



Article

COVID-19 Tweets Classification Based on a Hybrid Word Embedding Method

Yosra Didi ^{1,2,*} , Ahlam Walha ^{1,2} and Ali Wali ²

¹ Department of Computer Science, Umm Al-Qura University, Makkah 24243, Saudi Arabia; aawalha@uqu.edu.sa

² REsearch Groups in Intelligent Machines (REGIM-Lab), National Engineering School of Sfax, University of Sfax, Sfax 3038, Tunisia; ali.wali@ieee.org

* Correspondence: ymdidi@uqu.edu.sa

Abstract: In March 2020, the World Health Organisation declared that COVID-19 was a new pandemic. This deadly virus spread and affected many countries in the world. During the outbreak, social media platforms such as Twitter contributed valuable and massive amounts of data to better assess health-related decision making. Therefore, we propose that users' sentiments could be analysed with the application of effective supervised machine learning approaches to predict disease prevalence and provide early warnings. The collected tweets were prepared for preprocessing and categorised into: negative, positive, and neutral. In the second phase, different features were extracted from the posts by applying several widely used techniques, such as TF-IDF, Word2Vec, Glove, and FastText to capture features' datasets. The novelty of this study is based on hybrid features extraction, where we combined syntactic features (TF-IDF) with semantic features (FastText and Glove) to represent posts accurately, which helps in improving the classification process. Experimental results show that FastText combined with TF-IDF performed better with SVM than the other models. SVM outperformed the other models by 88.72%, as well as for XGBoost, with an 85.29% accuracy score. This study shows that the hybrid methods proved their capability of extracting features from the tweets and increasing the performance of classification.



Citation: Didi, Y.; Walha, A.; Wali, A. COVID-19 Tweets Classification Based on a Hybrid Word Embedding Method. *Big Data Cogn. Comput.* **2022**, *6*, 58. <https://doi.org/10.3390/bdcc6020058>

Academic Editor: Paolo Trunfio

Received: 8 April 2022

Accepted: 7 May 2022

Published: 18 May 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/>).

Keywords: COVID-19; features extraction; machine learning; sentiment analysis; text classification; Twitter

1. Introduction

The impact of the COVID-19 outbreak [1] severely affected many nations' economies and societies [2,3]. To control the spread of pandemic, all countries followed precautionary measures such as lockdowns, wearing face masks, and social distancing, and provided quick solutions to contain the disease [4].

During the pandemic, users shared their opinions, news, and their experiences in facing this virus on a daily basis through social media [5] which is considered a big data centre. Social networking sites (SNSs) such as Twitter have been considered as valuable sources of different event detection and tracking, such as disease outbreaks. This online platform prompted researchers to analyse, in real time, the tweets that contain peoples' feelings [6] and reactions in many subjects, such as election voting, the stock market, crime, and hate speech [7].

Furthermore, the objective of artificial intelligence (AI) in this current crisis has clearly contributed to studies of the change of human reactions and concerns in correlation with COVID-19 patients and deaths during and after the pandemic. Thus, many COVID-19 surveillance models search for an effective approach of text processing and extracting knowledge from COVID-19-related posts, generating reports earlier, which can be decisive for outbreak prevention. This process is namely sentiment analysis (SA) [8], or emotions mining [9], which classifies the opinions of different sentences as negative, neutral, or

positive. These sentences are preprocessed with natural language processing (NLP) and classified with machine learning (ML) [10–13]. These sentiments are very useful for building faster disease surveillance systems. Many studies contributed to analyse English-language tweets but they did not take into consideration the relationship between syntactic and semantic information and ML methods based on feature types. Thus, in this research, we studied Twitter sentiment analysis on a dataset in relation to COVID-19 to predict and monitor the disease outbreak by the application of supervised machine learning methods. We employed TF-IDF N-gram and word-level for the syntactic analysis, whereas for the semantic analysis, we used Word2vec, FastText, and Glove. Accordingly, the main areas of the research were the following subjects:

1. We present five different extant feature extraction methods: TF-IDF N-gram, TF-IDF uni-gram, Word2vec, Glove, and FastText. We also present two novel methods: hybrid TF-IDF with Glove, and hybrid TF-IDF-based FastText.
2. We compare machine learning methods performance with different features extraction for English-language tweets classification.
3. We choose the best methods combination and fusion to enhance the previously compared performances of machine learning classifiers.

In order to study these two areas, we collected a geo-tagged tweet dataset from IEEE port. It included the tweet IDs and users' sentiment scores, since Twitter's policy does not provide access to streaming complete tweets to be published to third parties. After the hydration of the tweet ID to obtain the meaningful text, the tweets were cleaned using preprocessing techniques. Then, we calculated the sentiment scores using the TextBlob toolkit which classified the tweets as negative, neutral, or positive [14]. We proposed the use of feature extraction methods, such as TF-IDF, Word2vec, FastText, and GloVe, for improving accuracy. The novelty of the present study is in gaining the benefits of these techniques, combined together, in the same ensemble. We combined two methods, first TF-IDF with FastText features and second TF-IDF with Glove, in order to enhance classification accuracy. Therefore, seven machine learning methods, decision tree (DT), random forest (RF), logistic regression (LR), XGBoost classifier, AdaBoost, naïve Bayes, and support vector machine (SVM) were applied, then we compared their performances in the testing phase. Furthermore, we applied one deep learning model, a convolution neural network (CNN), with different word embedding, in comparison with the other models. We evaluated the performance with the following metrics: accuracy, AUCC score, precision, recall, and F1-score. The article is organised as follows. A literature review is presented in Section 2. Materials and methods are presented in Section 3. In Section 4, we discuss performance evaluation. The discussion is presented in Section 5. Lastly, in Section 6, we conclude the paper.

2. Literature Review

Analysing Twitter data is catching the attention of data science researchers, because the data are rich with information on COVID-19 and peoples' attitudes towards the virus. Sentiment analysis, also called emotional extraction or opinion mining, was studied by researchers to understand disease behaviour and its relation with official cases or deaths [15,16].

Researchers, such as Rajput et al. [17], have applied an approach to analyse tweets about the COVID-19 outbreak, based on words frequency and sentiment analysis. Their approach is based on word-level, bi-gram, and tri-gram frequencies to represent word rates by power law distribution. Three tweets classes were obtained accordingly: negative, positive, and neutral.

Samuel et al. [11] proposed machine learning models: naïve Bayes and logistic regression to categorise sentiment tweets into two classes—positive and negative. In their paper, they test the performance of these models on two categories of data with different lengths of characters, less than 77 characters for the first category and 120 characters per tweet in the second category. Naïve Bayes outperformed logistic regression in both categories,

the shorter tweets achieved 91.43% accuracy by NB, while this was 74.29% for LR, and the longer tweets achieved 57.14% accuracy by NB, while LR assessed an accuracy of 52%.

Muthausami et al. [18] have introduced a research based on analysing and visualising the worldwide influence of COVID-19. They classified the tweets into three classes based on machine learning methodology. The classes are positive, neutral, and negative. They utilised different classifiers, such as SVM, naïve Bayes, random forest, decision tree, LogitBoost and MaxEntropy. The proposed method showed that the LogitBoost ensemble classifier achieved better results than the other algorithms.

A study conducted by Jelodar et al. [19] implemented an approach to classify sentiments based on deep learning models, such as LSTM recurrent neural networks (LSTM RNN). The classifier was implemented using NLP for COVID-19 topic modelling expressed on social media.

Aljameel et al. [20] analysed a large Arabic COVID-19-related tweets dataset. The authors built a machine learning model to predict and classify Saudi Arabian citizens' responsiveness toward government measures and pandemic control. They applied uni-gram and bi-gram TF-IDF with SVM, naïve Bayes and KNN classifiers to enhance accuracy. The output results showed that SVM outperformed KNN and naïve Bayes with 85% accuracy.

Al-sukkar et al. [21] introduced a sentiment analysis approach to analyse Arabic tweets as negative or positive with two machine learning classifiers: SVM and naïve Bayes. To enhance the accuracy of classifiers, they applied N-gram TF-IDF with 10-fold cross-validation. Experimental results proved that, using uni-gram, SVM showed the highest accuracy of 83.16%, whereas naïve Bayes achieved an accuracy of 81.93% using bi-gram and tri-gram.

