , and the elementwise multiplication stands for \odot .

3.6. Evaluation Criteria

We use four standard metrics, namely accuracy, precision, recall, and F_1 -score [30].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

$$F_1 - \text{score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (13)$$

In the above equations, *TP* is the true positive and predicted correctly, *FP* is the false positive and predicted incorrectly, *TN* is the true negative and predicted correctly, and *FN* is the false negative and predicted incorrectly.

4. Experimental Results Analysis

This section presents evaluation metrics in accuracy, precision, recall, and f1-score—further, a brief discussion of the results is made.

4.1. Setup for the Experiment

We utilized the Keras [61] deep learning platform in the experiments, which uses Tensorflow [23] for deep learning method implementation as a back-end. We trained our model using Google Colab, a free cloud service with a free GPU (Graphics processing unit) that comes in handy when working with big datasets.

Parameters Setting

We used TF-IDF and bag-of-words to convert our tweets into numeric vectors to construct machine learning models. While using both TF-IDF and bag-of-words, we ignored terms that appeared less than 1000 times in the documents. We used the Adam optimization algorithm to train deep learning models, which incorporate two stochastic gradient descent extensions such as AdaGrad and RMSProp. Furthermore, we used ReLUs activation functions, `sparse_categorical_crossentropy` for the loss function, and the softmax activation function for ternary classification.

4.2. Sentiment Analysis

We used the Twitter dataset, as is seen in Table 1, for this experiment. This experiment was carried out to classify user tweets into three categories: neutral, positive, and negative. We have explored people's emotions towards COVID-19 by looking at the tweets. People are mostly curious regarding COVID-19, whose tweets fall into the neutral category. According to the experiment, 61.8% of people's emotions are neutral, 20.6% are positive, and just 17.6% are negative, as seen in Figure 2.

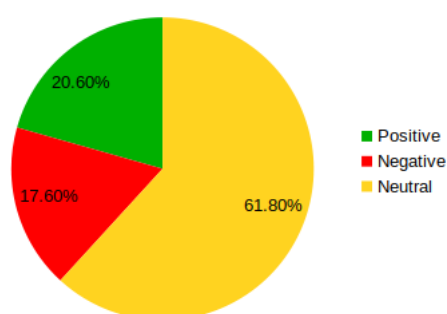


Figure 2. User sentiments.

4.3. Machine Learning Analysis

After extracting the features, we performed a train-test-split on the dataset. This process involves taking the dataset and dividing it into two subsets. We have chosen an 80:20 ratio, i.e., 80% of the data for the training dataset and 20% for the test dataset. We used the seven machine learning algorithms to train our models: logistic regression, support vector machine (SVM), decision tree, random forest, Naïve Bayes, *k*-nearest neighbors (*k*-NN), and XGBoost. For each algorithm, the accuracy of the test dataset was determined. Table 3 represents the confusion matrix, precision, recall, and F_1 -score values used to verify performance.

Table 3. Accuracies including evaluation matrices measurement of ML approaches.

Feature Extraction	Algorithms	Accuracy	Precision	Recall	F ₁ -Score
bag-of-words	Logistic Regression	93.76%	95.08%	89.28%	91.85%
	SVM	93.97%	95.49%	89.66%	92.25%
	Naïve Bayes	90.44%	89.10%	86.05%	87.46%
	k-NN	88.84%	93.29%	81.06%	85.73%
	Decision Tree	95.69%	94.98%	94.39%	94.68%
	Random Forest	97.09%	97.68%	95.21%	96.39%
	XGBoost	84.25%	91.07%	72.17%	78.06%
TF-IDF	Logistic Regression	90.88%	93.46%	84.18%	87.98%
	SVM	93.75%	95.10%	89.27%	91.85%
	Naïve Bayes	90.83%	91.79%	84.67%	87.70%
	k-NN	90.23%	93.04%	83.96%	87.68%
	Decision Tree	95.91%	95.44%	94.41%	94.91%
	Random Forest	97.11%	97.83%	95.13%	96.41%
	XGBoost	83.46%	90.91%	70.82%	76.88%

The accuracy for random forest with TF-IDF is 97.11%, which is the highest among all. With bag-of-words random forest, decision tree, logistic regression, SVM, Naïve Bayes, k-NN, and XGBoost obtain 97.08% 95.69%, 93.76%, 93.97%, 90.44%, 88.84%, 84.25% respectively. Accuracy with TF-IDF for logistic regression, SVM, Naïve Bayes, k-NN, decision tree, XGBoost were 90.88%, 93.75%, 90.83%, 90.23%, 95.91%, 83.46% respectively as shown in Figure 3a.

