IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Emotion processing by applying a fuzzy-based Vader lexicon and a parallel deep belief network over massive data

FATIMA ES-SABERY [1], IBRAHIM ES-SABERY [1], ABDELLATIF HAIR [1], BEATRIZ SAINZ-DE-ABAJO [2], BEGONYA GARCIA-ZAPIRAIN [3], (Member, IEEE).

[1]Department of computer science, Faculty of Sciences and Technology, Sultan Moulay Slimane University, 23000 Beni Mellal, Morocco
[2]Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid, 47011 Valladolid, Spain
[3]eVIDA Research Group, University of Deusto, 48007 Bilbao, Spain

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

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

ABSTRACT Emotion processing has been a very intense domain of investigation in data analysis and NLP during the previous few years. Currently, the algorithms of the deep neural networks have been applied for opinion mining tasks with good results. Among various neuronal models applied for opinion mining a deep belief network (DBN) model has gained more attention. In this proposal, we have developed a combined classifier based on fuzzy Vader lexicon and a parallel deep belief network for emotion analysis. We have implemented multiple pretreatment techniques to improve the quality and soundness of the data and eliminate disturbing data. Afterward, we have performed a semi-automatic dataset labeling using a combination of two different methods: Mamdani's fuzzy system and Vader lexicon. As well, we have applied four feature extractors, which are: GloVe, TFIDF (Trigram), TFIDF (Bigram), TFIDF (Unigram) with the aim of transforming each incoming tweet into a digital value vector. In addition, we have integrated three feature selectors, namely: The ANOVA method, the chi-square approach and the mutual information technique with the objective of selecting the most relevant features. Further, we have implemented the DBN as classifier for classifying each inputted tweet into three categories: neutral, positive or negative. At the end, we have deployed our proposed approach in parallel way employing both Hadoop and Spark framework with the purpose of overcoming the problem of long runtime of massive data. Furthermore, we have carried out a comparison between our newly suggested hybrid approach and alternative hybrid models available in the literature. From the experimental findings, it was found that our suggested vague parallel approach is more powerful than the baseline patterns in terms of false negative rate (1.33%), recall (99.75%), runtime (32.95s), convergence, stability, F1 score (99.53%), accuracy (98.96%), error rate (1.04%), kappa-Static (99.1%), complexity, false positive rate (0.25%), precision rate (97.59%) and specificity rate (98.67%). As a conclusion, our vague parallel approach outperforms baseline and deep learning models, as well as certain other approaches chosen from the literature.

INDEX TERMS Hadoop, Sentiment analysis, Extractors of features, HDFS, Selectors of features, MapReduce, Fuzzy logic, Deep belief neural network.

I. INTRODUCTION

OPINION mining is a data extraction technique and an automatic language processing. For a piece of text, identify its sentimental score as positive, neutral, or negative and provide multiple ways, resources, and performance criteria to perform this task [1]. The sentimental score of a sentiment can be determined based on various threshold values and can be considered as different categories. With the growth of consumer-posted texts in the micro-blogging sites and the social networking like Facebook, Instagram,YouTube, Ticktock, Trip Advisor, Twitter, Whatsapp and Amazon. The analysis of sentiments in social media and websites has become increasingly popular in many scientific and industrial research communities [2].

Opinion mining is important active research field in NLP. Indeed, the past several years have testified an augmentation in the range of text-based sentiment data resources becoming widely available on the World Wide: web users' comments, which are more and more centralized by forums, customer investigations carried out by the leading brands, search engines, and social networks [3]. With such a wealth of data and resources available, automating the aggregation of various sentiments is becoming essential to efficiently obtain a comprehensive view of sentiments on a particular topic. The value of this data is enormous, both for organizations that want to get the feedback of the customer on their brand image or their products/services, and for individual who want to find out about an outing, a trip or a purchase.

Micro-blog platforms have recently attracted the interest of researchers and users due to the fact of their easiness and quickness of data exchange [4]. These platforms can be deemed as a giant storehouse of data with many millions of written posts and messages, usually arranged in complicated networks with users interchanging with each other at particular moments. Because of its widespread popularity, The Twitter is known as the first micro-blogging network in the entire world, it provides APIs for freely collecting data that can then be utilized for performing analysis or developing new applications, That's the reason why we have selected it for our diverse experiences [5].

Many research papers have been published on opinion mining on Twitter in various fields: natural disasters, politics, marketing, etc. Actually, Twitter is nowadays one of the greatest chances for a company to increase its visibility and accessibility among its prospective customers [6]. Marketers are noting the numerous new opportunities that Twitter provides and are beginning to introduce novel digital social innovations at an incredible ratio. As a consequence, the worldwide brands have recognised Twitter as a fully integrated advertising and marketing platform and are utilizing it in innovative ways to feed their promotional campaigns [7]. Twitter is equally being used as a political campaign platform by making it an incorporated media at the core of the policy communications strategies [8].

From the perspective of sentiment analysis tools, this liberty to hold expression is a critical challenge, as the goal is to extract the preoccupations of respondents in non-structured data. This explains the significant work carried out on this topic in the NLP area, adapted to the resolution of this kind of data extraction [9]. In essence, the high-noise content in data, typified by the occurrence of misspellings, formatting of content, syntactical mistakes whether unintentional or intentional poses a challenge at several points in the analysis of the data, from data-preprocessing (lemmatization, categorization, grammatical, word and sentence segmentation) to word/sentiment retrieval [10].

Deep learning patterns have the potential to be used to capture meaningful knowledge that is unobserved in the daily generated social network content [11]. There exists many different deep learning patterns that aid in the learning process, such as Restricted Boltzmann Machines (RBMs), Long Short Term Memory Networks (LSTMs), Recurrent Neural Networks (RNNs), Deep Belief Networks (DBNs), Generative Adversarial Networks (GANs), Autoencoders, Multilayer Perceptrons (MLPs), Self Organizing Maps (SOMs), Convolutional Neural Networks (CNNs), Radial Basis Function Networks (RBFNs), and FeedForward Neural Networks (FFNNs). These deep learning methods work on most all type of data and require higher levels of computation and learning capacity to resolve challenging problems [12].

In this proposal, we have designed a combined classifier based on fuzzy Vader lexicon and a parallel deep belief network for emotion analysis, which integrates the strengths of deep belief networks, Vader lexicon and fuzzy system of Mamdani to carry out the classification of feelings with high efficiency. Moreover, this proposal is implemented in parallel way utilizing both Hadoop and Spark framework with the purpose of overcoming the problem of long runtime of massive data. Consequently, the principal suggestions of our proposal can be summarised as indicated below:

1) Our combined classifier based on fuzzy Vader lexicon and a parallel deep belief network classifies the collected Tweets into 3 different classes: neutral, negative or positive.
2) Multiple pretreatment techniques like negation procedure, stop words, lemmatization process and warping process are implemented to improve the quality and soundness of the data and eliminate disturbing data.
3) A semi-automatic dataset labeling using a combination of two different methods: Mamdani's fuzzy system and Vader lexicon is applied over the Sentiment140 dataset.
4) Application of four feature extractors, which are: GloVe, TFIDF (Trigram), TFIDF (Bigram), TFIDF (Unigram) with the aim of transforming each incoming tweet into a digital value vector.
5) Integration of three feature selectors, namely: The ANOVA method, the chi-square approach and the mutual information technique with the objective of selecting the most relevant features.
6) Implementation of the DBN as classifier for classifying each inputted tweet into three labels: negative, neutral or positive.
7) Implementation of our proposed approach in parallel way employing the Hadoop framework with the purpose of overcoming the problem of long runtime of massive data
8) A comparison between our newly suggested hybrid approach and alternative hybrid models available in the literature is carried out.
9) Our suggested vague parallel approach is more powerful than the baseline patterns in terms of false negative rate, recall, runtime, convergence, stability, F1 score, accuracy, error rate, kappa-Static, complexity, false positive rate, precision rate and specificity rate.