Imran et al. [22] have used a deep learning algorithm LSTM to perform classification of sentiments related to COVID-19 tweets. The application of LSTM on the sentiment 140 dataset was improved with pre-trained Glove Twitter embedding. The main objective of this method was to compute sentiment polarity and users' emotions from tweets. Accordingly, the authors proved that there is a high correlation of sentiment polarity between neighbouring countries.

Alam et al. [23] employed SVM, FastText, and BERT on 218 Arabic tweets and 504 English tweets. The FastText model provided the best result for Arabic text.

In the approach presented by Alqurashi et al. [24], different machine learning classifiers were applied on Arabic Tweets to identify misinformation related to COVID-19 and they employed TF-IDF, Word2Vec, and FastText feature embedding techniques to enhance accuracy of classifiers. The results show that FastText produced high accuracy of 86.8% with traditional classifier, XGBoost, whereas Word2Vec obtained better accuracy with deep learning classifiers, achieving 85.7% with CNN.

Naseem et al. [25] correspondingly proposed the use of various pre-trained embedding representations—FastText, GloVe, Word2Vec, and BERT—to extract features from a Twitter dataset. Furthermore, for the classification, they applied deep learning methods Bi-LSTM and several classical machine learning classifiers, such as SVM and naïve Bayes. The TF-IDF model and FastText outperformed other feature extraction methods with traditional classifiers SVM and RF.

Furthermore, Basiri et al. [26] presented a model that combine five models such as naïve Bayes support vector machines (NBSVM), FastText, DistilBERT, CNN, and bidirectional gated recurrent unit (BiGRU) on COVID-19 tweets in eight highly affected countries. Their approach, that is improved by a meta learning method, achieved a high accuracy of 85.80% in classifying sentiments.

The authors in [27] proposed a COVID-19 tweets classification approach based on several traditional machine learning algorithms: decision tree, XGBoost, extra tree classifier (ETC), random forest, and LSTM. To better represent the text, they used bag-of-words (BOW) and TF-IDF methods. The experimental results showed that ETC achieved higher accuracy with 93%.

Moreover, Nemes and Kiss [28] implemented a model to classify tweets into positive and negative with an RNN model and the TextBlob method [29]. Their proposed work outperformed TextBlob.

For sentiment tweet classification related to COVID-19, Kau et al. [30] proposed a hybrid heterogeneous SVM method (HH-SVM). The results show that the proposed approach outperformed RNN.

As well, Li et al. [31] presented an approach to classify microblog reviews sentiments that included emojis with an emoji-text-incorporating bi-LSTM (ET-BiLSTM) model. The emojis, represented in vectors, were passed to the proposed model. The results show that ET-BiLSTM enhances the performance of sentiment classification.

The work proposed by Umair and Masciari [31] is to classify COVID-19 tweets related to vaccine. They implemented the BERT model for sentiment classification. Then, they investigated the distribution of sentiments towards the vaccine across the world by analysing the hot-spot regions and the application of kernel density estimation. The proposed approach achieved 55%, 69%, and 58% for precision, recall, and F-score for the positive sentiments, respectively, while negative sentiments achieved 54%, 85%, and 64% for precision, recall, and F-score, respectively.

Another study during COVID-19 was proposed by Balli et al. [32] to classify public datasets and SentimentSet data, manually labelled for positive and negative Turkish tweets. Two different libraries were used to preprocess the dataset: Zemberek [33] library and SnowBall library. Furthermore, the data were tokenized by TF-IDF to be passed to ML algorithms, such as LR, RF, Bayesian, and stochastic gradient descent (SGD), while for LSTM, the data were presented by the tokenizer class. It was observed that the models applied on SentimentSet have better performances and the negatively weighted data accuracy was higher than the positively weighted data accuracy.

In another study, Sitaula and Shahi [34] designed a hybrid feature to represent Nepali tweets. They used the fusion of two-word representations such as the bag-of-words (BOW) with FastText-based and domain-specific methods. The combined representations were passed to a multi-channel convolutional neural network (MCNN). The results showed that feature combination outperformed individual features with 69.7% accuracy, and the MCNN model achieved 71.3% accuracy compared with classical algorithms.

Singh et al. [35] conducted a study to classify sentiment Twitter data related to COVID-19 using enhanced feature weighting with the attention mechanisms of LSTM-RNN. TF-IDF were applied to extract tweets features. The experimental results showed that the method proposed outperformed the rest of classical ML algorithms, such as: RF, SVM, NB, and LR, with an accuracy of 84.56%.

In a study by Parimala et al. [36], the researchers used LSTM with feature extraction method to classify tweets related to catastrophe events. They proposed to use risk assessment sentiment analysis (RASA) algorithm. The results show that RASA achieved high accuracy compared with XGBoost and binary classifiers [36].

In all previous studies, sentiment analysis is a very valuable source of information, and researchers seek a method which will enhance accuracy. In this present study, we present a comparative analysis between machine learning algorithms using TF-IDF, Word2vec, FastText, and Glove word embedding models. Accordingly, to achieve high accuracy, we present a hybrid method that combines TF-IDF and two effective word representations, Glove and FastText embedding.

3. Materials and Methods

After conducting the literature review, we can observe that different researchers have developed different models using machine learning methods towards COVID-19 detection and pattern analysis. The proposed framework, as shown in Figure 1, is divided into five stages: data collection, preprocessing phase, sentiment analysis, features extraction, and classifiers application. First, we start by collecting tweets dataset freely available on IEEE data port [37]. The second stage is preprocessing data by removing punctuation, symbols,

hashtags, and stop-words, using a natural language processing toolkit (NLTK) [38]; this stage also includes stemming and tokenizing steps. After that, the following stage is performed in two parts. The first part includes calculation of sentiment scores and the second part includes categorisation of sentiment type. Next, in the features extraction stage, we applied different widely used word representation tools, such as TF-IDF, word2Vec, Glove, FastText, and a combination of word embedding models. The result of this stage is transferred to several classifiers for evaluation. Accordingly, each stage is explained in-depth in the following sections.

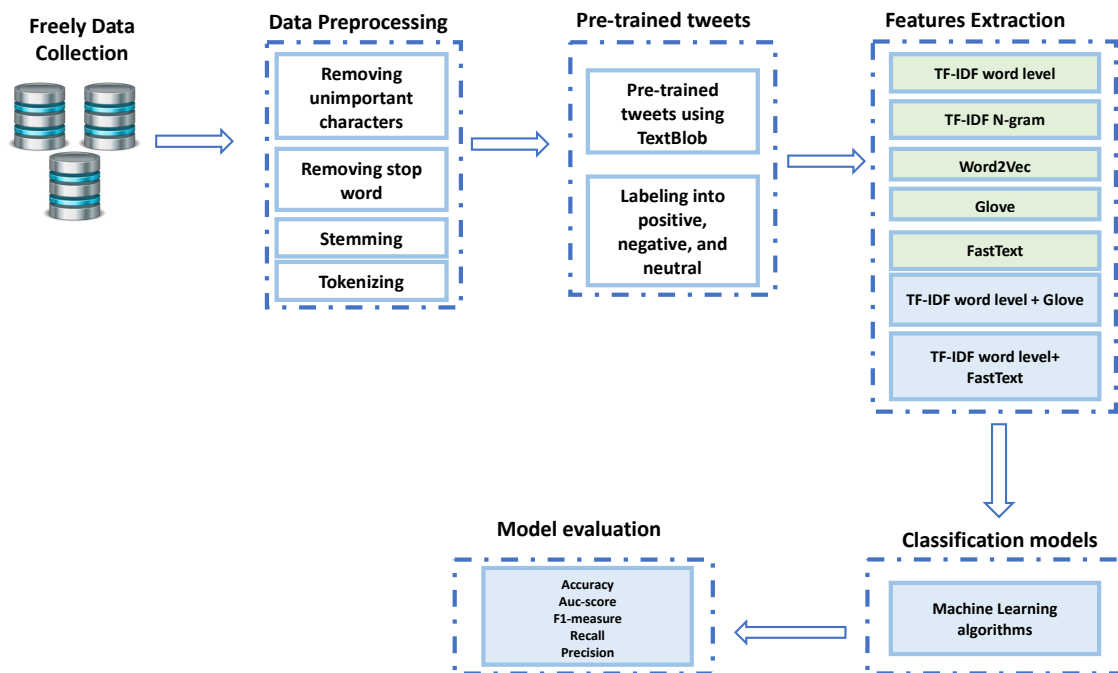


Figure 1. Framework of the proposed model.