Random forest with TF-IDF achieved the highest F₁-score, which is 96.41% while random forest with bag-of-words achieved 96.39%. With bag-of-words, logistic regression, SVM, Naïve Bayes, k-NN, decision tree, and XGBoost reach 91.85%, 92.25%, 87.46%, 85.73%, 94.68%, and 78.06 % F₁-scores, respectively. With TF-IDF, logistic regression, SVM, Naïve Bayes, k-NN, decision tree, and XGBoost gain 87.98%, 91.85%, 87.70%, 87.68%, 94.91%, and 76.88% F₁-score, respectively which shown in Figure 3b.

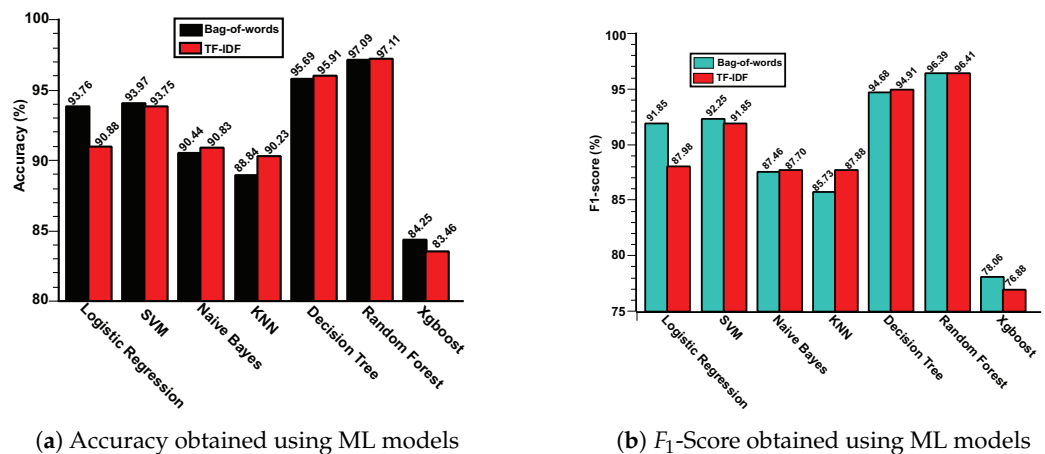


Figure 3. Accuracy and F₁-Score obtained using ML models for testing dataset.

4.4. Deep Learning Analysis

After extracting the DL features, we adopted the same 80:20 approach of train-test-split to divide our dataset to evaluate the deep neural networks models. For CNN and LSTM algorithms, the accuracy of the test dataset was determined. Table 4 represents the confusion matrix, precision, recall, and F₁-score values used to verify performance.

Figure 4a shows the accuracy of two DL models with different techniques, namely: CNN and LSTM. The accuracy of CNN using GloVe embedding with 100d is 98.5%. Moreover, the accuracy of CNN using GloVe embedding with 200 and 300 dimensions are 98.9% and 99.1%, respectively. We also find that CNN’s accuracy using word2vec embedding is

99.9%, and the accuracy of CNN using the encoding technique is 99.3%. Similarly, the accuracy of LSTM using GloVe embedding with three different dimensions like 100,200 and 300 are identical, which is about 61.7%. Furthermore, CNN provides 99.9% and 99.2% accuracy using word2vec and encoding techniques.

The F_1 -Score of CNN and LSTM models using word2vec embedding were highest, which are about 99.99%. CNN with GloVe embedding(100d, 200d and 300d) and Encoding techniques reach 98.00%, 98.00%, 98.00% and 99.00% F_1 -scores, respectively. On the other hand, LSTM with GloVe embedding (100d, 200d as well as 300d) and Encoding technique give 26.00%, 26.00%. 26.00% and 99.00% F_1 -score, respectively which shown in Figure 4b.