The remainder of this work is structured in this way: The

2nd section discusses the driving forces behind the creation of this contribution, the 3rd section introduces the earlier published research studies, the 4th section outlines all steps in this contribution, the 5th section describes the results achieved in the different performed experiments, and finally, the 6th section makes the synthesis of the proposed hybrid approach and formulates a few guidelines for further work.

II. MOTIVATION

Nowadays, eCommerce online platforms allow their clients to publish reviews or comments on the items they have purchased. The insights offered by consumer feedback are important in helping other potential consumers decide whether or not to buy a product based on the opinions and experiences of other consumers about a specific product. Additionally, companies can also gather consumer feedback through online comments to enhance the quality of their products. Nevertheless, as the number of consumers purchasing items grows, the number of comments also rises over time and thus it is impossible for manufacturers or users to review all the comments of previous consumers on a certain item. Furthermore, certain customer feed-backs are very lengthy, which makes it challenging for users to identify both positive and negative product characteristics when considering if a product is actually worth purchasing, or for producers to know if the item requires enhancement. A sentiment rating process, which analyzes whether a consumer provides a negative or positive rating of a particular product, is very important and strongly recommended to potential consumers and manufacturers because it enables them to conveniently collect precious information about products through a diversity of feedback, that assists them in decision-making based on others' opinions. Motivated by the significant impact of sentiment analysis on our daily routine. In this proposal, we have developed a combined classifier based on fuzzy Vader lexicon and a parallel deep belief network that is employed to carry out the classification of sentiments. This contribution incorporates NLP techniques for performing the data preprocessing, Mamdani fuzzy system + Vader lexicon for carrying out the dataset semi-automatic labelling, feature extractor for transforming each tweet into digital vector, feature selector for choosing the most appropriate features and for reducing the high dimensional vector space of every extracted feature, DBN for performing the classification of each tweet into 3 categories(positive, neutral, or negative). Finally, both Hadoop and Spark framework are implemented for overcoming the problem of long runtime of massive data

The initial stage of our suggested methodology is the application of data pre-processing techniques. Therefore, data pre-processing step has a significant influence on the data classification process, as discussed in this article [13]. Their authors have supplied us with a comparative analysis process to assess the impact of data pre-treatment technologies on tweets classification by measuring the accuracy. Experimental findings indicated that applying the data preprocessing techniques on the linguistic data significantly enhanced the classification efficiency. Also, many other papers in literature [14], [15] have proven that the data preprocessing technologies have positively influenced the data classification procedure in terms of precision, recall, and accuracy. We were thus motivated by the strong performance that was reported on data pre-processing technologies, and we have incorporated these pre-processing techniques into this proposal.

After the preprocessing process, the following stage is a semi-automatic dataset labeling using a combination of two different methods: Mamdani's fuzzy system and Vader lexicon. The subsequent stage is the data mapping in which we implemented 4 feature extractors, including GloVe, TFIDF (Trigram), TFIDF (Bigram), and TFIDF (Unigram), in order to map every tweet into a digital vector. Then, we have incorporated 3 feature selectors, which are ANOVA approach, chi-square technique and mutual information method. In addition, we have implemented the DBN for classifying each tweet into 3 categories (negative, neutral, or positive). In the final stage, we have deployed our combined proposal in parallel way by employing both Hadoop and Spark frameworks.

In summary, the goal of this research is to raise the classification efficiency of feeling analysis through integrating the strengths of data pretreatment approaches in improving the thoroughness of tweets by removing noisy and unsuitable data, of the feature extraction techniques which transform every tweet into a number vector and captures the most pertinent characteristics of the tweet, of the feature selection techniques which minimize the high dimensional characteristics obtained in the previous stage and choose the most interesting characteristics, of the DBN in classifying every tweet and improving the performance of data classification, and that of both Hadoop and Spark frameworks for overcoming the problem of long runtime of massive dataset.

III. PREVIOUS RESEARCH

Here are some examples of existing published works that have applied many diverse deep learning models to tackle opinion retrieval problems in a diverse range of languages.

Es-sabery et *al.* [16] suggested a new classifier based on CNNs, FFNNs and Mamdani Fuzzy System (MFS). Firstly, they used the CNN as an efficient automatic procedure to retrieve and choose the most appropriate features. Then, they applied the FFNN to calculate the negative and positive emotional values. Finally, they employed the MFS as a classifier to categorize the outputs of the employed patterns (FFNNs+CNNs) into 3 categories, namely: negative, neutral, and positive. The empirical findings proved that their suggested fuzzy parallel approach is more efficient than the baseline patterns in terms of false negative rate, recall, runtime, convergence, stability, F1 score, accuracy, error rate, kappa-Static, complexity, false positive rate, precision rate and specificity rate.

The authors of the paper [17] proposed an hybrid deep learning model for opinion mining of Malayalam Tweets. Their suggested approach combine Bi-LSTMs, CNNs, Gated Recurrent Unit (GRU) and LSTMs. They applied the CNNs

for extracting and choosing the most relevant features. Then, they employed LSTMs to eliminate long-term dependency but retain some valuable data. In addition, they used Bi-LSTM to generate the precise copy of LSTM in the opposite sense. Finally, they applied GRU to decrease the architecture complexity of the LSTM. Their experimental results show that CNN+GRU achieved a maximum classification rate of 87.23% and CNN+BiLSTM achieved a classification rate of 74%.

In the paper [18], the authors designed a new hybrid model of deep learning structures, namely CNNs and LSTMs. Their hybrid model is provided for the classification of the opinions of reviews published in various fields. The authors chose to apply deep CNNs because of their high efficiency in selecting local features, and they applied LSTM because of its efficiency in sequentially processing a long text. Their suggested Co-LSTM approach has two primary purposes in the analysis of sentiment. Firstly, it is ideally suited for addressing large social data with scalability is taking into account, and secondly, in contrast to traditional machine learning algorithms, it is independent of any specific field. The experimental findings demonstrate that the overall suggested pattern exceeds other machine learning algorithms in terms of accuracy and other metrics.