3.1. Data Collection

Since Twitter API does not allow access to stream old data for more than one week, our study is based on a Twitter dataset [39] freely available on IEEE’s website [37] from 20 March 2020. The extracted dataset mostly contained geo-tagged tweets with only the tweet IDs of the users and filtered based on keywords related to COVID-19, since Twitter’s policy does not allow third parties to publish tweets or access streaming of complete tweets. To capture complete tweet information, such as tweet ID, tweet text, location, time created, and more, we need to hydrate the IDs using the DocNow [40] hydrator tool, which is a desktop application that allows hydration of tweets in JSON as well as CSV format. The hydrated tweets from 20 March 2020 to 26 May 2021 were downloaded into a CSV file. Furthermore, the extracted tweets are only presented in English language in this current study.

3.2. Data Preprocessing

The preprocessing phase is very important in this stage for text classification. We needed to clean the tweets by removing the special characters, emoticons, hashtags, punctuation, URLs, numbers, mention, and symbols from the text which were not necessary for analysis purposes using the NLP toolkit (NLTK) [35,38]. Then, the tweets were converted into lower text. We then proceed to tokenizing the tweets [41], which is essentially splitting the text into a list of words using methods available in the NLP library [42]. Finally, and after removing English stop-words, we used the stemming process to reduce words to their roots with the Porter stemming algorithm [43]. It is one of the most well known stemming algorithms. After the cleaning process, we visualised the most significant words

3.3.1. Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a fundamental step used for transforming the tweet text data into numbers before applying any classification model [27]. It is performed in two statistical methods: first, TF, which is the total word number appearance in a document; the second method is IDF, that refers to the total terms occurrences in the document. The weight is based on the product of TF and IDF to measure the relevance and how the term is important in a given document. Equations (1)–(3) present the formula to calculate TF, IDF, and their product. Where t refers to the term with frequency n , d represents the document, and N refers to documents' frequency, d , containing the term, t .

$$TF(t, d) = \frac{n_t}{n} \quad (1)$$

$$IDF(d) = \frac{N_d}{N} \quad (2)$$

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \quad (3)$$

We constructed TF-IDF vectors twice: the first with word level and the other one with N-grams.

3.3.2. Word2Vec

Word2Vec [44] is the most widely used technique to learn word embedding utilising neural network. The trained model applied mathematical operation on the text corpus to place similar words in the same vector. There are two main methods for Word2Vec, one called the skip-gram-based method, where the main idea is predicting the context based on a word, and the second is continuous bag of words (CBOW), in which the predicted term depends on the context. The algorithm built in this study is CBOW, trained on the corpus with window sizes $W = 5$, minimum word frequency = 5, and dimension $D = 100$.

3.3.3. FastText

FastText [45], provided by the Facebook team, is an approach of word embedding using the skip-gram-based model, where each word is transformed to N-grams character. The words in the training corpus are associated with vector representation sum of each character N-gram, even misspelled or rare words, not presented in the dictionary, will have an embedding. We applied pre-trained FastText embedding methods to produce one vector for each word of a specific tweet. The model used in this study was 1 million word vectors trained on Wikipedia 2017 with 1 billion tokens. FastText is an extension to Word2Vec and has been shown more accuracy compared with Word2Vec [46].

3.3.4. Glove

The global vector for word representation [47] is mostly used for feature extraction. The Glove technique generates feature matrix based on feature-feature co-occurrence. In this study, we used pre-trained word vectors freely available on corpora, which is the combination of Gigaword5 and Wikipedia2014, with 6 billion tokens from Common Crawl.

3.3.5. Hybrid Word Embedding Techniques with TF-IDF

Despite the use of various embedding techniques and tools, we look to enhance machine learning algorithms performance and to deal with challenges to optimise classification process. Consequently, we provide a hybrid approach that combined TF-IDF [27] features with FastText [45] on the one hand and TF-IDF features with GloVe [47] on the other hand. In the experimental results, TF-IDF, FastText, and Glove proved their capability of extracting features from the tweets and increasing the performance of classification.

First, we applied TF-IDF on the dataset to represent tweets syntactically. This technique generates a scores vector from each word in each tweet. Then, to capture the semantic feature, we multiplied the word embedding Glove or FastText with word TF-IDF-scores

for all the words in the sentence to acquire the vector used for the classification phase. Equation (4) for matrices multiplication output is shown as follows:

$$F_{ij} = \sum_{k=1}^P a_{ik}b_{kj} \quad (4)$$

We note that the number of tokens is P . a and b represent the TF-IDF matrix and semantic matrix, respectively. F represents the final multiplication matrix. The combined techniques syntactic and semantic are complementary to each other and the fusion is helpful for models aiming to improve their accuracies and to benefit from the power of two techniques in same model.

3.4. Classification

In the phase of classification, we applied several machine learning classifiers. Those classifiers are XGBoost, random forest, AdaBoost, decision tree, logistic regression, naïve Bayes, and SVM. Deep learning models such as convolutional neural network (CNN) were used in addition to traditional ML. We chose several algorithms since they are widely applied in sentiment analysis for their high accuracy. We compared the performance of each classifier to obtain the best model.

3.4.1. Decision Tree

Decision tree is a machine learning model frequently applied in classification [48]. To find the outcomes, DT recursively splits the independent variables into groups. It is structured as a tree, where the features of the dataset are represented by the internal nodes, the outcomes are associated with each leaf node, and the decision rules are denoted in the branches.

3.4.2. Random Forest

The random forest [49] classifier is a bagging technique that belongs to the ensemble techniques. It contains a number of decision trees instead of one decision tree, which are considered as the base learners. The predictive accuracy from each tree is improved by the average of the dataset. Thus, the trees are trained independently. The ensemble concepts of different decision trees in RF classifier lead to higher accuracy and prevent the problem from over fitting.

3.4.3. XGBoost

XGBoost is the abbreviation of the extreme gradient boosting algorithm which belongs to the ensemble method that is based on boosting trees. This successful machine learning method proposed by Tianqi Chen [50] is based on a gradient boosting algorithm. The principal idea of XGBoost is to learn from the previous error performed by the model which improves the next performances.

3.4.4. AdaBoost

Adaptive boosting, known as AdaBoost [51], is an important ensemble boosting classifier. AdaBoost trains a weak learning algorithm with equal weights set to the instance of dataset producing poorly performing classifiers. After choosing the coefficient, α , depending on the weak learning classifier performance, misclassified points may be produced and their weights had to be increased, while the weights of accurately classified points had to be reduced before running the weak learning algorithms again. As a result, the new weighted data is obtained for the weak classifiers. To obtain correctly classified data points, this process is repeated until the maximum level is reached. The AdaBoost classifier is developed when arriving at the final step.

3.4.5. Naïve Bayes

Naïve Bayes is very popular model and it has been used widely for text classification since 1960 [52]. This simple algorithm is effective and based on the Bayes theorem. Aiming to predict the class, NB uses maximum posteriori estimation that has high conditional probability. The NB formula is shown in (5):

$$C_{NB} = \operatorname{argmax}_{c \in C} \log P(c) + \sum_{i \in \text{positions}} \log P(w_i|c) \quad (5)$$

where c is the estimated class among all the classes C , \mathbb{P} is the posterior probabilities, and w_i is considered as the word index in the document.

3.4.6. Logistic Regression

David Cox developed logistic regression in 1958 [53] and it is considered as one of the popular methods of machine learning. Using probabilities to describe the outcomes, LR is suitable for predicting categorical classification. Here we applied multinomial logistic regression for multi-class classification that adapted multinomial probability distribution. The rule of LR is to predict the class with the highest posterior probability. The decision rule is shown in (6):

$$\hat{y} = \operatorname{argmax}_{k \in \{0, \dots, k-1\}} \mathbb{P}(y = k|x) \quad (6)$$

where \mathbb{P} is the posterior probabilities, \hat{y} is the predicted label, k is the total labels, and x is the input text.

3.4.7. SVM

SVM is a method based on supervised techniques which can be used for the classification [54]. The input features are represented as vectors and projected onto larger dimension space. SVM is applicable with different type of function such as Gaussian/radial or kernel (linear, polynomial) type. Accordingly, in this research, we applied kernel function [55].

3.4.8. Convolutional Neural Network

CNN is a very popular deep learning algorithm in the domain of image classification. It can capture the patterns with very high accuracy in computer vision. Recently, Kim [56] demonstrated the efficient use of CNN for natural language processing and text analysis on various benchmark tasks. Therefore, CNNs are able to capture patterns in text.

