Sentiment Analysis of Covid19 Tweets Using A MapReduce Fuzzified Hybrid Classifier Based On C4.5 Decision Tree and Convolutional Neural Network

Fatima Es-sabery[1*], *Khadija* Es-sabery[1], *Hamid* Garmani[1] and *Abdellatif* Hair[1]

[1]Department of Computer Science, Faculty of Sciences and Technology, Sultan Moulay Slimane University, Beni Mellal 23000, Morocco

Abstract. This contribution proposes a new model for sentiment analysis, which combines the convolutional neural network (CNN), C4.5 decision tree algorithm, and Fuzzy Rule-Based System (FRBS). Our suggested method consists of six parts. Firstly we have applied several pre-processing techniques. Secondly, we have used the fastText method for vectoring the analysed tweets. Thirdly, we have implemented the CNN for extracting and selecting the pertinent features from the tweets. Fourthly, we have fuzzified the CNN output using the *Gaussian Fuzzification* (GF) method for coping with vague data. Then we have applied fuzziness C4.5 for creating the fuzziness rules. Finally, we have used the *General Fuzziness Reasoning* (GFR) approach for classifying the new tweets. In summary, our method integrates the advantages of CNN and C4.5 techniques and overcomes the shortcomings of ambiguous data in the tweets using FRBS, which is consists of three-phase: fuzzification phase using GF, inference mechanism using fuzziness C4.5, and defuzzification phase using GFR. Also, to give our approach the ability to deal with the massive data, we have implemented it on the Hadoop framework of five computers. The experiential findings confirmed that our model operates excellently compared to other chosen models form the literature.

1 Introduction

By nature, humans communicate with each other. In humankind's history, communication is deemed an essential tool to resolve problems and strengthen social commitment and social engagement. Nevertheless, today's human communication has radically changed from the human communication of the past. Presently, social media platforms are frequently used by all society segments as a primary communication manner [1]. YouTube, Twitter, WhatsApp, Facebook, Tiktok, and Instagram are the preeminent social network platforms.

The Twitter platform comprises worthy data toward a variety of areas, such as economic, commercial, social, governmental, and political applications [2]. The analysis manually of Twitter's massive volume of data for extracting valuable information is very challenging. In this case, Twitter opinion mining tools have been demonstrated very useful. Twitter opinion mining's primary goal is to get an idea of how Twitter user's sense toward a particular topic and their ideas and proposals.

In the area of NLP, scientific researchers have carried out Twitter opinion mining by performing five tasks which are: data collection, data cleaning, data vectorization, feature extraction and data classification. In data classification, they employed several algorithms picked out from various types of approaches, such as lexicon-based technique, rule-based fuzzy system, machine learning, hybrid strategies and deep learning.

Deep learning approaches are coming to overcome the problems of machine learning and lexicon-based techniques by using extensive engineering features, that is to say, they extract automatically the relevant and complete features which is amelioration in accuracy. This motivated us to carry out Twitter opinion mining for the English language by applying a hybrid classifier that incorporates CNN and C4.5 algorithms. Although applying the most efficient machine learning and deep learning approaches, NLP's inherent vagueness requires more solutions. Numerous works from the literature [3], [4-5] prove that fuzzy logic theories are the appropriate techniques to handle ambiguous, uncertain and imprecise information.

Therefore, in this proposal, we have implemented the convolution neural network to extract the relevant and accurate features from the tweets. We have applied fuzzy C4.5 as a classifier. And we have utilized the fuzzy rule-based system to deal with imprecision and ambiguous data in each tweet. Because of the ever-increasing amount of data, we have implemented our hybrid method using the cluster of Hadoop. Our approach aims to predict the sentimental rate of each tweet.

The remainder of this contribution is arranged in the following order: Section 2 discusses certain existing research. Section 3 discusses our proposed hybrid method. Section 4 summarizes the findings of the performed experiments and makes comparisons. Section

* Fatima Es-sabery: fatima.essabery@gmail.com

5 concludes with conclusions and several recommendations for future contributions.

2 Literature reviews

The pioneer research works picked up from the literature will be briefly discussed in this section, including machine learning algorithms for opinion classification, deep learning for opinion classification, and fuzzy approaches for sentiment analysis.

Kanakaraddi *et al.* [6] have introduced an analysis of diverse supervised machine learning methods for opinion mining such as SVM, random forest, max entropy, and naive bayes. Amongst all these methods, the SVM gives a better classification rate equal to 79.90%.

Authors of the research paper [7] applied five types of machine learning approaches on the movie review dataset, which consists of 2000 reviews. Hence, the employed supervised classifiers in this work are decision tree algorithms (C4.5, CART, and ID3), Bernoulli and multinomial naive bayes, SVM, and maximum entropy. The experimental results outline that multinomial naive Bayes delivers a strong performance in terms of classification rate (88.5%), F-score (87.87%), and precision (92.94%). While, the SVM achieves the higher performance in terms of recall (89.33%).

In [8], the authors examine the restaurant customers' reviews using multiple machine learning classifiers with supervision such as k-nearest neighbor, C4.5 algorithm, SVM, naïve bayes, and random forest. Their simulation outcomes revealed that the SVM classifier has a greater classification rate of 94.56 % for the used dataset than other classifiers.

Liao *et al.* [9] applied CNN for classifying Twitter dataset Due to its ability to capture, detect and extract global features based on the linguistic and lexical relationship amongst these global features.

In [10], the authors improved the convolutional neural network by combining it with multi-Head attention technique. Their suggested approach integrates features to create diverse features channels and applies a CNN based on multi-channels to detect opinion words from various aspects. Then they applied the multi-head attention technique to extract the pertinent features from diverse dimensions. The empirical outcomes prove that the proposed approach attained the highest classification precision equal to 86.32% at the number of heads of the multi-attention mechanism equal to eight.