Bodapati et *al*. [19] applied two models for conducting feelings analysis. One pattern is constructed using RNNs and LSTMs architecture and another with CNNs. In their first pattern, the RNNs + LSTMs pattern were employed to detect syntactic and semantic relationships among words in a review using the word2vec word embedding method. In their second pattern, uni-dimensional CNNs were utilized for learning the structure in a set of terms and the feature-specific position. They applied both models to the IMDB movie review dataset and the experimental findings were compared. Both models performed extremely well.

The authors of the paper [20] developed a new hybrid model incorporating RNNs+LSTMs and logistic regression to perform sentiment classification. The purpose of their research was to familiarize themselves with the concept of sentiment analysis and the manner in which social media acts as an essential part of it. Additionally, they used YouTube and Twitter web scraping to select a standard data set to do more analysis. Experimental results have shown that the RNNs+LSTMs model is more accurate than logistic regression with an accuracy equal to 83.25%.

In the paper [21], the authors proposed a new hybrid approach of deep learning by combining the two bidirectional models LSTM and GRU to solve the high dimension problem of the feature space. They used two distinct layers which are GRU and bidirectional LSTM layers to extract future and past features by attaching two opposites hidden layers at the same level of background. Also, they used the group-wise improvement technique over the set of features retrieved by the bi-LSTM layer, that splits the features into several categories, improving the relevant features of each category, while reducing the less valuable ones. The introduced pattern utilizes both pooling and convolution layers to retrieve a set of features and to minimize the high dimensional space of features. Experimental findings reveal that the introduced convolutional two-way RNN infrastructure with a group-wise improvement technique perform better than the state-of-the-art results for opinion mining.

Subhashini et *al*. [22] introduced a new decision-making model in which negative, positive, and boundary areas are categorized through the use of fuzzy logic concepts to overcome the limitations of ML approaches in managing uncertainties in people's opinions. They then applied the CNN to additional classification of vague concepts initially attributed to the boundary area. Their proposed framework utilizes some formal concepts for representing the uncertainties, and the CNNs categorize the concepts of boundary areas into either negative or positive opinions. Experimental results show that the proposed decision-making with a 3-way scheme deals effectively with opinion uncertainties.

The authors of the paper [23] implemented and evaluated several deep learning models such RNNs, LSTMs, gated recurrent unit (GRU), Group LSTMs, and update recurrent unit (URU). Then, they combined all evaluated deep learning models with several feature extractor, namely Skip-grams, FastText, word2vec and Glove. The 5 diverse deep learning models with the 3 feature extractors are assessed on the basis of F1 score, precision, recall, and accuracy for the unbalanced and balanced datasets. Their experimental results show that for the balanced dataset, the LSTM model achieved a maximum accuracy of 88.39%. And for the unbalanced dataset the GRU model combined with the FastText word embeddings approach obtained the best accuracy of 93.75%.

In the paper [24], the authors proposed a new hybrid CNN-LSTM. In the first step of their approach, they applied Word2vec word embedding to transform each word-based text into a digital vector. Once the word embedding process is done, in the next step they applied the convolution layer and the max-pooling layer to extract and select the most relevant features with long-term relationships. The model they offer also utilizes drop-out technique, standardization and a rectified linear unit to improve accuracy. Experimental findings show that their newly developed hybrid CNN-LSTM model surpasses both conventional machine and deep learning approaches in terms of precision, accuracy, recall, and F1-score.

IV. MATERIALS AND METHODS

We will discuss in the next subsection of this document, the reasons why we are proposing and developing this hybrid parallel pattern. In addition, the underlying architecture of this suggested hybrid parallel pattern is composed of six steps; The first step is to collect data using the massive Sentiment140 dataset in order to evaluate our newly suggested hybridized pattern. The 2nd step, referred to as data pre-processing, is designed to eliminate noisy and unwanted data. The 3rd phase is known as semi-automatic dataset labeling using a combination of two different methods: Mamdani's

fuzzy system and Vader lexicon. The 4th phase is feature extraction, that converts the data-based text into digital vectors. The 5th phase is the characteristic selection to minimize high dimensionality of the retrieved characteristics in data extraction phase. In step six, we set up the DBN as a classifier so that every tweet is classified into 3 categories (neutral, negative or positive). Finally, we have deployed our proposed approach in parallel way employing both Hadoop and Spark framework with the purpose of overcoming the problem of long runtime of massive data.

As illustrated in the figure 1, our combined classifier based on fuzzy Vader lexicon and parallel DBN's general structure is made up of 5 steps: data collection stage, data pre-treatment stage, data representation stage, characteristic extraction stage, characteristic selection stage, data classification, data parallelization employing Hadoop.

A. DATA COLLECTION STAGE

Opinion mining systems need a corpus of user comments to train a categorizer or to assess it. The most commonly used corpuses have been gathered through social networking websites, since the available contents are free, easy and instantaneous. People can share and discuss their thoughts in public. In this proposal, we have employed the massive Sentiment140 corpus. It includes 1,600,000 gathered tweets with all emoticons in this corpus suppressed. For each single tweet, it was tagged by using 2 labels: positive and negative, wherein 4 indicates the positive feedback label while 0 denotes the negative feedback tag. Sentiment140 corpus comprises six features that are outlined below:

- **User:** indicates the username who tweeted the message.
- **Location:** displays the right location where the tweet has been published.
- **Text:** introduces the full text of every tweet.
- **Date:** denotes the precise time the tweet has been published (Tuesday July 05 05:49:31 +0202 1999).
- **Flags:** is used to indicate the request content of the username. While the string "NO QUERY" represents the value of flag variable in the case when the user has not published a request.
- **Ids:** is a unique number (542369871) which uniquely identifies every tweet.
- **Target:** Recognizes the category tag for every tweet, wherein 4 denotes the positive feedback tag while 0 denotes the negative feedback tag.

We focus on examining opinions in this contribution. It also means that we collect every viewpoint offered by all Twitter user in any tweet posted. As a result, the other attributes of this corpus have no bearing on the training objective. We have preserved the attributes "Text" and "Target" from the dataset while discarding the attributes "User," "Date," "Ids," and "Flag." The dataset chosen for this contribution is divided into two subsets, the formation and test subsets. Consequently, we used both subsets to demonstrate, in comparison to previous existing methods in the literature, the effectiveness of our hybrid classifier based on fuzzy Vader lexicon and a parallel deep belief network.

Figure 2 presents the proportion of tweets from the formation and evaluation sub-set that are neural, favorable, and unfavorable. 1120,000 tweeted posts were taken as the total number of tweets in the formation phase. And 480000 tweeted posts we taken as the total number of tweets in the test phase. As a result, the test subset accounts for 30% of the overall tweeted posts in the original corpus.

B. TWEETS PRETREATMENT STEP

Numerous applications that employ raw and unstructured data make use of the preprocessing task, which has been the subject of extensive research. The necessity for pretreatment is considerably increased when research focuses more on published posts in social networks since many posts are misspelled and improperly written. Data pre-processing methods are thus required to obtain a cleaner corpus and the next classification operation performs more effectively when the dataset has been cleaned.

