IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 24, 2021, accepted April 11, 2021, date of publication April 14, 2021, date of current version April 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3073215

A MapReduce Opinion Mining for COVID-19-Related Tweets Classification Using Enhanced ID3 Decision Tree Classifier

FATIMA ES-SABERY[1], KHADIJA ES-SABERY[2], JUNAID QADIR[3], BEATRIZ SAINZ-DE-ABAJO[4], ABDELLATIF HAIR[1], BEGOÑA GARCÍA-ZAPIRAIN[5], (Member, IEEE), AND ISABEL DE LA TORRE-DÍEZ[4]

[1]Department of Computer Science, Faculty of Sciences and Technology, Sultan Moulay Slimane University, Beni Mellal 23000, Morocco
[2]Department of Computer Science, National School of Applied Sciences, Cadi Ayyad University, Marrakech 40000, Morocco
[3]Department of Electronics, Quaid-i-Azam University, Islamabad 45320, Pakistan
[4]Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid, 47011 Valladolid, Spain
[5]eVIDA Research Group, University of Deusto, 48007 Bilbao, Spain

Corresponding authors: Beatriz Sainz-de-Abajo (beasai@tel.uva.es) and Fatima Es-Sabery (fatima.essabery@gmail.com)

This work was supported by the eVida Research Group, University of Deusto, Bilbao, Spain, under Grant IT 905-16.

ABSTRACT *Opinion Mining* (OM) is a field of *Natural Language Processing* (NLP) that aims to capture human sentiment in the given text. With the ever-spreading of online purchasing websites, micro-blogging sites, and social media platforms, OM in online social media platforms has picked the interest of thousands of scientific researchers. Because the reviews, tweets and blogs acquired from these social media networks, act as a significant source for enhancing the decision making process. The obtained textual data (reviews, tweets, or blogs) are classified into three different class labels which are negative, neutral and positive for analyzing and extracting relevant information from the given dataset. In this contribution, we introduce an innovative MapReduce improved weighted ID3 decision tree classification approach for OM, which consists mainly of three aspects: Firstly We have used several feature extractors to efficiently detect and capture the relevant data from the given tweets, including N-grams or character-level, Bag-Of-Words, word embedding (GloVe, Word2Vec), FastText, and TF-IDF. Secondly, we have applied a multiple feature selector to reduce the high feature's dimensionality, including Chi-square, Gain Ratio, Information Gain, and Gini Index. Finally, we have employed the obtained features to carry out the classification task using an improved ID3 decision tree classifier, which aims to calculate the weighted information gain instead of information gain used in traditional ID3. In other words, to measure the weighted information gain for the current conditioned feature, we follow two steps: First, we compute the weighted correlation function of the current conditioned feature. Second, we multiply the obtained weighted correlation function by the information gain of this current conditioned feature. This work is implemented in a distributed environment using the Hadoop framework, with its programming framework MapReduce and its distributed file system HDFS. Its primary goal is to enhance the performance of a well-known ID3 classifier in terms of accuracy, execution time, and ability to handle the massive datasets. We have carried out several experiences that aims to assess the effectiveness of our suggested classifier compared to some other contributions chosen from the literature. The experimental results demonstrated that our ID3 classifier works better on COVID-19_Sentiments dataset than other classifiers in terms of Recall (85.72 %), specificity (86.51 %), error rate (11.18 %), false-positive rate (13.49 %), execution time (15.95s), kappa statistic (87.69 %), F1-score (85.54 %), classification rate (88.82 %), false-negative rate (14.28 %), precision rate (86.67 %), convergence (it convergent towards the iteration 90), stability (it is more stable with mean deviation standard equal to 0.12 %), and complexity (it requires much lower time and space computational complexity).

INDEX TERMS ID3 decision tree, opinion mining, Hadoop, HDFS, MapReduce, feature extractors, feature selectors, DataMining, big data, information gain.

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

I. INTRODUCTION

Opinion mining (also recognized as sentiment analysis) is an intense research issue in the domain of NLP. It pursues at

classifying, studying, selecting, and evaluating the opinions, attitudes, emotions, and reactions from user-posted texts in social media platforms towards entities, such as organization, service, individual, product, topic, event, and issue [1]. With the spread of user-generated text in social networks and microblogging websites such as YouTube, Trip Advisor, Tiktok, Instagram, Twitter, Facebook, Amazon, and Whatsapp; sentiment analysis in web sites and social networks has acquired rising popularity amongst many scientific research and industry communities [2].

OM is noted to be performed in three diverse aspects, which are feature aspect, document level, and sentence level [3]. At the feature level, we classify the sentences/documents as negative, positive, or neutral based on the extracted opinion words of those sentences/documents. This task is generally renowned as feature-level sentiment analysis [4], [5]. Document-level is a sentiment classification task that endeavors to classify the entire document as negative, neutral, or positive polarity [5]. Sentence-level is the sentiment mining operation that pursues to calculate the sentimental polarity of the given sentence. That is to say; the sentence will be classified as negative, positive, or neutral according to the expressed opinion in this sentence [6]. In this proposal, sentence-level sentiment classification has been taken into consideration.

OM is employed in several other applications like predicting the future customer trends or behaviors [7], proposing the new future marketing strategies [8], improving the e-learning techniques [9], and stock market prognosis [10]. One of OM's most substantial applications is that it provides governmental parties or political parties with many pieces of information, which are analyzed to guess the opportunities of their winning in forthcoming political elections. Also, it determines if the public people are contented with their policies as performed during US elections 2020.

Twitter is the most communal social network in real-time to express an individual or group of individuals' opinions and ideas about a specific topic through short messages of 280 characters called tweets. Due to the limitation in terms of length of a tweet, the users tend to use Slang, Informal language, many abbreviations, URLs, short forms, and heavy use of emoticons along with Twitter-specified expressions like hashtags and user mentions [2], and that pose considerable defies for Twitter sentiment classification. Therefore, it is mandatory to apply intelligent mechanisms to capture useful knowledge from tweets. In this proposal, we use a parallel supervised technique to handle the tweets in such a manner as to conquer the above defies.

The number of Twitter users reached 330 million monthly active users, 145 million daily active Twitter users in 2020, and half a billion tweets are sent out each day, that equates to 5,787 tweets per second [11]. Due to this massive generated data per day on the Twitter platform, the scientific researchers considered Twitter as an instance of Big Data [12]. Because it has the same Big Data characteristics such as (i) *Volume:* the average number of tweets to announce a news incident is at least 1000000; (ii) *Variety:* each tweet has consisted of diverse materials (such as short words, slang, abbreviations, URLs and emoticons); (iii) *Velocity:* the tweets about a topic are extremely dynamic. For example, the new coming tweets every day about an event are more than 500TB on Twitter. (iv) *Value:* the rich data hidden in the generated tweets are a hot research area for scientific researchers in the Big Data and sentiment analysis field, and also a robust tool for organizations and governments for making- decisions or universal strategies. In our work, Hadoop Big Data framework is applied to parallelize the improved contribution (ID3 decision tree + Word embedding) in order to deal with massive Twitter data.

The Twitter platform makes smooth the interactions between customers and organizations or institutions. The freedom to utilize the Twitter social network offers chances for their users to write feedback that express their opinion about certain situations, products, events, and services [13]. These feedbacks are expressed predominantly based on the user's experience with these products or services, which may be negative, positive, or neutral judgments on products or services. Extracting user opinions that negatively impact product or service from the expressed feedback on the Twitter platform is an essential task that helps the owner organizations improve their products or services, thereby reaping more profit [14]. Therefore, it is substantial to assess user feedback gathered from social network platforms. OM is an efficient tool for determining the polarity (negative, neutral, or positive) of users' judgments about a service or product via analyzing microblog data. In OM, the sub-operations in Fig. 1 are performed by evaluating user feedback posted on the different social network platforms. These sub-operations aim to extract useful information from social media data that indicates the polarity of (positive, negative, neutral) of analyzed social networks text. Machine learning techniques can be utilized to reveal valuable knowledge which is hidden in such noisy social media data created on daily basis [15]. There are numerous techniques of machine learning that support learning, such as *K-Nearest Neighbors* (KNN) [16], *Support Vector Machines* (SVM) [17], ID3 decision tree [18], *Logistic Regression* (LR) [19], *Naive Bayesian* (NB) [20], or *Random*

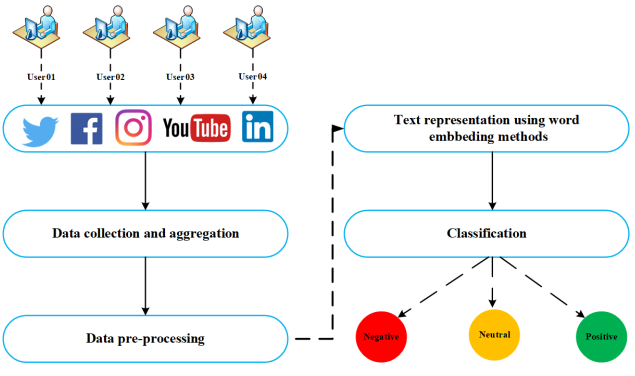


FIGURE 1. Basic steps of opinion mining on social network platforms.

Forest (RF) [21]. These algorithms have largely been used in various areas like banking [22], cyberhate detection [23], bioinformatics [24], abusive language detection [25], social media [26], and cyberbullying identification [27].

The machine learning techniques are divide into two types, which are supervised and unsupervised learning algorithms [5]. Supervised learning employs a labeled dataset to construct a classification model, which is thereafter utilized to forecast the class labels for unlabeled testing dataset labels. On the other hand, unsupervised learning algorithms like clustering aims to form unlabeled data into different clusters by calculating the similarity. Supervised learning algorithms have widely been utilized for OM [28]–[32]. In this work, we will use our improved ID3 supervised learning algorithm, because it is executed only on labeled data. So, our work described in this paper carries out opinion mining for English language content applying an improved ID3 decision tree implemented by Hadoop technology.

The principal proposals of our study can be summed as follows. As we said previously, our work classifies the collected tweets into three class labels: negative, positive, or neutral. The tweets are generally in the unstructured text format, consisting of slang, abbreviations, short words, URLs, stop words, etc. So, the first step of our work is applying preprocessing techniques on the tweets to remove unwanted and noisy information for further analysis. After the preprocessing step, these tweets will be passed via an operation of representation in which tweets are turned into numerical values, which taking the matrix form as the second step of our work. Then we minimize the dimensionality of the obtained matrix in the second step employing multiple feature selectors. The reduced matrix in the third step will be the given input of our proposed improved ID3 [33] for classification of the tweets. Finally, we parallelize all precedent steps using the Hadoop ecosystem diagram, with its MapReduce programming model and its HDFS distributed file systems. Various parameters are then utilized to assess the performance of our proposal. Therefore, our contribution has mainly consisted of seven aspects

1) Data preprocessing techniques like lemmatization operation, stemming process, and effect of negation technique are applied to improve the tweets quality by removing the noise and unwanted data.
2) Representation approaches like TF-IDF, FastText, word embedding (Word2Vec, GloVe), Bag-Of-Words, N-grams are applied on the used dataset of the tweets to convert text-based data (tweets) into text-based numerical.
3) Feature selectors like Gini Index, Information Gain, Chi-Square, and Gain Ratio are employed to reduce the high feature dimensionality.
4) Our Improved ID3 decision tree algorithm with numerous parameters is used for classifying the analyzed tweet into a neutral, negative or positive class label.
5) Our introduced model is implemented on parallel manner using Hadoop Big Data framework to prevent the long execution time problem and improve our proposal's ability to deal with the massive dataset.
6) Comparing the performance of the suggested procedure with other chosen methods from the literature.
7) Our proposed model outperforms baseline approaches by a significant margin in terms of Recall, specificity, error rate, false-positive rate, execution time, kappa statistic, F1-score, false-negative rate, accuracy and precision rate.

The remainder of our work is constructed as follows: previous literature researches are described in the ''Previous Research'' section; the ''Methodology of our proposed approach'' section introduces our developed approach in detail; in ''Experiment and results,'' we introduce the empirical setup and obtained results. And the ''Conclusions'' section summarizes the proposed method and recommends future work.

II. PREVIOUS RESEARCH

A thorough discussion has been performed on OM for different language utilizing machine learning algorithms, some of which are presented below as follows:

Here are some works that employ different machine learning technique for carried out the opinion mining in diverse languages such as in (Sharma *et al.* [32]; Patil *et al.* [34]; Shein *et al.* [35]; Gamallo *et al.* [36]; Anjaria *et al.* [37]; Duwairi *et al.* [38]; Soni *et al.* [39]; Ngoc *et al.* [40]), and as described in the Table 1 and 2. The authors in Patil *et al.* [34] developed a new hybrid technique that integrates the TF-IDF vectorization method with the SVM algorithm to find out the polarity of analyzed textual data. This work proved that SVM could extract high dimensional feature space from textual data, which cancels the need for other feature selection techniques. This suggested contribution in Shein *et al.* [35] integrates formal concept analysis ontology, and SVM for labelling the collected software reviews to negative, neutral, or positive.

There are many works concerning the ID3 decision tree algorithm and their improvements like Es-Sabery *et al.* [33]; Yu-Xun *et al.* [41]; Chai *et al.* [42]; Elyassami *et al.* [43]; Zou *et al.* [44]; Srinivasan *et al.* [45]; Chen *et al.* [46]; Ding *et al.* [47]; Zhu *et al.* [48]; Kaewrod *et al.* [49]; Rajeshkanna *et al.* [50]; and as introduced in the Table 3 and 4. In [33], we proposed a novel enhanced ID3 decision tree algorithm, which integrates the weighted theory and information gain criterion. In the split learning process, the improved algorithm considers the association between the immediate conditional feature, decision feature, and the other conditional features to compute the weighted information gain, then find the best split feature. Unlike in the traditional ID3, we consider only the association between the current feature and decision feature to calculate the information gain. This improved proposed ID3 decision Tree algorithm is used in this work as the classifier. Elyassami *et al.* [43], the authors applied the Fuzzy ID3 algorithm for software price prediction; it is implemented by combining the basic concepts of

TABLE 1. Machine learning technique that carried out the opinion mining in diverse languages.

Authors	OM	Language	Algorithm	Vectorization	Approach	Advantage(s)	Disadvantage(s)	Year	Ref
Sharma *et al.*	Yes	English	KNN+ , ME+, C4.5+, NB+, A+, W+, SVM	DF+, IG+, CHI+, GR+, RF+	This work discusses the utilisation of five generally applied feature selectors approaches in the data mining area (IG, DF, GR, CHI, R-F, and GR) and seven machine learning algorithms (NB, SVM, ME, C4.5, KN, W, A) for sentiment classification on online movie reviews dataset.	-This paper presents a comparative study between DF, GR, CHI, NB+ GR+ IG, and R-F feature selectors, and it deduces that GR performs better than other feature selectors. - Also, this paper carries out a comparative study between NB, SVM, ME, C4.5, KN, W, A classifier, and it deduces that SVM gives good accuracy for the sentiment analysis, while the NB approach demonstrates a better classification rate when applied with fewer feature vector spaces.	Sparse feature vector spaces. - It achieved lower performance when it applied on huge amount of data.	2012	[32]
Patil *et al.*	Yes	English	SVM	TF-IDF	The proposed approach combine the preprocessing techniques, SVM and TF-IDF methods for classifying English textual data are negative positive, or neutral.	-This developed model gives good performance than neural network models when they applied them on lower size of data. -It has the ability to propagate high dimensional feature space. -It extracts little number of the irrelevant feature -SVM cancels the necessity of using feature selector.	-Sparse feature vector spaces. -It is inefficient on high size of data.	2007	[34]
Shein *et al.*	Yes	English	SVM	FCA	This suggested contribution integrates NLP techniques, formal concept analysis ontology, and SVM for labelling the collected software reviews to negative, neutral, or positive	- It extracts more relevant concepts and features. - It achieves good performance on lower size of data. - Ontology affords information about a particular field that is understandable by both scientific researchers and computers	Sparse feature vector. - It has a lower classification rate compared to neural networks model when it was applied to Big Data	2010	[35]

TABLE 2. Machine learning technique that carried out the opinion mining in diverse languages (Continue).

Authors	OM	Language	Algorithm	Vectorization	Approach	Advantage(s)	Disadvantage(s)	Year	Ref
Gamallo *et al.*	Yes	Spanish	NB	U+B	-The proposed system in this work combines the Naive-Bayes classifier with unigrams+bigrams to detect Spanish tweets'polarity. Experimental results display that this work achieved an accuracy equal to 67 %.	-This proposed method achieved the best performance by applying a binary level classifier which is trained to capture just two polarity scores: positive and negative.	-The major shortcoming of this work is the application of thresholds for capturing polarity scores: high polarity scores (strong positive and negative), as well as low polarity score (neutral).	2013	[36]
Anjaria *et al.*	Yes	English+, Hindi.	ME+, SVM+, NB+, ANN.	U, B, U+B.	This work carried out the sentiment classification over of US Presidential Elections 2012 and Karnataka State Assembly Elections (India) 2013 Twitter data using machine-learning algorithms,such as SVM, ME, NB, ANN and unigram,unigram+bigram bigram, feature extractor.	-This suggested approach developed a new feature extractor to detect the terms influencing the event. -It also evolved a method to find the impact factor produced with re-post over Twitter. -It achieved the classification rate equal to 88 % in the case of US Presidential Elections 2012.	The hybrid proposed classifiers require to be parallelized to afford real-time high performance computing outcomes to portend any event.	2014	[37]
Duwairi *et al.*	Yes	Arabic	NB+, SVM	TF-IDF+, BM	This contribution has performed the sentiment analysis on Arabic Twitter dataset in the existence of dialectical words.	-NB and SVM classifiers performed an excellent job in eliminating false instances, and thence they achieved high precision.	They are less well at predicting true instances, therefore they achieved low recall without applying the dialect lexicon.	2015	[38]
Soni *et al.*	Yes	English	ID3	NLP parser	In this proposal, an improved ID3 technique is proposed to avoid the shortcomings of classical ID3 Then this novel ID3 is used to perform the sentiment analysis on English dataset.	–The designed algorithm avoids the performance to be dependent on the volume of the classified training dataset -It attempts to elect a more optimal feature for the decision tree.	-It is inefficient on high size of data, it requires to be parallelized using a Big Data environment. -It is not useful for multi-domain and multi-format data.	2017	[39]

TABLE 3. Works concerning the ID3 decision tree algorithm and their improvements.

Authors	Algorithm	Approach	Advantage(s)	Disadvantage(s)	Year	Ref
Es-sabery*et al.*	ID3	An Enhanced ID3 Classification Algorithm Based On Weighted Correlation Function	-This new version of the ID3 decision takes into consideration all conditions attributes and decision attributes in the operation of measuring the information gain, Unlike the traditional ID3.	-No mention.	2019	[33]
Yu-xun *et al.*	ID3	A new effective classification architecture for ID3	-This amended ID3 algorithm is employing the weighted attribute importance in the process of calculating the information gain of every feature and choose the feature with te highest information gain value as the splitting feature.	-Resolve the classification deficiency of inclining to choose features with more values.	2010	[41]
Chai *et al.*	ID3	A new effective classification architecture for ID3	-This novel ID3 proposed new schemes to resolve the shortcoming of inclining to select features which have multiple values.	-The new proposed architecture has shorter consuming time and higher accuracy rate than ID3 algorithm.	2010	[42]
Elyassami *et al.*	ID3	Applying Fuzzy ID3 Decision Tree for Software Effort Estimation	-In this work, the authors have experimentally implemented and analyzed two Fuzzy ID3 algorithms for evaluating the software effort estimation. This contribution combines the ID3 algorithm decision tree with the fuzzy set theory in order to deal with ambiguous and uncertain data.	-No mention.	2011	[43]
Zou *et al.*	ID3	ID3 Decision Tree in Fraud Detection Application	-In this paper, the authors applied the ID3 algorithm decision to construct the fraud detection model. The utilization of ID3 proved that this algorithm could perform the data classification very well, and it supplies decision-makers with a set of decision rules.	-No mention.	2012	[44]
Srinivasan *et al.*	ID3+, SVM	Fuzzy fast classification algorithm with hybrid of ID3 and SVM	- In this work, the authors designed the Fast Fuzzy classification method for getting better performances of classification. They also have incorporated the advantages of the ID3 decision tree and the SVM algorithm, for improving the accuracy and for getting a fast classification result.	-ID3 is not suited for the massive dataset, and SVM has less classification rate of choosing random attribute.	2013	[45]
Chen *et al.*	ID3	An enhanced ID3 Decision Tree technique	-Novel feature selector is proposed using the attribute importance, the technique implemented in [41], the association function used in [47] ,and adding the number of values of each attribute.	- This improved algorithm is insufficient in detecting the relevance between all elements and their attributes.	2014	[46]

TABLE 4. Works concerning the ID3 decision tree algorithm and their improvements (Continue).