4. Experimental Results

In this phase, we analyse the experimental approach performance with the five performance metrics, accuracy score, precision score, recall score, F1-score, and AUCC score. These metrics definitions are briefly outlined as follows:

- Accuracy indicates the weighted harmonic mean of both precision and recall. The accuracy equation is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

- Precision score represents the percentage of positively classified tweets that actually correct. The precision is mathematically expressed as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (8)$$

- Recall score indicates the ability of the classifiers to classify all positive instances correctly. The recall is mathematically expressed as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (9)$$

- F1-score indicates the weighted harmonic mean of both precision and recall. The F1-score is mathematically expressed as follows:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (10)$$

- AUCC score indicates the classifiers' ability to distinguish between classes through the probability curve (ROC). The AUCC is defined as follows:

$$AUCC = \sum_{i \in (TP+TN+FP+FN)} \frac{(TPR_i + TPR_{i-1}) \times (FPR_i + FPR_{i-1})}{2} \quad (11)$$

where *TP* refers to true positives: the correct prediction number of positive class. *TN* is true negatives: the correct predictions number of negative class. *FP* refers to false positives: the incorrect positive predictions number of a class. Furthermore, *FN* refers to false negatives: the incorrect negative predictions number of a class.

The Tweet dataset collected from the open source comprised about 396,452 tweets, which were shuffled for the classification process. We split the data into 80% for training phase and 20% for the testing phase, as an input for the machine learning algorithms. Many researchers have followed this partition of datasets, such as Ben Jabeur et al. [57] and Antunes et al. [58]. Another method proposed by Gholamy et al. [59] proves that we obtain better results when the data is split in 20–30% for testing, and 70–80% of the remaining data for training.

Before applying any algorithms, we observed that our dataset was imbalanced, thus we applied the synthetic minority oversample technique (SMOTE) algorithm proposed by Farquad and Bose [60] to generate the synthetic samples for the class with minor number. This method overcomes the overfitting problem due to random oversampling.

4.1. Machine Learning Algorithms with Simple Feature Extraction Methods

We trained the traditional classifiers and deep learning algorithm using uni-gram and N-gram TF-IDF feature representations. We also report the results based on Word2Vec, FastText, and Glove embedding methods. Table 1 shows the accuracy, AUCC score, precision, recall, and F1-score results of the traditional classifiers for word-level and N-gram TF-IDF feature representations. The SVM classifier with uni-gram feature representations outperformed the other algorithms with the best accuracy of 85.44%, AUCC score of 96.04%, and F-score of 0.85. SVM also outperformed all algorithms when the TF-IDF was based on N-gram, the highest performance reached 85.2% accuracy, 95.7% AUCC score, and 0.85 for F-score. RF also achieved high performance after SVM, with 81.32% accuracy and 93.38% AUCC score, while NB achieved the lowest accuracy in both word embedding methods, with 54.3% accuracy in uni-gram TF-IDF and 53.12% with N-gram level.

Table 1. Traditional classifier performance using TF-IDF.

		Machine Learning Algorithms						
		DT	RF	XGBoost	AdaBoost	NB	LR	SVM
TF-IDF word-level	Accuracy	64.12%	81.32%	78.87%	60.52%	54.3%	64.66%	85.44%
	AUCC	73.09%	93.38%	92.76%	76.05%	71.84%	83.36%	96.07%
	F1-Score	0.64	0.81	0.79	0.61	0.52	0.65	0.85
	Precision	0.64	0.81	0.79	0.61	0.56	0.65	0.85
	Recall	0.64	0.81	0.79	0.61	0.54	0.64	0.85
TF-IDF N-gram level	Accuracy	63.99%	81.02%	80.11%	60.24%	53.12%	63.84%	85.2%
	AUCC	72.99%	92%	92.98%	75.39%	71.03%	82.04%	95.7%
	F1-Score	0.64	0.81	0.80	0.60	0.51	0.64	0.85
	Precision	0.64	0.82	0.80	0.60	0.56	0.64	0.85
	Recall	0.64	0.81	0.80	0.60	0.53	0.64	0.85

All the classifiers achieved slightly high performance when using TF-IDF uni-gram compared with the N-gram technique, except for XGBoost, while the best performance was achieved with the N-gram method with 80.11% accuracy and 92.98% AUCC score. Figure 6 illustrates the ROC curve and the accuracy of word-level TF-IDF, Glove, and FastText.

Table 2 shows the evaluation results of the traditional classifiers for Word2Vec, Glove, and FastText feature embedding. The experimental results show that the AUCC and F1-score slightly increase for NB, AdaBoost, and LR with Word2Vec compared with TF-IDF, while it decreases with FastText, when compared with Word2Vec.

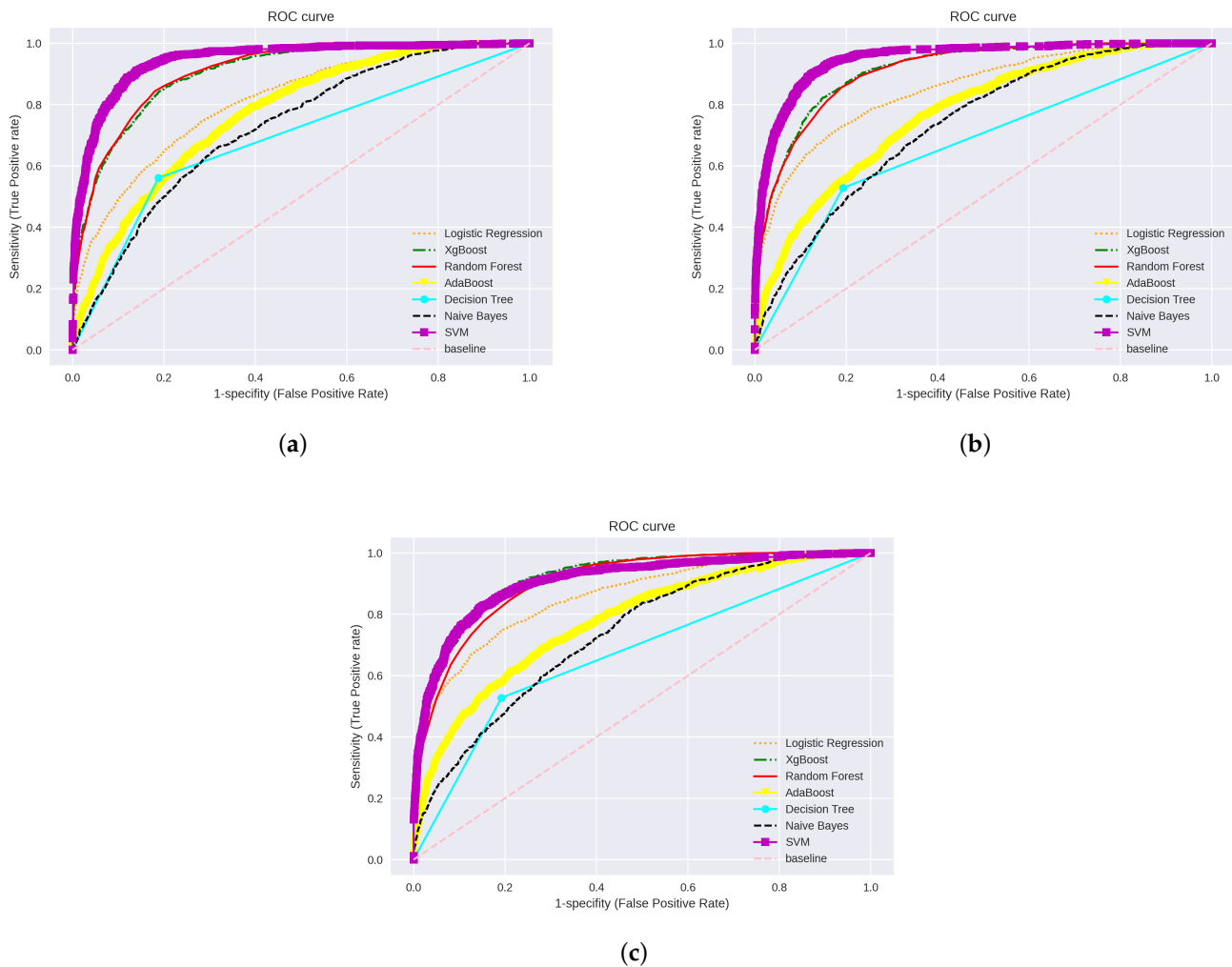


Figure 6. Visualisation of the ROC curves of traditional classifiers using (a) TF-IDF, (b) Glove, and (c) FastText word embedding techniques.