Table 4. Accuracy’s including evaluation matrices measurement of DL approaches.

Algorithms	Feature Extraction	Accuracy	Precision	Recall	F_1 -Score
CNN	GloVe with 100d	98.51%	98.00%	98.00%	98.00%
	GloVe with 200d	98.89%	98.00%	98.00%	98.00%
	GloVe with 300d	99.07%	99.00%	98.00%	98.00%
	Encoding Techniques	99.34%	99.00%	99.00%	99.00%
	word2vec	99.89%	99.91%	99.99%	99.99%
LSTM	GloVe with 100d	61.74%	21.00%	33.00%	26.00%
	GloVe with 200d	61.74%	21.00%	33.00%	26.00%
	GloVe with 300d	61.74%	21.00%	33.00%	26.00%
	Encoding techniques	99.24%	99.00%	99.00%	99.00%
	word2vec	99.88%	99.99%	99.99%	99.99%

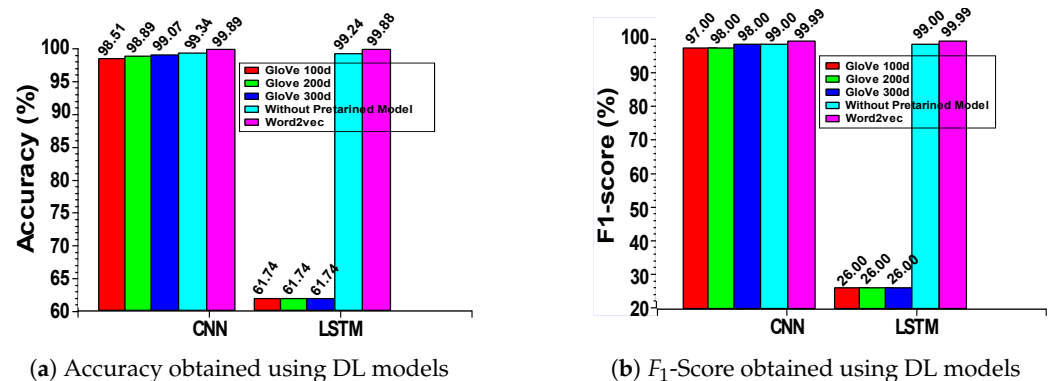


Figure 4. Accuracy and F_1 -Score obtained using DL models for testing dataset.

4.5. *Infected COVID-19 Cases vs. Estimated COVID-19 Cases Using Twitter Dataset*

We are curious to know the analogies between the real COVID-19 infected cases (i.e., Dataset-II) and the estimated COVID-19 cases using the Twitter dataset (i.e., Dataset-I). Figure 5 shows the experimental results between COVID-19 cases and COVID-19 related tweets. In this experiment, we consider the semi-log scale (i.e., the Y-axis log scale). The results show that when COVID-19 cases increase, people post more COVID-19-related tweets on social media in all states except California and Georgia. The results show that California has the highest number of COVID-19 cases, but there are relatively few tweets about COVID-19 on Twitter, while Georgia has the opposite. We believe that this happens in case of tweets are posted to raise awareness about COVID-19. However, in our next work, we will focus on understanding the relationship between COVID-19-related tweets and the number of COVID-19 cases for further analysis.