Authors	Algorithm	Approach	Advantage(s)	Disadvantage(s)	Year	Ref
Ding *et al.*	ID3+ Rough theory	A New Decision Tree Algorithm Based on Rough Set Theory	-In this research paper, the authors designed a novel enhanced ID3,which used as a splitting measure in the rough set theory instead of the information gain measure in traditional ID3.	-This improved ID3 based on rough set theory becomes inefficient in the case of data set with missing data.	2015	[47]
Zhu*et al.*	ID3	Refinement of Decision Tree ID3 technique	-It proposed a new improved ID3 which is based on information gain average and with different parameters, to a specific range for avoiding the multi-valued attribute bias problem.	-No mention.	2017	[48]
Kaewrod*et al.*	ID3	Enhancing ID3 technique by Ignoring Minor examples	-The principal goal of this work is to evolve a new method to relieve the rigorousness of the traditional ID3 algorithm by removing minor examples.	-No mention.	2018	[49]
Rajeshkanna *et al.*	ID3	ID3 Decision Tree algorithm:An Algorithmic Perspective based on Error rate	-It implement the ID3 technique with various UCI datasets and is also evaluated utilizing different statistical measures.	-No mention.	2020	[50]

the ID3 algorithm and the fuzzy set theory principles providing the model with the ability to deal with uncertain and ambiguous data, which can enhance the classification rate greatly. Zou *et al.* [44] applied the ID3 algorithm decision to construct the fraud detection model. The utilization of ID3 proved that this algorithm could perform the data classification very well, and it supplies decision-makers with a set of decision rules. Srinivasan *et al.* [45] designed the Fast Fuzzy classification method for getting better performances of classification. They also have incorporated the advantages of the ID3 decision tree and the SVM algorithm, for improving the accuracy and for getting a fast classification result. In this work [46], an improved ID3 (called AFI-ID3) This new version of ID3 has based on an original is proposed. feature selection method instead of information gain used in the traditional ID3. This novel feature selection method is computing using the attribute importance, the association function, and adding the number of values of each attribute. Ding *et al.* [47] designed a novel enhanced ID3,which used as a splitting measure in the rough set theory instead of the information gain measure in traditional ID3. This improved ID3 based on rough set theory becomes inefficient in the case of data set with missing data. Zhu *et al.* [48] It proposed a new improved ID3 which is based on information gain average with different parameters, to a specific range for avoiding the multi-valued attribute bias problem. The principal target of the work [49] is to evolve a new procedure to relieve the rigorousness of the traditional ID3 algorithm by removing minor examples. Rajeshkanna *et al.* [50] implemented the ID3 technique using various UCI training datasets and it is also evaluated utilizing different statistical measures.

The newest innovative research papers for the OM are Lakshmi *et al.* [51]; Guerreiro *et al.* [52]; Mehta *et al.* [53]; Zhang [54]; Lopez-Chau *et al.* [55]; AitAddi *et al.* [56]; Patel *et al.* [57]; Wang *et al.* [58] as presented in the Table 5. In [51], the authors proposed a new contribution that intends to classify reviews using two classifiers and determine which of the both perform better performance. These both classifiers are DT, and NB. Guerreiro *et al.* [52] designed a new text mining approach, which is applied to online collected reviews in order to extract the drivers behind each explicit recommendations. In [53], the author implemented nine separate algorithms which are: Long short-term memory (LSTM), NB, NLP, Multilayer perceptron (MLP), Max entropy, Convolutional neural network (CNN), RF, XGBoost, DT, SVM to classify tweets and compare their accuracy on preprocessing data. The experimental result proved that the convolutional neural network outperforms other applied approaches by achieving 79 % accuracy. Zhang [54] evolved a new approach that used the Term Frequency Inverse Document Frequency (TF-IDF) as the feature extractor and it employed Chi Square and Mutual Information as feature selectors. Then the extracted features are fed into Logistic Regression, Linear SVC, and Multinomial Naive Bayes classifiers for performing the sentiment classification. The authors of the research paper Lopez-Chau *et al.* [55] analyzed the data sets collected from Twitter about the earthquake topic on September 19, 2017, applying OM tools and supervised machine learning. They built three classifiers to find out the sentiment of tweets that appear on the same topic. The experimental result proved that the SVM, and NB achieved the best classification rate of classifying the emotions. The author of [56] suggested an innovative method based on a three-level binary tree structure for multi-class hierarchical opinion mining in Arabic text. empirical outcomes show that their proposed method obtains considerable improvements over other literature approaches. In Patel *et al.* [57], The authors performed a sentiment analysis on Twitter data for World Cup soccer 2014 held in Brazil to extract the emotions of the people everywhere in the world, utilizing machine learning algorithms. After the application of NLP operations like part-of-speech, word tokenization, lemmatization, etc., the SVM, NB, KNN have been applied as the classifiers to extract the emotions from the tweet. The empirical result proved that the NB achieves best accuracy equal to 88.17 %. In the paper [58] a novel model for aspect-level opinion mining is proposed. This developed model uses the paradigm of Gradual Machine Learning for performing automatic precise machine labeling. The experimental results have proved that the accuracy of the suggested technique is considerably better than its unsupervised algorithms. For performing the sentiment classification task based on deep neural network, there are several works in the literature as described in the Table 6, and for example: Baccouche *et al.* [59] developed a novel technique for carrying out an automatic classification health-related Twitter annotation for three languages: Arabic, French, and English. In particular, the authors of this work implement a Recurrent Neural Network (RNN), CNN, and LSTM for performing sentiment analysis. The empirical result shows that LSTM-RNN performs better than another literature review for both English and Arabic datasets in terms of classification rate, F1-Score, recall, and precision. In [60] and [61], the authors proposed a novel computationally functional method for classifying the positive and negative emotions by utilizing publicly available movie review datasets, namely, Stanford Sentiment Tree and Movie Review datasets. They combine a global maximum pooling layer (GMPL) with one bidirectional LSTM layer and obtained an F1 score equal to 85.78 % and 80.21 % for Stanford Sentiment Tree and Movie Review datasets, respectively. Zhang proposed a new sentiment classification approach based on CNN and SVM learning algorithm in the paper [62]. Which is consists of a two-aspects: Firstly, the authors utilize the artificial plus sensitive dictionary approach to label the original e-commerce review dataset. And then, they classified the labeled dataset using the CNN classifier and the SVM model. In the paper [63], a new multi-task learning method is developed that combines multi-scale CNN and LSTM for performing multi-task and multi-scale opinion mining. Chugh *et al.* [64] suggested a new sentiment classification method that integrates the deep recurrent neural network (RNN), SentiWordNet, fuzzy K-Nearest neighbor (Fuzzy

TABLE 5. Newest innovative research papers for the OM.

Authors	OM	Language	Algorithm	Dataset	CR	Year	Ref
Lakshmi *et al.*	Yes	English	Naïve Bayes+Decision Tree	IMDB movie reviews that contains 25250 reviews	65 %	2020	[51]
Guerreiro *et al.*	Yes	English	logistic regression+ CHAID decision tree	Academic Yelp that contains 1112708 reviews	63.2 %	2020	[52]
Mehta *et al.*	Yes	English	NB+SVM	Electronic products that contains 1200 tweets	90 %	2020	[53]
Zhang	Yes	Chinese	NB+ SVM+ logistic regression	Movie Short V2 that contains 480000 reviews	68 %	2020	[54]
López-Chau *et al.*	Yes	Mexican	NB+SVM	Earthquake of September 19, 2017	59 %	2020	[55]
AitAddi *et al.*	Yes	Arabic	CART	IMDB movie reviews that contains 35070 reviews	74 %	2020	[56]
Patel *et al.*	Yes	English	NB+ SVM+ KNN+ random forest	World Cup2014 that contains 1415958 tweets	88.17 %	2020	[57]
Wang *et al.*	Yes	English	paradigm of Gradual Machine Learning	LAP16, RES16, LAP15 and RES dataset	67.01 %	2020	[58]

TABLE 6. Deep learning classifiers for performing the OM.

Authors	OM	Language	Algorithm	Dataset	CR	Year	Ref
Baccouche *et al.*	Yes	English+ Arabic	RNN+CNN+LSTM	Health-Related Twitter dataset	83 %	2018	[59]
Hameed *et al.*	Yes	English	BiLSTM+GMPL	MR, SST2 and IMDb datasets	80.50 %	2020	[60], [61]
Zhang	Yes	English	CNN+SVM	E-commerce dataset	87.24 %	2020	[62]
Jin *et al.*	Yes	English	CNN+LSTM	Electronic machines dataset which contains 12 000 reviews	85.50 %	2020	[63]
Chugh *et al.*	Yes	English	RNN+SentiWordNet, Fuzzy KNN	Amazon unlocked mobile reviews dataset which 400 000 reviews	97.70 %	2021	[64]
Jelodar *et al.*	Yes	English	LSTM+RNN	COVID-19-related dataset	81.15 %	2020	[65]
Dong *et al.*	Yes	English+ Chinese	VCPCNN	Stanford emotion dataset and UCI sentiment labelled sentences dataset	49.59 %	2020	[66]
Kumar *et al.*	Yes	English	IGCN	SemEval 2014 datasets	81.34 %	2020	[67]
Es-Sabery *et al.*	Yes	English	FFNN+CNN +MFS	COVID-19_Sentiments and the Sentiment140 datasets	99.97 % and 99.83 %	2021	[68]
Sadr *et al.*	Yes	English	Multi-view learning+ CNN+RNN	SST1 and SST2 datasets	52.94 % and 91.93 %	2020	[69]
El-Affendi *et al.*	Yes	Arabic	MPAN	IMDB movie review dataset	96.13 %	2021	[70]

KNN) for performing sentiment classification, and the experimental result proved that this model achieved an accuracy equal to 97.70 %. The authors in [65] introduced a new technique, which is applied to examine COVID-19–related remarks from sub-reddits. This paper's main contribution is to incorporate both deep learning algorithms, which are LSTM and RNN, for carrying out the COVID-19–related sentiment classification. Dong *et al.* [66] suggested a new deep learning model called variable convolution and pooling convolution neural network (VCPCNN), which combines CNN with many convolution layers and several pooling operations for performing sentiment analysis. In [67], the authors proposed a new hybrid approach called IGCN that integrates a bidirectional gating mechanism and CNN for prophesying the opinion of a target. Es-Sabery *et al.* [68] proposed a new fuzzy deep learning classifier for performing opinion mining. This hybrid approach combines the feedforward and convolutional neural network (FFNN + CNN) and Mamdani fuzzy system (MFS). The empirical result demonstrates that this approach achieves a better accuracy, which equals 99.97 % and 99.83 % on the COVID-19_Sentiments and the Sentiment 140 datasets. In [69], Sadr *et al.* developed a new multi-view deep learning classifier based on convolutional and recursive neural networks for classifying the text sentiment. The experimental results proved that the suggested multi-view deep learning classifier surpasses the single-view deep learning classifier in terms of accuracy. The authors of the paper [70] introduce a new deep learning-based multilevel parallel attention neural (MPAN) classifier that employs a simplistic positioning binary embedding system to concurrently measure contextualized embeddings at the character, word, and sentence levels. The experimental result proved that the proposed approach achieved an accuracy equals to 96.13 %.

III. MATERIALS AND METHODS

In the subsequent sections of this section, we will present the reasons that we are motivated to propose and to evolve this approach. Generally, the fundamental structure of this proposed paradigm is constituted of five-phase; the first stage is the data collection in which we used two massive datasets in order to evaluate our suggested classifier. Second stage called data pre-processing which aims to remove the unwanted and noisy data. Third phase is data representation phase which converts the tweets into numerical data. Fourth step is the feature selection stage for reducing the dimensionality of extracted features. Finally, we have applied our improved ID3 [33] in order to classify each tweet sentence into the negative, or neutral, or positive label.

A. MOTIVATION

Opinion mining is an important field of research that endeavor to design computational technique to detect, capture and evaluate people's ideas and opinions about an entity and its diverse sides and to extract the emotions expressed in those ideas and opinions. These expressed sentiments about a product, event, or service have a significant commercial worth. For example, the user-written reviews about products help new users in decision-making, such as buying this new product or no. And are extremely valuable for big companies/ organizations in the supervision of their products/events/ services, consolidating best relationships with their clients, designing and evolving efficient marketing strategies, enhancing and devising their products/events/services. Motivated by the significant importance of sentiment analysis in different domains of our daily life. We evolved in this contribution an innovative technique that serves to carry out the sentiment analysis. This approach combines NLP techniques for performing the text pre-processing, vectorization techniques for converting tweets into numerical values, selection methods for capturing the essential features, our improved ID3 for performing the classification [33], and the Hadoop framework to parallelize our work.

The first stage of this contribution is the text pre-processing task. Therefore, the pre-processing plays an essential role in the text classification process as introduced in the paper [71]. In where the authors provided us with a comparative study that aims to evaluate the influence of text pre-processing steps on text classification in terms of accuracy. The empirical result proved that the implementation of the text pre-processing techniques on linguistic data performed a considerable improvement in classification performance. Many literature studies [72]–[76] proved that the preprocessing steps have a positive impact on text classification task in terms of accuracy. Thence we motivated by the presented positive result about the text preprocessing steps, and we have applied these preprocessing techniques in this work.

After the preprocessing phase, the next phase is the representation stage that serves to turn out the tweets into numerical values. In this step of our work, we used the most common methods such as word embeddings (Word2Vec, GloVe) [77], N-grams or character-level [78], FastText [79], Bag-Of-Words [80], and we did a comparative study to find the better one amongst them. Then the next stage of our work is the feature selection phase, most of the time uses the filter approach like information gain [81], mutual information [82], chi-square [83], document frequency IF-TDF [84], and Gini index [85]. Also in this stage we did a comparative study to find the efficient feature selection method.

Finally, in the classification stage we used our improved ID3. So why we use our improved ID3 proposed in [33] as a classifier in this work? We used our proposed improved ID3 because of two reasons. Firstly it has been achieved good performance as presented in the paper [33], and secondly, rule-based opinion mining (ID3) often has many advantages, such as the structure of the rule-based algorithm is very simple, which makes it easy to understand by scientific researchers. The ID3 rule-based algorithm is pertinent in large-scale datasets. It often has a high predictive classification rate, and the rules are widespread in the DataMining field.

In the literature, many scientific researchers have endeavored to discover several manners to apply DataMining generated rules in opinion mining and find out several diverse relationships between NLP and DataMining field. The ID3 decision tree is the most common and essential algorithm in the DataMining field; thus, the produced rules using the ID3 decision tree technique are very accurate. This will result in various scientific researches, hence the motivation for this work.

As a summarized conclusion, this study's goal is to raise the classification performance of opinion mining by incorporate the strength points of text preprocessing techniques in improving the data quality by removing the unwanted and noisy data, feature extraction methods in converting the text data into numerical values and extracting the most relevant feature, feature selectors in reducing the high dimensionality of extracted feature in the preceding step and selecting the most interesting feature and our improved ID3 decision tree algorithm in classifying the input sentence and improving the classification accuracy. As depicted in Fig. 2, the universal architecture of the suggested approach in this paper be composed of five stages, which are the collection of data step, text pre-processing phase, data representation stage, feature selection methods, and our improved ID3 decision tree algorithm.

B. DATA COLLECTION STAGE

In this stage, we selected two massive datasets to assess the effectiveness of the suggested classifier for opinion mining. The first dataset is named **sentiment140**, which is downloaded from this link https://www.kaggle.com/kazanova/sentiment140 employing Twitter application programming interfaces (API). This dataset contains 1600000 collected tweets and all emoticons in this dataset were eliminated. Each tweet have been classified using two class labels positive, and negative, where the value 4 indicates the positive label and the

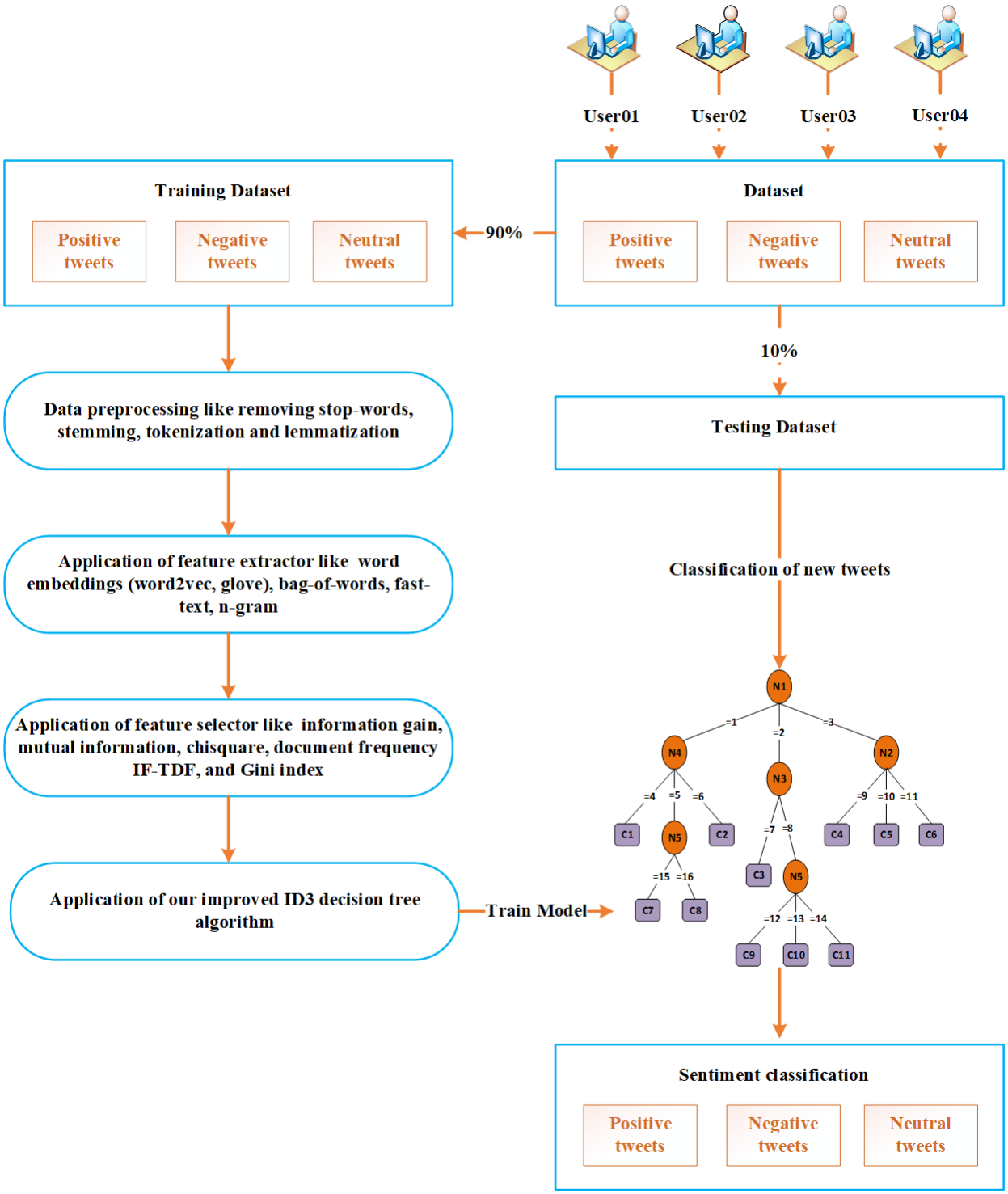


FIGURE 2. Universal architecture of the suggested approach.

value 0 indicates the negative label. It comprises six features which are presented below:

- **Target**: determine the class label of each tweet, where the value 4 indicates the positive label and the value 0 indicates the negative label.
- **Ids**: is a unique number (2356221408) that identify each tweet.
- **Date**: for example this feature takes this expression (Mon April 22 19:15:33 PDT 2005) as value which indicates the exact time when the tweet is posted.
- **Flag**: describes the text of the user-query. The 'NO_QUERY' value is assigned to this feature in the case the user does not posted any query.

- **User**: indicates the username that posted the tweet (username:Merissa).
- **Text**: introduces the text of each tweet, for example "thought sleeping in was an option tomorrow but realizing that it now is not. evaluations in the morn…" is the text tweeted by the user Merissa.

In our contribution, we are focused on opinion mining. That is means, obtain every opinion communicated by the Twitter-user in every posted tweet. In consequence, the other features in this dataset have not any impact on the training operation. Consequently, we eliminated the "Date", "Flag", "Ids", "User" features, and we saved the "Target" and "Text" features from the dataset. The "Target" feature apportionment of the dataset is equitable apportionment, because 50 % of the data being labelled negative class label, ranging from the row number 0 to 799999*th* row, and another 50 % of the data labelled positive class label, which are ranging from the row number 800000 to 1600000*th* row. In this work, the given dataset is splitted into two subsets which are training and testing subsets. Hence, we have utilized these two subsets to demonstrate our proposed method's classification effectiveness compared to other suggested techniques picked from the literature. Fig. 3 presents the number of user tweets in each training and testing subsets, where a total of 160000 posted tweets were utilized in the testing operation and 1440000 posted tweets were utilized in the training operation.