With Word2Vec, the best accuracy achieved was by LR, compared with the rest of the classifiers, with 74.09% accuracy and 0.72 F-score, while NB performed with the lowest accuracy of 58%. FastText and Glove performances were better with classification except for NB, LR, and Adaboost, which performed worse than the other classifiers. SVM achieved a higher performance in both techniques, FastText and Glove, in which it achieved 86.17% and 82% with Glove and FastText, respectively. However, NB achieved the lowest accuracy with 56.88% and 55.69% accuracy for Glove and FastText methods, respectively. With Glove, the second-best classifier was RF with 80.52%, while the third-best classifier was XGBoost, with 80.04%. However, the XGBoost classifier outperformed RF with FastText, where it achieved 80.84%, while RF performance was 79.05%. We can deduce that the model that achieved the highest performance with the TF-IDF, Glove, and FastText was SVM. Figure 6

shows the ROC curve generated by the traditional classifiers using TF-IDF, Glove, and FastText. A visualisation of the accuracy for each ML methods using TF-IDF, Glove, and FastText is shown in Figure 7.

Table 2. Traditional classifier performance using Word2Vec, Glove, and FastText.

		Machine Learning Algorithms						
		DT	RF	XGBoost	AdaBoost	NB	LR	SVM
Word2Vec	Accuracy	59.34%	71.04%	69.69%	66.25%	58%	74.09%	69.72%
	AUCC	63.23%	82.65%	80.1%	75.89%	73.12%	84.12%	81.45%
	F1-Score	0.59	0.67	0.64	0.62	0.48	0.72	0.69
	Precision	0.59	0.75	0.71	0.63	0.56	0.72	0.69
	Recall	0.59	0.71	0.69	0.66	0.58	0.74	0.69
Glove	Accuracy	62.65%	80.52%	80.04%	61.74%	56.88%	70.47%	86.17%
	AUCC	71.98%	93.19%	93.22%	79.54%	75.48%	86.7%	96.22%
	F1-Score	0.62	0.81	0.80	0.62	0.57	0.70	0.86
	Precision	0.62	0.81	0.80	0.62	0.57	0.70	0.86
	Recall	0.63	0.81	0.80	0.62	0.57	0.70	0.86
FastText	Accuracy	62.73%	79.05%	80.84%	64.85%	55.69%	71.29%	82%
	AUCC	72.05%	92.54%	93.66%	80.64%	75.02%	87.7%	93.21%
	F1-Score	0.63	0.79	0.81	0.64	0.55	0.71	0.80
	Precision	0.63	0.80	0.81	0.64	0.57	0.71	0.81
	Recall	0.63	0.79	0.81	0.64	0.56	0.71	0.81

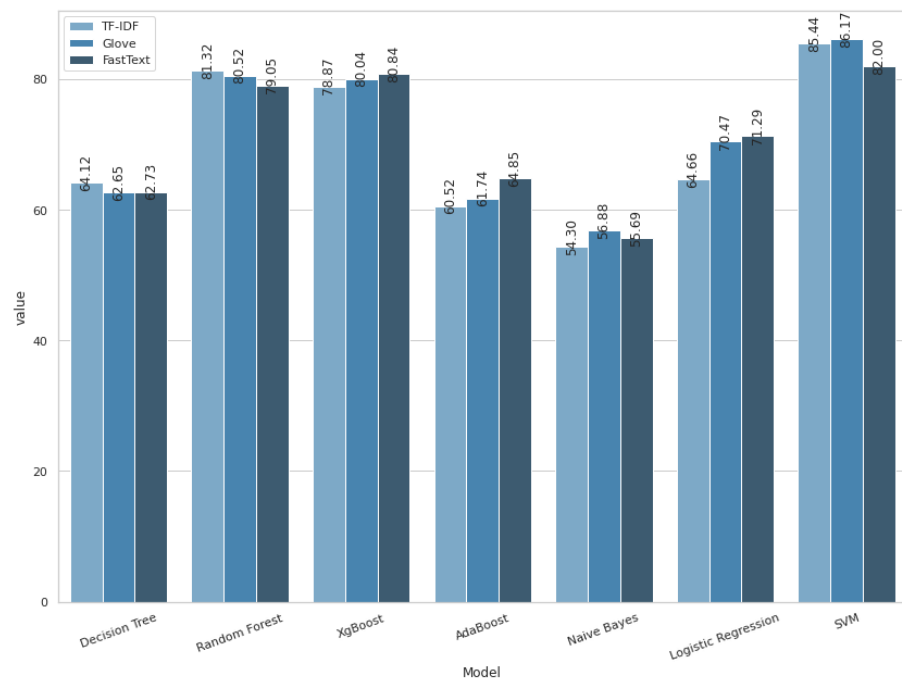


Figure 7. Visualisation of the accuracy of traditional classifiers using TF-IDF, Glove, and FastText word embedding techniques.

We trained the deep learning classifier CNN using the Adam optimiser to learn the model parameters. We reported the results with the four pre-trained word embedding methods. Table 3 shows the accuracy, AUCC, F1 measure, recall, and precision with the pre-trained word embedding. CNN achieved the highest accuracy with Glove, reaching a 79.83% score, while the lowest score was with FastText, with 73.59%. However, the highest AUCC score was achieved by TF-IDF uni-gram with 86.52% followed by Glove, with a close AUCC score of 82.18%. With the pre-trained TF-IDF uni-gram, Word2Vec, Glove,

and FastText embedding, the performance increased. The CNN with Glove had the best improvement as it outperformed all other traditional classifiers with the best accuracy, F1-score, recall, and precision. The performance shows an improvement over the existing results.

Table 3. CNN classifier performance with feature extraction.

	Deep Learning Classifier CNN				
	Accuracy	AUCC	F1-Score	Precision	Recall
TF-IDF word-level	76.01%	86.52%	0.76	0.76	0.76
Word2Vec	74.33%	81.42%	0.74	0.74	0.74
Glove	79.83%	82.18%	0.79	0.79	0.79
FastText	73.59%	80.45%	0.71	0.71	0.71

4.2. Machine Learning Algorithms with Hybrid Feature Extraction Techniques

Through a final analysis, the highest performing techniques were chosen to optimise the classification phase, which was the objective of our proposed method. Where the word-level TF-IDF, Glove, and FastText have the best improvement with the classifiers, the TF-IDF N-gram showed the lowest performance, followed by Word2Vec. We choose to follow two experiments, first combining TF-IDF and Glove features and second combining TF-IDF and FastText with different classifiers aiming to obtain high performance from the machine learning methods. Almost all algorithms’ performances were improved with the hybrid methods.

As shown in Figures 8 and 9 and Table 4, SVM accuracy increased to 88.72%, as well as for XGBoost, with 85.29% accuracy score, LR with 81.85%, and RF with 81.85%, when the input to the classifier was TF-IDF with FastText features. All the other classifiers also outperformed their previous results. Using the hybrid TF-IDF with Glove, the accuracy reached 86.16%, 83.47%, 83.13%, and 81.05% accuracy scores for SVM, LR, XGBoost, and RF, respectively. The results were enhanced because the feature set size increased with the hybrid features, so the model learnt from the newly established features and improved its accuracy. The predicted accuracy, AUCC, F1-score, precision, and recall metrics of each model can be observed in Table 4. In a comparison between AUCC scores, we observed that the models showed better performances for hybrid TF-IDF and FastText (95.88%, 89.63% 87.74%, and 87.03%, on SVM, XGBoost, RF, and LR, respectively) compared with hybrid TF-IDF and Glove, with 96.42%, 88.72%, 88.58%, and 87.66% on SVM, LR, XGBoost, and RF, respectively. This was thanks to their capability to represent words vocabulary effectively, in comparison with other tested methods. Consequently, we can observe that the performance of both experiments is slightly close.

Table 4. Traditional classifier performance using hybrid features.