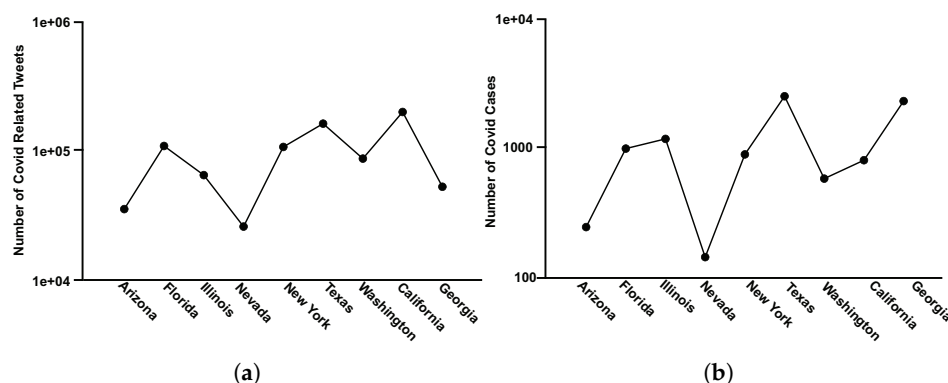


Figure 5. Comparison between the (a) number of COVID-19 related tweets in each state, and (b) the number of COVID-19 cases in each state.

5. Discussion and Conclusions

This research aims to evaluate user sentiment by creating ML and DL models that can effectively forecast sentiment and compare COVID-19 infection cases and COVID-19 associated tweets. From 1 April to 15 April 2020, we gathered data from Twitter using the search keywords CORONAVIRUS and COVID-19 from nine states of the USA.

It is concluded from the research that most of the user sentiments are neutral. Both TF-IDF and the traditional bag-of-words feature extraction techniques work well at classifying user sentiments in machine learning models. Random Forest with bag-of-words and TF-IDF worked exceptionally well with other ML models. However, the random forest classifier generated the most stable and reliable results when combined with the TF-IDF feature extraction technique. Logistic Regression and SVM perform better with traditional bag-of-words, while TF-IDF extracts features for the other models. In DL, features are automatically trained and extracted, achieving higher precision and efficiency than ML versions. We used GloVe embedding with three dimensions, Word2Vec embedding, and an encoding technique to convert input data before feeding it into DL models. CNN and LSTM architectures were examined and paired with various methods to conduct sentiment analysis. We ran several tests on the dataset, including tweets, to compare CNN and LSTM models. After analyzing, we identified that the deep learning model constructed using word2vec and encoding feature extraction techniques outperformed Glove embedding feature extraction techniques. Although the best performance was obtained using the LSTM with the word2vec feature extraction technique. After evaluating the experiments, we observed that CNN surpasses the LSTM model.

In the future, we will focus on multiple social networking platforms such as Facebook, Instagram, and LinkedIn to create an effective model capable of classifying user sentiments more accurately. The construction model would then be compared to other established models to improve sentiment classification accuracy.

Author Contributions: Conceptualization, N.Y., N.I.M. and B.C.S.; methodology, N.Y., N.I.M. and B.C.S.; software, N.Y. and N.I.M.; validation, N.Y. and N.I.M.; formal analysis, B.C.S., M.K.B., Z.A. (Zulfikar Alom), M.A.A. and Z.A. (Zeyar Aung); investigation, N.Y. and N.I.M.; data curation, N.Y. and N.I.M.; writing—original draft preparation, N.Y., N.I.M., M.K.B., B.C.S. and Z.A. (Zulfikar Alom); writing—review and editing, B.C.S., M.K.B., Z.A. (Zulfikar Alom), M.A.A. and Z.A. (Zeyar Aung); supervision, B.C.S.; project administration, B.C.S., Z.A. (Zulfikar Alom), M.A.A. and Z.A. (Zeyar Aung); funding acquisition, Z.A. (Zeyar Aung). All authors have read and agreed to the published version of the manuscript.