Behera *et al.* [11] designed an innovative hybrid procedure by integrating the CNN and LSTM for carrying out opinion mining of reviews posted at different areas. It gave a classification precision equal to 83.13% on movie review dataset.

Bedi *et al.* [4] suggested an innovative hybrid fuzziness deep learning method that incorporates the abilities of LSTM in automating feature engineering discovery from addressed data and the capabilities of fuzzy logic theory in dealing with the present uncertainty and ambiguity in the analyzed data to afford a more suitable sentiment forecast to the user.

In [12], the authors offered a novel convolutional neural-fuzziness network that combines fuzzy logic theory and a convolutional neural network. The incorporation brings the benefits of both fuzziness logic theory and the CNN in deriving valuable high-level global and local features from imprecise and uncertain data collectively.

The paper [13] suggests a technique for doing sentiment classification employing a multilayer perceptron back-propagation network and fuzzy logic theory. In this proposed approach, the input online comments are fuzzified utilizing the Gaussian fuzzification method, and the fuzzification matrix is produced. This produced matrix is reversed and passed to the implemented multilayer perceptron back-propagation network.

3 Our proposed approach

This section elaborates on the proposed hybrid approach that consolidates the convolutional neural network, C4.5 decision tree algorithm, and rule-based fuzziness system and Hadoop platform for performing the English sentence level classification. Primarily, our principal focus is to increase the classification rate of the sentiments analysis by applying the CNN as feature extractor, by using the rule-based fuzziness system for handling the vagueness and uncertainty that existed in the human expressed sentiment at the level of every tweet, and by implementing the proposed approach in a parallel manner employing the Hadoop cluster. In general, our proposed hybrid model comprises seven steps: data acquisition, data cleaning, data vectorization, features extraction and selection, data fuzzification, fuzziness C4.5, data classification, data parallelization.

3.1.1 Data acquisition

The implementation of our proposed hybrid model is performed employing Python language version 3.10. 0a5. and we execute our model on two big datasets as presented as follows:

Sentiment140 is downloaded using the link https://www.kaggle.com/kazanova/sentiment140. It initially consists of 1600000 tweets annotated with 800 000 positive labels, and 800 000 negative labels. In this dataset, the decision attribute takes either value 4 or 0. The decision attribute value 4 designates that the tweet is positive, and the decision attribute value 0 designates that the tweet is negative. This dataset permits us to identify the emotion and attitude over a product, service, topic, or brand on the Twitter Platform.

COVID-19 Sentiments is also a massive dataset downloaded using the link https://www.kaggle.com/abhaydhiman/covid19-sentiments. It consists of 259 458 neutral tweets, 120 646 negative tweets, and 257 874 positive tweets. So, this dataset contains 637 978 as the total number of collected tweets. The decision attribute value in this dataset is annotated as either negative, neutral, or positive. The neutral value takes 0, the negative value takes a number between -1 and 0, and the positive value takes a number between 0 and 1.

3.1.2 Data cleaning

Generally, tweets are unstructured and semi-unstructured data. Also, it contains an amount of incomplete, inconsistent, undesired, and noisy data. Therefore, to avoid these shortcomings of tweets and extract valuable knowledge from these unstructured data, it is essential to carry out extensive data pre-processing techniques on tweets to become proper for implementing text mining and natural language processing methods. Consequently, the following data preprocessing tactics are trained in this contribution to assure tweets quality:

- Expand abbreviation, replace slang, and correct spelling mistakes.
- Remove usernames, numbers, white-spaces, hashtags, special characters, URLs, and punctuation.
- Convert all existed uppercase letters to lowercase.
- Replace extended words and remove stop-words.
- Tokenization, lemmatization, and stemming.

After the text preprocessing stage, the next step is the text vectorization, which transforms the input tweet into a numeric vector. In this, the FastText word embedding method is implemented because it achieved good classification performance compared to other approaches as described in the paper [14]. In this contribution, the data preprocessing phase outputs will be the inputs of the text vectorization phase.

3.1.3 Data vectorization

The purpose of this proposal is to recognize the sentiment polarities of the collected tweets from the Twitter platform. Therefore, the preprocessed tweets should be expressed with machine language using word vectorization methods for later analysis, process and classification. In the literature, there are several word vectorization methods such as FastText, N-gram, Word2Vec, GloVe, IF-IDF and Bag-Of-Words, which are achieved a classification rate equal to 87.13 %, 51.76 %, 77.43 %, 72.23 %, 71.05 % and 64.24 %, respectively, according to the comparative study performed in the paper [14].

In this phase, we have conducted unsupervised training using data vectorization method FastText that converts each word with n-gram=2 into a low dimensional vector. FastText vectorizes every tweet by using a bag of n-grams characters or employing a bag of terms and process it using the skip-gram approach or the continuous bag-of-terms approach to get its low dimensional.

Based on the comparative study performed in [15] for comparing the skip-gram approach or the continuous bag-of-terms approach. This study's experimental findings proved that the skip-gram approach is more efficient than the continuous bag-of-words for representing words but needs a long time to be trained. Therefore, we have used the skip-gram model in our work because we have resolved its shortcoming concerning the time by implementing the Hadoop ecosystem.

After applying FastText as the data vectorization phase of our work, in which we convert the tweets into a matrix of low dimensional embedding vectors. The next stage is the feature extraction and selection using the CNN, as described in the following subsection.

3.1.4 Features extraction and selection

Extracting and selecting the most relevant features is deemed as an essential phase in different utilization of natural language processing and many research works have been performed on generating robust, complete, and appropriate features. After, we have studied different research works from the literature [16-18]. We have deduced that the recent works have been given much attention to feature extraction engineering adopted by the deep learning models instead of handcrafted features adopted by traditional machine learning algorithms.