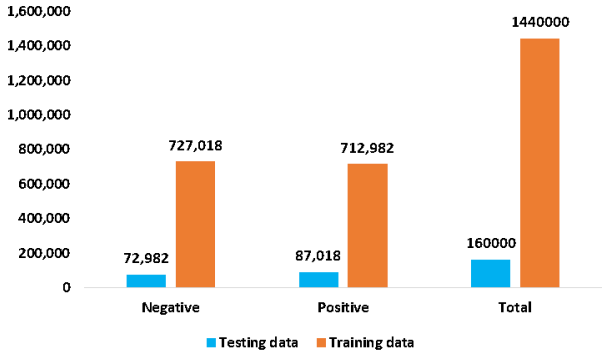


FIGURE 3. Number of positive and negative tweets for the sentiment140 dataset.

The second dataset is termed **COVID-19_Sentiments**, which is downloaded from this link https://www.kaggle.com/abhaydhiman/covid19-sentiments employing Twitter application programming interfaces (API). This dataset contains 637978 of collected tweets. It classified collected tweets into three class labels which are neutral that takes the value 0, negative that takes a value in the interval $[-1, 0]$, and positive that takes a value in the interval $[0, 1]$. It comprises five features which are introduced below:

- **Target**: determine the class label of each tweet, where neutral class label takes the value 0, negative class label takes a value in the interval $[-1, 0]$, and positive class label takes a value in the interval $[0, 1]$.

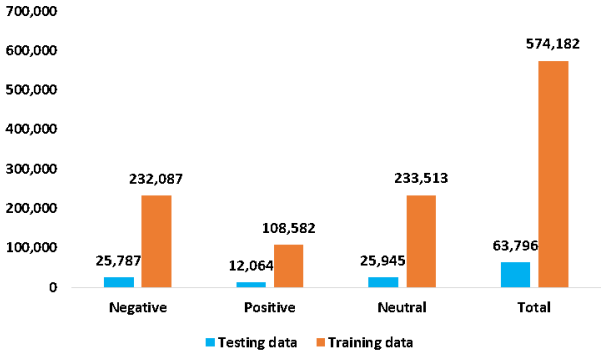


FIGURE 4. Number of neutral, negative, and positive tweets for the COVID-19_Sentiments dataset.

- **Ids**: is a unique number (520442) that identify each tweet.
- **Date**: indicates the exact data when the tweet is posted (Sat June 26 05:43:52 + 0101 2019).
- **Location**: indicates the exact location where the tweet is posted (New Delhi, India).
- **Text**: introduces the text of each tweet, for example "Delhi government will pay salaries to all contract workers, daily wage labourers, guest teachers a cause of Corona Virus".

The essential features in this contribution are the "Target" and "Text" features. Consequently, all other features were eliminated. Besides, the "Target" feature divided the data of this given dataset in an inequitable apportionment, where 259458 is the total number of neutral tweets, the number of negative tweets equal to 120646, and the number of positive tweets is equal to 257874. Fig. 4 shows the number of neural, negative and positive posted tweets inside of training and testing subset. In the training operation, we have used the number of tweets equal to 574182 tweets. In the testing operation, we have used the number of tweets equal to 63796 tweets. In other terms, the testing subset represents 10 % of the total number of tweets in this dataset.

C. DATA PREPROCESSING STAGE

To examine the semantic data, noise or abnormal data must be purified, so that the accurate emotions and meaning can be procured from the examined linguistic text. At this point, the text preprocessing techniques play a primary role. In this contribution, the type of analyzed data is the tweets extracted from the Twitter platform. These tweets often are in an unregulated form of text and contain inefficient, noise, and undesired knowledge. Data preprocessing techniques normalize the semantic data, purifier noise, eliminate unwanted information, and clarify the vocabulary employed to analyze emotions from the tweets. The Fig. 5 depicts the most common vision for data preprocessing techniques.

As Fig. 5 illustrated, There are numerous steps to be carried out to reappraise the data accurately and extract the real sense behind it via data processing. Thence these followed up preprocessing steps in this study are introduced below:

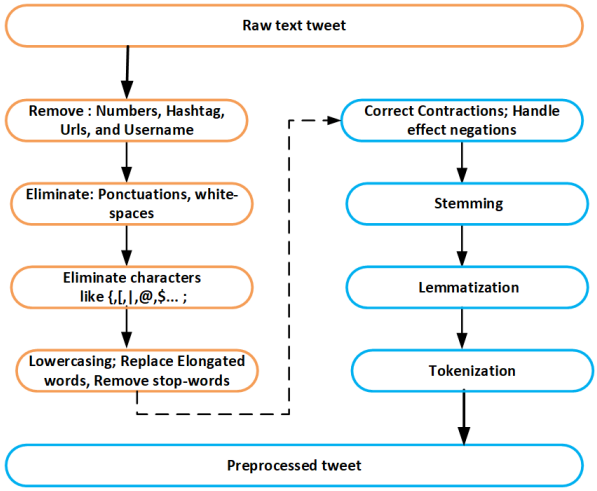


FIGURE 5. Basic steps of data preprocessing stage.

1) REMOVE USERNAMES, NUMBERS, HASHTAGS, AND URLs

It is a common technique to remove the usernames, numbers, hashtags, and URLs from the tweet since they do not express any sentiments.

2) REMOVE WHITE-SPACES, PUNCTUATION, AND SPECIAL CHARACTERS

The foremost stage to perform is removing all occurred white-spaces in the tweet, then we delete exclamation, stop and question punctuation marks. Finally, as we know the special characters do not hold any negative or positive effect on communicated opinion in the given tweet, hence all existed special characters are eliminated. After the implementation of all text pre-processing tasks introduced formerly, we preserved only the uppercase and lowercase letters.

3) LOWER-CASING

After the application of both formerly presented stages, all special characters have been removed and only the letters have been preserved. Therefore the next stage is the application of lower casing process. That means all the uppercase letters in the given tweet were converted to lower case, and that minimize the dimensionality of each terms.

4) REPLACE ELONGATED WORDS

This process aims to remove the letter, which is appeared at least more than two times in the word like the elongated term ''Saaaad''. So, after effecting this technique, the elongated word ''Saaaad'' becomes ''Saad'' and stamped with at most two characters.

5) REMOVE STOP-WORDS

It is the technique that removes the words which are frequently confronted in texts without dependence on a specific topic (e.g., ''a'', ''an'',''of'', ''on'' ''the'', ''in'', ''I'', ''she'', ''it'', ''your'', ''and'' etc.). These frequent use words are often ineffective and meaningless for the data classification task and deleted beforehand of the classification. Stop-words are particular to the studied language, as in the situation of stemming technique.

6) STEMMING

It is a technique that aims to reduce the size of each word by converting the derived word into its root, or stem form. Since the stem word is semantically analogous to its derived word. The stemming process is ordinarily referred to as stemming algorithms or stemmers. A straightforward stemming algorithm looks up the stem form of the derived word in a lookup table. This type of procedure is fast and easy to understand. Its shortcoming the lookup table does not contain all inflected forms of the stem word. For example, the words ''using'', ''usage'', ''used'' must be reduced to only one root form, which is ''use''.

7) LEMMATIZATION

it is a technique that acts as a Stemming algorithm. Its main goal is to find the root or stem form of words in the analyzed sentences. The only difference between these similar techniques is in the followed algorithm to capture the lemma or stem words, since the lemmatization takes into consideration the morphological analysis of the derived words.

8) TOKENIZATION

it is a technique that divides each input tweet into a set of meaningful words since each word is named a token. For example, a piece of text is split into sentences or words. In this contribution, we employed an NLTK tokenizer developed by Python.

In this work, the stage that follows up the pre-processing data phase is the data representation phase. In other terms, the obtained text after the utilisation of all text pre-processing tasks previously described will be the input of the data representation methods.

D. DATA REPRESENTATION STAGE

Most machine learning classifiers can only deal with numerical data; the kind of data we have in this work is text-based data. Therefore, to address our contribution for handling the obtained textual data after applying the preprocessing data stage employing machine learning algorithms, these textual data needs to be turned out into numerical values. This process is termed text vectorization or feature extraction phase. It is one of the fundamental issues in NLP that allowing machine learning techniques for examining text-based data. As we said previously, Feature extraction is the operation of turning out the text-based input data into a set of numerical features. The effectiveness of each machine learning classifier is depended significantly on the extracted features. Therefore, it is critical to select the better features that have a positive impact on classification accuracy. There are several diverse feature extractors to represent the textual data in numerical data, including Bag-Of-Words, N-grams, GloVe, Word2Vec, FastText, and TF-IDF. These different feature extractors will lead to different analysis results and influence differently the classification performance. This stage's

primary purpose is to summarize and convert text-based input data into a set of feature vectors that operate agreeably to the used machine learning classifier. On the other hand, our principal purpose is to apply all previously presented feature extractors and compare them to determine the better method that positively influences the classification rate. This section introduces in detail the previously different cited vectorization methods.

1) N-GRAMS OR CHARACTER-LEVEL

In the areas of probability, computational linguistics, and statistic, an N-grams is an adjacent sequence of items, since each item contains N characters obtained by dividing the analyzed text or word using the N-grams algorithm. For example, the word ''WORD'' would have the following N-grams:

- Bigrams (N = 2): _W, WO, OR, RD, D_
- Trigrams (N = 3): _WO, WOR, ORD, RD_, D__
- Quadrigrams (N = 4): _WOR, WORD, ORD_, RD__, D___

N-grams are handcrafted features which extensively act as distinctive features in text classification [86], opinion mining [87], cyber bullying detection [88], spam filtering [89], Authorship attribution and verification [90], plagiarism checker [91], and various other application fields. The N-grams methods aid in forming useful word representation vector for unknown words, thus enhancing classification accuracy in tasks based on textual data, where a significant portion of unknown words appears, e.g., in opinion mining. The primary benefit of N-grams methods is language independence, that is to say, the potential of porting a data representation method and a machine learning algorithm from one to another language is not taken into consideration. The problem of N-grams methods is they generate a significant number of features, and the implementation of the Hadoop framework can avoid this issue.

2) BAG-OF-WORDS

Bag-Of-Words feature extractor is one of the commonly used approaches in data representation task, which is called feature extractor. It is considered as a valuable and straightforward approach for data representation that maps a set of sentences to be classified into a numerical vector as $S_v[x_1; x_2; \ldots; x_n]$, where x_j describes the occurrence of the jth item in the collected vocabulary from the given dataset, i.e. it is only taken into consideration the word duplicates and disregarding the morphology and position of the words. This approach is hot research and possesses an excellent ability for picking out and assorting the features by constructing bags for each example kind; for data, it used in many application fields such as Natural Language Processing [92], Information Retrieval [93], Document classification [94], Sentence classification [95], Computer vision [96] and Opinion mining [97]. For example, we have the three book reviews [98] as described below:

- Review A: This book is very long and boring
- Review B: This book is not boring and is shortened
- Review C: This book is good and enjoyable

The vocabulary of this three movie reviews consists of eleven words which are: 'This', 'book', 'is', 'very', 'boring', 'and', 'long', 'not', 'shortened', 'good', 'enjoyable' as introduced in. Therefore the numerical vector of each review is created by the bag-of-word method as follows:

- **Vector of Review A**:[This:1, book:1, is:1, very:1, boring:1, and:1, long:1, not:0, shortened:0, good:0, enjoyable:0]
- **Vector of Review B**: [This:1, book:1, is:1, very:0, boring:1, and:1, long:0, not:1, shortened:1, good:0, enjoyable:0]
- **Vector of Review C**: [This:1, book:1, is:1, very:0, boring:0, and:1, long:0, not:0, shortened:0, good:1, enjoyable:1]

3) TF-IDF

In this study, TF-IDF is taken as one of the techniques to vectorize the text-based data of tweets. Each tweet is handled as a document. TF-IDF is used for data representation because it takes into account the frequency of a word over the whole list of documents. Within each sentence (document), each term is weighted according to its relevance in that sentence, i.e. every word is assigned a weight according to how pertinent it is to that sentence. Therefore, if a word occurs in numerous sentences, the weight assigned to that word is decreased, as it is not beneficial for discerning the sentences.TF-IDF forms a matrix in which rows depict the sentences, columns describe the terms, and values indicate the importance of the terms in the sentences.

TF meant term frequency that calculates how many times a word appears in a given sentence. IDF meant inverse document frequency that counts how many times that word appears within a set of sentences. If a term frequently appears in a given sentence, but it also appears in various other sentences, that term is not helpful to discriminate any given sentence.

$$W_{w,s} = \mathrm{tf}_{\mathrm{w,s}} \times \log(\frac{\mathrm{N}}{\mathrm{df_w}}) \quad (1)$$

where:

- $W_{w,s}$: weight of word w in sentence s.
- $\mathrm{tf}_{\mathrm{w,s}}$: number of occurrences of word w in sentence s.
- $\mathrm{df_w}$:number of sentences containing the word w.
- N: total number of sentences.

From the numerical formula preceding, it can be observed that if a term frequently appears in a given sentence, but it does not occur in various other sentences. Its TF rate will be high; thus, its IDF will be very approaching to 1. Hence, its TF-IDF rate will be high. In contrast, if a term frequently appears in a given sentence, but it further appears in various other sentences, even although its TF rate will be high, its IDF rate will be approaching 0. Consequently, its TF-IDF rate will be very low, if a term frequently appears in a given sentence and also is existed in all other sentences, its IDF rate will be equal to 0. Consequently, its TF-IDF rate will be equal to 0 as well. Relevant terms, which frequently appear

in a given sentence, will have an elevated TF-IDF rate, whilst less relevant terms, which frequently appear in both a given sentence and various other sentences, will have a soft TF-IDF rate. These rates represent the features that will be used later by machine learning classifiers in the learning process.

4) GloVe

Pennington *et al.* [99] are evolved the GloVe approach which means Global Vectors for Word representation, and the Stanford University shored this approach because of its important in the data representation. By definition, the GloVe is a paradigm that incorporates the benefits of the two prominent pattern families in the literature, which are global matrix factorization and local context window approaches. This training paradigm is an unsupervised model. Its main goal is measuring the representation of vector for given terms. The GloVe process is accomplished by computing the semantic similarity between terms, subsequently producing a matrix called the term-term co-occurrence count. Each value in this produced matrix indicate how times the term in the row and the correspond term in the column appear collectively in the set of sentences (documents). Because that, the GloVe model is also described as the pattern based on the count process. Term embedding matrix of this pattern is created by gathering the generated count-based co-occurrence matrix from a set of sentences (corpus). The obtained term embedding matrix describes for every term in vector space a significant linear substructure. Fig. 6 illustrates the linear substructures of set of words.

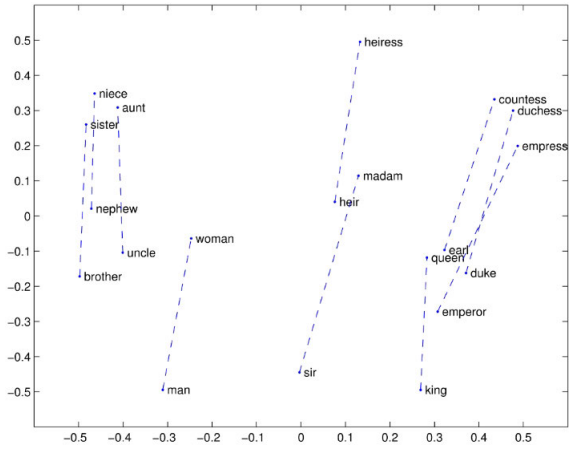


FIGURE 6. Illustration of linear substructures of a set of terms when applying the word embedding GloVe.

The representation graphic illustrated in Fig. 6 is a result of the application of an evaluation scheme based on word analogies which is adopted by GloVe to find the word vector space of an unknown word is used. For example, the relationship ''the brother is to sister as uncle is to aunt'' are represented using the vector representation space which is computed by the vector equation **brother-sister = uncle-aunt**. This evaluation system prefers patterns that provide dimensions of meaning, thence picking out the multi-clustering concept of distributed vectorizations.

5) Word2Vec

Word2Vec approach was suggested by Mikolov *et al.* [100], and was supported Google. Word2Vec is a method for natural language processing. This method employs a feedforward neural network model that contains only one hidden layer model to learn term embedding matrix from a large input corpus of text. Once trained, such a pattern can identify similar terms or suggest different terms for a partial sentence. As the name reveals, Word2Vec symbolizes each specific term with a particular list of numbers termed a vector. The vectors are taken accurately such that a straightforward numerical function designates the level of semantic similarity among the terms represented by those vectors. This approach combines the Continuous Bag-of-Words pattern, that forecasts the immediate target term using the vocabulary terms, and the Skip-Gram pattern, that forecasts the immediate vocabulary terms using the target terms. The substantial intent of this method is to evolve the predictive capability for data representation. This approach inputs a big corpus of data and outputs a matrix. In the produced matrix, every row describes the vector with hundred dimensions, which is the term representation of the one-hot vector of the inputted stem.

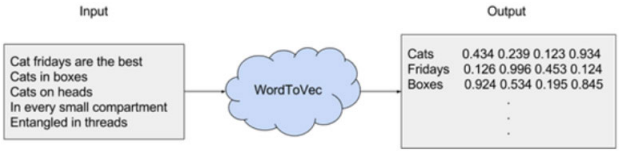


FIGURE 7. Illustration of Word2Vec data representation method.

In general, Word2Vec approach is a feedforward neural network with only one hidden layer, its fundamental objective is to regulate its weights for decreasing the inaccuracy average by minimizing the error function. Its hidden weights are utilized as the term embedding. Word2Vec achieves good accuracy in opinion mining, and its accuracy is good on big datasets.

6) FastText

Facebook AI Research team proposed new word embedding approach which is named FastText in order to improve the learning effectiveness of word representation. This technique is considered as a new version of the Word2Vec word embedding technique; So, instead of learning the word embedding vector of a set of terms immediately as in the Word2Vec approach, FastText approach learns the word embedding vector of a set of N-grams of characters. for instance, the vectorization of the term ''Happy'' employing the FastText approach with N-grams = 2 is (H, Ha, ap, pp, py, y), and the brackets indicate the starting and ending of the vectorized term. The advantage of the FastText approach permits to capture the meaning of shorter terms and aids the word embedding process to pick up the prefixes and suffixes of the learned

term. Therefore, the next stage after dividing the inputted term by using the character N-grams is the application of either continuous Bag-Of-Words or skip-gram in order to generate term embedding. FastText approach performs better with unseen terms and the terms out-of-lexicon than other word embedding techniques. Therefore, even if the term is out-of-lexicon in the former learning stages, this approach divide this term into multiple N-grams characters to learn its word embedding vector. Word2Vec and GloVe approaches both be unsuccessful to create word embedding vector for the out-of-lexicon terms, unlike FastText approach that has the ability to create the word embedding vector of the unseen terms. This is the powerful advantage of this technique in comparing with other approaches.

After the word embedding stage that endeavors to turn out the data into numerical values, which it inputs a pre-processed text, and it outcomes a term embedding vector. The following phase is the feature selection, as outlined in the next subsection.

E. FEATURE SELECTION

After the features extraction stage, in which the features are derived from the text corpus through the application of the FastText word embedding approach. It is worth taking into account the dimensionality of features. The reason is that the high feature dimensionality requires more costly computational resources. And also may probably decrease the accuracy and effectiveness of the applied learning algorithm. That is named ''The Curse of Dimensionality'' [101].

The cause why high features dimensionality may lead to the decreasing in the accuracy of applied patterns is because when the vector space of the features expands bigger and bigger by appending further and further features, the vector space of the features becomes further and further sparse. The further sparse in the vector space of the features causes further over-fitting in the applied machine learning algorithm on the training dataset. Therefore, suppose the vector space of the features is overly sparse. In that case, applied learning algorithms will over-fit the training dataset and thus became inefficient in classifying the unknown instances in the testing dataset. On the other hand, the try of decreasing dimensions of the features vector space may lead to decreasing in the performance as well, as this might eliminate some relevant features and cause to under-fitting of the trained algorithms to the data.

In the literature, There are two techniques for decreasing the dimensionality of the feature space. The first technique is feature projection, and the second technique is the feature selection. The distinction between these techniques is that the former technique shrinks the dimensionality of the vector space of the features by projecting the features in high-dimensional vector space upon low-dimensional vector space. While the latter technique eliminates irrelevant features. A new subset of features is created with the former techniques, while a subset of features is kept with the latter techniques.

In this paper, Chi-Square, Information Gain, Gain Ratio, Mutual Information, and Gini Index as feature selection techniques are employed for reducing the dimensionality of feature vector spaces for text opinion mining in order to ameliorate the accuracy and diminish the execution time of used machine learning algorithms.