Funding: This research is partially funded by Khalifa University, Abu Dhabi, United Arab Emirates.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available at <https://www.kaggle.com/>, accessed on 30 November 2021.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, H.; Wang, Z.; Dong, Y.; Chang, R.; Xu, C.; Yu, X.; Zhang, S.; Tsamlag, L.; Shang, M.; Huang, J. Others Phase-adjusted estimation of the number of coronavirus disease 2019 cases in Wuhan, China. *Cell Discov.* **2020**, *6*, 10. [CrossRef] [PubMed]
2. World Health Organization. Novel Coronavirus (2019-nCoV): Situation Report. Available online: https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200207-sitrep-18-ncov.pdf?sfvrsn=fa644293_2/ (accessed on 9 February 2020).
3. Twitter Usage Statistics. Internet Live Stats Website. Available online: <http://www.internetlivestats.com/twitter-statistics/> (accessed on 11 October 2016).
4. Soriano, C.; Roldan, M.; Cheng, C.; Oco, N. Social media and civic engagement during calamities: The case of Twitter use during typhoon Yolanda. *Philipp. Political Sci. J.* **2016**, *37*, 6–25. [CrossRef]
5. Van Lent, L.; Sungur, H.; Kunneman, F.; Van De Velde, B.; Das, E. Too far to care? Measuring public attention and fear for Ebola using Twitter. *J. Med Internet Res.* **2017**, *19*, e193. [CrossRef] [PubMed]
6. Nair, M.; Ramya, G.; Sivakumar, P. Usage and analysis of Twitter during 2015 Chennai flood towards disaster management. In Proceedings of Procedia Computer Science, Cochin, India, 22–24 August 2017; pp. 350–358.
7. Fu, K.; Liang, H.; Saroha, N.; Tse, Z.; Ip, P.; Fung, I. How people react to Zika virus outbreaks on Twitter? A computational content analysis. *Am. J. Infect. Control* **2016**, *44*, 1700–1702. [CrossRef]
8. Pang, B.; Lee, L. Opinion mining and sentiment analysis Foundations and Trends. *Inf. Retr.* **2008**, *2*, 1–2.
9. Liu, B. Sentiment analysis and opinion mining. *Synth. Lect. Hum. Lang. Technol.* **2012**, *5*, 1–167. [CrossRef]
10. Huang, Q.; Chen, R.; Zheng, X.; Dong, Z. Deep sentiment representation based on CNN and LSTM. In Proceeding of the International Conference On Green Informatics (ICGI), Fuzhou, China, 15–17 August 2017; pp. 30–33.
11. Sethi, M.; Pandey, S.; Trar, P.; Soni, P. Sentiment identification in COVID-19 specific tweets. In Proceedings of the 2020 International Conference On Electronics And Sustainable Communication Systems (ICESC), Coimbatore, India, 2–4 July 2020; pp. 509–516.
12. Shamantha, R.; Shetty, S.; Rai, P. Sentiment Analysis Using Machine Learning Classifiers: Evaluation of Performance. In Proceedings of the 2019 IEEE 4th International Conference On Computer And Communication Systems (ICCCS), Singapore, 23–25 February 2019; pp. 21–25.
13. Singh, B.C.; Carminati, B.; Ferrari, E. Learning Privacy Habits of PDS Owners. In Proceedings of the IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 151–161.