		Machine Learning Algorithms						
		DT	RF	XGBoost	AdaBoost	NB	LR	SVM
Hybrid 1 ¹	Accuracy	76.58%	81.85%	85.29%	74.73%	64.30%	82.40%	88.72%
	AUCC	74.68%	87.74%	89.63%	82.54%	74.07%	87.03%	95.88%
	F1-Score	0.75	0.82	0.84	0.74	0.62	0.81	0.85
	Precision	0.76	0.81	0.85	0.74	0.69	0.82	0.85
	Recall	0.76	0.81	0.85	0.74	0.64	0.82	0.85
Hybrid 2 ²	Accuracy	73.28%	81.05%	83.13%	72.38%	66.11%	83.47%	86.16%
	AUCC	72.46%	87.66%	88.58%	80.02%	77.99%	88.72%	96.42%
	F1-Score	0.72	0.81	0.83	0.72	0.63	0.83	0.86
	Precision	0.73	0.81	0.83	0.72	0.74	0.83	0.86
	Recall	0.63	0.71	0.73	0.62	0.56	0.73	0.86

¹ Hybrid TF-IDF with FastText. ² Hybrid TF-IDF with Glove.

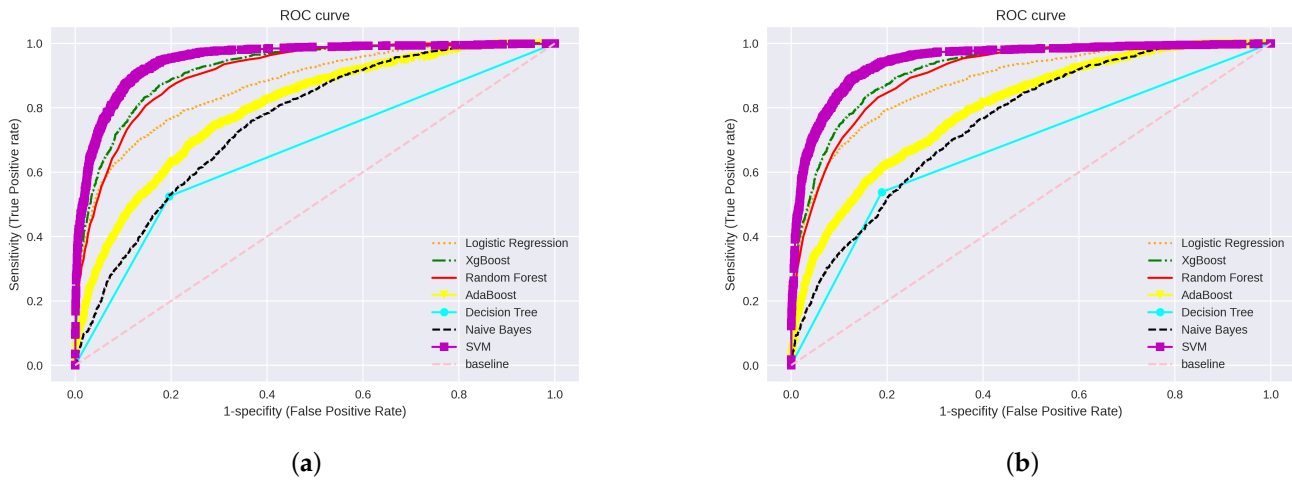


Figure 8. Visualisation of the ROC curves of traditional classifiers using hybrid: (a) TF-IDF with Glove and (b) TF-IDF with FastText word embedding techniques.

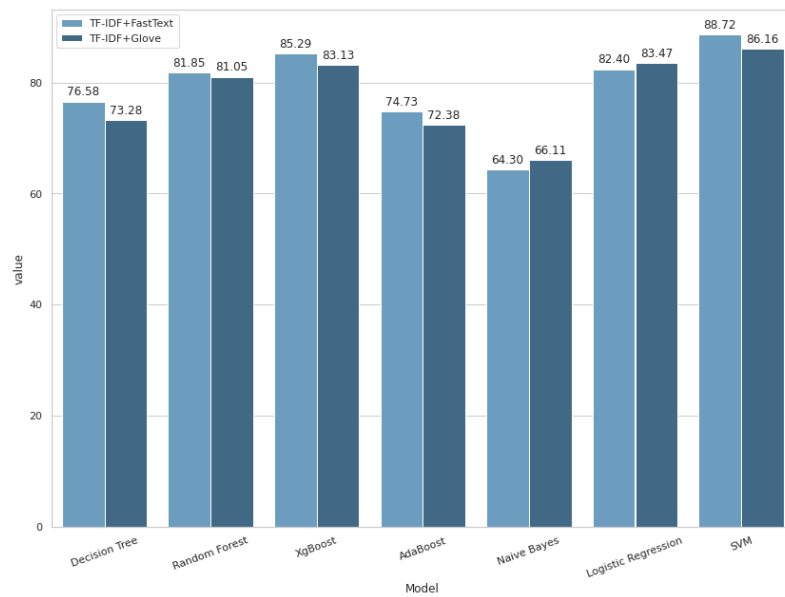


Figure 9. Visualisation of the accuracy of traditional classifiers for hybrid methods: TF-IDF with FastText and TF-IDF with Glove word embedding techniques.

In Table 5, the results of hybrid features with previous methods are illustrated and a comparison is presented, to show the significance of this study. We observe that our hybrid method provides an AUCC score higher than other state-of-the-art methods based on traditional and DL methods, with over 10%, and accuracy over 2%. Our proposal has achieved better classification evaluation results compared with other learning models.

Table 5. Summary of the reviewed COVID-19 classifiers.

Study	Classifier Name	Accuracy	AUCC Score	F1-Score
Alqurashi et al. [24]	FastText + XGBoost	86.8%	85.4%	0.39
Naseem et al. [25]	TF-IDF + RF	84.5%	NA ³	NA ³
Imran et al. [22]	FastText + LSTM	82.4%	NA ³	0.82
Sitaula and Shahi [34]	FastText + ds + BOW + MCNN	71.3%	NA ³	0.50
Singh et al. [35]	Improved LSTM-RNN with attention mechanisms	84.56%	NA ³	0.81
Hybrid Method 1 ¹	TF-IDF and Glove + SVM	86.16%	96.42%	0.86
Hybrid Method 2 ²	TF-IDF and FastText + SVM	88.72%	95.88%	0.86

¹—Hybrid TF-IDF with Glove. ²—Hybrid TF-IDF with FastText. ³—Not applicable.

5. Discussion

To build a classification model with better accuracy and efficiency, features extraction and supervised machine learning methods were evaluated. In this research, we performed sentiment analysis on the ongoing COVID-19 pandemic. Accordingly, different textual analytic visualisation plots were employed to better understand the data. Sentiment analysis mostly helps to analyse people's feelings in a specific event. First, analysis was conducted to categorise the dataset into three different categories (neutral, positive, and negative) using the library python NLTK. It was observed that the dataset was imbalanced and the performance of the methods was biased towards the majority class (positive), thus we used the SMOTE algorithm to generate the synthetic samples for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling. The second analysis phase was based on feature extraction and word embedding; therefore, this step was crucial to enhance classification accuracy. Word embedding is the representation and transformation of the words semantically and syntactically within a document as real-valued vectors before feeding them to the model. TF-IDF, Word2Vec, Glove, and FastText were applied. In order to enhance the classification models, we obtained the best of both worlds: using TF-IDF as a first method to weigh how much each token contributes to the sentence embedding, and Glove word presentation using matrix factorisation techniques; as a second method, we used TF-IDF with FastText to represent each word as an n-gram of characters. Both features are considered to complement each other in representing the tweets. The extensive experiments with different methods evaluation measures suggests that SVM model has performed significantly better compared with other supervised and neural models with different word embedding techniques. Moreover, the proposed method to combine the two-word embedding helped in increasing the accuracy and AUCC score for most of machine learning algorithms. In these the best performance was achieved by TF-IDF, weighted with FastText, fed into the SVM, with 88.72% accuracy, against the lowest accuracy which was achieved by NB when the feature embedding was the TF-IDF N-gram model (where the accuracy was 53.12%). The key difference between Word2Vec and FastText was that, during the learning phase, FastText presented each word as a group of n-grams characters, while Word2Vec considered words as the smallest unit. As shown by the results, all classifiers performed better with the help of pre-trained embedding. Overall, the available machine learning methods can deliver a high performance compared with deep learning. Accordingly, traditional classifiers have better performances, with higher AUCC values.