1) INFORMATION GAIN

Information gain (IG) has been utilized commonly as a term (feature) goodness criterion in the text classification based on machine learning techniques such as C4.5 and ID3. It gauges the information needed in bits for class label prediction of a given sentence (or document) by measuring the entropy that is computed based on the absence or existence of a feature word in that sentence. IG is obtained by measuring the effect of the feature word's inclusion on diminishing overall entropy. The predictable information required to predict the class label of an example in partition PA is recognized as entropy and is computed as follows:

$$\text{Entropy(PA)} = -\sum_{i=1}^{C}(P_i) \times log_2(P_i) \tag{2}$$

where:

- C indicates the overall number of the class labels in the corpus.
- $P_i = \frac{|C_{i,\text{PA}}|}{|\text{PA}|}$ is the probability that an arbitrary example sentence in partition PA has the class label C_i.

To predict the class label of an example sentence in PA on some feature $F\{f_1, f_1, \ldots, f_v\}$, and the partition PA is divided into k sub-partitions $\{\text{PA}_1, \text{PA}_2, \ldots, \text{PA}_k\}$. Therefore, the information required to get an exact prediction is calculated by:

$$\text{Entropy}_F\text{(PA)} = -\sum_{j=1}^{k}\frac{|\text{PA}_j|}{|\text{PA}|} \times \text{Entropy(PA}_j) \tag{3}$$

where:

- $\frac{|\text{PA}_j|}{|\text{PA}|}$ is the weight of the jth partition PA_j.
- Entropy(PA_j) is the entropy of jth partition.

Finally, IG by dividing up on feature F is measured by the following equation:

$$\text{IG}(F) = \text{Entropy(PA)} - \text{Entropy}_F\text{(PA)} \tag{4}$$

In this research, before reducing the feature vector space dimension, each feature within the sentence is ordered depending on their relevance for the text classification in decreasing order employing the IG technique. Thus, in the learning process of text classification, features that have lesser importance are neglected, and dimension decrease techniques are exercised to the features that have higher importance (i.e. IG).

2) GAIN RATIO

Gain Ratio (GR) aims to improve the IG by normalizing the supplying of all terms to the final decision of the classification

process for a given sentence (because in our study, we work on sentence-level classification). An iterative procedure picks out smaller groups of terms (features) in decremental by applying GR rate. The iterative process ends when the previously defined number of terms remain. GR is employed as a disparity ratio, and a high GR rate means that the chosen terms will be valuable for text classification. GR was proposed in the C4.5 decision tree algorithm. The applied normalization rate is called the split information rate. The split information rate is represented by the prospective knowledge acquired by dividing up the training partition PA into k subpartitions, corresponding to k conclusions on the attribute F:

$$\text{SplitInfo}_F(\text{PA}) = -\sum_{j=1}^{k} \frac{|\text{PA}_j|}{|\text{PA}|} \times \log_2 \frac{|\text{PA}_j|}{|\text{PA}|} \tag{5}$$

where a high SplitInfo rate indicates that the partitions are uniform and a low SplitInfo rate indicates few partitions contain most of the instances. Therefore, GR is computed as follows:

$$\text{GR}(F) = \frac{\text{IG}(F)}{\text{SplitInfo}(F)} \tag{6}$$

3) CHI-SQUARE

Chi-Square is a statistical technique (CHI) for estimating how dependent an attribute variable on the corresponding target variable. A high Chi-Square rate indicates that the association between a feature variable and the corresponding target variable is very strong. Whereas a low Chi-Square rate indicates that the association between a feature variable and the corresponding target variable is very weak. And CHI is equal to 0 in the case that the feature variable is independent of the corresponding target variable. CHI score is computed following up the described steps below:

1) Determine the null assumption that shows that the feature variable and the corresponding target variable are independent, and define the alternative assumption that indicates that the feature variable is dependent on the corresponding target variable
2) Design a table that indicates how the corresponding class variables in rows and attributes in columns are spread out. The ratio of freedom for the created table is (row $-$ 1) $\times$ (column $-$ 1).
3) Compute the predicted values for all the cells of the formed table previously using the following formula: $e = n \times p$. Where e is the predicted value; n is the number of times that the value of each cell appears in the whole table; and p is the joint probability between an attribute and its corresponding class.
4) Compute the CHI statistic using the following formula: $x_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$. Where c is the degree of freedom; O is the observed value; and E is the expected value.
5) The CHI rate calculated above can be examined against the Chi-Square created table in step 2 to check whether it occurs in the acceptance or rejection cell. If it occurs in the refusal cell, then the null hypothesis is declined, and the alternative hypothesis is admitted. If it occurs in the approval cell, then it is the other way around. Consequently, whether the attribute and the corresponding class are independent of each other can be captured. If they are autonomous, then the attribute can be excluded.

4) GINI INDEX

Gini index is an impurity-based measure that computes a particular class's purity after dividing along with a specific feature. The best separating raises the purity of the sub-partition resulting from the separation. If D is a given dataset with C the total number of different class labels, GINI is computed using the following formula:

$$\text{Gini}(D) = 1 - \sum_{i=1}^{C} (P_i)^2 \tag{7}$$

where P_i is a relative frequency if class label i in dataset D. The smaller its value, i.e., the lesser the ''impurity'', the better the attribute, and we save it. The bigger its value, i.e., the higher the ''impurity'', the attribute is inefficient, and we remove it.

After the application of the feature selectors in order to decrease the dimensionality of the attribute vector space by removing the irrelevant attributes from the set of extracted attributes in the feature extraction stage. The next stage is the classification phase using our improved ID3 [33], as introduces in the following subsection.

F. CLASSIFIER

In this work, and after the feature selection phase, the next phase is the classification task, in which we used improved ID3 in order to classify each input sentence into the negative or neutral or positive class label. ID3 decision tree technique is a classification paradigm that pursues to measure the IG criterion described by the equation (4) for creating a decision tree. This decision tree technique computes the IG to pick out the better splitting feature in every algorithm's round. The computing IG process considers only a present conditional feature and class label feature, and the other conditional features cannot be employed to gauge the feature importance. Thus, we proposed a novel enhanced ID3 based on weighted adjusted IG. The weighted modified IG considers the association between the current conditional feature, the feature decision, and the other conditions features in the learning process's progress. The principal purpose of our enhanced ID3 based on weighted adjusted IG is to gauge the influence of the other conditional features on the IG criterion for the present conditional features in the training operations. Or rather, our enhanced ID3 approach takes into account the association between the present analyzed feature and the other features and the association between each conditional feature and the class label feature in the progress of the training operation. The weighted modified IG is computed using weighted attribute and the weighted correlation function.

G. WEIGHTED ATTRIBUTE

Let $F = \{F_1, F_2, \ldots, F_k\}$ be a be a collection of K conditional features. We suppose the occurrence number of feature $F_i (i = 1, 2, \ldots, K)$ is K_i. Thus, the frequency of F_i can be described as:

$$\mathrm{TF}_i = \frac{K_i}{K} \tag{8}$$

Then the weight of the feature F_i can be calculated as:

$$\mathrm{WF}_i = \frac{\mathrm{TF}_i}{\sum_{i=1}^{K} \mathrm{TF}_i} \tag{9}$$

Due to the conditions of weight theory, the total importance of all features satisfy the equation described below:

$$\mathrm{WF}_i = 1 \tag{10}$$

H. WEIGHTED CORRELATION FUNCTION

Let $F = \{F_1, F_2, \ldots, F_k\}$ be a collection of K conditional features with an interval of values $\{\mathrm{RV}_1, \mathrm{RV}_2, \ldots, \mathrm{RV}_k\}$, respectively. Let C be a class label feature with an interval of values RV_C. $F_{i\in(i=1,2,\ldots,K)}$ is one of conditional features of the collection and has N values, whereas $\mathrm{RV}_i \in (i = 1, 2, \ldots, K) = f_1, f_2, \ldots, f_N$. C is the class label feature and has Y values $\mathrm{RV}_C = \{C_1, C_2, \ldots, C_Y\}$. Thus, the weighted correlation function (WCF) between the conditional feature A_k and the class label feature Y is computed using the following formula:

$$\mathrm{CF}(F_i, C) = \frac{\sum_{v=1}^{N} \left| |F_{vj}| - \sum_{j=2}^{Y} |F_{vj}| \right|}{N} \tag{11}$$

where N is the number of values of F_i ; $|F_{vj}|$ is the examples' number of that vth value of F_i be associated with the jth category of class label feature C. Then WCF of the feature F_i can be calculated as:

$$\mathrm{WCF}(F_i, C) = \frac{\mathrm{CF}(F_i, C)}{\sum_{i=1}^{K} \mathrm{CF}(F_i, C)} \tag{12}$$

I. WEIGHTED MODIFIED INFORMATION GAIN

Weighted modified information gain (WMIG) is measured using the calculated IG in the equation (4) and the computed WCF in the equation (12):

$$\mathrm{WMIG}(F) = \mathrm{IG}(F) \times \sum_{i=1}^{N} \mathrm{WCF}(F). \tag{13}$$

Subsequently, due to the IG criterion and the weight representation theory, the particular implementation of our improved ID3 decision tree technique based on WMIG is introduced in detail in the algorithm 1

J. PARALLELIZATION OF OUR PROPOSED APPROACH

The appearance of the Big Data epoch poses a defy to conventional machine learning algorithms. In information technology, big data is a set of datasets that contain a massive amount of data, which makes it very complicated to process employing popular database management devices or conventional data treatment applications. Big data is generally constituted of datasets with sizes exceeding the capability of ordinarily utilized software devices, which are incapable of capturing, managing, or processing such data within a reasonable and feasible execution time. Big data challenges involve acquisition, storage, exploration, distribution, examination, and visualization. Consequently, both the time and space effectiveness of conventional machine learning algorithms diminish dramatically when handling Big Data. To remedy these challenges in this study, we have applied the Big Data Hadoop framework [102] as depicted in Fig. 8 that represents the implementation of our proposal using Hadoop framework with its distributed file system and mapreduce programming model.

As illustrated in the Fig. 8, Hadoop is a framework employed for storing and treating a massive amount of data. Our dataset is stored on five inexpensive machines: four slave computing nodes and one master computing node that run as a cluster. Data storage is effectuated using a Hadoop distributed file system, which enables simultaneous processing and fault tolerance. On the other hand, data processing is carried out by the MapReduce programming model, which provides a kind of distributed parallel computation environment for retrieval and processing the massive data which is stored in the Hadoop distributed file system. MapReduce divides the computation process of the massive amount of data into Map and Reduce steps, which match to the implementation of a mapper () method and a reducer () method, respectively. The MapReduce process takes as input the chunk tweet in the format of <key, value> pairs, where the ''key'' variable represents the serial number of the inputted chunk tweet, and the ''value'' variable describes the data value of the inputted chunk tweet. In the Map step, MapReduce splits data into partitions of equal sizes and treats each partition in the form of <key, value> pairs <Key1, Value1> to determine the formal input. It applies the mapper () method to produce intermediate outcomes in the form of <Key2, Value2>, which are arranged according to the data value of Key2. The ''Value2'' values whose ''key2'' serial number are the same are merged to create a novel list <Key2, list(Value2)> and, subsequently, gathered according to the serial number ''Key2'' for starting the execution of Reduce tasks. In the Reduce step, the outcomes of the Map jobs are combined and arranged, <Key2, list(Value2)> is used as the input, and the reduce () method is performed to get the <key, value> pairs <Key3, Value3>, which is output to store in the Hadoop distributed file systems.

IV. EXPERIMENT AND RESULTS

The previous sections were consecrated to the general introduction, previous research, materials and methods of our proposed classifier. As we said early, our proposed classifier consists of five steps; data collection in which we have chosen Sentiment140 (https://www.kaggle.com/kazanova/sentiment140 and COVID-19_Sentiments https://www.kaggle.com/abhaydhiman/covid19-sentiments datasets to apply our contribution to them. After the data collection

Algorithm 1: Our Enhanced Weighted Algorithm ID3

Input:

. **Training feature set:** $F = \{F_1, F_2, \ldots, F_K\}$ be a collection of K conditional features with an interval of values $\{\mathrm{RV}_1, \mathrm{RV}_2, \ldots, \mathrm{RV}_N\}$, respectively. And C is a class label feature and has RV_Y values $\mathrm{RV}_Y = \{C_1, C_2, \ldots, C_Y\}$.

. **Training example set:** $E = \{(e_i, c_i) \| e_i \in \mathrm{RV}_1 \times \mathrm{RV}_2 \times \ldots \times \mathrm{RV}_n, \mathrm{c_i} \in \mathrm{RV}_Y\}$ is an example picked from an unknown distribution, where e_i has an outcome c_i related with it.

. **Ending Condition:** is the condition to end the training operation.

1 All samples in the training dataset relate to a unique value of c.

2 All feature values of E are the same or the feature set F is empty.

Output: DT (Decision tree)

TreeCreate(E, F) : generate a novel decision tree DT with a single node called root

if EndingCondition(E) $= 1$ **then**

  Tick the root node as a leaf node labelling the class C.

  return

else if EndingCondition(E) $= 2$ **then**

  Tick DT as a leaf node with the most prevalent value of class label feature C in the training examples E as a label

  return

else

  for $F_i \in F$ **do**

    Compute the IG employing the next equation

    $\mathrm{IG}(F_i)[i] \leftarrow \mathrm{Entropy}_{F_i}(C) - \mathrm{Entropy}_{F_i}(F_i, E)$

  end for

  $\mathrm{SImp} \leftarrow 0$

  for featureValues $\mathrm{FV_i} \in \mathrm{RV_i}$ **do**

    $\mathrm{PImp} \leftarrow 1$

    Compute the importance of each feature employing the training set E_i of each value FV_i of feature F_i

    $S \leftarrow 0$

    for featureValues $F_k \in F \setminus \{F_i\}$ **do**

      compute the frequency employing the following equation

      • $\mathrm{CF}_{(F_k,C)}[k] \leftarrow \frac{\sum_{v=1}^{|\mathrm{FV}_i|} ||F_{vy}| - \sum_{c=2}^{Y} |F_{vy}||}{V_k}$ • $S \leftarrow S + \mathrm{CF}_{(F_k,C)}[k]$

    end for

    for featureValues $F_k \in F \setminus \{F_i\}$ **do**

      compute the importance employing the following equation

      • $\mathrm{WCF}_{(F_j,C)}[k] \leftarrow \frac{\mathrm{CF}_{(F_k,C)}[k]}{S}$ • $\mathrm{PImp} \leftarrow \mathrm{PImp} \times \mathrm{WCF}_{(F_j,C)}[k]$

    end for

    $\mathrm{SImp} \leftarrow \mathrm{SImp} + \frac{|E_i|}{|E|} \times \mathrm{PImp}$

  end for

  $\mathrm{IG}(F_i)[i] \leftarrow \mathrm{IG}(F_i)[i] \times \mathrm{SImp}$

end if

Determining the better dividing feature F_{better} that has the maximum weighted adjusted $\mathrm{IG}(F_i)[i]$

$F_{\mathrm{better}} \leftarrow \mathrm{argmax}_F \mathrm{IG}(F_i)[i]$

Attach F_{better} into DT

for featureValues $v \in F_{\mathrm{better}}$ **do**

  create an edge that links the new node with the previous created DT, and E_v describes the subset of the examples in E of which the F_{better} feature is v

  if $E_v =$ null **then**

    Tick the edges of the new node as the values labels of the preceding leaf node, and its class label is ticked as the class label that contains the highest number of examples in E

    return

  else

    Recursion of **TreeCreate**(E_v, $F \setminus \{F_{\mathrm{better}}\}$) continues

  end if

end for

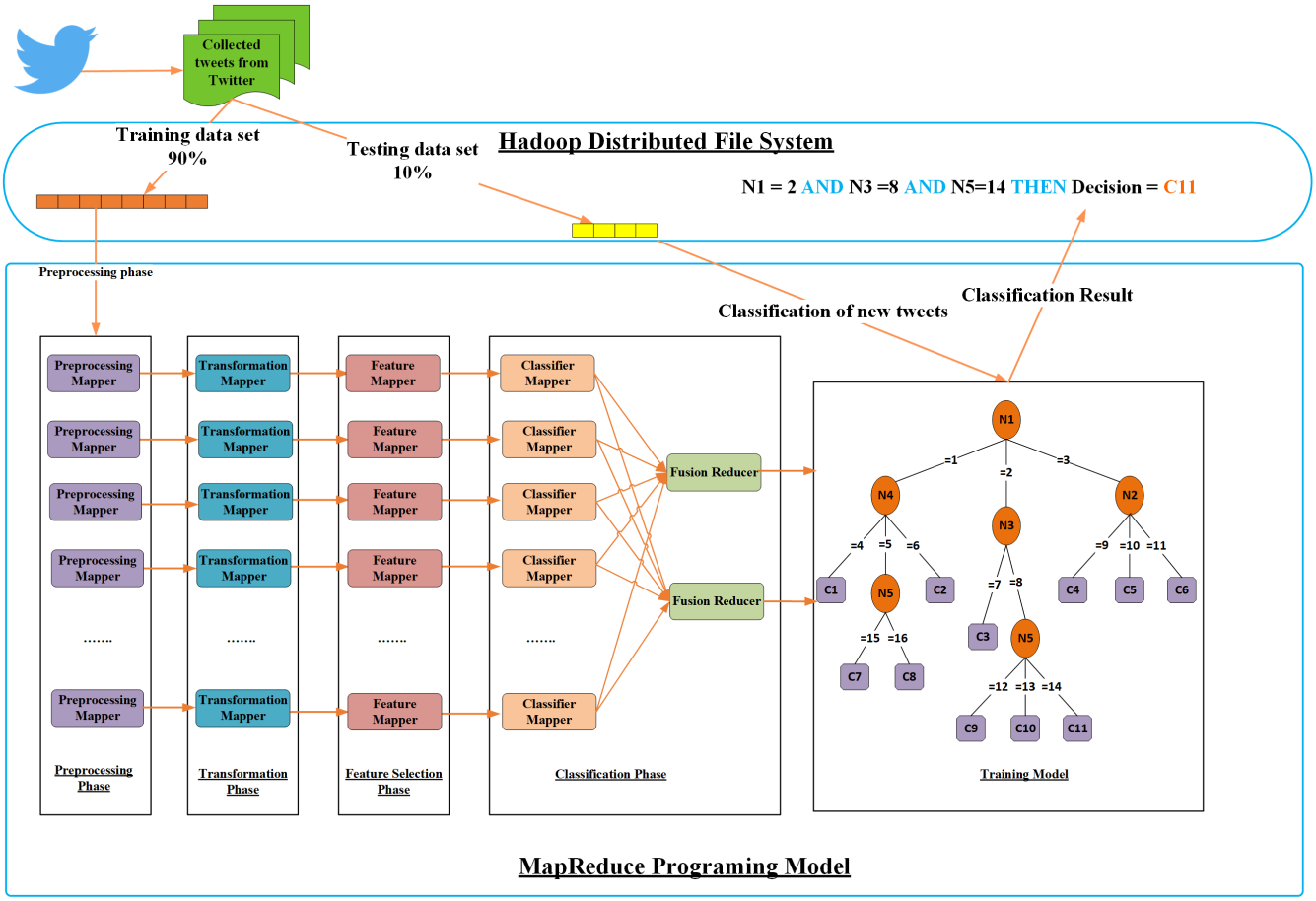


FIGURE 8. Basic steps of opinion mining on social network platforms using Hadoop framework.

phase, the next phase is the text preprocessing phase, in which we have applied several preprocessing tasks to reduce the noisy data and remove unwanted information for improving the data quality. The data representation stage follows up the text-preprocessing stage, which is an operation of converting the text-based input data into a set of numerical features. In this phase, we have applied several techniques, including N-grams, Bag-of-words, TF-IDF, GloVe, Word2Vec, Fast-Text. After the data representation stage, the next phase is the feature selection phase aiming to reduce the high feature dimensionality. Also, in this phase, we have applied many feature selectors such as Chi-square, IG, Gain Ratio, and Gini Index. Finally, and after the feature selection phase, the next phase is the application of our improved ID3 as presented in the subsection "Classifier". In this section, we perform five experiments to demonstrate our suggested classifier's correctness and effectiveness compared to other literature's methods. And to assess its effectiveness, we have selected nine evaluation criterion as presented in subsection "Evaluation metrics".

A. EVALUATION METRICS

To assess our text classification method, we principally compute ten assessment metrics: *True Positive Rate* (TPR), *True Negative Rate* (TNR), *Kappa Statistic* (KS), *False Positive Rate* (FPR), *Precision* (PR), *False Negative Rate* (FNR), *Classification Rate or Accuracy* (AC), *Error Rate* (ER), *Time Consumption* (TC), and *F1-score* (FS) [103]. These evaluation metrics are computed as described in Table 7 and based on the confusion matrix for binary classification [68] as given in Fig. 9.

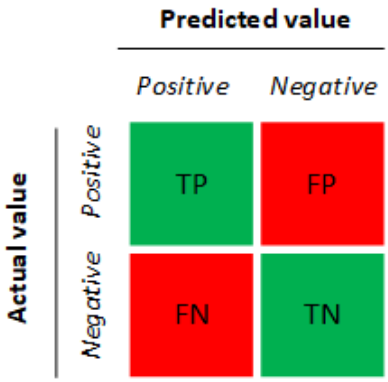


FIGURE 9. Confusion matrix for a binary classification task.