14. Singh, B.C.; Carminati, B.; Ferrari, E. Privacy-Aware Personal Data Storage (P-PDS): Learning how to Protect User Privacy from External Applications. *IEEE Trans. Dependable Secur. Comput.* **2021**, *18*, 889–903. [CrossRef]
15. Baowaly, M.K.; Kibirige, G.W.; Singh, B.C. Co-Comment Network: A Novel Approach to Construct Social Networks within Reddit. *Comput. Sist.* **2022**, *26*, 311–323.
16. Shin, W.Y.; Singh, B.C.; Cho, J.; Everett, A.M. A new understanding of friendships in space: Complex networks meet Twitter. *J. Inf. Sci.* **2015**, *41*, 751–764. [CrossRef]
17. Singh, B.C.; Alom, Z.; Hu, H.; Rahman, M.M.; Baowaly, M.K.; Aung, Z.; Azim, M.A.; Moni, M.A. COVID-19 Pandemic Outbreak in the Subcontinent: A Data Driven Analysis. *J. Pers. Med.* **2021**, *11*, 889. [CrossRef]
18. Jain, A.; Dandannavar, P. Application of machine learning techniques to sentiment analysis. In Proceedings of the 2nd International Conference On Applied And Theoretical Computing And Communication Technology (iCATccT), Bangalore, India, 21–23 July 2016; pp. 628–632.
19. Chen, E.; Lerman, K.; Ferrara, E. Tracking Social Media Discourse About the COVID-19 Pandemic: Development of a Public Coronavirus Twitter Data Set. *JMIR Public Health Surveill.* **2020**, *6*, e19273. [CrossRef]
20. Neogi, A.S.; Garg, K.A.; Mishra, R.K.; Dwivedi, Y.K. Sentiment analysis and classification of Indian farmers' protest using twitter data. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100019. [CrossRef]
21. Shofiya, C.; Abidi, S. Sentiment Analysis on COVID-19-Related Social Distancing in Canada Using Twitter Data. *Int. J. Environ. Res. Public Health* **2021**, *18*, 5993. [CrossRef]
22. Naseem, U.; Razzak, I.; Khushi, M.; Eklund, P. W.; Kim, J. COVIDSenti: A large-scale benchmark Twitter data set for COVID-19 sentiment analysis. *IEEE Trans. Comput. Soc. Syst.* **2021**, *8*, 1003–1015. [CrossRef]
23. Stringhini, G.; Kruegel, C.; Vigna, G. Detecting spammers on social networks. In Proceedings of the 26th Annual Computer Security Applications Conference, Austin, TX, USA, 6–10 December 2010; pp. 1–9.
24. Kabir, M.; Madria, S. CoronaVis: A Real-time COVID-19 Tweets Analyzer. *arXiv* **2020**, arXiv:2004.13932.
25. Pokharel, B. Twitter Sentiment analysis during COVID-19 Outbreak in Nepal. 2020. Available online: <https://ssrn.com/abstract=3624719> (accessed on 15 June 2020).
26. Sharma, M.K.; Dhiman, N.V.; Mishra, V.N. Mediative fuzzy logic mathematical model: A contradictory management prediction in COVID-19 pandemic. *Appl. Soft Comput.* **2021**, *105*, 107285. [CrossRef]
27. Sharma, M.K.; Dhiman, N.; Mishra, V.N.; Mishra, L.N.; Dhaka, A.; Koundal, D. Post-symptomatic detection of COVID-2019 grade based mediative fuzzy projection. *Comput. Electr. Eng.* **2022**, *101*, 108028. [CrossRef]