The CNN is recognized as one among the most common kinds of deep learning systems developed to give an appropriate representation of their inputs. According to its overall architecture, the convolutional neural network can be deemed a good choice for extracting and selecting the most appropriate features in this work. Generally, the simple version of the CNN comprises of a convolutional layer, a pooling layer, and a dense layer.

Convolution layer: This first layer identifies and extracts the most relevant features from the constructed word embedding matrix E in the data vectorization phase. The convolution layer consists of multiple convolution operations. At every convolution operation, a slid filter (S) is implemented over each word embedding matrix window (EW) picked up from the E, and a feature map is produced as a result. Therefore, multiple convolution operations indicate multiple filters with changing window size are implemented over E, and multiple feature maps are provided as outputs of these convolution operations. We assume that the $EW = [v_1; v_2; \ldots; v_n]$ with v_i in R^m, a feature map M_i is constructed by applying S_i over a EW with size $V_{i:i+x-1}$ by employing the following equation:

$$M_i = ReLU\,(S_i \,.\, V_{i:i+x-1} + a) \quad (1)$$

Where the term $ReLU$ refers to a non-linear activation method named a rectified linear unit as illustrated in (2); a in R indicates the applied bias and x represents the size of the employed filter S. Thus, the feature map $M_0 = [F_0; F_1; \ldots; F_{i+x-1}]$ is constructed by the implementation of (1) in all selected window EW from the matrix E. Several filters $S_{i:1\to z}$ are exercised to create a set of feature maps $MF_{i:1\to z}$.

The produced set of feature maps $MF_{i:1\to z}$ is activated, i.e., convert the linear set of feature map to a non-linear set of feature map by the application of the $ReLU$ over the linear set of feature map. The $ReLU$ is computed utilizing the next equation (2):

$$ReLU(y) = max\,(0; y) \quad (2)$$

In general, if the ReLU method receives a negative value as income it generates the value 0 as output. Furthermore, if this method receives a positive value as income, it generates a positive value as output.

After applying all operations of the convolution layer for extracting the most appropriate features from the produced embedded matrix in the data vectorization phase, we have obtained a set of non-linear feature map. The next phase is to pass these obtained sets of non-linear feature map into the pooling layer for selecting the most relevant features and decrease the high feature dimensionality.

Pooling layer: After implementing the convolution layer for extracting the appropriate features by applying several slid filters over the produced embedding matrix in the data vectorization phase, the next stage is implementing the pooling layer to pick out the most pertinent and appropriate features by reducing the extracted feature maps dimensionality in the convolution phase. The pooling layer applied either max-pooling or average-pooling operation over the extracted set of feature maps for selecting the pertinent features. The average-pooling function computes the average of all features of the obtained feature maps in the preceding operation and considers the result as the pooled feature. The max-pooling function deems the pooled feature as the feature that has the highest possible value of the obtained feature maps in the previous convolution layer and drops the remainder.

In our proposal, we have employed the max-pooling function. The max-pooling function is exercised at every feature map M_i and selects the feature with the highest value in the feature map as the pooled feature $pf = max\ [F_i]$. This operation produces a set of pooling features with its size equal to the feature maps' number N in the pooling layer input. And the obtained set of optimum features is further passed to dense layer which also called fully connected layer

Dense layer or fully connected layer: is applied in this work to convert the pooling layer outputs to linear outputs using the Softmax activation function. Its operation may be summarized as a linear operation in which every input is weighted differently and connected to all outputs. The dense layer transforms the pooled feature maps into linear output employing the following equation (3):

$$Lo = soft\ (W_c * P f + A) \tag{3}$$

Where Lo is the computed linear value, W_c is the applied matrix of the weight, Pf indicates the pooled feature maps which is generated from the pooling layer, A is the applied bias and $soft$ is the Softmax activation method, that is defined as in the equation (4):

$$soft(y) = \frac{e_j^n}{\sum_{i=1}^{I} e_j^n} \tag{4}$$

Where $soft$ is the Softmax activation procedure, n is the entered neuron value, e_j^n indicates the ordinary exponential method of the entered neuron, and I indicates the total number of categories in the trained set of data.

After the application of CNN for extracting and selecting the most appropriate features in this work. The next phase is the application of Gaussian membership function for fuzzifying the obtained features in CNN phase. We applied the fuzzification approach in this work for giving our suggested hybrid model the capacity to cope with ambiguous and unclear data and then improving the accuracy of our suggested approach.

3.1.5 Data fuzzification

After extracting and selecting the most relevant and consistent features by applying the convolutional neural network. The next stage is the text fuzzification step which aims to fuzzify the set of features obtained in the preceding phase. This stage's main purpose is to fuzzify the features in order to apply the fuzzy C4.5 decision tree over them and give our model the ability to deal with uncertain and vague data. In this work, we apply the fuzzification function to turn out the neuron values of denser layer to a set of fuzzy values by measuring the membership degree of each neuron value employing Gaussian membership function. We have chosen the Gaussian membership function instead of triangular of trapezoidal membership functions because of the experimental result presented in the paper [3], which proved that the Gaussian membership function achieves a good accuracy equal to 94.87% compared to trapezoidal membership function that reaches an accuracy equal to 91.21% and triangular membership function that gives an accuracy equal to 90.14%. The Gaussian function is defined by two variables r is the central value, and $d > 0$ indicates the standard deviation and the membership degree of variable z is computed employing the next equation (5).

$$\mu_A(z) = e^{-\frac{(z-r)^2}{2.d^2}} \tag{5}$$

After we fuzzified the neuron values of the dense layer using the Gaussian membership function we get the fuzzy neuron values. Hence, the next stage is the implementation of fuzziness C4.5 method which aims to compute the fuzziness information gain ratio.