Almost all of the above listed sentiment analysis studies [9] - [15], the applied pretreatment methods are integrated. The pretreatment methods used are easy to understand, such as correction of simple error, punctuation, filtering of letters and of words. Lexicons are used to repair common faults like misspelled words and repeated letters. And the dictionaries are used to fix mistakes. Similar to this, abbreviations and acronyms are turned to words from a comprehensive dictionary.

A substitute strategy that has been used is removing the pointless content. Like stop words and punctuation which are removed from tweets to reduce the diversity of words used because they do not significantly affect the sentimental score [25]. The overuse of letters is one example of a filtering technique. A good illustration of vowel recurrence is "Weeeeeeeell". Repeated punctuation is an illustration of this example "well !!!!!!!!!!!!!". By spotting the overuse of more than two of the following letters, these filtering techniques can be used [26].

The most widely used method for filtering and replacing strings is the regular expression [27]. It is an useful method for locating sub-strings in a string and allows for the identification and removal of errors and abuses. Regular expressions are used in many "search and replace" functionalities of programs because they are seen to be a particularly powerful matching technique for strings [28].

Tokenization is a useful additional tactic that must be used to reduce the broad variety of phrases. It is a technique for stripping verb forms down to their root [29]. The reduction of "liked" to "like" is an example of tokenization. This reduces the variety of verb conjugations that a word can have, which in turn reduces the amount of data [30].

Most of the methods described are language-specific. The language of the text is crucial for pretreatment and categorization [31]. It is essential to have the clearest feasible corpus. When we look at different languages, we find that each

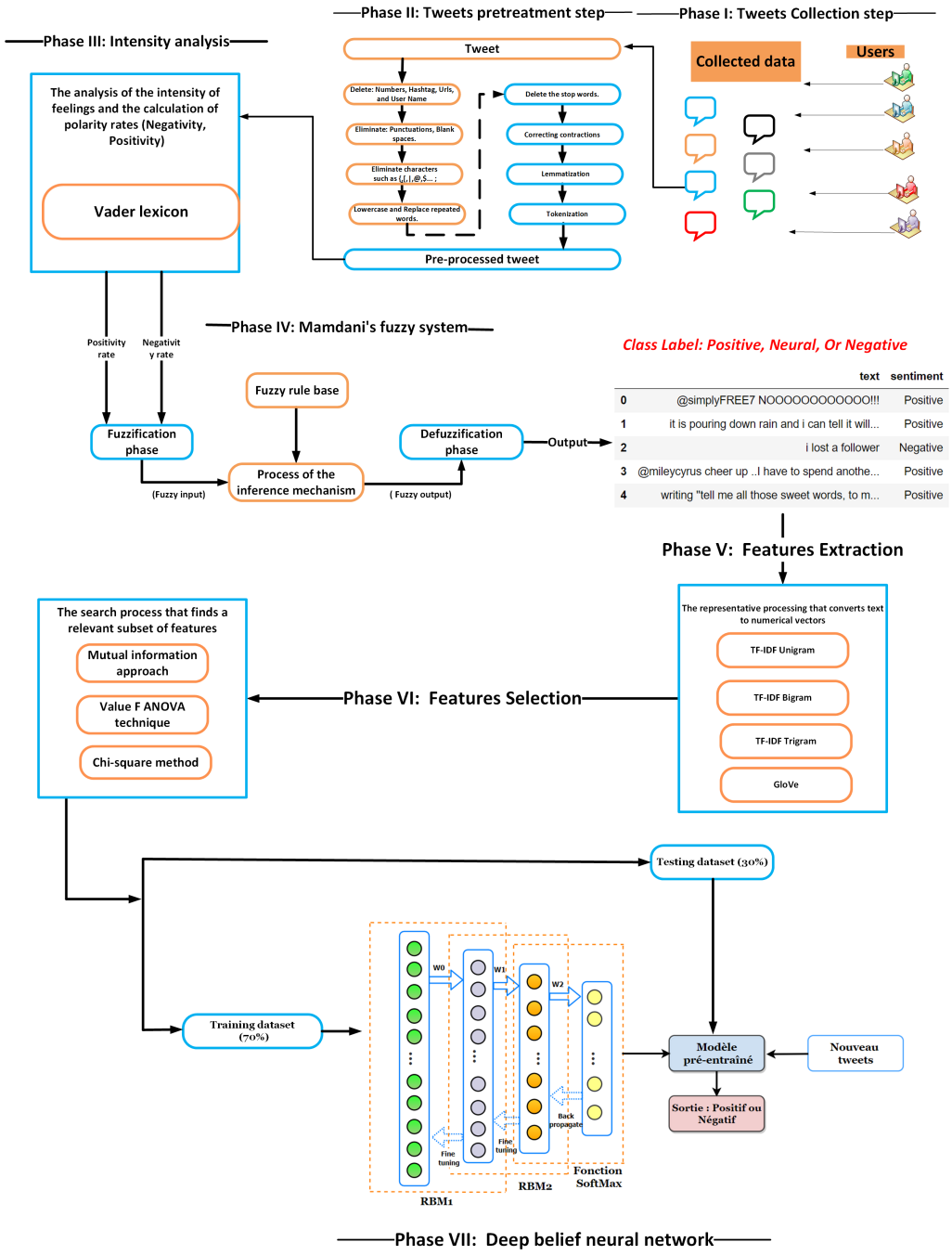

FIGURE 1. Global architecture of our combined classifier based on fuzzy Vader lexicon and a parallel deep belief network.

one has a unique term vocabulary, syntax, and grammatical form. Therefore, it is vital to identify the language for select a confirmed grammatical context. This is the main area of study for lingual recognition, which has received an extensive research [32].

When a message's language has been determined, the pattern that will be trained can proceed as though all tweeted posts originate from that language. Therefore, it is safe to conclude that the tweeted posts contain just words from the recognized language [33].

1) Data exploration:

The word cloud is a traditional form of data representation in autonomous word recognition [34]. The most frequently employed terms can be easily displayed in this format. The recurrence of terms from the dataset utilized is relative to the size of the terms in concern in the image, with the exception of colors, which serve just as decoration (as depicted in figure 3).

On the two offered graphical representations, we have charted the original data both with and without pre-treatment.

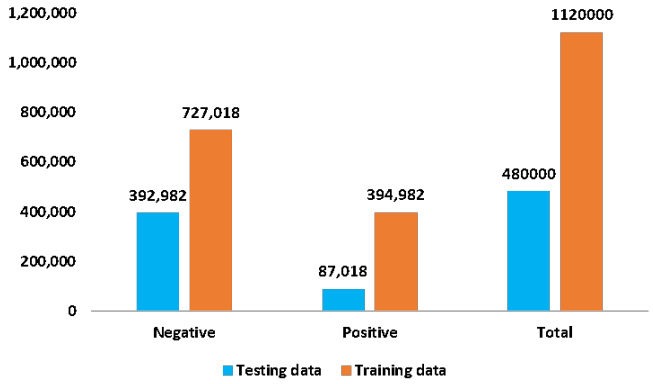

FIGURE 2. Proportion of positive, neutral, and negative tweeted posts in the Sentiment140 corpus.

FIGURE 3. Word cloud.