6. Conclusions

During the COVID-19 pandemic, tweets have represented a source of information and could be reliable as a trigger for disease surveillance models, with the situation and peoples' responses and emotions changing continuously during this critical period. Analysing tweets can help public health services in their early responses when providing signals ahead of outbreaks and providing early warnings before the pandemic spreads. In this research, we proposed a study of COVID-19 sentiments expressed in tweets, which is both

a reliable and a valuable source of information for analysing a large amount of data and studying peoples' behaviour. We applied preprocessing techniques with TextBlob and features embedding methods to enhance machine learning algorithm performances. In the purpose of enhancing classification and machine learning models accuracies, we proposed a combination of TF-IDF with FastText and TF-IDF with Glove to determine the highest performance. The syntactic representation by TF-IDF and semantic representation of text by FastText or Glove showed their complementarity in capturing the tweets information more effectively when they are integrated together in the same model. The study concluded that SVM outperformed other models and reached higher performance with our two features fusion approach compared with other machine learning models. From all seven features embedding methods used, TF-IDF uni-gram, TF-IDF N-gram, Word2Vec, FastText, Glove, and the two hybrids, our approach showed the best results. We also presented a comparison of our models performance of the best feature techniques with previous studies. We concluded that our proposed approaches are slightly better. This can be related to the ability of two effective word representations combined to represent words vocabulary and extract features effectively, in comparison with the other tested methods. Despite the positives, our research has some limitations. The first limitation of our approach is the reliance only on global English-language tweets; for further study, we can extend the model to classify tweets by group of countries that have similar languages, such as Arabic, Spanish, and French, or target societies by including important COVID-19-specific keywords. We can also include data from other platforms depending on their popularity in different regions [61], such as a Google Trend dataset for comparison to analyse and evaluate this model [62]. The second limitation of this study is in ignoring the bad performance of CNN. In the future, this work can focus on optimising the hyper-parameters of DL algorithms to achieve better results.

Author Contributions: Conceptualization, Y.D.; methodology, Y.D.; software, Y.D.; validation, Y.D.; formal analysis, Y.D.; investigation, Y.D.; resources, Y.D.; data curation, Y.D.; writing—original draft preparation, Y.D.; writing—review and editing, Y.D., A.W. (Ahlam Walha) and A.W. (Ali Wali); visualization, Y.D.; supervision, A.W. (Ahlam Walha) and A.W. (Ali Wali). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on IEEE data Port at <https://iee-dataport.org/open-access/coronavirus-COVID-19-geo-tagged-tweets-dataset> (accessed on 26 May 2021).

Acknowledgments: The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4360867DSR01).

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

Abbreviations

The following abbreviations are used in this manuscript:

AdaBoost	adaptive boosting
AI	artificial intelligence
CNN	convolutional neural network
DT	decision tree
FP	false positives

FN	false negatives
KNN	k-nearest neighbours
LR	logistic regression
LSTM	long short-term memory
ML	machine learning
NB	naïve Bayes
NLP	natural language processing
RF	random forest
SA	sentiment analysis
SGD	stochastic gradient descent
SMOTE	synthetic minority oversample technique
SNS	social network sites
SVM	support vector machine
TF-IDF	term frequency-inverse document frequency
TP	true positives
TN	true negatives
URL	uniform resource locator
XGBoost	extreme gradient boosting

References

- Worldometer. Available online: <https://www.worldometers.info/coronavirus> (accessed on 26 May 2021).
- Adamu, H.; Lutfi, S.L.; Malim, N.H.A.H.; Hassan, R.; Di Vaio, A.; Mohamed, A.S.A. Framing twitter public sentiment on Nigerian government COVID-19 palliatives distribution using machine learning. *Sustainability* **2021**, *13*, 3497. [[CrossRef](#)]
- Huang, H.; Peng, Z.; Wu, H.; Xie, Q. A big data analysis on the five dimensions of emergency management information in the early stage of COVID-19 in China. *J. Chin. Gov.* **2020**, *5*, 213–233. [[CrossRef](#)]
- Chakraborty, K.; Bhatia, S.; Bhattacharyya, S.; Platos, J.; Bag, R.; Hassaniien, A.E. Sentiment Analysis of COVID-19 tweets by Deep Learning Classifiers—A study to show how popularity is affecting accuracy in social media. *Appl. Soft Comput.* **2020**, *97*, 106754. [[CrossRef](#)] [[PubMed](#)]
- Depoux, A.; Martin, S.; Karafillakis, E.; Preet, R.; Wilder-Smith, A.; Larson, H. The pandemic of social media panic travels faster than the COVID-19 outbreak. *J. Travel Med.* **2020**, *27*, taaa031. [[CrossRef](#)]
- Pappa, S.; Ntella, V.; Giannakas, T.; Giannakoulis, V.G.; Papoutsis, E.; Katsounou, P. Prevalence of depression, anxiety, and insomnia among healthcare workers during the COVID-19 pandemic: A systematic review and meta-analysis. *Brain Behav. Immun.* **2020**, *88*, 901–907. [[CrossRef](#)]
- Kabir, M.; Madria, S. CoronaVis: A real-time COVID-19 tweets data analyzer and data repository. *arXiv* **2020**, arXiv:2004.13932.
- Taboada, M. Sentiment analysis: An overview from linguistics. *Annu. Rev. Linguist.* **2016**, *2*, 325–347. [[CrossRef](#)]
- Beigi, G.; Hu, X.; Maciejewski, R.; Liu, H. An overview of sentiment analysis in social media and its applications in disaster relief. In *Sentiment Analysis and Ontology Engineering*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 313–340.
- Sailunaz, K.; Alhaji, R. Emotion and sentiment analysis from Twitter text. *J. Comput. Sci.* **2019**, *36*, 101003. [[CrossRef](#)]
- Samuel, J.; Ali, G.; Rahman, M.; Esawi, E.; Samuel, Y. COVID-19 public sentiment insights and machine learning for tweets classification. *Information* **2020**, *11*, 314. [[CrossRef](#)]
- Liu, R.; Shi, Y.; Ji, C.; Jia, M. A survey of sentiment analysis based on transfer learning. *IEEE Access* **2019**, *7*, 85401–85412. [[CrossRef](#)]
- Tyagi, P.; Tripathi, R. A review towards the sentiment analysis techniques for the analysis of twitter data. In Proceedings of the 2nd International Conference on Advanced Computing and Software Engineering (ICACSE), Sultanpur, India, 8–9 February 2019.
- Saura, J.R.; Palacios-Marqués, D.; Ribeiro-Soriano, D. Exploring the boundaries of open innovation: Evidence from social media mining. *Technovation* **2022**, 102447. [[CrossRef](#)]
- Mackey, T.; Purushothaman, V.; Li, J.; Shah, N.; Nali, M.; Bardier, C.; Liang, B.; Cai, M.; Cuomo, R. Machine learning to detect self-reporting of symptoms, testing access, and recovery associated with COVID-19 on Twitter: Retrospective big data infoveillance study. *JMIR Public Health Surveill.* **2020**, *6*, e19509. [[CrossRef](#)] [[PubMed](#)]
- Wan, S.; Yi, Q.; Fan, S.; Lv, J.; Zhang, X.; Guo, L.; Lang, C.; Xiao, Q.; Xiao, K.; Yi, Z.; et al. Relationships among lymphocyte subsets, cytokines, and the pulmonary inflammation index in coronavirus (COVID-19) infected patients. *Br. J. Haematol.* **2020**, *189*, 428–437. [[CrossRef](#)] [[PubMed](#)]
- Rajput, N.K.; Grover, B.A.; Rathi, V.K. Word frequency and sentiment analysis of twitter messages during coronavirus pandemic. *arXiv* **2020**, arXiv:2004.03925.
- Muthusami, R.; Bharathi, A.; Saritha, K. COVID-19 outbreak: Tweet based analysis and visualization towards the influence of coronavirus in the world. *Gedrag Organ. Rev.* **2020**, *33*, 8–9.
- Jelodar, H.; Wang, Y.; Orji, R.; Huang, S. Deep sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: Nlp using lstm recurrent neural network approach. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2733–2742. [[CrossRef](#)] [[PubMed](#)]