3.1.6 Fuzziness C4.5

After the text fuzzification phase, in which we fuzzified the dense layer's crisp neuron values. The next stage aims to construct the fuzziness decision tree by applying the fuzziness C4.5 decision tree method, then extract the fuzziness rules from the created fuzziness tree and store them in the rule base. Our classifier in this contribution combines the principle of the C4.5 decision tree and fuzzy set theory.

Generally, every feature in every used dataset has multiple values, and these values are represented by the fuzziness sets in the fuzziness logic theory. In this theory, the membership function describes every fuzzy set. We assume E is the set of all dataset instances. $F^{(n)} = \{n = 1, 2, ..., K\}$ indicates the total number of the features in the analyzed dataset, and each feature takes multiple fuzzy values represented by multiple fuzzy set or linguistic term $F_c^{(n)} = \{c = 1, 2, ..., l\}$, $D_{F_c^{(n)}}$ represents the membership degree of the fuzziness set $F_c^{(n)}$, and the decision feature in the used dataset is represented by the linguistic terms $Z_a = \{a = 1, 2, ..., m\}$.

Let D_{Z_a} signifies the membership rate of the linguistic word Z_a.

The level of membership of the target feature YD of the *cth* linguistic term of the nth feature $F_c^{(n)}$ concerning the *ath* fuzzy decision feature value Z_a is measured utilizing the next equation (6):

$$YD_{F_c^{(n)}}(Z_a) = \frac{\sum_{y \in Y_a} D_{F_c^{(n)}}(y^{(n)})}{\sum_{y \in Y} D_{F_c^{(n)}}(y^{(n)})} \quad (6)$$

Therefore the entropy based on fuzzy logic (*EF*) of the cth linguistic term of the *nth* feature $F_c^{(n)}$ is computed using the following equation (7):

$$EF_{F_c^{(n)}} = -\sum_{a=1}^{m} YD_{F_c^{(n)}}(Z_a)\ log\ (YD_{F_c^{(n)}}(Z_a)) \quad (7)$$

Furthermore, the entropy based on fuzzy logic (*EF*) of the nth feature $F^{(n)}$ is measured as a sum of the weighted of $EF_{F_c^{(n)}}$:

$$EF_{F^{(n)}} = \sum_{c=1}^{l} \frac{\sum_{y \in Y} D_{F_c^{(n)}}(y^{(n)})}{\sum_{j=1}^{l} \sum_{y \in Y} D_{F_c^{(n)}}(y^{(n)})} \cdot EF_{F_c^{(n)}} \quad (8)$$

On the other hand, the membership degree of the decision feature *YD* of the set of training examples concerning the ath fuzzy decision feature value Z_a is determined utilizing the following equation (9):

$$YD(Z_a) = \frac{\sum_{y \in Y_a} D_{Z_a}(y)}{\sum_{y \in Y} D_{Z_a}(y)} \quad (9)$$

Therefore, the entropy based on fuzzy logic (*EF*) of the set of training examples is measured accordingly to the following equation (10):

$$EF = -\sum_{a=1}^{m} YD(Z_a) \log(YD(Z_a)) \quad (10)$$

Hence, the information gain based of fuzzy set theory (*IGF*) of the nth feature in terms of instances of training datasets is computed by employing the following equation (11):

$$IGF_{F^{(n)}} = EF - EF_{F^{(n)}} \quad (11)$$

Hence, the split information $SI_{F^{(n)}}$ of the nth feature is computed by applying the following equation (12):

$$SI_{F^{(n)}} = \sum_{c=1}^{l} -\left(\frac{\sum_{y \in Y} D_{F_c^{(n)}}(y^{(n)})}{\sum_{j=1}^{l} \sum_{y \in Y} D_{F_c^{(n)}}(y^{(n)})}\right) * \log\left(\frac{\sum_{y \in Y} D_{F_c^{(n)}}(y^{(n)})}{\sum_{j=1}^{l} \sum_{y \in Y} D_{F_c^{(n)}}(y^{(n)})}\right) \quad (12)$$

Therefore the fuzziness ratio of information gain of the *nth* feature is computed by applying the next equation (13):

$$IGRF_{F^{(n)}} = \frac{IGF_{F(n)}}{SI_{F(n)}} \quad (13)$$

After we have created the fuzzy decision tree by applying the fuzziness C4.5 based on the fuzzy information gain ratio. We have extracted all possible fuzzy rules from the constructed fuzzy decision tree and we store them in the rule base. Therefore the next stage of our work is the application of the general fuzzy reasoning method over the obtained rule base for classifying the new examples.

3.1.7 Data classification

After both stages of creating the fuzzy decision tree and extracting the fuzzy rules by implementing the fuzziness C4.5 decision tree method based on fuzziness ration of information gain. Therefore the trained model is constructed, and the next phase is evaluating our created learning model. I.e., we have applied the general fuzziness reasoning algorithm over the generated rule base for classifying the new input example and defining the class label it belongs to. The general fuzziness reasoning approach follows up the subsequently introduced steps below for classifying a new instance:

For classifying the new example $i_n = \{f_1, f_2, f_3, ?\}$, that has three feature, and an unknown decision feature "?" using the general fuzzy reasoning approach. We assume that the values of these f_1, f_2 *and* f_3 features are defined as the fuzzy sets, and we assume that the $D(f_1) = 0.54$, $D(f_2) = 0.25$ and $D(f_3) = 0.76$ are the computed membership degree using Gaussian membership function of f_1, f_2 *and* f_3 respectively, and we possess four fuzzy rules as described below:

FR1*:* **IF** *M* is m_1 **AND** *N* is n_1 **AND** *P* is p_1 **THEN** *Z* is z_1. With $D(m_1) = 0.63$ $D(n_1) = 0.19$ and $D(p_1) = 0.95$

FR2*:* **IF** *M* is m_2 **AND** *N* is n_2 **AND** *P* is p_2 **THEN** *Z* is z_2. With $D(m_2) = 0.11$, $D(n_2) = 0.89$ and $D(p_2) = 0.65$

FR3*:* **IF** *M* is m_3 **AND** *N* is n_3 **AND** *P* is p_3 **THEN** *Z* is z_1. With $D(m_3) = 0.21$, $D(n_3) = 0.98$ and $D(p_3) = 0.48$

FR4*:* **IF** *M* is m_4 **AND** *N* is n_4 **AND** *P* is p_4 **THEN** *Z* is z_2. With $D(m_4) = 0.81$, $D(n_4) = 0.49$ and $D(p_4) = 0.27$.