Mostly in case of crude data, the tweet headers' structure is primarily to blame for the noise that was noticed. Once we have successfully removed a significant amount of noise from the pre-treated data, it becomes meaningful and can begin to be studied.

The semi-automatic tagging phase is the step in this approach that comes after the preprocessing stage for tweets. That indicates that the data obtained following the completion of all of the aforementioned activities for preprocessing of tweets will be fed to the mixed method fuzzy Vader lexicon.

C. SEMI-AUTOMATIC LABELING STEP

An important phase in ML (Machine Learning) is data labeling [35]. It is essential to label the data before using it to train an AI model. After carefully examining and analyzing the dataset, we discovered that certain tweets were incorrectly tagged. As a result, we chose to combine the Mamdani fuzzy system and Vader lexicon for performing the semi-automatic data tagging.

1) Vader lexicon

Vader, a rule-based lingual dictionary and web usage mining tool created to evaluate social network sentiments, and it stands for Valence Aware Dictionary and Sentiment Reasoner. It has an MIT license and is open source [36], [37]. It makes use of a range of technologies and tools. A sentiment vocabulary is a collection of lexical features (such as terms) that are classified as neutral, positive, or negative based on their sentimental score. It displays the intensity of a negative or positive mood in addition to the rates of positivity and negativity [38]. VADER maintains the benefits of traditional vocabulary like LIWC. It is more substantial, simple to understand, quick to use, and simple to grow. VADER's emotion dictionary is quality gold-standard and has been approved by professionals. VADER differs from LIWC in its awareness of feeling expressions in the context of social media and also in its more favorable generalization to other fields. In this proposal, we have used the VADER lexicon to compute the negativity and positivity rates of each tweet before feeding it into the Mamdani fuzzy system. For example, we have applied the VADER onto the tweet described in the table 1, and we got as positivity rate equal to 0.575% and negativity rate equal to 0.425%.

TABLE 1. Polarity intensity.

Tweet	Positivity score	Negativity score
Deep neural networks are incredibly effectivvvvvvve at processing data, but they are inefficient when dealing with the inherent ambiguity in NLP, which requires more answers.	0.575	0.425

2) Fuzzy logic system of Mamdani:

Once both sentimental scores NS (Negativity score) = 0.425 and PS (Positivity score) = 0.575 have been computed. The implementation of Mamdani's fuzzy logic, which is composed of three stages as illustrated in the figure 4, is the following stage.

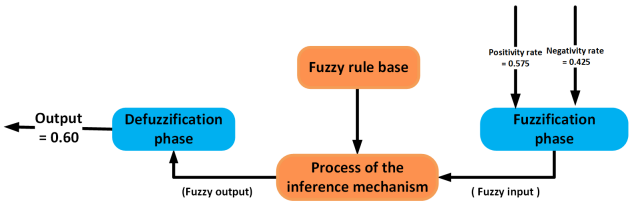

FIGURE 4. Stages of Mamdani fuzzy system.

A first stage prior to the fuzzification operation is the setting the in and out lingual variables and the setting of the lingual words for every lingual variable [39]. Therefore, in our contribution, we have implemented the fuzzy system of Mamdani as a categorizer on the two variables PS and NS obtained in the preceding stage of Vader lexicon. Moreover, the variables linguistic inputs are NS and PS and every linguistic variable is assigned three lingual words which are Lower (between 0.0 and 0.35), Medium (between 0.35 and 0.65) and Higher (between 0.65 and 1). Further, the output linguistic variable is the class label that takes three lingual words positive (is between 0.65 and 1.0), negative (is between 0.0 and 0.35) and neutral (is between 0.35 and 0.65). In conclusion, the table 3 depicts the inputs and outputs linguistic variables.

TABLE 2. Lingual input and output metrics of Mamdani's fuzzy system.

Variables	Lingual variables	Plage	Lingual words	metrics
Input	PS	0-1	Lower	0.0-0.35
			Medium	0.35-0.65
			Higher	0.65-1
Input	NS	0-1	Lower	0.0-0.35
			Medium	0.35-0.65
			Higher	0.65-1
Output	CT	0-1	Negative	0.0-0.35
			Neutral	0.35-0.65
			Positive	0.65-1

Fuzzification phase: After the setting of lingual variables and lingual words, the subsequent stage is the implementation of fuzzification operation [40] to the net rates of PS and NS, employing one of the membership functions (MFs) detailed by the equations (1), (2), (3), (4) and (5) for computing the membership degree μ of both sentimental scores PS and NS in the Lower, Medium, and Higher fuzzy sets.

A triangular membership function [41] is defined by a low value **lv**, a modal value **mv**, a high value **hv** and $lv < mv < hv$. It is defined as follows:

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq lv \\ \frac{x-lv}{mv-lv} & \text{si } lv \leq x \leq mv \\ \frac{hv-x}{hv-mv} & \text{si } mv \leq x \leq hv \\ 0 & \text{si } hv \leq x \end{cases} \quad (1)$$

A trapezoidal membership function [41] is defined by a low value **lv**, a high value **hv** and two values **vp** and **vz** which represent the boundaries of its kernel. The formula for the trapezoidal membership function is represented as follows:

$$\mu_A(x) = \begin{cases} 0 & \text{si } (x < lv) or (x > hv) \\ \frac{x-lv}{vp-lv} & \text{si } lv \leq x \leq vp \\ 1 & \text{si } vp \leq x \leq vz \\ \frac{hv-x}{hv-vz} & \text{si } vz \leq x \leq hv \end{cases} \quad (2)$$

A monotonically increasing membership function [42] is defined by two metrics **d** and **p**. It is defined by the next equation:

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq d \\ \frac{x-d}{p-d} & \text{si } d \leq x \leq p \\ 1 & \text{si } p \leq x \end{cases} \quad (3)$$

A monotonically decreasing membership function [42] is defined by two metrics **d** and **p**. Its formula is represented as follows:

$$\mu_A(x) = \begin{cases} 1 & \text{si } x \leq d \\ \frac{p-x}{p-d} & \text{si } d \leq x \leq p \\ 0 & \text{si } p \leq x \end{cases} \quad (4)$$

A Gaussian function is defined [41] by its modal value **m** and by a value **k > 0**. It reaches 1 only for the modal value **m**. The formula associated to the Gaussian membership function is described as follows:

$$\mu_A(x) = e^{-\frac{(x-k)^2}{2.m^2}} \quad (5)$$

For example, we apply the triangular MF (1) to measure the degrees of membership of the variables PS and NS to the Lower, Medium, and Higher fuzzy sets. The computational process is presented as follows:

In the case of the linguistic word **Lower**, and the optimum scalar metrics are lv=0; mv=0.175; and hv=0.35; then, we used these metrics to measure the degrees of membership of the two linguistic parameters PS and NS to the Lower fuzzy set. The outcomes are the following:

- There is PS=0.575 ≥ hv=0.35 So, $\mu_{Lower}(PS) = 0$
- There NS=0.425 ≥ hv=0.35 So, $\mu_{Lower}(NS) = 0$

Consequently, the metric values of each used membership function have been found empirically, and we choose the optimum values of these metrics that yield the better classification results.