Where:

- TP means *True Positive* which is the total number of sentiment sentences that are currently positive and classified to be positive.

TABLE 7. Measures for binary and multi-class classification using the notation of Fig. 9.

Measure	Binary-class Formula	Multi-class Formula
TPR	$\frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fn}}$	$\frac{\sum_{i=1}^{l}\frac{\mathrm{tp}_i}{\mathrm{tp}_i+\mathrm{fn}_i}}{l}$
TNR	$\frac{\mathrm{tn}}{\mathrm{tn}+\mathrm{fp}}$	$\frac{\sum_{i=1}^{l}\frac{\mathrm{tn}_i}{\mathrm{tn}_i+\mathrm{fp}_i}}{l}$
FPR	$\frac{\mathrm{fp}}{\mathrm{fp}+\mathrm{tn}})$	$\frac{\sum_{i=1}^{l}\frac{\mathrm{fp}_i}{\mathrm{fp}_i+\mathrm{tn}_i}}{l}$
FNR	$\frac{\mathrm{fn}}{\mathrm{fn}+\mathrm{tp}})$	$\frac{\sum_{i=1}^{l}\frac{\mathrm{fn}_i}{\mathrm{fn}_i+\mathrm{tp}_i}}{l}$
MER	$\frac{\mathrm{fp}+\mathrm{fn}}{\mathrm{tp}+\mathrm{fn}+\mathrm{tn}+\mathrm{fp}}$	$\frac{\sum_{i=1}^{l}\frac{\mathrm{fp}_i+\mathrm{fn}_i}{\mathrm{tp}_i+\mathrm{fn}_i+\mathrm{tn}_i+\mathrm{fp}_i}}{l}$
PR	$\frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}}$	$\frac{\sum_{i=1}^{l}\frac{\mathrm{tp}_i}{\mathrm{tp}_i+\mathrm{fp}_i}}{l}$
AC	$\frac{\mathrm{tp}+\mathrm{tn}}{\mathrm{tp}+\mathrm{fn}+\mathrm{tn}+\mathrm{fp}}$	$\frac{\sum_{i=1}^{l}\frac{\mathrm{tp}_i+\mathrm{tn}_i}{\mathrm{tp}_i+\mathrm{fn}_i+\mathrm{tn}_i+\mathrm{fp}_i}}{l}$
KS	$\frac{P_0-P_e}{1-P_e}$	$\frac{\sum_{i=1}^{l}\frac{P_{0\mathrm{i}}-P_{\mathrm{ei}}}{1-P_{\mathrm{ei}}}}{l}$
FS	$\frac{2\times\mathrm{PR}\times\mathrm{TPR}}{\mathrm{PR}+\mathrm{TPR}}$	$\frac{\sum_{i=1}^{l}\frac{2\times\mathrm{PR}_i\times\mathrm{TPR}_i}{\mathrm{PR}_i+\mathrm{TPR}_i}}{l}$

- FN means *False Negative* which is the total number of sentiment sentences that are currently positive and classified to be negative.
- TN means *True Negative* which is the total number of sentiment sentences that are currently negative and classified to be negative.
- FP means *False Positive* which is the total number of sentiment sentences that are currently negative and classified to be positive.

B. RESULTS AND DISCUSSION

In this subsection, we have performed five experiments. The first experiment aims to evaluate the effectiveness of each applied pre-processing tasks. The second experiment explores the most efficient feature extractor among all employed methods such as TF-IDF, Bag-Of-Words, N-grams, GloVe, Word2Vec, and FastText. The third experiment assesses the Chi-square, Information Gain, Gain Ratio, and Gini Index approaches in order to determine the approach with high classification rate. The fourth experiment is did to prove the effectiveness of our suggested approach in terms of AC, TPR, ER, TNR, FS, FPR, KS, FNR, PR, and TC compared to ID3, C4.5, Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58] approaches. Finally, the fifth experiment is performed to proved the performance of our classifier in terms of complexity, stability and convergence compared to Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58] classifiers.

C. EXPERIMENT 1

In this first experiment, we analyzed the performance of all applied data preprocessing techniques in terms of dataset size and execution time before and after the implementation of the Hadoop framework. Therefore, the main goal of applying data preprocessing techniques on the tweets dataset is to remove the noise, improve the data quality, and diminish the dataset size, as depicted in Fig. 10 and 11.

Fig. 10 and 11 represent the dataset size and the data execution time of each data preprocessing task before and after the implementation of the Hadoop framework on the Sentiment140 and COVID-19_Sentiments dataset, respectively. As an outcome, the processing data can achieve high

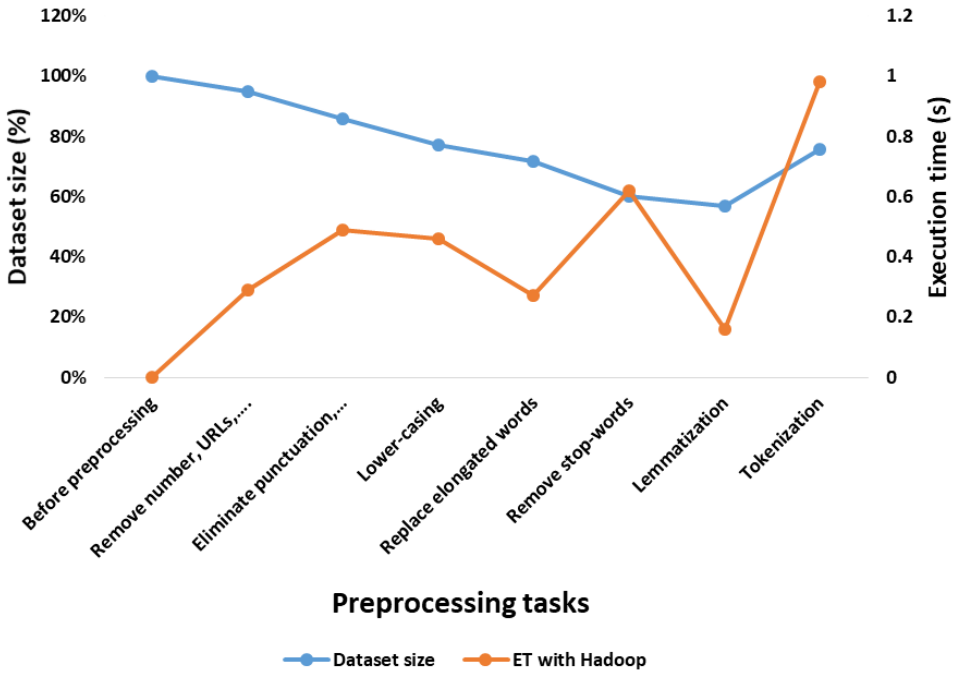


FIGURE 10. Dataset size and data execution time of each data preprocessing task applied on the Sentiment140 dataset.

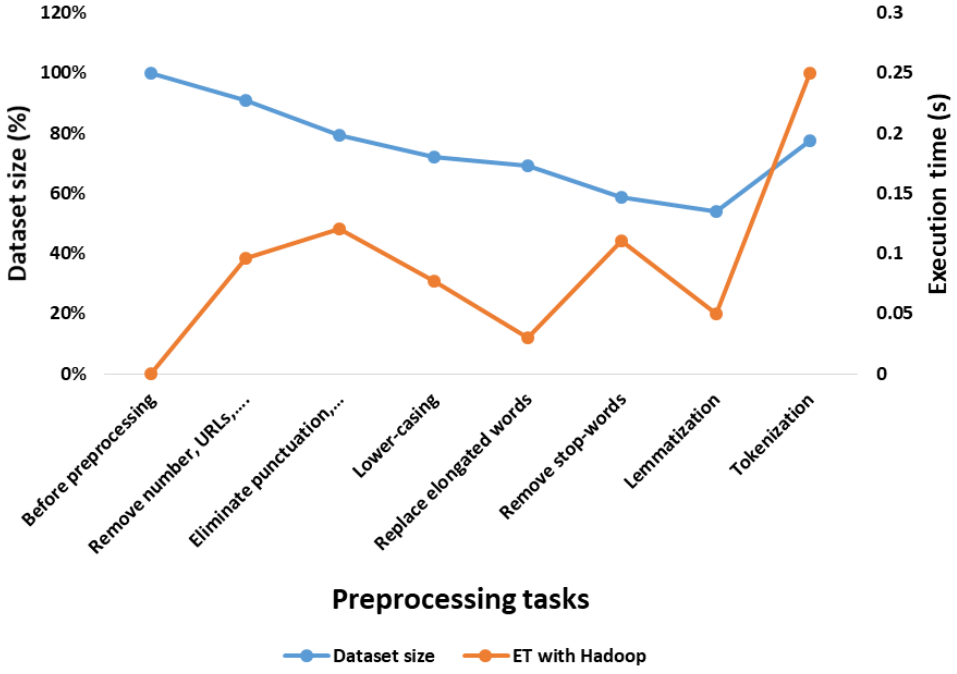


FIGURE 11. Dataset size and data execution time of each data preprocessing task applied on COVID-19_Sentiments dataset.

TABLE 8. Analysis of dataset size and data execution time in the case of the application of each data preprocessing task on the Sentiment140 dataset.

Preprocessing techniques	Dataset size (KB)	Dataset size (%)	ET without Hadoop (s)	ET with Hadoop (s)
Before preprocessing	283234.71	100 %	——	——
Remove usernames, numbers, hashtags, and URLs	268251.59	94.71 %	1.49	0.29
Remove white-spaces, special characters, and punctuation	242618.85	85.66 %	2.45	0.49
Lower-casing	218204.02	77.04 %	2.33	0.46
Replace elongated words	203532.46	71.86 %	1.38	0.27
Remove stop-words	169997.47	60.02 %	3.08	0.62
Lemmatization	161302.17	56.95 %	0.80	0.16
Tokenization	214748.55	75.82 %	4.92	0.98

classification performance in evaluating and examining data when one of the supervised learning algorithms is trained. The dataset size is decreased after each data preprocessing task. Table 8 and 9 display the reduction in dataset file size after the application of each data preprocessing task on Sentiment140 and COVID-19_Sentiments dataset, respectively.

As can be seen in Fig. 10, we note that the application of "*Remove usernames, numbers, hashtags, and URLs*" preprocessing task reduces the Sentiment140 dataset's size by 5.29 %, and it takes the time 0.29 s. Also, "*Remove punctuation, white-spaces, and special characters*" preprocessing task diminishes the Sentiment140 dataset's size by 9.05 %, and it consumes the time 0.49 s. The task "*Lower-casing*" minimizes the Sentiment140 dataset's size by 8.62 % and it expends the time 0.46 s. "*Replace elongated words*" preprocessing task reduces the Sentiment140 dataset's size by 5.18 % and it takes the time 0.27 s. For the task "*Remove stop-words*" diminishes the Sentiment140 dataset's size by 11.84 %, and it consumes the time 0.62 s. The preprocessing task "*Lemmatization*" decreases the Sentiment140 dataset's size by 3.07 %, and it expends the time 0.16 s. Finally, the preprocessing task "*Tokenization*" increases the

TABLE 9. Analysis of dataset size and data execution time in the case of the application of each data preprocessing task on the COVID-19_Sentiments dataset.

Preprocessing techniques	Dataset size (KB)	Dataset size (%)	ET without Hadoop (s)	ET with Hadoop (s)
Before preprocessing	112935.94	100 %	——	——
Remove usernames, numbers, hashtags, and URLs	102647.45	90.89 %	0.49	0.096
Remove white-spaces, special characters, and punctuation	89693.72	79.42 %	0.61	0.12
Lower-casing	81347.75	72.03	0.39	0.077
Replace elongated words	78095.20	69.15 %	0.15	0.03
Remove stop-words	66270.81	58.68 %	0.54	0.11
Lemmatization	60883.76	53.91 %	0.26	0.05
Tokenization	87468.88	77.45 %	1.23	0.25

Sentiment140 dataset's size by 18.87 %, and it consumes the time 0.80 s.

According to Table 8, we observe that the Sentiment140 dataset's size before applied pre-processing tasks is equal to 283234.71 KB. After applying all used pre-processing tasks except Tokenization, the size of the dataset became 161302.17 KB. Therefore the dataset's size is reduced by 43.05 %, which represents the percentage of noisy and unwanted data. After applying the Tokenization pre-processing task, the dataset's size increases to 214748.55 KB because this technique serves to divide each input sentence into a set of tokens, raising the dataset's size. Also, the time consuming after applying all pre-processing tasks is equal to 16.45 s, but the application of the Hadoop framework reduces this value to 3.27 s.

From Fig. 11, we remark that the application of "*Remove usernames, numbers, hashtags, and URLs*" preprocessing technique decreases the COVID-19_Sentiments dataset's size by 9.11 % and it consumes the time 0.096 s. In addition, "*Remove white-spaces, punctuation, and special characters*" preprocessing technique reduces the COVID-19_Sentiments dataset's size by 11.47 %, and it expends the time 0.12 s. The task "*Lower-casing*" minimizes the COVID-19_Sentiments dataset's size by 7.39 %, and it expends the time 0.077 s. "*Replace elongated words*" preprocessing task reduces the COVID-19_Sentiments dataset's size by 2.88 %, and it takes the time 0.03 s. For the task "*Remove stop-words*" diminishes the COVID-19_Sentiments dataset's size by 10.47 %, and it consumes the time 0.11 s. The preprocessing task "*Lemmatization*" decreases the COVID-19_Sentiments dataset's size by 4.77 %, and it expends the time 0.05 s. Finally, the preprocessing task "*Tokenization*" increases the COVID-19_Sentiments dataset's size by 23.54 %, and it consumes the time 0.25 s.

According to Table 9, we remark that the COVID-19_Sentiments dataset's size before applied pre-processing tasks is equal to 112935.94 KB. After applying all used pre-processing tasks except Tokenization, the size of the dataset became 60883.76 KB. Therefore the dataset's size is reduced by 46.09 %, which represents the percentage of noisy and unwanted data. After applying the Tokenization pre-processing task, the dataset's size increases to 87468.88 KB because this technique serves to divide each input sentence into a set of tokens, increasing the dataset's size. Also, the time consuming after applying all pre-processing tasks is equal to 3.67 s, but the application of the Hadoop framework reduces this value to 0.733 s.

Another experiment is performed to evaluate the efficiency of all used data pre-processing tasks on COVID-19_Sentiments and Sentiment140 datasets by computing the error rate with and without the utilisation of text pre-processing mechanisms. Table 10 shows the experimental results of the calculation of error rate (ER %) without and with the application of data pre-processing approaches on both datasets.

From the empirical results, as introduced in Table 10, we concluded that the text pre-processing tasks minimize the ER. Since the ER in the case of the Sentiment140 dataset

TABLE 10. Error rate (ER %) without and with text pre-processing mechanisms.

Name of dataset	ER without data pre-processing	ER with data pre-processing
Sentiment140	39.81	13.47
COVID-19_Sentiments	30.12	11.18

diminishes from 39.59 % to 13.47 %, and it reduces from 30.04 % to 11.18 % in the case of the COVID-19_Sentiments dataset. Therefore, it is recommended to apply the text pre-processing mechanisms.

1) EXPERIMENT 2

This second experiment aims to determine the most efficient feature extractor in terms of classification rate. We have applied several feature extractors in this work, including N-grams, Bag-Of-Words, TF-IDF, GloVe, Word2Vec, FastText. As we said earlier, these feature extractors serve to convert the tweets' input data into a set of numerical features. Fig. 12 introduces the accuracy of sentiment classification tasks depending on the length of character N-grams and obtained on the Sentiment140 dataset using the Hadoop framework.

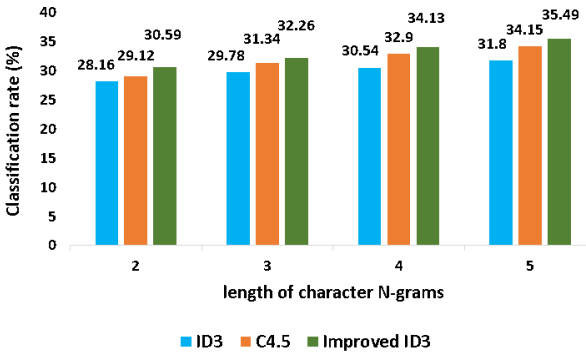


FIGURE 12. Accuracy of sentiment classification task depending on the length of character N-grams and obtained on the Sentiment140 dataset using the Hadoop framework.

As can be seen in Fig. 12, we note that N-gram = 5 achieved good classification performance compared to other characters N-grams (N-grams = 2, N-grams = 3, N-grams = 4). Since the N-grams = 5 reached classification accuracy is equal to 31.80 %, 34.15 %, and 35.49 % in the case of the three used machine learning algorithms C4.5, ID3, and our improved ID3, respectively.

Fig. 13 describes the accuracy of sentiment classification tasks depending on the length of character N-grams and obtained on the COVID-19 Sentiments dataset using the Hadoop framework.

As can be seen in Fig. 13, we also remark in this experiment that N-grams = 5 achieved good classification performance compared to other length characters N-grams and which is equal to 51.76 %, 54.41 % and 54.99 % in the case of the three used machine learning algorithms C4.5, ID3, and our improved ID3 respectively.

By comparing the obtained results in Fig. 12 and 13, we deduce that the N-grams representation method is inefficient in the case of Big Data because it achieved an accuracy less than 36 % in the case of the big Sentiment140 dataset. In both experiments, we remark that the accuracy increases when augmenting the length of character N-grams.

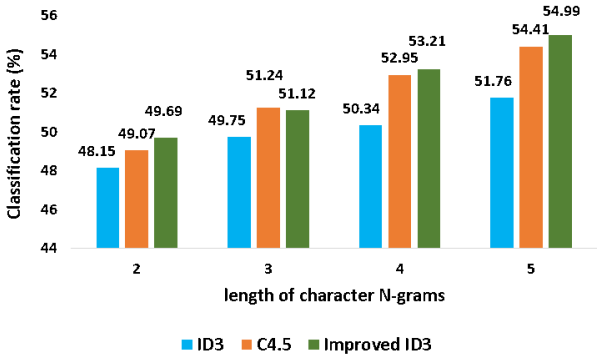


FIGURE 13. Accuracy of sentiment classification task depending on the length of character N-grams and obtained on the COVID-19 Sentiments dataset using the Hadoop framework.

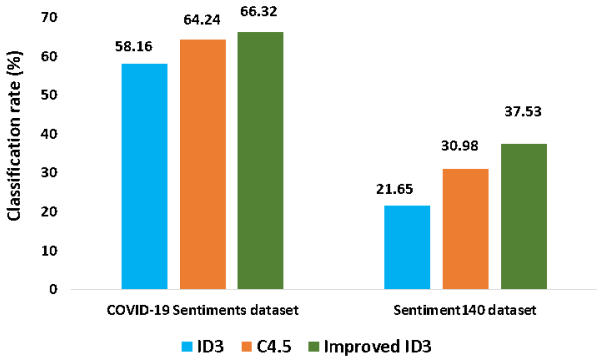


FIGURE 14. Accuracy of opinion mining task obtained after the utilisation of Bag-Of-Words on the COVID-19 Sentiments and Sentiment140 datasets using the Hadoop framework.

Fig. 14 presents the accuracy of sentiment classification tasks obtained on the Sentiment140 and COVID-19 Sentiments datasets using the Bag-Of-Words extractor method and the Hadoop framework. As can be seen in Fig. 14, we note that the application of Bag-Of-Words on the COVID-19 Sentiments dataset is more efficient than the application on Sentiment140 datasets because the Sentiment140 dataset contains a big amount of data. Since the Bag-Of-Words extractor method achieved 66.32 % in the accuracy rate after it implemented on COVID-19 Sentiments dataset, and it achieved 37.53 % in the accuracy rate after it applied on Sentiment140 datasets. This experiment proved that the Bag-Of-Words extractor method is not scalable. In addition, the Bag-Of-Words extractor method is more efficient than the N-grams extractor method with N-grams = 5 as shown in Fig. 12, 13, and 14.