Stage 1: compute the rate *r* that input example (f_1, f_2, f_3) matches every fuzz rule term (m_1, m_2, m_3, m_4, n_1 , n_2, n_3, n_4, p_1, p_2, p_3, p_4), and then we will use these computed rates to calculate the compatibility rate (CR) for every fuzzy rule.

1. r(f_1,m_1) = min(D(f_1),D(m_1)=min(0.54,0.63)=0.54
2. r(f_2,n_1) = min(D(f_2),D(n_1)=min(0.25,0.19)=0.25
3. r(f_3,p_1) = min(D(f_3),D(p_1)=min(0.76,0.95)=0.76
4. r(f_1,m_2) = min(D(f_1),D(m_2)=min(0.54,0.11)=0.11
5. r(f_2,n_2) = min(D(f_2),D(n_2)=min(0.25,0.89)=0.25
6. r(f_3,p_2) = min(D(f_3),D(p_2)=min(0.76,0.65)=0.65
7. r(f_1,m_3) = min(D(f_1),D(m_3)=min(0.54,0.21)=0.21
8. r(f_2,n_3)= min(D(f_2),D(n_3)=min(0.25,0.98)=0.25
9. r(f_3,p_3) = min(D(f_3),D(p_3)=min(0.76,0.48)=0.48
10. r(f_1,m_4) = min(D(f_1),D(m_4)=min(0.54,0.81)=0.54
11. r(f_2,n_4) = min(D(f_2),D(n_4)=min(0.25,0.49)=0.25
12. r(f_3,p_4) = min(D(f_3),D(p_4)=min(0.76,0.27)=0.27

Therefore:
- min(1,2,3) = min(0.54,0.25,0.76)= 0.25
- min(4,5,6)= min(0.11,0.25,0.65)= 0.11
- min(7,8,9) = min(0.21,0.25,0.48)= 0.21
- min(10,11,12)= min(0.54,0.25,0.27)= 0.27

Hence the *CR* between instance in and every fuzzy rule FR_1, FR_2, FR_3 and FR_4, equals : $CR(i_n, FR_1) = 0.25$, $CR(i_n, FR_2)=0.11$, $CR(i_n, FR_3)=0.21$ and $CR(i_n;FR_4) =0.27$.

Stage 2: For every decision feature value, measure the classification degree CD_z. So, in our example we possess two decision feature value z_1 and z_2 such as:

- $CD_{z_1} = f\{CR(i_n, FR_x)|z_1\} = CR(i_n, FR_1) + CR(i_n, FR_3) = 0.25+0.21 = 0.46$

- $CD_{z_2} = f\{CR(i_n, FR_x)|z_2\} = CR(i_n, FR_2) + CR(i_n, FR_4) = = 0.11+0.27 = 0.38$

Stage 3: Attach the decision feature value z_1 to the input example $i_n = \{f_1, f_2, f_3, z_1\}$, where z_1 is the decision feature value with the greatest sum (CD_{z_1}= 0.46) computed in the stage 2.

We have chosen in this contribution to apply the general fuzzy reasoning approach for classifying new input examples instead of applying the classic fuzzy reasoning approach because of the introduced experimental result in the paper [5] which proved that the general fuzzy reasoning approach gives better accuracy equals 86.56% than the classic fuzzy reasoning method which achieves an accuracy equals 63.96%. So, after we have explained how we have applied the general fuzzy reasoning method in this work. The next stage is the clarification of how we have used the Hadoop framework in this contribution.

3.1.8 Data parallelization

The Hadoop framework [19] is used to store and handle a large-scale dataset. Our given dataset is partitioned into testing subset and training subset saved in a distributed manner on five computers using the HDFS with its NameNode and DataNodes. The data learning process is also conducted in a parallel manner employing the MapReduce with its JobTracker and TasksTrackers.

Therefore, in the first MapReduce job, the training dataset is divided into chunks. Every chunk is inputted to every Mapper function for applying on it the data preprocessing techniques, FastText word embedding for converting the chunk into a numerical vector, then application of CNN for extracting and selecting the most relevant features, after utilization of Gaussian membership function for fuzzifying the selected feature, finally the application of fuzziness C4.5 for creating the fuzziness decision tree of this chunk of data. At every Mapper node level, we get a small fuzzy decision tree, so we used two reducer nodes to aggregate the obtained Mapper nodes results. The output result of both reducer nodes is the complete fuzzy decision tree.

In the second MapReduce job, we have executed our generated fuzzy decision tree in the first MapReduce job over the testing dataset using five mapper nodes. The mapper tasks aim to apply the general fuzzy reasoning approach for classifying the testing dataset, and the classification results are stored in HDFS.