In the case of the linguistic word **Medium**, and the optimum scalar metrics are lv=0.35; mv=0.5; and hv=0.65; then, we used these metrics to measure the degrees of membership of the two linguistic parameters PS and NS to the Medium fuzzy set. The outcomes are the following:

- There is mv=0.5 ≤ PS=0.575 ≤ hv=0.65 So, $\mu_{Medium}(PS) = \frac{0.65-0.575}{0.65-0.5} = 0.5$
- There is lv=0.35 ≤ NS=0.425 ≤ mv=0.5 So, $\mu_{Medium}(NS) = \frac{0.425-0.35}{0.5-0.35} = 0.5$

In the case of the linguistic word **Higher**, and the optimum scalar metrics are lv=0.65; mv=0.825; and hv=1; then, we used these metrics to compute the degrees of membership of the two linguistic parameters PS and NS to the Middle fuzzy set. The outcomes are the following:

- There is PS=0.575 ≤ lv=0.65 So, $\mu_{Higher}(PS) = 0$
- There is NS=0.425 ≤ lv=0.65 So, $\mu_{Higher}(NS) = 0$

Base of Fuzzy rules: The establishment of the fuzzy rules (FR) for IF-THEN statements comes next after the fuzzification procedure as described below.

FR 1: IF PS is Lower **&** NS is Lower **THEN** CT = neutral
FR 2: IF PS is Lower **&** NS is Medium **THEN** CT = Negative
FR 3: IF PS is Lower **&** NS is Higher **THEN** CT = Negative
FR 4: IF PS is Medium **&** NS is Lower **THEN** CT = Positive
FR 5: IF PS is Medium **&** NS is Medium **THEN** CT = Neutral
FR 6: IF PS is Medium **&** NS is Higher **THEN** CT =

Negative
FR 7: IF PS is Higher **&** NS is Lower **THEN** CT = Positive
FR 8: IF PS is Higher **&** NS is Medium **THEN** CT = Positive
FR 9: IF PS is Higher **&** NS is Higher **THEN** CT = Neutral

Inference mechanism: Once the IF-THEN fuzzy rules has been made. The next stage is the application of the inference procedure [43] which is a technique for gathering the data of a particular model using a defined collection of rules for the representation of any issue. Every rule provides a part of its conclusion that is later merged with the rest of the rules in order to get a full conclusion. In general, three rules govern the inference operation: Application, Implication, and Aggregation. Which are introduced below.

Application sub-stage : this step of a inference mechanism matches the fuzzy membership degrees of each rule's inputs to a firing strength for that rule [44]. The firing strength rate of each rule is measured by the intersection of the antecedent block for the fuzzy rule. Where intersection (conjunctive) expressed in the logic connective "OR" by t-norm = maximum. And in the logic connective "AND" it denoted by t-norm= minimum. This process is defined by both equations (6) and (7) below:

$$\mu_{Ap} = \mu_i(PS) \ \ \mathbf{A}ND \ \ \mu_i(NS) = Min(\mu_i(PS), \mu_i(NS)) \tag{6}$$

$$\mu_{Ap} = \mu_i(PS) \ \ \mathbf{O}R \ \ \mu_i(NS) = Max(\mu_i(PS), \mu_i(NS)) \tag{7}$$

For example we have:

FR 1: IF (PS is Lower) = 0 **&** (NS is Lower) = 0 **THEN** (CT = neutral) = Min(0,0) = 0
FR 2: IF (PS is Lower) = 0 **&** (NS is Medium) = 0.5 **THEN** (CT = Negative) = Min(0,0.5) = 0
FR 3: IF (PS is Lower) =0 **&** (NS is Higher) = 0 **THEN** (CT = Negative) = 0
FR 4: IF (PS is Medium) = 0.5 **&** (NS is Lower) = 0 **THEN** (CT = Positive) = Min(0.5,0) = 0
FR 5: IF (PS is Medium) = 0.5 **&** (NS is Medium) = 0.5 **THEN** (CT = Neutral) = Min(0.5,0.5) = 0.5
FR 6: IF (PS is Medium) = 0.5 **&** (NS is Higher) = 0 **THEN** (CT = Negative) = Min(0.5,0) = 0
FR 7: IF (PS is Higher) = 0 **&** (NS is Lower) = 0 **THEN** (CT = Positive) = Min(0,0) = 0
FR 8: IF (PS is Higher) = 0 **&** (NS is Medium) = 0.5 **THEN** (CT = Positive) = Min(0, 0.5) = 0
FR 9: IF (PS is Higher) = 0 **&** (NS is Higher) = 0 **THEN** (CL = Neutral) = Min(0,0)= 0

Implication sub-step: this step aims to apply an implication operator at every IF-THEN activated fuzzy rule and this implication operator applied mostly the operation minimum between the consequent block of every rule and fuzzy results given by the previous application step [45]. The following formula 8 presents this implication sub-stage:

$$\mu_{Im}(CL_r) = \mathbf{m}in(\mu_{Ap}, \mu_j(CL_r) = 1) \tag{8}$$

For example we have:
FR 1: $\mu_1(neutral)$ = minimum(0,1) = 0
FR 2: $\mu_2(negative)$ = minimum(0,1) = 0
FR 3: $\mu_3(negative)$ = minimum(0,1) = 0
FR 4: $\mu_4(positive)$ = minimum(0,1) = 0
FR 5: $\mu_5(neutral)$ = minimum(0.5,1) = 0.5
FR 6: $\mu_6(negative)$ = minimum(0,1) = 0
FR 7: $\mu_7(positive)$ = minimum(0,1) = 0
FR 8: $\mu_8(positive)$ = minimum(0,1) = 0
FR 9: $\mu_9(neutral)$ = minimum(0,1) = 0

Aggregation sub-step: the last sub-stage in the inference process is the combination of the results given by the implication sub-step. In other terms, all IF-Then Fuzzy Rules having the same class label will be aggregated together [46]. There are various aggregation indicators, such as mean,maximum, arithmetic mean, geometric mean and minimum. A widely utilized operator is the maximum that is defined by the next formula (9):

$$\mu_{Ag}(CL_r) = \mathbf{m}ax(\mu_{J1}(CL_r), \mu_{J2}(CL_r), ..., \mu_{Jn}(CL_r)) \tag{9}$$

For example we have:
$\mu(neutral) = \mu_1(neutral) \vee \mu_5(neutral) \vee \mu_9(neutral)$ = **Max(0,0.5,0) = 0.5**
$\mu(negative) = \mu_2(negative) \vee \mu_3(negative) \vee \mu_6(negative)$ = **Max(0,0,0) = 0**
$\mu(positive) = \mu_4(positive) \vee \mu_7(positive) \vee \mu_8(positive)$ = **Max(0,0,0) = 0**

Defuzzification phase: is Mamdani's fuzzy system's final step. It is the procedure of yielding a measurable outcome in Crisp logic, based on the corresponding fuzzy sets and membership degrees [47]. It is the procedure that transforms a fuzzy result into a crisp result. There are several defuzzification approaches like Mean-Max Membership (MMM), last of maxima (FMLM), Max-Membership Principle (MMP), Centre of Largest Area (CLA), Centroid Method (CM), Centre of Sums (CS), Weighted Average Method (WAM), and first of maxima. The following outlines the defuzzification techniques employed:

Max-Membership Principle : This defuzzification approach is also termed as the height approach [48]. It is restricted to peak outcome functions and it is described by the next algebraic formula.

$$\mu(y^*) \geq \mu(y) | y \in Y \tag{10}$$

Where $\mu(y)$ indicates the membership rate of the element x and $\mu(y^*)$ is the membership rate of the defuzzified element y^*. The representation graphic of the MMP defuzzification method is illustrated in the figure 5.