Fig. 15 introduces the accuracy of opinion mining task obtained by the implementation of TF-IDF extractor on the Sentiment140 and COVID-19 Sentiments datasets using the Hadoop framework. As can be seen in Fig. 15, we notice that the TF-IDF extractor achieved good performance result in both COVID-19 Sentiments and Sentiment140 datasets. Since it achieved an accuracy equal to 74.67 % after it applied on COVID-19 Sentiments dataset, and it achieved

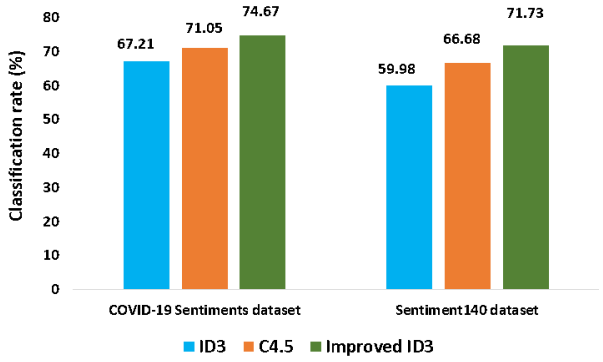


FIGURE 15. Accuracy of opinion mining task obtained by the implementation of TF-IDF extractor on the COVID-19 Sentiments and Sentiment140 datasets using the Hadoop framework.

an accuracy equal to 71.73 % after it applied on Sentiment140 dataset. Then, we remark that the TF-IDF extractor attained an approximate accuracy rate (74.67 %; 71.73 %) when it applied on both used datasets. Thence this TF-IDF extractor is scalable compared to N-grams and Bag-Of-Words extractor methods.

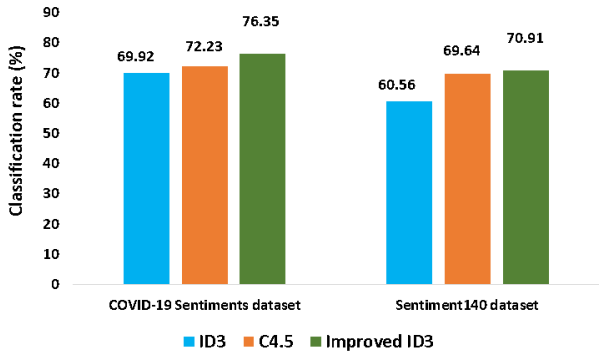


FIGURE 16. Accuracy of opinion mining task obtained by the implementation of GloVe extractor on the COVID-19_Sentiments and Sentiment140 datasets using the Hadoop framework.

Fig. 16 shows the classification rate of sentiment classification tasks obtained by the application of GloVe extractor method on the COVID-19_Sentiments and Sentiment140 datasets using the Hadoop framework. As can be seen in Fig. 16 we deduce that the GloVe attained good classification performance on both COVID-19_Sentiments and Sentiment140 datasets. Since it attained an accuracy equal to 76.35 % after it applied on COVID-19 Sentiments dataset, and it achieved an accuracy equal to 70.91 % after it applied on Sentiment140 dataset. Then, we remark that the GloVe extractor achieved an approximate accuracy rate (76.35 %; 70.91 %) when it applied on both used datasets. Therefore the GloVe extractor is scalable and its accuracy rate is approximated to the obtained TF-IDF accuracy (74.67 %; 71.73 %) on both used datasets.

Fig. 17 presents the classification rate of sentiment classification tasks obtained by applying the Word2Vec

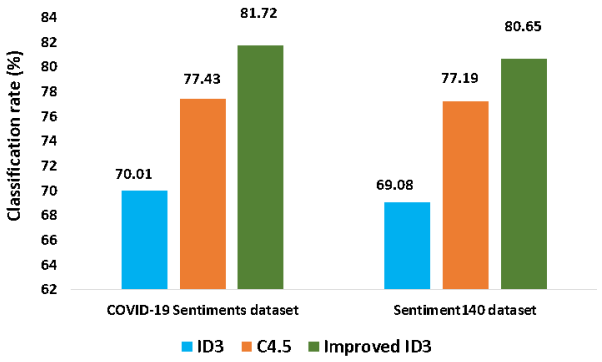


FIGURE 17. Accuracy of opinion mining task obtained by the implementation of Word2Vec extractor on the COVID-19_Sentiments and Sentiment140 datasets using the Hadoop framework.

extractor method on the COVID-19_Sentiments and Sentiment140 datasets using the Hadoop framework. As can be seen in Fig. 17, we notice that the Word2Vec extractor method is very efficient on a large dataset. Since it achieved 80.65 % when it applied on the Sentiment140 dataset and 81.72 % when it applied on COVID-19_Sentiments. Therefore, the Word2Vec extractor method is scalable, and it outperforms the N-grams, Bag-Of-Words, TF-IDF, and GloVe extractor methods in terms of classification rate. Until present, the most efficient extractor method is Word2Vec. The next experiment aims to compare the Word2Vec with FastText extractor method and find out to most efficient among them.

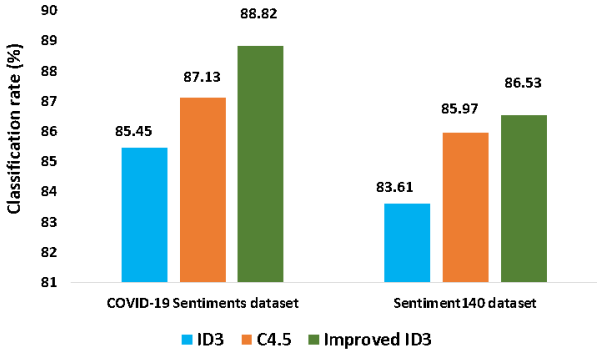


FIGURE 18. Accuracy of opinion mining task obtained by the implementation of FastText extractor on the COVID-19_Sentiments and Sentiment140 datasets using the Hadoop framework.

Fig. 18 shows the classification rate of sentiment classification tasks obtained by applying the FastText extractor method on the COVID-19_Sentiments and Sentiment140 datasets using the Hadoop framework. As can be seen in Fig. 18, we observe that the FastText gives a good classification rate on both used datasets compared to all used extractor methods (N-grams, Bag-Of-Words, TF-IDF, GloVe, and Word2Vec). Since it achieved 86.53 % when it applied on the Sentiment140 dataset and 88.82 % when it applied on COVID-19_Sentiments, according to all comparative studies performed in this subsection "Experiment 2", we deduce that the FastText

TABLE 11. Accuracy achieved by applying ID3, C4.5, and our improved ID3 decision tree algorithms using different feature selectors.

	Sentiment140			COVID-19_Sentiments		
	ID3	C4.5	Improved ID3	ID3	C4.5	Improved ID3
Chi-square	67.14	70.91	73.35	76.45	77.95	79.20
Gain Ratio	75.70	77.26	80.16	84.67	86.51	88.13
Information gain	83.61	85.97	86.53	85.45	87.13	88.82
Gini index	57.02	59.41	60.62	67.59	72.32	74.70

outperforms all other methods. So in the rest of this work, we will use only the FastText as a data representation method.

2) EXPERIMENT 3

As we introduced previously, the next phase after the extraction feature phase is the selection feature stage, in which we have applied many techniques, including Chi-square, Information Gain, Gain Ratio, and Gini Index. Therefore, this experiment aims to find the best feature selector among all used selection methods in terms of accuracy.

Table 11 describes the accuracy achieved by applying ID3, C4.5, and our improved ID3 decision tree algorithms on both Sentiment140 and COVID-19_Sentiments datasets using different feature selection techniques presented previously. As can be seen in Table 11, we remark that the IG used as a feature selector outperforms other feature selectors in terms of accuracy since it achieved an accuracy equal to 86.53 % by applying our improved ID3 on Sentiment140 and accuracy equal to 88.82 % by applying the improved ID3 on COVID-19_Sentiments. Therefore in the remainder of this contribution, we will only use the IG as a feature selector. After multiple experiments-we deduce that in the data representation stage - the **FastText** technique outperforms all other methods (N-grams, Bag-Of-Words, TF-IDF, GloVe, Word2Vec). It achieved an accuracy equal to 88.82 % when applied on COVID-19_Sentiments data and accuracy equal to 86.53 % when applied on the Sentiment140 dataset. Then in the feature selection phase, The empirical results proved that the **Information Gain** used as a feature selection method outperforms all other feature selectors (Gini Index, Gain Ratio, and Chi-square.) in terms of accuracy since it achieved an accuracy equal to 86.53 % when applied on Sentiment140 and accuracy equal to 88.82 % when applied on COVID-19_Sentiments. Finally, we have applied our improved ID3 [33] as a classifier. Fig. 19 depicts the final structure of our suggested classifier:

3) EXPERIMENT 4

In this experiment, we are going to introduce the empirical outcomes of our proposed classifier. These empirical results are achieved by practicing our proposed classifier and other methods such as ID3, C4.5, Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58], on both chosen datasets as described in subsection "Data Collection Stage". To demonstrate which

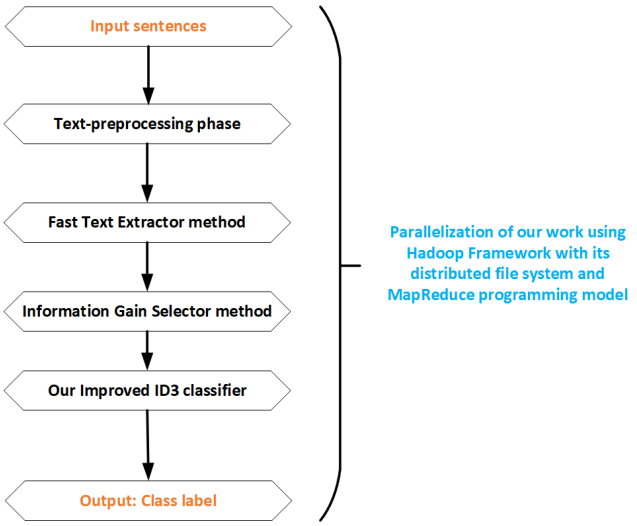


FIGURE 19. Final structure of our suggested classifier using the Hadoop framework.

of these methods is more efficient and has better performance, we measure nine evaluation metrics outlined previously in Table 7. Our classifier will be performed parallelly, employing the Hadoop framework with the Hadoop distributed file system and the MapReduce programming paradigm. The Hadoop cluster comprises four salve nodes and one master node.

Fig. 20, depicts the experimental result achieved for the classification rate applying our suggested classifier on Sentiment140 and COVID-19_Sentiments datasets, and we compare the empirical outcome reached with other approaches like ID3, C4.5, Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58]. As can be seen in Fig. 20, we remark that our proposed classifier outperforms the other implemented approaches in terms of classification rate. As the Fig. 20 depicted, our proposed classifier achieved a higher accuracy equal to 86.53 %, and 88.82 % when it applied on Sentiment140 and COVID-19_Sentiments datasets, respectively. And the AitAddi *et al.* [56] has the lower accuracy equal to 43.02 %, and 31.08 % in this experiment.

Fig. 21, illustrates the empirical result obtained for the error rate applying our suggested classifier and other previously chosen approaches on Sentiment140 and COVID-19_Sentiments datasets. As can be seen in Fig. 21,we notice

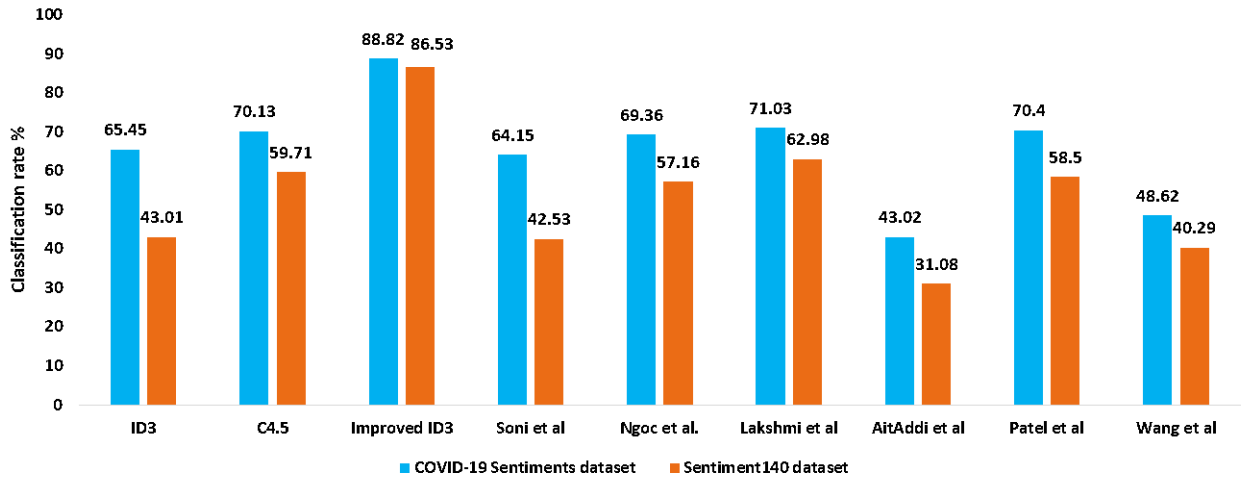


FIGURE 20. Classification rate obtained by implementing our classifier and other approaches.

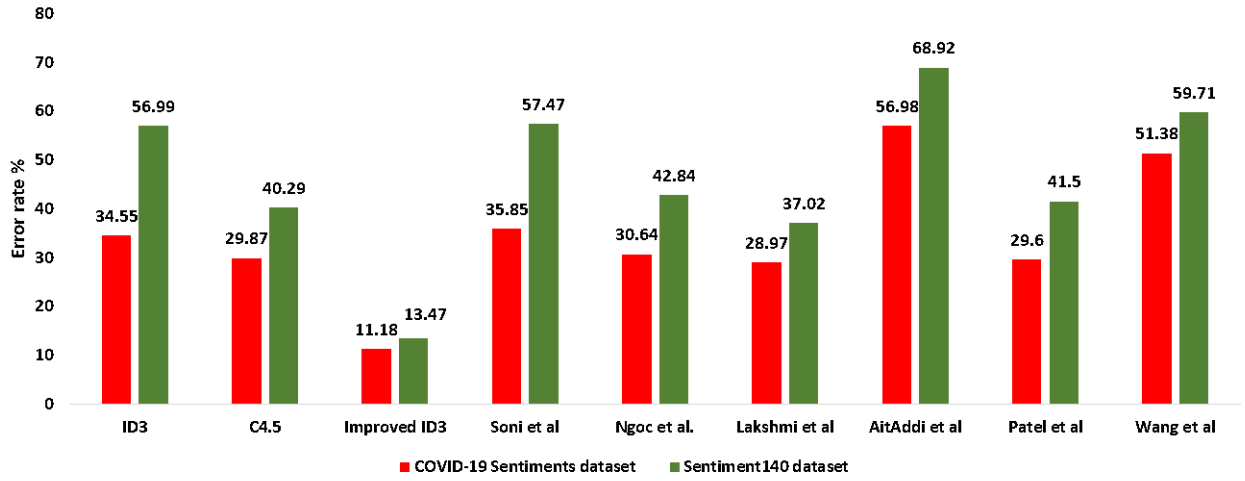


FIGURE 21. Error rate obtained by implementing our classifier and other approaches.

that our proposed classifier has lower error rate compared to other implemented approaches. And if we compared our classifier with AitAddi *et al.* [56] approach, we note that our classifier reduce the error rate from 56.98 %, and 68.92 % (AitAddi *et al.* [56]) to 11.18 %, and 13.47 % (our improved ID3) for the Sentiment140, and COVID-19_Sentiments datasets respectively.

Another experiment is conducted to examine the execution time between our classifier and the other chosen methods. Fig. 22 displays the experimental result reached after applying all proposals on both elected datasets. Without forgetting that our classifier is executed in a parallel mode on five computers utilizing framework Hadoop. As can be seen in Fig. 22, we observe that our developed model has a lower execution time rate compared to other implemented methods. And if we compared our classifier with the Patel *et al.* [57] method, we remark that our classifier reduces the execution time from 5402.15 s (Patel approach) to 45.21 s (our improved ID3) for the Sentiment140 dataset and from 842.91 s (Patel approach) to 15.95 s (our improved ID3) for the COVID-19_ Sentiments dataset. The comparison between ID3 and our classifier confirms that the implementation of our classifier using the Hadoop cluster of five machines is a more efficient tool to reduce the consumption time.

For more proving the performance of our classifier, we have computed other evaluation measures such as TPR, FNR, TNR, FPR, PR, KS, and FS as previously presented in Table 7. Table 12 depicts the experimental result reached.

As can be seen in the Table 12, we deduce that our classifier outperforms all other classifiers applied on both used datasets COVID-19_Sentiments and Sentiment140. Our classifier achieved higher values at the level of all computed evaluation measures.

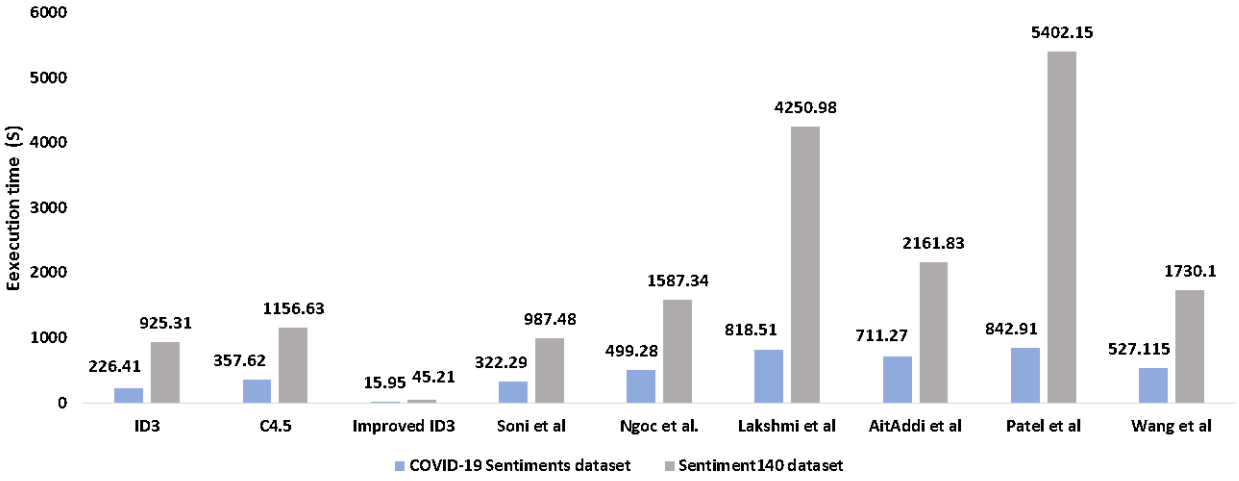


FIGURE 22. Execution time obtained by implementing our classifier and other approaches.

TABLE 12. Experimental outcome of TPR, FNR, TNR, FPR, PR, KS, and FS.

		TPR	FNR	TNR	FPR	PR	KS	FS
COVID-19_Sentiments dataset	Our classifier	85.72	14.28	86.51	13.49	86.67	87.69	85.54
	Soni *et al.* [39]	64.57	35.43	59.12	40.88	60.76	61.89	60.48
	Ngoc *et al.* [40]	69.08	30.92	67.15	32.85	66.51	65.70	66.84
	Lakshmi *et al.* [51]	71.46	28.54	70.89	29.11	72.32	73.51	70.54
	AitAddi *et al.* [56]	43.54	56.46	44.62	55.38	42.27	43.25	42.92
	Patel *et al.* [57]	71.92	28.08	70.52	29.48	72.69	70.56	71.96
	Wang *et al.* [58]	49.11	50.89	48.17	51.83	47.92	49.43	48.69
Sentiment140 dataset	Our classifier	81.41	18.59	82.33	17.67	83.04	84.58	83.87
	Soni *et al.* [39]	43.56	56.44	42.71	57.29	40.19	41.79	42.64
	Ngoc *et al.* [40]	56.24	43.76	57.19	42.81	56.92	54.61	55.24
	Lakshmi *et al.* [51]	63.78	36.22	64.50	35.5	63.29	62.26	64.73
	AitAddi *et al.* [56]	32.91	67.09	33.13	66.87	32.92	34.51	33.85
	Patel *et al.* [57]	58.27	41.73	59.62	40.38	57.92	58.41	57.86
	Wang *et al.* [58]	40.91	59.09	43.78	56.22	41.92	40.54	42.81

4) EXPERIMENT 5

In this Fifth experience, we have assessed the performance of the designed classifier by computing of stability, convergence, and complexity. The main objective of this last experiment is comparing our suggested classifier with ID3, C4.5, Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58], and to find out the most effective classifier among all assessed classifiers in terms of stability, complexity, and convergence.

5) COMPLEXITY

By definition, the complexity rate of a classifier is a criterion to measure the space employing and time consuming by a classifier. in this experiment we have measured the space complexity and time complexity of our suggested classifier, Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58]. In summary, Table 13 describes the obtained the space complexity results after we computed the size of executing

TABLE 13. Space complexity of the designed classifier, Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58] techniques.

Name of Dataset	Techniques	No. instructions	No. parameters
COVID-19_Sentiments	Our classifier	12.60 M	9.57 M
	Soni *et al.* [39]	13.51 M	11.03 M
	Ngoc *et al.* [40]	12.95 M	10.23 M
	Lakshmi *et al.* [51]	16.73 M	14.36 M
	AitAddi *et al.* [56]	15.41 M	12.98 M
	Patel *et al.* [57]	19.62 M	17.19 M
	Wang *et al.* [58]	17.25 M	15.48 M
Sentiment140	Our classifier	29.106 M	20.192 M
	Soni *et al.* [39]	32.71 M	26.58 M
	Ngoc *et al.* [40]	31.21 M	24.65 M
	Lakshmi *et al.* [51]	38.64 M	32.88 M
	AitAddi *et al.* [56]	35.59 M	29.72 M
	Patel *et al.* [57]	45.32 M	39.36 M
	Wang *et al.* [58]	39.84 M	35.45 M

TABLE 14. Time complexity of the designed classifier, Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58] techniques.