28. Day, M.; Lee, C. Deep learning for financial sentiment analysis on finance news providers. In Proceedings of the IEEE/ACM International Conference On Advances In Social Networks Analysis And Mining (ASONAM), San Francisco, CA, USA, 18–21 August 2016; pp. 1127–1134.
29. Heikal, M.; Torki, M.; El-Makky, N. Sentiment analysis of Arabic Tweets using deep learning. In Proceedings of the Procedia Computer Science, Dubai, United Arab Emirates, 17–19 November 2018; pp. 114–122.
30. Goularas, D.; Kamis, S. Evaluation of deep learning techniques in sentiment analysis from Twitter data. In Proceedings of International Conference On Deep Learning And Machine Learning In Emerging Applications (Deep-ML), Istanbul, Turkey, 26–28 August 2019; pp. 12–17.
31. Ain, Q.; Ali, M.; Riaz, A.; Noureen, A.; Kamran, M.; Hayat, B.; Rehman, A. Sentiment analysis using deep learning techniques: A review. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 6.
32. Cliche, M. Bb_twt at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms. *arXiv* **2017**, arXiv:1704.06125.
33. Chen, N.; Wang, P. Advanced combined LSTM-CNN model for twitter sentiment analysis. In Proceedings of the 5th IEEE International Conference On Cloud Computing And Intelligence Systems (CCIS), Nanjing, China, 23–25 November 2018; pp. 684–687.
34. Ali, N.; Abd El Hamid, M.; Youssif, A. Sentiment analysis for movies reviews dataset using deep learning models. *Int. J. Data Min. Knowl. Manag. Process (IJDKP)* **2019**, *9*, 42–49.
35. Maas, A.; Daly, R.; Pham, P.; Huang, D.; Ng, A.; Potts, C. Learning Word Vectors for Sentiment Analysis. In Proceedings of the 49th Annual Meeting of the Association For Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
36. Sosa, P. Twitter sentiment analysis using combined LSTM-CNN models. *Eprint Arxiv* **2017**, 1–9.
37. Your machine learning and Data Science Community. Kaggle. (n.d.). Retrieved 30 November 2021. Available online: <https://www.kaggle.com/> (accessed on 30 November 2021).
38. Straka, M.; Straková, J. *Tokenizing, pos Tagging, Lemmatizing and Parsing ud 2.0 with Udpipe*; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 88–99.
39. Lovins, J. Development of a stemming algorithm. *Mech. Transl. Comput. Linguist.* **1968**, *11*, 22–31.
40. Loria, S. TextBlob: Simplified Text Processing. Release ver. 0.15.2. Available online: <https://textblob.readthedocs.org/en/dev/index.html> (accessed on 26 March 2020).
41. El-Din, D. Enhancement bag-of-words model for solving the challenges of sentiment analysis. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 99. [[CrossRef](#)]
42. Aizawa, A. An information-theoretic perspective of tf-idf measures. *Inf. Process. Manag.* **2003**, *39*, 45–65. [[CrossRef](#)]
43. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the Conference on Empirical Methods In Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
44. Kleinbaum, D.; Dietz, K.; Gail, M.; Klein, M.; Klein, M. *Logistic Regression*, 3rd ed.; Springer: New York, NY, USA, 2002; p. 702.
45. Stoltzfus, J. Logistic regression: A brief primer. *Acad. Emerg. Med.* **2011**, *18*, 1099–1104. [[CrossRef](#)]
46. Joachims, T. *SvmLight: Support Vector Machine*; University of Dortmund: Dortmund, Germany, 1999; Volume 19, p. 25. Available online: <http://svmlight.joachims.org/> (accessed on 9 February 2020).
47. Noble, W.S. What is a support vector machine? *Nat. Biotechnol.* **2006**, *24*, 1565–1567. [[CrossRef](#)]
48. Tan, S. An effective refinement strategy for KNN text classifier. *Expert Syst. Appl.* **2006**, *30*, 290–298. [[CrossRef](#)]
49. Rish, I. An empirical study of the naive Bayes classifier. *IJCAI 2001 Workshop Empir. Methods Artif. Intell.* **2001**, *3*, 41–46.
50. Dai, W.; Xue, G.; Yang, Q.; Yu, Y. Transferring naive bayes classifiers for text classification. *AAAI* **2007**, *7*, 540–545.
51. Kibriya, A.; Frank, E.; Pfahringer, B.; Holmes, G. *Multinomial Naive Bayes for Text Categorization Revisited*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 488–499.
52. Priyam, A.; Abhijeeta, G.; Rathee, A.; Srivastava, S. Comparative analysis of decision tree classification algorithms. *Int. J. Curr. Eng. Technol.* **2013**, *3*, 334–337.
53. Xu, B.; Guo, X.; Ye, Y.; Cheng, J. An Improved Random Forest Classifier for Text Categorization. *J. Comput.* **2012**, *7*, 2913–2920. [[CrossRef](#)]
54. Belgiu, M.; Drăguț, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 24–31. [[CrossRef](#)]
55. Chen, Z.; Jiang, F.; Cheng, Y.; Gu, X.; Liu, W.; Peng, J. XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud. In Proceedings of the IEEE International Conference On Big Data And Smart Computing (bigcomp), Shanghai, China, 15–17 January 2018; pp. 251–256.
56. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.
57. Severyn, A.; Moschitti, A. Unitn: Training deep convolutional neural network for twitter sentiment classification. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, CO, USA, 4–5 June 2015; pp. 464–469.
58. Wang, X.; Liu, Y.; Sun, C.; Wang, B.; Wang, X. Predicting polarities of tweets by composing word embeddings with long short-term memory. In Proceedings of the 53rd Annual Meeting of The Association For Computational Linguistics and the 7th International Joint Conference On Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; pp. 1343–1353.
59. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]

-
60. Rojas-Barahona, L. Deep learning for sentiment analysis. *Lang. Linguist. Compass* **2016**, *10*, 701–719. [[CrossRef](#)]
 61. Lee, K.; Caverlee, J.; Webb, S. Uncovering social spammers: Social honeypots+ machine learning. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 19–23 July 2010; pp. 435–442.