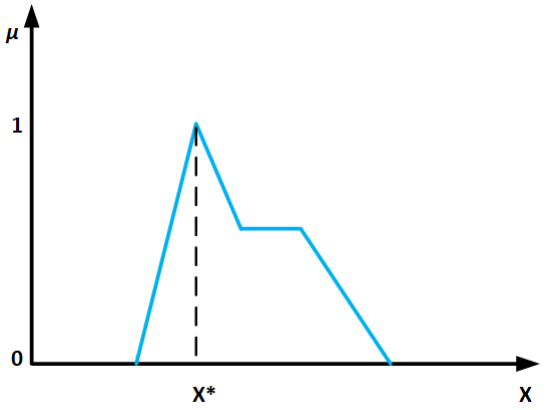

FIGURE 5. Max-Membership Principle.

Centroid approach : this method is also renowned as the centre of mass, of gravity, or of area [49]. It is the most frequently applied defuzzification technique. Its core idea is to identify the point x^* at which a vertical boundary line would split the aggregate into two independent equal masses. It is defined by the following equation (11)

$$x^* = \frac{\sum_{i=1}^{n} x_i.\mu(x_i)}{\sum_{i=1}^{n} \mu(x_i)} \tag{11}$$

Where x_i denotes the element in the instance,$\mu(x_i)$ is the membership rate of the variable x_i , and n presents the overall number of the variables in the used example. The representation graphic of the CM defuzzification approach is illustrated in the figure 6.

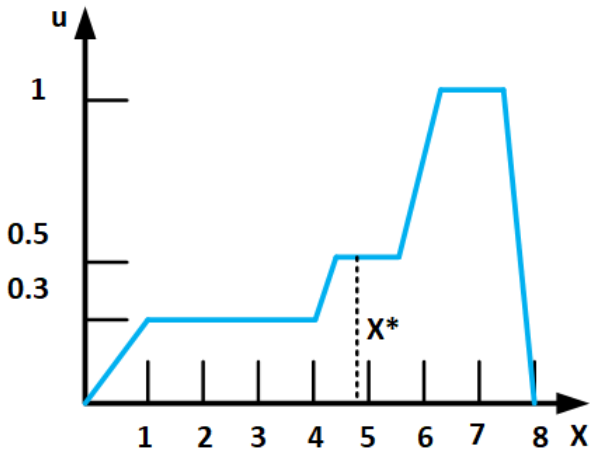

FIGURE 6. Centroid Approach.

Weighted Average Approach : is the simplest and, most commonly applied defuzzification method [50]. This technique is also referred as the "Sugeno defuzzification" approach. it can be formed by averaging every function of the outcome by its corresponding maximum belonging degree. This approach is also useful for fuzzy sets with symmetric outcome belonging functions and gives results quite comparable to the output of the CLA approach. And it is defined by the next algebraic equation (12).

$$x^* = \frac{\sum \mu_i(x).x}{\sum \mu_i(x)} \tag{12}$$

Where x represents an element of the instance and $\mu_i(x)$ indicates the membership degree of the element x. The following figure 7 illustrates the representation graphic of the Weighted Average defuzzification approach

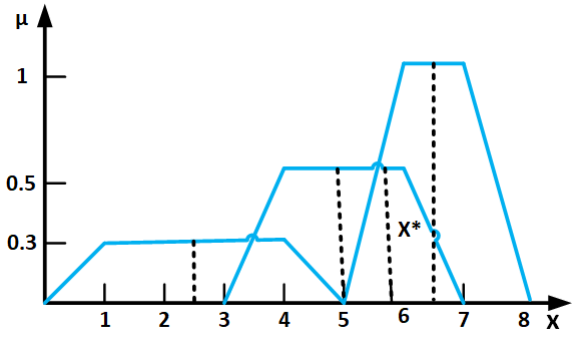

FIGURE 7. Weighted Average Method.

Mean-Max Membership : this approach is also referred to as the medium of maxima procedure [51]. It is very closely linked to the maximum membership function, with the exception that the peak membership positions can be non-unique. The defuzzified outcome here is expressed by the subsequent equation (13):

$$x^* = \frac{\sum_{x \in N}(x)}{N} \tag{13}$$

Where x represents the maximum membership degree and N indicates the overall count of elements in the instance. The figure 8 depicts the representation graphic of the Mean-Max Membership defuzzification approach.

Centre of Sums approach : in this procedure, the overlapping region is covered several times, whereas the centroid procedure only does so once [52]. It utilizes the algebraic summation of the single fuzzy subsets rather than their fusion and it defined by the following equation (14).

$$x^* = \frac{\sum_{ii=1}^{n} x_{ii} \cdot \sum_{j=1}^{k} \mu_{i_j}(x_{ii})}{\sum_{ii=1}^{n} \cdot \sum_{j=1}^{k} \mu_{i_j}(x_{ii})} \tag{14}$$

Where *K* represents the number of fuzzy lingual terms, *n* indicates the overall count of the fuzzy sets, and μ_{i_j} represents the *jth* fuzzy set' membership degree. The representation graphic of the centre of Sums defuzzification method is depicted in the figure 9.

Centre of Largest Area : It can be applied where the outcome

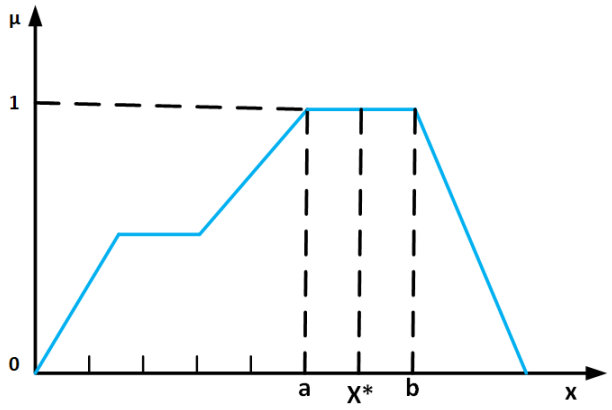

FIGURE 8. Mean-Max representation graphic.