4 Experiments and results

This section introduces in depth the performance of the suggested hybrid system that combines several pre-processing techniques, FastText word embedding method [20], convolutional neural network, gaussian membership function, fuzziness C4.5 decision tree, general fuzziness reasoning method and Hadoop ecosystem. The performance of our suggested hybrid system and other hybrid approaches selected from the literature (Xu *et al.* [21], Liao *et al.* [9], Maheswari *et al.* [22], and Es-sabery *et al.* [14]) is assessed by measuring five assessment criteria as discussed in below:

$$Accuracy = \frac{TruePositiveTweets + TrueNegativeTweets}{TruePositiveTweets + FalsePositiveTweets + TrueNegativeTweets + FalseNegativeTweets}$$

$$Error = \frac{FalsePositiveTweets + FalseNegativeTweets}{TruePositiveTweets + FalsePositiveTweets + TrueNegativeTweets + FalseNegativeTweets}$$

$$Precision = \frac{TruePositiveTweets}{TruePositiveTweets + FalsePositiveTweets}$$

$$Recall = \frac{TruePositiveTweets}{TruePositiveTweets + FalseNegativeTweets}$$

$$F\text{–}Measure = 2 * \frac{Recall * Precision}{Recall + Precision}$$

Fig.1 represents the obtained accuracy and error rate after the implementation of our suggested hybrid model on the Sentiment140 and COVID-19 Sentiments datasets.

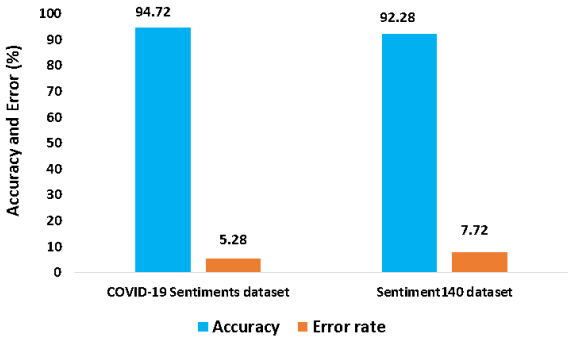

Fig. 1. Accuracy and Error rate of our hybrid model.

The experiential results in Fig.1 proved that our suggested hybrid model operates excellently on the Sentiment140 and COVID-19 Sentiments datasets in terms of error rate (7.72%, 5.28%), and accuracy (92.28%, 94.72%) respectively.

Fig.2 represents the obtained results in terms of precision, recall and f-measure after the implementation of our suggested hybrid model on the Sentiment140 and COVID-19 Sentiments datasets.

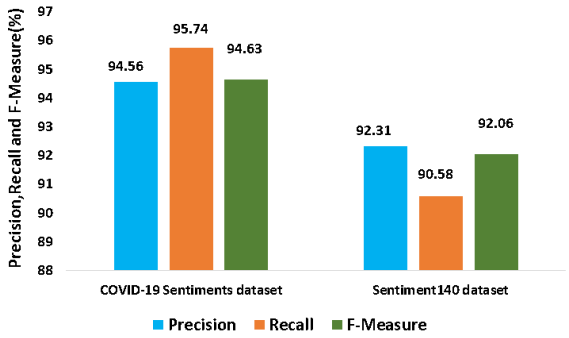

Fig. 2. Precision, Recall, and F-Measure rate of our hybrid model.

From Fig.2, we deduced that our proposed hybrid model achieved (92.31%, 94.56%) in precision, (90.58%, 95.74%) in recall, and (92.06%, 94.63%) in f-measure on the Sentiment140 and COVID-19 Sentiments datasets respectively.

Fig.3, illustrates the empirical results attained in terms of accuracy implementing our proposed hybrid model, Xu et al. [21], Liao et al. [9], Maheswari et al. [22], and Es-sabery et al. [14] classifiers over Sentiment140 and COVID-19 Sentiments datasets in order to prove the efficiency of our suggested hybrid classifier by comparing its achieved performance to the experimental performance attained with selected classifiers.

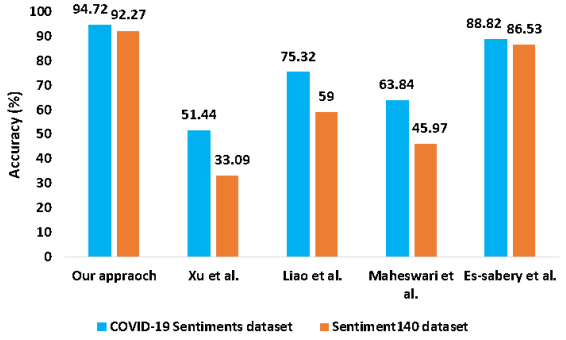

Fig. 3. Accuracy obtained by implementing our proposed hybrid model and other methods.

As depicted in Fig. 3, Xu et al. [21] has achieved a lower accuracy (51.44 %, and 33.09 % in the case COVID-19 Sentiments and Sentiment140 datasets, respectively) compared to other classifiers because the authors of this approach [21] do not give great importance for text preprocessing tasks. Maheswari et al. [22] has achieved an accuracy equals 63.84 %, and 45.97 % in the case COVID-19 Sentiments and Sentiment140 datasets, respectively. Its performance is better than Xu et al. [21], because it Mamdani fuzzy system as the classifier, which is very efficient in dealing with inherent and ambiguous data. Although this approach [22] has lower performance compared to the remainder evaluated approaches because the AFINN word dictionary has a limited capacity for detecting all relevant features. Liao et al. [9] has achieved an accuracy equals 75 .32 %, and 59 % in the case COVID-19 Sentiments and Sentiment140 datasets, respectively, which is a good performance compared to both [21], and [22] classifiers. Because this approach applied the convolution neural network as a classifier. Furthermore, this classifier [9] has lower accuracy compared to our suggested hybrid model in this work and the approach [14], because it applied the one-hot-vector as word embedding method, which has a lower accuracy compared to FastText word embedding method used in the work [14] and do not apply fuzzy logic theory for dealing with inherent and uncertain data as performed in our suggested hybrid model. Es-sabery et al. [14] approach has attained an accuracy equals 88.82 %, 86.53 % in the case of COVID-19 Sentiments and Sentiment140 datasets, respectively. This approach has a better performance compared to all previously studied approaches because the authors this approach apply a comparative study amongst several approaches for choosing the most efficient technique to perform every task. But this approach [14] has lower accuracy compared to our suggested hybrid model in this work. Our proposed hybrid model has achieved higher accuracy than all previously evaluated classifiers, which equal 94.72 % and 92.27 % in the case of COVID-19 Sentiments and Sentiment140 datasets, respectively. Therefore, the good performances achieved by our suggested hybrid model compared to all other approaches are due to the utilization of convolution neural network, which has a higher ability to elicit and pick out the most relevant features accurately and the combination of fuzzy logic theory and decision tree to cope with ambiguous and unclear data in order to improve the efficiency of the classification process.