Name of Datasets	Techniques	Training time	Testing time
COVID-19_Sentiments	Our classifier	11.96 s	3.98 s
	Soni *et al.* [39]	241.71 s	80.57 s
	Ngoc *et al.* [40]	374.46 s	124.82 s
	Lakshmi *et al.* [51]	613.88 s	204.62 s
	AitAddi *et al.* [56]	533.45 s	177.81 s
	Patel *et al.* [57]	632.18 s	210.72 s
	Wang *et al.* [58]	395.33 s	131.77 s
Sentiment140	Our classifier	33.90 s	11.30 s
	Soni *et al.* [39]	740.61 s	246.87 s
	Ngoc *et al.* [40]	1190.50 s	396.83 s
	Lakshmi *et al.* [51]	3250.98 s	1062.74 s
	AitAddi *et al.* [56]	1621.37 s	540.45 s
	Patel *et al.* [57]	4051.61 s	1350.53 s
	Wang *et al.* [58]	1297.57 s	432.52 s

instructions, and the size of the classifier parameters of the suggested classifier and other selected classifiers.

As can be seen in Table 13, we remark that the suggested classifier has carried out several instructions which are occupied a memory size equal to (29.106 M, 12.60 M) for Sentiment140 and COVID-19_Sentiments datasets, respectively. The size of the designed classifier' parameters is equal to (20.192 M, 9.57 M) for Sentiment140 dataset and COVID-19_Sentiments, respectively. As the empirical result offers, our suggested classifier needs much lower space computational complexity if we compare it with Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58] methods.

Table 14 presents the obtained time complexity results after measuring the training and testing time consuming of our suggested classifier and other selected classifiers.

As can be seen in Table 14, we remark that the suggested classifier has expended a training time equal to 33.90 s, and 11.26 s for Sentiment140 and COVID-19_Sentiments datasets, respectively. Also our suggested classifier has consumed a testing time equal to 11.30 s, and 3.98 s for Sentiment140 and COVID-19_Sentiments datasets, respectively. As the obtained empirical consequence described, our suggested classifier consumes much lower time computational complexity if we compare it with Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58] approaches.

D. CONVERGENCE

the suggested classifier will be demonstrated if it is convergent or not convergent by finding a particular number of training rounds in which the proposed classifier meets the condition depicted in equation 14. This equation defines the condition of the convergent trend:

$$E_{\mathrm{rp}} - E_{\mathrm{rc}} \geq T_{\mathrm{re}} \tag{14}$$

where E_{rp} is our classifier average error of the former learning round, E_{rc} is the classifier average inaccuracy of the current learning round, and T_{re} is the sill value that set the convergence rate value, and after we carried out several experiments, we fixed this sill rate to 0.0001. Therefore, the suggested classifier average inaccuracy is measured using the following

equation:

$$E = \frac{1}{2} \times \frac{\sum_{j=1}^{S} \sum_{i=1}^{C} (y - y_{\text{label}})^2}{S} \tag{15}$$

where S is the total number of instances in the used dataset, C is the total number of classifier class labels (in our case there three class labels which are negative, Neutral, and positive), y is the wanted classification decision in the output, and y_{label} the obtained output classification decision. If the previously described equation (14) is verified, our suggested classifier can be considered converging, and the algorithm is executed till the classifier's average inaccuracy meets the condition. Oppositely, our suggested classifier is not converging.

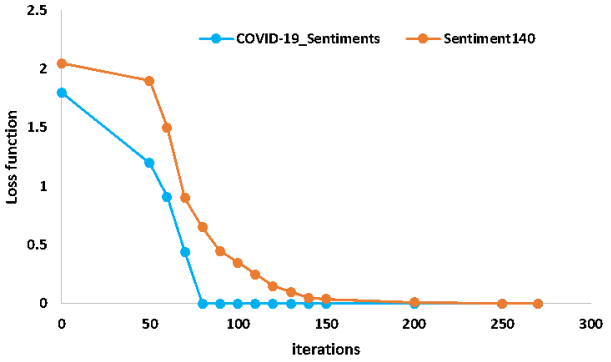


FIGURE 23. Convergence rate of our suggested classifier when it applied to COVID-19_Sentiments and Sentiment140 dataset.

Fig. 23 depicts our suggested classifier's convergence rate when it applied to Sentiment140 and COVID-19_Sentiments datasets. As can be seen in Fig. 23, we perceive that the suggested classifier converged towards the sill value 0.0001 after our classifier's algorithm reached 90 and 270 rounds when it applied to COVID-19_Sentiments and Sentiment140 dataset, respectively.

Table 15 introduces the convergence rate value of our suggested classifier, Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58] classifiers. As can be seen in Table 15, we conclude that our suggested classifier converges very speedy compared to other given classifiers.

E. STABILITY

In this step, we computed the mean standard deviation (MSD) of our classifier, Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58] classifiers over different five cross-validations of the used dataset. The main goal of this step is to determine the more stable classifier among all applied classifiers. Table 16 presents the obtained mean deviation standard and average accuracy (AVA) of our classifier compared to Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58] classifiers over the five cross-validations of both employed datasets in this contribution.

TABLE 15. Convergence rate of our proposed classifier, Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58] approaches.

Datasets	Algorithms	iterations
COVID-19_Sentiments	Our classifier	90
	Soni *et al.* [39]	150
	Ngoc *et al.* [40]	102
	Lakshmi *et al.* [51]	215
	AitAddi *et al.* [56]	184
	Patel *et al.* [57]	426
	Wang *et al.* [58]	357
Sentiment140	Our classifier	270
	Soni *et al.* [39]	461
	Ngoc *et al.* [40]	317
	Lakshmi *et al.* [51]	645
	AitAddi *et al.* [56]	552
	Patel *et al.* [57]	1278
	Wang *et al.* [58]	1071

TABLE 16. Stability of our classifier compared to Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58] classifiers over different five cross-validations.

Datasets	Algorithms	AVA (%)	MSD (%)
COVID-19_Sentiments	Our classifier	88.82	0.12
	Soni *et al.* [39]	64.15	1.95
	Ngoc *et al.* [40]	69.36	1.06
	Lakshmi *et al.* [51]	71.03	0.91
	AitAddi *et al.* [56]	43.02	4.08
	Patel *et al.* [57]	70.40	0.98
	Wang *et al.* [58]	48.62	3.56
Sentiment140	Our classifier	86.53	0.26
	Soni *et al.* [39]	42.53	3.45
	Ngoc *et al.* [40]	57.16	3.68
	Lakshmi *et al.* [51]	62.98	2.31
	AitAddi *et al.* [56]	31.08	7.82
	Patel *et al.* [57]	58.50	2.55
	Wang *et al.* [58]	40.29	4.48

As can be seen in Table 16, we remark that our classifier is more stable than other classifier because it achieved higher average accuracy (88.82 %, and 86.53 %) with a very low mean standard deviation (0.12 %, and 0.26 %) when it applied to COVID-19_Sentiments and Sentiment140 dataset, respectively.

V. CONCLUSION

In this study, we have proposed an innovative approach to classify tweets into positive, negative, or neutral based on social media Big Data. The suggested system consists of five parts: data collection, data preprocessing, data representation, data classification, feature selection, and application of the Hadoop framework. In the data collection phase we have chosen Sentiment140 and COVID-19_Sentiments datasets to evaluate our proposal. For the data preprocessing phase, we have applied multiple preprocessing tasks on both chosen datasets, and we carried out the first experiment to evaluate the effectiveness of the applied data preprocessing tasks on both used datasets. The experimental results show that the data preprocessing tasks have a significant

impact since they reduce the classification error rate from (39.59 %, and 30.04 %) to (13.47 %, and 11.18 %) for Sentiment140 and COVID-19_Sentiments respectively. Then, in data representation, we have applied several techniques to convert the textual data in numerical data, including N-grams or character-level, Bag-Of-Words, word embedding (GloVe, Word2Vec), FastText, and TF-IDF. Also, the second experiment is performed to evaluate the performance of each used method. The empirical results show that the accuracy of N-grams, Bag-Of-Words, TF-IDF, GloVe,Word2Vec, and FastText is equal to 54.99 %, 66.32 %, 74.67 %, 76.35 %, 81.72 %, and 88.82 %, which are obtained after the application of our improved ID3 on COVID-19_Sentiments dataset. Thus the most efficient method is the FastText extractor. After the feature extraction, the next phase is the feature selection in which we have applied Chi-Square, Information Gain, Gain Ratio, and Gini Index methods in order to reduce the dimensionality of the feature space. We carried out the third experiment to evaluate the used feature selectors. The experimental results show that Chi-Square, Gain Ratio, Information Gain, and Gini Index achieved an accuracy equal to 79.20 %, 88.13 %, 88.82 %, and 74.70 % respectively, which are obtained after the application of our improved ID3 on COVID-19_Sentiments dataset. Thence the empirical result proved that the information gain is the most effective feature selector. Finally we have applied our improved ID3 classifier on both used datasets and we obtained an accuracy equal 86.53 %, and 88.82 % for Sentiment140 and COVID-19_Sentiments respectively. Our proposal is parallelized using Hadoop Framework (MapReduce + HDFS). In the last experiment we have compared our proposal with other approaches like ID3, C4.5, Soni *et al.* [39], Ngoc *et al.* [40], Lakshmi *et al.* [51], AitAddi *et al.* [56], Patel *et al.* [57], and Wang *et al.* [58]. The experimental outcome show that our proposal outperforms all other classifiers applied on both used datasets COVID-19_Sentiments and Sentiment140 in terms of Recall, specificity, false-positive rate, false-negative rate, error rate, precision rate, classification rate, kappa statistic, F1-score, execution time, convergence, stability, and complexity.

Our future work is to merge the fuzzy logic theory and our proposal in this paper in order to handle with continuous-valued features, taking into consideration various parameters concerning the feature extractors and feature selectors. Utilization of Mamdani fuzzy system as a classifier for sentiment analysis in order to deal with uncertainty and vagueness sentiments held in the data expressed by social media users. Integration of fuzzy rule-based model with our MapReduce improved ID3 decision tree for inferring the sentiment expressed in speech cues on the social media networks.

REFERENCES

[1] D. Tang, B. Qin, and T. Liu, "Deep learning for sentiment analysis: Successful approaches and future challenges," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 5, no. 6, pp. 292–303, Nov. 2015, doi: 10.1002/widm.1171.

[2] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2015, pp. 959–962, doi: 10.1145/2766462.2767830.

[3] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, pp. 82–89, Apr. 2013, doi: 10.1145/2436256.2436274.

[4] R. Sharma, S. Nigam, and R. Jain, "Opinion mining of movie reviews at document level," *Int. J. Inf. Theory*, vol. 3, no. 3, pp. 13–21, Jul. 2014, doi: 10.5121/ijit.2014.3302.

[5] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using N-gram machine learning approach," *Expert Syst. Appl.*, vol. 57, pp. 117–126, Sep. 2016, doi: 10.1016/j.eswa.2016.03.028.

[6] K. K. Pawar, P. P. Shrishrimal, and R. R. Deshmukh, "Twitter sentiment analysis: A review," *Int. J. Sci. Eng. Res.*, vol. 6, no. 4, pp. 957–964, 2015, Mar. 20, 2021. [Online]. Available: http://www.ijser.org/

[7] H. Lu and J. C.-C. Lin, "Predicting customer behavior in the market-space: A study of Rayport and Sviokla's framework," *Inf. Manag. J.*, vol. 40, no. 1, pp. 1–10, Oct. 2002, doi: 10.1016/S0378-7206(01)00131-8.

[8] J. R. Saura, P. R. Palos-Sanchez, and M. B. Correia, "Digital marketing strategies based on the E-business model: Literature review and future directions," in *Organizational Transformation and Managing Innovation in the Fourth Industrial Revolution*, A. G. Guerra, Ed. Hershey, PA, USA: IGI Global, 2019, pp. 86–103, doi: 10.4018/978-1-5225-7074-5.ch005.

[9] Z. Kechaou, M. Ben Ammar, and A. M. Alimi, "Improving E-learning with sentiment analysis of users' opinions," in *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, Apr. 2011, pp. 1032–1038, doi: 10.1109/EDUCON.2011.5773275.

[10] T. H. Nguyen and K. Shirai, "Topic modeling based sentiment analysis on social media for stock market prediction," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process. (Long Papers)*, vol. 1, 2015, pp. 1354–1364, doi: 10.3115/v1/P15-1131.

[11] Y. Lin. (2020). *10 Twitter Statistics Every Marketer Should Know in 2020 [Infographic]*. Marketing Your Store. Accessed: Mar. 20, 2021. [Online]. Available: https://www.oberlo.com/blog/twitter-statistics

[12] D. Jiang, X. Luo, J. Xuan, and Z. Xu, "Sentiment computing for the news event based on the social media big data," *IEEE Access*, vol. 5, pp. 2373–2382, 2017, doi: 10.1109/ACCESS.2016.2607218.

[13] M. U. Salur and I. Aydin, "A novel hybrid deep learning model for sentiment classification," *IEEE Access*, vol. 8, pp. 58080–58093, 2020, doi: 10.1109/ACCESS.2020.2982538.

[14] F. Zhu and X. Zhang, "Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics," *J. Marketing*, vol. 74, no. 2, pp. 133–148, Mar. 2010, doi: 10.1509/jm.74.2.133.

[15] J. Han, M. Kamber, and J. Pei, "Getting to know your data," in *Data Mining: Concepts and Techniques*, 3th ed. San Francisco, CA, USA: Morgan Kaufmann, 006, pp. 39–78.

[16] N. O. F. Daeli and A. Adiwijaya, "Sentiment analysis on movie reviews using information gain and K-nearest neighbor," *J. Data Sci. Appl.*, vol. 3, no. 1, pp. 1–7, May 2020, doi: 10.34818/jdsa.2020.3.22.

[17] N. Zainuddin and A. Selamat, "Sentiment analysis using support vector machine," in *Proc. Int. Conf. Comput., Commun., Control Technol. (I4CT)*, Sep. 2014, pp. 333–337, doi: 10.1109/I4CT.2014.6914200.

[18] M. Rathi, A. Malik, D. Varshney, R. Sharma, and S. Mendiratta, "Sentiment analysis of tweets using machine learning approach," in *Proc. 11th Int. Conf. Contemp. Comput. (IC3)*, Aug. 2018, pp. 1–3, doi: 10.1109/IC3.2018.8530517.

[19] W. P. Ramadhan, S. T. M. T. A. Novianty, and S. T. M. T. C. Setianingsih, "Sentiment analysis using multinomial logistic regression," in *Proc. Int. Conf. Control, Electron., Renew. Energy Commun. (ICCREC)*, Sep. 2017, pp. 46–49, doi: 10.1109/ICCEREC.2017.8226700.

[20] C. Troussas, M. Virvou, K. J. Espinosa, K. Llaguno, and J. Caro, "Sentiment analysis of Facebook statuses using naive Bayes classifier for language learning," in *Proc. IISA*, Jul. 2013, pp. 1–6, doi: 10.1109/IISA.2013.6623713.

[21] Y. Al Amrani, M. Lazaar, and K. E. El Kadiri, "Random forest and support vector machine based hybrid approach to sentiment analysis," *Procedia Comput. Sci.*, vol. 127, pp. 511–520, Jan. 2018, doi: 10.1016/j.procs.2018.01.150.

[22] P. S. Patil and N. V. Dharwadkar, "Analysis of banking data using machine learning," in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Analytics Cloud (I-SMAC)*, Feb. 2017, pp. 876–881, doi: 10.1109/I-SMAC.2017.8058305.

[23] P. Burnap and M. L. Williams, "Us and them: Identifying cyber hate on Twitter across multiple protected characteristics," *EPJ Data Sci.*, vol. 5, no. 1, pp. 1–15, Dec. 2016, doi: 10.1140/epjds/s13688-016-0072-6.

[24] P. Larrañaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armañanzas, G. Santafé, A. Pérez, and V. Robles, "Machine learning in bioinformatics," *Briefings Bioinf.*, vol. 7, no. 1, pp. 86–112, Mar. 2006, doi: 10.1093/bib/bbk007.

[25] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proc. 25th Int. Conf. World Wide Web*, Apr. 2016, pp. 145–153, doi: 10.1145/2872427.2883062.

[26] I. Habernal, T. Ptáček, and J. Steinberger, "Sentiment analysis in czech social media using supervised machine learning," in *Proc. 4th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, Jun. 2013, pp. 65–74. [Online]. Available: https://www.aclweb.org/anthology/W13-1609

[27] K. Reynolds, A. Kontostathis, and L. Edwards, "Using machine learning to detect cyberbullying," in *Proc. 10th Int. Conf. Mach. Learn. Appl. Workshops*, Dec. 2011, pp. 241–244, doi: 10.1109/ICMLA.2011.152.

[28] M. S. Neethu and R. Rajasree, "Sentiment analysis in Twitter using machine learning techniques," in *Proc. 4th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2013, pp. 1–5, doi: 10.1109/ICCCNT.2013.6726818.

[29] E. Boiy and M.-F. Moens, "A machine learning approach to sentiment analysis in multilingual Web texts," *Inf. Retr.*, vol. 12, no. 5, pp. 526–558, Oct. 2009, doi: 10.1007/s10791-008-9070-z.

[30] G. Gautam and D. Yadav, "Sentiment analysis of Twitter data using machine learning approaches and semantic analysis," in *Proc. 7th Int. Conf. Contemp. Comput. (IC3)*, Aug. 2014, pp. 437–442, doi: 10.1109/IC3.2014.6897213.

[31] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine learning-based sentiment analysis for Twitter accounts," *Math. Comput. Appl.*, vol. 23, no. 1, p. 11, Feb. 2018, doi: 10.3390/mca23010011.

[32] A. Sharma and S. Dey, "A comparative study of feature selection and machine learning techniques for sentiment analysis," in *Proc. ACM Res. Appl. Comput. Symp. (RACS)*, Oct. 2012, pp. 1–7, doi: 10.1145/2401603.2401605.

[33] F. Es-SABERY and A. Hair, "An improved ID3 classification algorithm based on correlation function and weighted attribute," in *Proc. Int. Conf. Intell. Syst. Adv. Comput. Sci. (ISACS)*, Dec. 2019, pp. 1–8, doi: 10.1109/ISACS48493.2019.9068891.

[34] G. Patil, V. Galande, V. Kekan, and K. Dange, "Sentiment analysis using support vector machine," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 2, no. 1, pp. 2607–2612, 2014. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1084.6923&rep=rep1&type=pdf

[35] K. P. P. Shein and T. T. S. Nyunt, "Sentiment classification based on ontology and SVM classifier," in *Proc. 2nd Int. Conf. Commun. Softw. Netw.*, Feb. 2010, pp. 169–172, doi: 10.1109/ICCSN.2010.35.

[36] P. Gamallo, M. Garcia, and S. Fernandez-Lanza, "TASS: A Naive-Bayes strategy for sentiment analysis on Spanish tweets," in *Proc. 29th Congreso de la Sociedad Española de Procesamiento de Lenguaje Natural*, Mar. 2013, pp. 126–132. [Online]. Available: https://gramatica.usc.es/~gamallo/artigos-web/TASS2013.pdf

[37] M. Anjaria and R. M. R. Guddeti, "A novel sentiment analysis of social networks using supervised learning," *Social Netw. Anal. Mining*, vol. 4, no. 1, p. 181, Dec. 2014, doi: 10.1007/s13278-014-0181-9.

[38] R. M. Duwairi, "Sentiment analysis for dialectical arabic," in *Proc. 6th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2015, pp. 166–170, doi: 10.1109/IACS.2015.7103221.

[39] V. K. Soni and S. Pawar, "Emotion based social media text classification using optimized improved ID3 classifier," in *Proc. Int. Conf. Energy, Commun., Data Anal. Soft Comput. (ICECDS)*, Aug. 2017, pp. 1500–1505, doi: 10.1109/ICECDS.2017.8389696.

[40] P. V. Ngoc, C. V. T. Ngoc, T. V. T. Ngoc, and D. N. Duy, "A C4.5 algorithm for english emotional classification," *Evol. Syst.*, vol. 10, no. 3, pp. 425–451, Sep. 2019, doi: 10.1007/s12530-017-9180-1.

[41] L. Yuxun and X. Niuniu, "Improved ID3 algorithm," in *Proc. 3rd Int. Conf. Comput. Sci. Inf. Technol.*, Jul. 2010, pp. 465–468, doi: 10.1109/ICCSIT.2010.5564765.