20. Aljameel, S.S.; Alabbad, D.A.; Alzahrani, N.A.; Alqarni, S.M.; Alamoudi, F.A.; Babili, L.M.; Aljaafary, S.K.; Alshamrani, F.M. A sentiment analysis approach to predict an individual's awareness of the precautionary procedures to prevent COVID-19 outbreaks in Saudi Arabia. *Int. J. Environ. Res. Public Health* **2021**, *18*, 218. [CrossRef]
21. Ghadeer, A.S.; Aljarah, I.; Alsawalqah, H. Enhancing the Arabic sentiment analysis using different preprocessing operators. *New Trends Inf. Technol.* **2017**, *113*, 113–117.
22. Imran, A.S.; Daudpota, S.M.; Kastrati, Z.; Batra, R. Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on COVID-19 related tweets. *IEEE Access* **2020**, *8*, 181074–181090. [CrossRef]
23. Alam, F.; Dalvi, F.; Shaar, S.; Durrani, N.; Mubarak, H.; Nikolov, A.; Martino, G.D.S.; Abdelali, A.; Sajjad, H.; Darwish, K.; et al. Fighting the COVID-19 infodemic in social media: A holistic perspective and a call to arms. *arXiv* **2020**, arXiv:2007.07996.
24. Alqurashi, S.; Hamoui, B.; Alashaikh, A.; Alhindi, A.; Alanazi, E. Eating garlic prevents COVID-19 infection: Detecting misinformation on the arabic content of twitter. *arXiv* **2021**, arXiv:2101.05626.
25. Naseem, U.; Razzak, I.; Khushi, M.; Eklund, P.W.; Kim, J. Covidsent: A large-scale benchmark Twitter data set for COVID-19 sentiment analysis. *IEEE Trans. Comput. Soc. Syst.* **2021**, *8*, 1003–1015. [CrossRef]
26. Basiri, M.E.; Nemati, S.; Abdar, M.; Asadi, S.; Acharya, U.R. A novel fusion-based deep learning model for sentiment analysis of COVID-19 tweets. *Knowl.-Based Syst.* **2021**, *228*, 107242. [CrossRef]
27. Rustam, F.; Khalid, M.; Aslam, W.; Rupapara, V.; Mehmood, A.; Choi, G.S. A performance comparison of supervised machine learning models for COVID-19 tweets sentiment analysis. *PLoS ONE* **2021**, *16*, e0245909. [CrossRef] [PubMed]
28. Nemes, L.; Kiss, A. Social media sentiment analysis based on COVID-19. *J. Inf. Telecommun.* **2021**, *5*, 1–15. [CrossRef]
29. Loria, S. Textblob Documentation. Available online: <https://buildmedia.readthedocs.org/media/pdf/textblob/dev/textblob.pdf> (accessed on 8 July 2021).
30. Kaur, H.; Ahsaan, S.U.; Alankar, B.; Chang, V. A proposed sentiment analysis deep learning algorithm for analyzing COVID-19 tweets. *Inf. Syst. Front.* **2021**, *23*, 1417–1429. [CrossRef]
31. Li, X.; Zhang, J.; Du, Y.; Zhu, J.; Fan, Y.; Chen, X. A Novel Deep Learning-based Sentiment Analysis Method Enhanced with Emojis in Microblog Social Networks. *Enterp. Inf. Syst.* **2022**, 1–22. [CrossRef]
32. Balli, C.; Guzel, M.S.; Bostanci, E.; Mishra, A. Sentimental Analysis of Twitter Users from Turkish Content with Natural Language Processing. *Comput. Intell. Neurosci.* **2022**, *2022*, 2455160. [CrossRef]
33. Zemberek, NLP Tools for Turkish. Available online: <https://github.com/ahmetaa/zemberek-nlp> (accessed on 20 September 2021).
34. Sitaula, C.; Shahi, T.B. Multi-channel CNN to classify nepali COVID-19 related tweets using hybrid features. *arXiv* **2022**, arXiv:2203.10286.
35. Singh, C.; Imam, T.; Wibowo, S.; Grandhi, S. A Deep Learning Approach for Sentiment Analysis of COVID-19 Reviews. *Appl. Sci.* **2022**, *12*, 3709. [CrossRef]
36. Parimala, M.; Swarna Priya, R.; Praveen Kumar Reddy, M.; Lal Chowdhary, C.; Kumar Poluru, R.; Khan, S. Spatiotemporal-based sentiment analysis on tweets for risk assessment of event using deep learning approach. *Softw. Pract. Exp.* **2021**, *51*, 550–570. [CrossRef]
37. Lamsal, R. Coronavirus (COVID-19) Geo-Tagged Tweets Dataset. 2020. Available online: <https://ieee-dataport.org/open-access/coronavirus-covid-19-geo-tagged-tweets-dataset> (accessed on 26 May 2021).
38. Loper, E.; Bird, S. Nltk: The natural language toolkit. *arXiv* **2002**, arXiv:0205028.
39. Lamsal, R. Design and analysis of a large-scale COVID-19 tweets dataset. *Appl. Intell.* **2021**, *51*, 2790–2804. [CrossRef] [PubMed]
40. Documenting the Now. [Computer Software]. 2020. Available online: <https://github.com/docnow/hydrator> (accessed on 7 July 2021).
41. Hedderich, M.A.; Lange, L.; Adel, H.; Strötgen, J.; Klakow, D. A survey on recent approaches for natural language processing in low-resource scenarios. *arXiv* **2020**, arXiv:2010.12309.
42. Python for NLP: Sentiment Analysis with Scikit-Learn. Available online: <https://stackabuse.com/python-for-nlp-sentimentanalysis-with-scikit-learn/> (accessed on 30 May 2021).
43. Willett, P. The Porter stemming algorithm: Then and now. *Program Electron. Libr. Inf. Syst.* **2006**, *40*, 219–223. [CrossRef]
44. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
45. Joulin, A.; Grave, E.; Bojanowski, P.; Douze, M.; Jégou, H.; Mikolov, T. Fasttext. zip: Compressing text classification models. *arXiv* **2016**, arXiv:1612.03651.
46. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. *arXiv* **2016**, arXiv:1607.01759.
47. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
48. Yang, H.; Fong, S. Optimized very fast decision tree with balanced classification accuracy and compact tree size. In Proceedings of the 3rd International Conference on Data Mining and Intelligent Information Technology Applications, Macao, China, 24–26 October 2011; pp. 57–64.
49. Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 1, pp. 278–282.
50. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.

51. Singer, G.; Marudi, M. Ordinal decision-tree-based ensemble approaches: The case of controlling the daily local growth rate of the COVID-19 epidemic. *Entropy* **2020**, *22*, 871. [[CrossRef](#)]
52. Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text classification algorithms: A survey. *Information* **2019**, *10*, 150. [[CrossRef](#)]
53. Cox, D.R. The regression analysis of binary sequences. *J. R. Stat. Soc. Ser. B* **1958**, *20*, 215–232. [[CrossRef](#)]
54. Naz, S.; Sharan, A.; Malik, N. Sentiment classification on twitter data using support vector machine. In Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Santiago, Chile, 3–6 December 2018; pp. 676–679.
55. Implementing SVM and Kernel SVM with Python’s Scikit-Learn. Available online: <https://stackabuse.com/implementing-svmand-kernel-svm-with-pythons-scikit-learn> (accessed on 30 June 2021).
56. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1746–1751. [[CrossRef](#)]
57. Jabeur, S.B.; Sadaoui, A.; Sghaier, A.; Aloui, R. Machine learning models and cost-sensitive decision trees for bond rating prediction. *J. Oper. Res. Soc.* **2020**, *71*, 1161–1179. [[CrossRef](#)]
58. Antunes, F.; Ribeiro, B.; Pereira, F. Probabilistic modeling and visualization for bankruptcy prediction. *Appl. Soft Comput.* **2017**, *60*, 831–843. [[CrossRef](#)]
59. Gholamy, A.; Kreinovich, V.; Kosheleva, O. Why 70/30 or 80/20 Relation between Training and Testing Sets: A Pedagogical Explanation. 2018. Available online: <https://www.cs.utep.edu/vladik/2018/tr18-09.pdf> (accessed on 31 July 2021)
60. Farquad, M.; Bose, I. Preprocessing unbalanced data using support vector machine. *Decis. Support Syst.* **2012**, *53*, 226–233. [[CrossRef](#)]
61. Singh, M.; Jakhar, A.K.; Pandey, S. Sentiment analysis on the impact of coronavirus in social life using the BERT model. *Soc. Netw. Anal. Min.* **2021**, *11*, 1–11. [[CrossRef](#)]
62. Pota, M.; Ventura, M.; Catelli, R.; Esposito, M. An effective BERT-based pipeline for Twitter sentiment analysis: A case study in Italian. *Sensors* **2020**, *21*, 133. [[CrossRef](#)]