Fig. 4, illustrates the empirical results attained in terms of error rate implementing our proposed hybrid model, Xu et al. [21], Liao et al. [9], Maheswari et al. [22], and Es-sabery et al. [14] classifiers over Sentiment140 and COVID-19 Sentiments datasets.

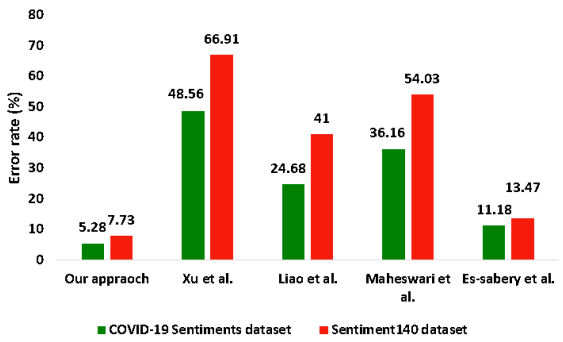

Fig. 4. Error rate obtained by implementing our proposed hybrid model and other methods.

As illustrated in Fig. 4, we remark that our suggested hybrid model reached an error rate equals to 5.28 % and 7.73 % in the case of COVID-19 Sentiments and Sentiment140 datasets, respectively. Xu et al. [21] approach achieved an error rate equals to 48.56 %, and

66.91 % in the case of COVID-19 Sentiments and Sentiment140 datasets, respectively. Liao et al. [9] method attained an error rate equals 24.68 %, and 41 % in the case of COVID-19 Sentiments and Sentiment140 datasets, respectively. Maheswari et al. [22] model has an error rate equals 36.16 %, 54.03 % in the case of COVID-19 Sentiments and Sentiment140 datasets, respectively. And Es-sabery et al. [14] classifier achieved an error rate equals 11.18 %, and 13.47 % in the case of COVID-19 Sentiments and Sentiment140 datasets, respectively. From these experimental outcomes, we notice that our suggested hybrid model has a lower error rate than all other evaluated approaches, and Xu et al. [21] approach has a higher error rate compared to all other assessed methods. Therefore and we said earlier, the good performances achieved by our suggested hybrid model are due to the followed steps in the learning process of sentiment analysis and the selected technique to implement at every step. Generally, our suggested hybrid model has given great importance to every step of the sentiment analysis process by choosing the most efficient method to do the necessary tasks in every step.

The second experiment is performed for computing the Precision, Recall, and F-Measure rate of our proposed hybrid model and the four other approaches selected from the literature in order to prove the effectiveness of our suggested hybrid model.

Tables 1 and 2 illustrates the experimental result in terms of Precision, Recall, and F-Measure rate reached by our suggested hybrid model, Xu et al. [21], Liao et al. [9], Maheswari et al. [22], and Es-sabery et al. [14] classifiers.

Table 1. Precision, Recall, and F-Measure rate of our proposed hybrid model and the four other approaches selected from the literature over Sentiment140 dataset.

	Precision (%)	Recall (%)	F-Measure (%)
Our approach	92.31	90.58	92.06
[21]	39.96	40.69	40.48
[9]	58.99	59.02	58.97
[22]	46	45.34	45.48
[14]	83.04	82.33	83.87

Table 2. Precision, Recall, and F-Measure rate of our proposed hybrid model and the four other approaches selected from the literature over COVID-19-Sentiments dataset.

	Precision (%)	Recall (%)	F-Measure (%)
Our approach	94.56	95.74	94.63
[21]	61.29	61.04	61.01
[9]	75.10	73.64	74.64
[22]	64.07	63.51	63.42
[14]	86.67	86.51	85.54

As depicted in Tables 1 and 2, our suggested hybrid model outperforms all other approaches in terms of Precision, Recall, and F-Measure rate. Our approach achieved a Recall equals 95.74 % and 90.58 % over COVID-19-Sentiments and Sentiment140 datasets, which indicates the proportion of negatives tweets that our developed model accurately classifies. Concerning the percentage of Precision rate gauges the closeness of the evaluation criteria to each other, and our approach has a higher closeness rate which equals 94.56 %, 92.31 % over COVID-19 Sentiments and Sentiment140 datasets than all other evaluated approaches. Finally, the F-Measure metric measures the accuracy and robustness of our suggested model, which equals 94.63 % and 92.06 % over COVID-19 Sentiments and Sentiment140 datasets, respectively. After analysed all computed evaluation metrics, we deduced that our suggested model is more powerful and efficient than all other evaluated approaches.

5 Conclusion

In this contribution, we have suggested a novel hybrid model for classifying tweets into class label positive, negative, or neutral based. The hybrid model consists of six phases: Data acquisition phase, in which we have selected both Sentiment140 and COVID-19 Sentiments datasets to assess the effectiveness of our contribution, data pre-processing phase by applying all necessary pre-processing tasks, data representation phase using FastText word embedding method, data extraction and selection phase employing CNN, data classification using Gaussian membership function for fuzzifying the extracted features, fuzziness C4.5 for creating the fuzziness tree, and general fuzzy reasoning method for classifying the new instances.