[42] R.-M. Chai and M. Wang, "A more efficient classification scheme for ID3," in *Proc. 2nd Int. Conf. Comput. Eng. Technol.*, 2010, pp. 329–332, doi: 10.1109/ICCET.2010.5486128.

[43] S. Elyassami and A. Idri, "Applying fuzzy ID3 decision tree for software effort estimation," 2011, *arXiv:1111.0158*. [Online]. Available: http://arxiv.org/abs/1111.0158

[44] K. Zou, W. Sun, H. Yu, and F. Liu, "ID3 decision tree in fraud detection application," in *Proc. Int. Conf. Comput. Sci. Electron. Eng.*, Mar. 2012, pp. 399–402, doi: 10.1109/ICCSEE.2012.241.

[45] V. Srinivasan, G. Rajenderan, J. Vandar Kuzhali, and M. Aruna, "Fuzzy fast classification algorithm with hybrid of ID3 and SVM," *J. Intell. Fuzzy Syst.*, vol. 24, no. 3, pp. 555–561, 2013, doi: 10.3233/IFS-2012-0574.

[46] X. J. Chen, Z. G. Zhang, and Y. Tong, "An improved ID3 decision tree algorithm," *Adv. Mater. Res.*, vols. 962–965, pp. 2842–2847, Jun. 2014, doi: 10.4028/www.scientific.net/AMR.962-965.2842.

[47] B. Ding, Y. Zheng, and S. Zang, "A new decision tree algorithm based on rough set theory," in *Proc. Asia–Pacific Conf. Inf. Process.*, Jul. 2009, pp. 326–329, doi: 10.1109/APCIP.2009.216.

[48] L. Zhu and Y. Yang, "Improvement of decision tree ID3 algorithm," in *Collaborate Computing: Networking, Applications and Worksharing* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 201. Cham, Switzerland: Springer, 2017, pp. 595–600, doi: 10.1007/978-3-319-59288-6_59.

[49] N. Kaewrod and K. Jearanaitanakij, "Improving ID3 algorithm by ignoring minor instances," in *Proc. 22nd Int. Comput. Sci. Eng. Conf. (ICSEC)*, Nov. 2018, pp. 1–5, doi: 10.1109/ICSEC.2018.8712762.

[50] A. Rajeshkanna and K. Arunesh, "ID3 decision tree classification: An algorithmic perspective based on error rate," in *Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*, Jul. 2020, pp. 787–790, doi: 10.1109/ICESC48915.2020.9155578.

[51] B. L. Devi, V. V. Bai, S. Ramasubbareddy, and K. Govinda, "Sentiment analysis on movie reviews," in *Advances in Intelligent Systems and Computing*, vol. 1054. Singapore: Springer, 2020, pp. 321–328, doi: 10.1007/978-981-15-0135-7_31.

[52] J. Guerreiro and P. Rita, "How to predict explicit recommendations in online reviews using text mining and sentiment analysis," *J. Hospitality Tourism Manage.*, vol. 43, pp. 269–272, Jun. 2020, doi: 10.1016/j.jhtm.2019.07.001.

[53] R. P. Mehta, M. A. Sanghvi, D. K. Shah, and A. Singh, "Sentiment analysis of tweets using supervised learning algorithms," in *Proc. 1st Int. Conf. Sustain. Technol. Comput. Intell.*, Singapore, vol. 1045, 2020, pp. 323–338, doi: 10.1007/978-981-15-0029-9_26.

[54] J. Zhang, *Sentiment Analysis of Movie Reviews in Chinese*. Uppsala, Sweden: Uppsala Univ., Mar. 2021. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1438431/FULLTEXT01.pdf

[55] A. López-Chau, D. Valle-Cruz, and R. Sandoval-Almazán, "Sentiment analysis of Twitter data through machine learning techniques," in *Software Engineering in the Era of Cloud Computing* (Computer Communications and Networks), M. Ramachandran and Z. Mahmood, Eds. Cham, Switzerland: Springer, 2020, pp. 185–209, doi: 10.1007/978-3-030-33624-0_8.

[56] H. A. Addi, R. Ezzahir, and A. Mahmoudi, "Three-level binary tree structure for sentiment classification in arabic text," in *Proc. 3rd Int. Conf. Netw., Inf. Syst. Secur.*, Mar. 2020, pp. 1–8, doi: 10.1145/3386723.3387844.

[57] R. Patel and K. Passi, "Sentiment analysis on Twitter data of world cup soccer tournament using machine learning," *IoT*, vol. 1, no. 2, pp. 218–239, Oct. 2020, doi: 10.3390/iot1020014.

[58] Y. Wang, Q. Chen, J. Shen, B. Hou, M. Ahmed, and Z. Li, "Aspect-level sentiment analysis based on gradual machine learning," *Knowl.-Based Syst.*, vol. 212, pp. 106509–106521, Jan. 2021, doi: 10.1016/j.knosys.2020.106509.

[59] A. Baccouche, B. Garcia-Zapirain, and A. Elmaghraby, "Annotation technique for health-related tweets sentiment analysis," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, Dec. 2018, pp. 382–387, doi: 10.1109/ISSPIT.2018.8642685.

[60] Z. Hameed and B. Garcia-Zapirain, "Sentiment classification using a single-layered BiLSTM model," *IEEE Access*, vol. 8, pp. 73992–74001, 2020, doi: 10.1109/ACCESS.2020.2988550.

[61] Z. Hameed, B. Garcia-Zapirain, and I. O. Ruiz, "A computationally efficient BiLSTM based approach for the binary sentiment classification," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, Dec. 2019, pp. 1–4, doi: 10.1109/ISSPIT47144.2019.9001781.

[62] M. Zhang, "E-commerce comment sentiment classification based on deep learning," in *Proc. IEEE 5th Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2020, pp. 184–187, doi: 10.1109/ICCCBDA49378.2020.9095734.

[63] N. Jin, J. Wu, X. Ma, K. Yan, and Y. Mo, "Multi-task learning model based on multi-scale CNN and LSTM for sentiment classification," *IEEE Access*, vol. 8, pp. 77060–77072, 2020, doi: 10.1109/ACCESS.2020.2989428.

[64] A. Chugh, V. K. Sharma, S. Kumar, A. Nayyar, B. Qureshi, M. K. Bhatia, and C. Jain, "Spider monkey crow optimization algorithm with deep learning for sentiment classification and information retrieval," *IEEE Access*, vol. 9, pp. 24249–24262, 2021, doi: 10.1109/ACCESS.2021.3055507.

[65] H. Jelodar, Y. Wang, R. Orji, and S. Huang, "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 Informat.*, vol. 24, no. 10, pp. 2733–2742, Oct. 2020, doi: 10.1109/JBHI.2020.3001216.

[66] M. Dong, Y. Li, X. Tang, J. Xu, S. Bi, and Y. Cai, "Variable convolution and pooling convolutional neural network for text sentiment classification," *IEEE Access*, vol. 8, pp. 16174–16186, 2020, doi: 10.1109/ACCESS.2020.2966726.

[67] A. Kumar, V. T. Narapareddy, V. A. Srikanth, L. B. M. Neti, and A. Malapati, "Aspect-based sentiment classification using interactive gated convolutional network," *IEEE Access*, vol. 8, pp. 22445–22453, 2020, doi: 10.1109/ACCESS.2020.2970030.

[68] F. Es-Sabery, A. Hair, J. Qadir, B. Sainz-De-Abajo, B. Garcia-Zapirain, and I. Torre-Diez, "Sentence-level classification using parallel fuzzy deep learning classifier," *IEEE Access*, vol. 9, pp. 17943–17985, 2021, doi: 10.1109/ACCESS.2021.3053917.

[69] H. Sadr, M. M. Pedram, and M. Teshnehlab, "Multi-view deep network: A deep model based on learning features from heterogeneous neural networks for sentiment analysis," *IEEE Access*, vol. 8, pp. 86984–86997, 2020, doi: 10.1109/ACCESS.2020.2992063.

[70] M. A. El-Affendi, K. Alrajhi, and A. Hussain, "A novel deep learning-based multilevel parallel attention neural (MPAN) model for multidomain arabic sentiment analysis," *IEEE Access*, vol. 9, pp. 7508–7518, 2021, doi: 10.1109/ACCESS.2021.3049626.

[71] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Inf. Process. Manage.*, vol. 50, no. 1, pp. 104–112, Jan. 2014, doi: 10.1016/j.ipm.2013.08.006.

[72] S. Vijayarani, J. Ilamathi, and M. Nithya, "Preprocessing techniques for text mining—An overview," *Int. J. Comput. Sci. Commun. Netw.*, vol. 5, no. 1, pp. 7–16, 2015.

[73] J. Abel and W. Teahan, "Universal text preprocessing for data compression," *IEEE Trans. Comput.*, vol. 54, no. 5, pp. 497–507, May 2005, doi: 10.1109/TC.2005.85.

[74] Z. Yao and C. Ze-wen, "Research on the construction and filter method of stop-word list in text preprocessing," in *Proc. 4th Int. Conf. Intell. Comput. Technol. Autom.*, Mar. 2011, pp. 217–221, doi: 10.1109/ICICTA.2011.64.

[75] H. Kruse and A. Mukherjee, "Preprocessing text to improve compression ratios," in *Proc. Data Compress. Conf. (DCC)*, Mar. 1998, p. 556, doi: 10.1109/DCC.1998.672295.

[76] M. Anandarajan, C. Hill, and T. Nolan, "Text preprocessing," in *Practical Text Analytics* (Advances in Analytics and Data Science), M. Anandarajan, C. Hill, and T. Nolan, Eds. Cham, Switzerland:, 2019, pp. 45–59, doi: 10.1007/978-3-319-95663-3_4.

[77] O. Levy and Y. Goldberg, "Dependency-based word embeddings," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics (Short Papers)*, vol. 2, 2014, pp. 302–308, doi: 10.3115/v1/P14-2050.

[78] M. Ghiassi, J. Skinner, and D. Zimbra, "Twitter brand sentiment analysis: A hybrid system using N-gram analysis and dynamic artificial neural network," *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6266–6282, Nov. 2013, doi: 10.1016/j.eswa.2013.05.057.

[79] I. Santos, N. Nedjah, and L. de Macedo Mourelle, "Sentiment analysis using convolutional neural network with fastText embeddings," in *Proc. IEEE Latin Amer. Conf. Comput. Intell. (LA-CCI)*, Nov. 2017, pp. 1–5, doi: 10.1109/LA-CCI.2017.8285683.

[80] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: A statistical framework," *Int. J. Mach. Learn. Cybern.*, vol. 1, nos. 1–4, pp. 43–52, Dec. 2010, doi: 10.1007/s13042-010-0001-0.

[81] C. Lee and G. G. Lee, "Information gain and divergence-based feature selection for machine learning-based text categorization," *Inf. Process. Manage.*, vol. 42, no. 1, pp. 155–165, Jan. 2006, doi: 10.1016/j.ipm.2004.08.006.

[82] P. A. Estevez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 189–201, Feb. 2009, doi: 10.1109/TNN.2008.2005601.

[83] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, Oct. 2017, doi: 10.1016/j.jksuci.2015.12.004.

[84] N. Wang, P. Wang, and B. Zhang, "An improved TF-IDF weights function based on information theory," in *Proc. Int. Conf. Comput. Commun. Technol. Agricult. Eng.*, Jun. 2010, pp. 439–441, doi: 10.1109/CCTAE.2010.5544382.

[85] H. Liu, M. Zhou, X. S. Lu, and C. Yao, "Weighted gini index feature selection method for imbalanced data," in *Proc. IEEE 15th Int. Conf. Netw., Sens. Control (ICNSC)*, Mar. 2018, pp. 1–6, doi: 10.1109/ICNSC.2018.8361371.

[86] Z. Wei, D. Miao, J.-H. Chauchat, R. Zhao, and W. Li, "N-grams based feature selection and text representation for chinese text classification," *Int. J. Comput. Intell. Syst.*, vol. 2, no. 4, pp. 365–374, Dec. 2009, doi: 10.1080/18756891.2009.9727668.

[87] A. Kaur and V. Gupta, "N-gram based approach for opinion mining of Punjabi text," in *Multi-Disciplinary Trends in Artificial Intelligence* (Lecture Notes in Computer Science), vol. 8875. Cham, Switzerland: Springer, 2014, pp. 81–88, doi: 10.1007/978-3-319-13365-2_8.

[88] S. Salawu, Y. He, and J. Lumsden, "Approaches to automated detection of cyberbullying: A survey," *IEEE Trans. Affect. Comput.*, vol. 11, no. 1, pp. 3–24, Jan. 2020, doi: 10.1109/TAFFC.2017.2761757.

[89] I. Kanaris, K. Kanaris, I. Houvardas, and E. Stamatatos, "Words versus character N-grams for anti-spam filtering," *Int. J. Artif. Intell. Tools*, vol. 16, no. 6, pp. 1047–1067, Dec. 2007, doi: 10.1142/S0218213007003692.

[90] K. Luyckx and W. Daelemans, "Authorship attribution and verification with many authors and limited data," in *Proc. 22nd Int. Conf. Comput. Linguistics (COLING)*, 2008, pp. 513–520, doi: 10.3115/1599081.1599146.

[91] A. Barrón-Cedeño and P. Rosso, "On automatic plagiarism detection based on N-grams Comparison," in *Advances in Information Retrieval* (Lecture Notes in Computer Science), vol. 5478, 2009, pp. 696–700, doi: 10.1007/978-3-642-00958-7_69.

[92] T. Walkowiak, S. Datko, and H. Maciejewski, "Bag-of-words, bag-of-topics and word-to-vec based subject classification of text documents in polish—A comparative study," in *Advances in Intelligent Systems and Computing*, vol. 761. Cham, Switzerland: Springer, 2019, pp. 526–535, doi: 10.1007/978-3-319-91446-6_49.

[93] R. Uma and B. Latha, "An efficient voice based information retrieval using bag of words based indexing," *Int. J. Eng. Technol.*, vol. 7, no. 2, pp. 622–627, Jun. 2018, doi: 10.14419/ijet.v7i2.33.14850.

[94] R. Zhao and K. Mao, "Fuzzy bag-of-words model for document representation," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 794–804, Apr. 2018, doi: 10.1109/TFUZZ.2017.2690222.

[95] Y. Shao, S. Taylor, N. Marshall, C. Morioka, and Q. Zeng-Treitler, "Clinical text classification with word embedding features vs. Bag-of-words features," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 2874–2878, doi: 10.1109/BigData.2018.8622345.

[96] M. Gabryel, "The bag-of-words method with different types of image features and dictionary analysis," *J. Univers. Comput. Sci.*, vol. 24, no. 4, pp. 357–371, 2018, doi: 10.3217/jucs-024-04-0357.

[97] N. Cummins, S. Amiriparian, S. Ottl, M. Gerczuk, M. Schmitt, and B. Schuller, "Multimodal bag-of-words for cross domains sentiment analysis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4954–4958, doi: 10.1109/ICASSP.2018.8462660.

[98] P. Huilgol. (Mar. 20, 2021.) *Quick Introduction to Bag-of-Words (BoW) and TF-IDF for Creating Features From Text*. Analytics Vidhya.com. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-TF-IDF/

[99] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543, doi: 10.3115/v1/D14-1162.

[100] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013, *arXiv:1310.4546*. [Online]. Available: http://arxiv.org/abs/1310.4546

[101] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality reduction: A comparative review," *J. Mach. Learn. Res.*, vol. 10, nos. 1–41, pp. 66–71, Oct. 2009, doi: 10.1080/13506280444000102.

[102] F. Es-Sabery and A. Hair, "Big data solutions proposed for cluster computing systems challenges: A survey," in *Proc. 3rd Int. Conf. Netw., Inf. Syst. Secur.*, Mar. 2020, pp. 1–7, doi: 10.1145/3386723.3387826.
[103] F. Es-sabery and A. Hair, "A MapReduce C4.5 decision tree algorithm based on fuzzy rule-based system," *Fuzzy Inf. Eng.*, pp. 1–28, Jun. 2020, doi: 10.1080/16168658.2020.1756099.

FATIMA ES-SABERY received the Technical University degree from the Department of Computer Science, Higher School of Technology, Casablanca, Morocco, in 2013, the professional license with option IT development from the Department of Computer Sciences, Faculty of Science, Casablanca, in 2014, and the master's degree in business intelligence from the Department of Computer Sciences, Sultan Moulay Sliman University, Beni Mellal, Morocco, in 2016. She has published several research papers in many international conferences and journals, i.e., *Fuzzy Information and Engineering*, the *International Journal of Informatics and Communication Technology*, The 3rd International Conference on Networking, Information Systems & Security, and the 2019 International Conference on Intelligent Systems and Advanced Computing Sciences. Her general research interests include data mining area, big data field, wireless sensor networks, fuzzy systems, machine learning, deep learning, and the Internet of Things.

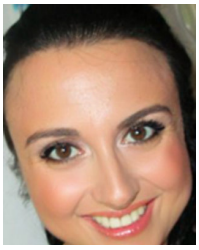
KHADIJA ES-SABERY received the Engineering degree from the Department of Computer Science, National School of Applied Sciences, Cadi Ayyad University, Marrakech, Morocco, in 2021. Her general interests include data mining area, big data field, wireless sensor networks, fuzzy systems, machine learning, deep learning, and the Internet of Things.

JUNAID QADIR received the M.Sc. degree in electronics from the Department of Electronics, University of Peshawar, in 2016, and the M.Phil. degree in electronics from Quaid-i-Azam University, Islamabad, Pakistan, in 2019. He is currently pursuing the Ph.D. degree with the Department of Electrical, Electronic and Telecommunications Engineering, and Naval Architecture (DITEN), University of Genova, Italy. He is also a Research Collaborator with the Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid, Spain. He has published many research articles in highly reputed international journals and conferences, such as IEEE Access, *Energies* (MDPI), *Journal of Intelligent & Fuzzy Systems*, the International Multitopic Conference (INMIC), and CISIS 2018, Matsue, Japan. His current research interests include underwater wireless sensor networks, wireless sensor networks, the Internet of Things, mobile edge computing (MEC), machine learning, and 5G mobile communication. He is a verified reviewer with a number of prestigious international publishers, such as IEEE Access, the IEEE Sensors Journal, the *International Journal of Distributed Sensor Networks* (IJDSN), *Journal of King Saud University* (Elsevier), Computer Methods and Programs in *Biomedicine & Pharmacotherapy* (Elsevier), *Heliyon*, IEEE International Wireless Communications and Mobile Computing Conference (IEEE IWCMC 2019), *Network Modeling and Analysis in Health Informatics and Bioinformatics*, and *Acta Acustica united with Acustica* journal of the European Acoustics Association (EAA).

BEATRIZ SAINZ-DE-ABAJO received the Ph.D. degree *(summa cum laude)* from the University of Cordoba, in 2009. She is currently an Associate Professor in telecommunications engineering with the University of Valladolid, Spain. Her fields of action include the development and evaluation of e-Health systems, m-Health, Medicine 2.0., and cloud computing. She focuses on topics related to electronic services for the information society. She belongs to the GTe Research Group, integrated within the UVa Recognized Research Group "Information Society." Among the lines of research, the group works to develop innovative solutions in the field of health that help patients improve their quality of life and facilitate the work of health professionals.

ABDELLATIF HAIR currently works as a Full Professor with the Department of Computer, FST Beni Mellal, Morocco, and a member of the LAMSC Laboratory. His research interests include object-oriented analysis/design, security of mobile agents, wireless sensor network (WSN), data warehousing, and machine learning (ML).

BEGOÑA GARCÍA-ZAPIRAIN (Member, IEEE) graduated in telecommunications engineering and specialized in telematics from the University of the Basque Country (UPV/EHU), Leioa, Spain, and the Advanced Program in Health Management at the Deusto Business School, Deusto University, Bilboa, Spain, in 2012. She received the Ph.D. degree *(summa cum laude)* in the pathological speech digital processing field in 2003, and the Executive M.B.A. degree from the University of the Basque Country, in 2011, with the Best Student Award. After spending five years at ZIV Company, she joined the Faculty of Engineering, University of Deusto, in 1997, as a Lecturer in signal theory and electronics, where she led the Department of Telecommunications, from 2002 to 2008. She received the Accesit to the Ada Byron Award to the Technologist Woman 2015. In recognition of the quality of its research activities, the research group she leads won the Research Award 2007 UDGrupo Santander, the ONCE Euskadi Solidarity Award 2007, the award for the best article in the Games 2009 International Congress, the prize for the best poster at ISIVC 2008, and was the finalist for the Social Innovation in Ageing—The European Award 2014.



ISABEL DE LA TORRE-DÍEZ is currently a Professor with the Department of Signal Theory and Communications and Telematic Engineering, University of Valladolid, Spain, where she is also the Leader of the GTe Research Group. Her research interests include design, development, and evaluation of telemedicine applications, services and systems, e-health, m-health, electronic health records (EHRs), EHRs standards, biosensors, cloud and fog computing, data mining, quality of service (QoS), and quality of experience (QoE) applied to the health field.