We have performed multiple experiments to evaluate the performance of our suggested hybrid model compared to other approaches. The experimental findings proved that our suggested model outperforms all other evaluated approaches in terms of accuracy, error rate, precision, recall, and f-measure.

Our future work is the utilization of deep learning model instead of fuzzy C4.5 decision tree algorithm for classifying the tweets, and searching for more feature extractors and feature selectors methods to compare their effectiveness and the CNN.

References

1. M. U. Salur and I. Aydin, *A Novel Hybrid Deep Learning Model for Sentiment Classification*, IEEE Access **8**, 58080-58093 (2020)
2. O. Almatrafi, S. Parack, and B. Chavan, *Application of location-based sentiment analysis using Twitter for identifying trends towards Indian general elections 2014*, in Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication, 8-10 January 2015, Bali, Indonesia (2015)
3. F. Es-Sabery, A. Hair, J. Qadir, B. Sainz-De-Abajo, B. Garcıa-Zapirain, and I. D. L. Torre-Dıez, *Sentence Level Classification Using Parallel Fuzzy Deep Learning Classifier*, IEEE Access 9, 17943-17985 (2021).
4. P. Bedi and P. Khurana, *Sentiment Analysis Using Fuzzy-Deep Learning*, in Proceedings of the 2019 International Conference on Emerging Trends in Information Technology, 21-22 June 2019, Delhi, India (2019).
5. F. Es-sabery and A. Hair, *A MapReduce C4.5 Decision Tree Algorithm Based on Fuzzy Rule-Based System*, Fuzzy Inf. Eng. **11**, 446-473 (2019).
6. S. G. Kanakaraddi, A. K. Chikaraddi, K. C. Gull, and P. S. Hiremath, *Comparison Study of Sentiment Analysis of Tweets using Various Machine Learning Algorithms*, in Proceedings of the International Conference on Inventive Computation Technologies (ICICT), 26-28 Feb. 2020, Coimbatore, India (2020).
7. A. Rahman and M. S. Hossen, *Sentiment Analysis on Movie Review Data Using Machine Learning Approach*, in Proceedings of the International Conference on Bangla Speech and Language Processing (ICBSLP), 27-28 Sept. 2019, Sylhet, Bangladesh (2019).
8. A. Krishna, V. Akhilesh, A. Aich, and C. Hegde, *Sentiment Analysis of Restaurant Reviews Using Machine Learning Techniques*, in Emerging Research in Electronics, Computer Science and Technology, 687-696 (2019).
9. S. Liao, J. Wang, R. Yu, K. Sato, and Z. Cheng, *CNN for situations understanding based on sentiment analysis of twitter data*, Procedia Comput. Sci. **111**, 376-381(2017).
10. Y. Feng and Y. Cheng, *Short Text Sentiment Analysis Based on Multi-Channel CNN With Multi-Head Attention Mechanism*, IEEE Access **9**, 19854-19863 (2021).
11. R. K. Behera, M. Jena, S. K. Rath, and S. Misra, *Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data*, Inf. Process. Manage. **58**, 102-435 (2021).
12. T.-L. Nguyen, S. Kavuri, and M. Lee, *A multimodal convolutional neuro-fuzzy network for emotion understanding of movie clips*, Neural Netw. **118**, 208-219 (2019).
13. J. B. Sathe and M. P. Mali, *A hybrid Sentiment Classification method using Neural Network and Fuzzy Logic*, in Proceedings of the 11th International Conference on Intelligent Systems and Control (ISCO), 5-6 Jan. 2017, Coimbatore, India (2017).
14. F. Es-Sabery, K. Es-Sabery, A. Hair, J. Qadir, B. Sainz-De-Abajo, B. Garcıa-Zapirain, and I. D. L. Torre-Dıez, *A MapReduce Opinion Mining for COVID-19-Related Tweets Classification Using Enhanced ID3 Decision Tree Classifier*, IEEE Access **9**, 58706-58739 (2021).
15. K. W. Church, *Word2Vec*, Nat. Lang. Eng. **23**, 155-162 (2017).
16. Y. Seddiq, Y. A. Alotaibi, S.-A. Selouani, and A. H. Meftah, *Distinctive Phonetic Features Modeling and Extraction Using Deep Neural Networks*, IEEE Access **7**, 81382-81396 (2019).
17. F. Es-sabery and A. Hair, *An Improved ID3 Classification Algorithm Based On Correlation Function and Weighted Attribute**, in Proceedings of the 2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS), 26-27 Dec. 2019, Taza, Morocco (2019).
18. M. Jogin, Mohana, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, *Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning*, in Proceedings of the 3rd IEEE International Conference on Recent Trends in Electronics, Information Communication Techn-ology (RTEICT), 18-19 May 2018, Bangalore, India (2018).
19. F. Es-Sabery and A. Hair, *Big Data Solutions Proposed for Cluster Computing Systems Challenges: A survey*, in Proceedings of the 3rd International Conference on Networking, Information Systems & Security, 31 March - 2 April 2020, Marrakech, Morocco (2020).
20. F. Es-sabery, K. Es-sabery, and A. Hair, *A MapReduce Improved ID3 Decision Tree for Classifying Twitter Data*, in Business Intelligence, Cham, 160-182 (2021).
21. F. Xu, Z. Pan, and R. Xia, *E-commerce product review sentiment classification based on a naive Bayes continuous learning framework*, Inf. Process. Manage. **57**, 102221 (2020).
22. S. U. Maheswari and S. S. Dhenakaran, *Aspect based Fuzzy Logic Sentiment Analysis on Social Media Big Data*, in Proceedings of the International Conference on Communication and Signal Processing (ICCSP), 28-30 July 2020, Chennai, India (2020).