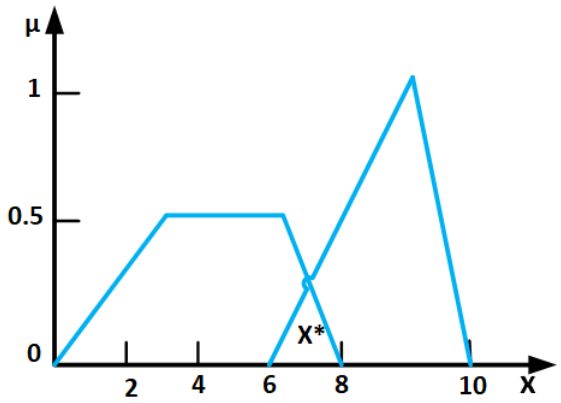

FIGURE 9. Centre of Sums.

has at least 2 non-overlapping fuzzy convex subsets. The outcome, in this scenario, is skewed to one side of a membership method [53]. Whenever the fuzzy outcome has two or more convex areas, so, the centroid of the convex fuzzy subarea with the biggest value is taken to get the defuzzified value x^*. The value is determined by the next equation (15).

$$y^* = \frac{\int_{\alpha}^{yBOA} \mu_i(y).yd_y}{\int_{yBOA}^{\beta} \mu_i(y).d_y} \tag{15}$$

Where $\alpha = min\{y; y \in Z\}$, $\beta = max\{y; y \in Z\}$ and $y = yBOA$ is the vertical axis dividing the zone between $y{=}\alpha$, $y{=}\beta$ $v{=}0$ *and* $v{=}\mu_i(y)$ into both zones which belongs to the same area, $\mu_i(y)$ represents the membership rate of the variable y, with y^* is the y variable' derivative. The representation graphic of the Centre of Largest Area defuzzification approach is depicted in the figure 10.

First of maxima: this approach aggregate the overall outcome or union of all outcome fuzzy sets C_i for finding the smallest value of the area that maximized the membership degree in the fuzzy sets C_i [53]. Therefore, the defuzzified

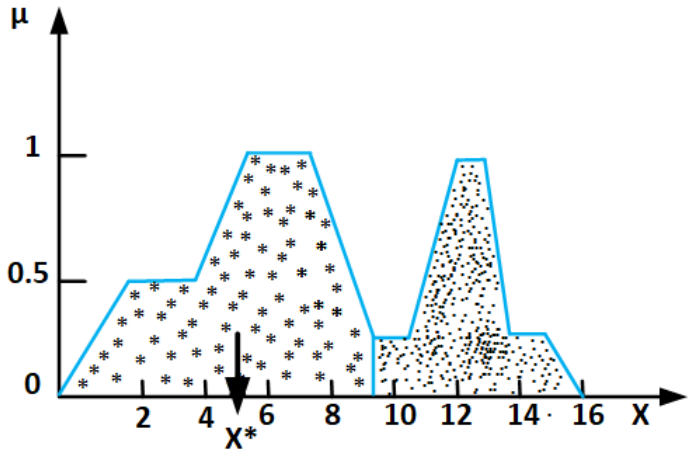

FIGURE 10. Centre of Largest Area.

value is described by the next formula (16)

$$x^* = min(x|C(x) = max_i C(i)) \tag{16}$$

The figure 11 depicts the representation graphic of the First of maxima Membership defuzzification approach.

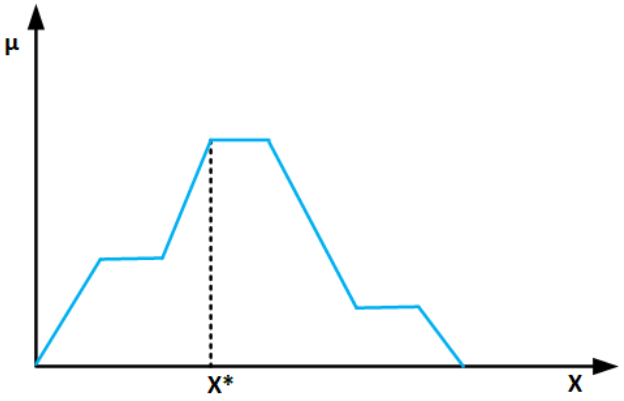

FIGURE 11. First of maxima.

Last of maxima method : It aggregate the overall outcome or union of all outcome fuzzy sets C_i for finding the greatest value of the area that maximized the membership degree in the fuzzy sets C_i [53]. Therefore, the defuzzified value is outlined by the next formula (17).

$$x^* = max(x|C(x) = max_i C(i)) \tag{17}$$

The figure 12 depicts the representation graphic of the Last of maxima Membership defuzzification approach.

D. TWEET REPRESENTATION STEP

So because attributes values should be multiplied by the system weights, feature extractor is often employed in machine learning models to turn the attributes into into real number

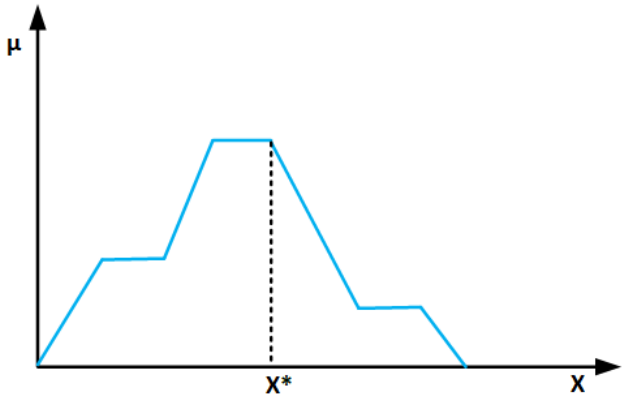

FIGURE 12. Last of maxima.

vectors. In our contribution, we have applied four feature extractors, which are: GloVe, TFIDF (Trigram), TFIDF (Bi-gram), TFIDF (Unigram) in order to discover which one provides great accuracy rate.

1) N-grams:

It is an n-element sub-strings created from a used string. The information theory work of Claude Shannon appears to be the source of the concept.His theory was that the probability ratio of the occurrence of the following letter could be determined, for instance, from a specified sequence of letters. It is simple to create a likelihood function for the following letter with a history of size n from a training corpus. N-grams are frequently employed in the analysis of natural language. Their application is predicated on the underlying premise that, provided a sequence of k items ($k \geq n$), Therefore, only the n-1 previous elements determine the likelihood of an element appearing at position i. We thus have:

$$P(OC_i|OC_1,...,OC_{i-1}) = P(OC_i|OC_{i-(n-1)}, OC_{i-(n-2)},..., OC_{i-1}) \quad (18)$$

With n = 3 (case of the trigram), we have:

$$P(OC_i|OC_1,...,OC_{i-1}) = P(OC_i|OC_{i-2}, OC_{i-1}) \quad (19)$$

The probability of the sequence is:

$$P(OC_1,k) = P(OC_1)\times P(OC_2|OC_1)\times P(OC_3|OC_1,OC_2) \times P(OC_k|OC_1,OC_2,...,OC_{k-1}) \quad (20)$$

$$P(OC_1,k) = P(OC_1) \times P(OC_2|OC_1) \prod_{i=3}^{n} P(OC_i|OC_{i-2}, OC_{i-1}) \quad (21)$$

2) TF-IDF:

It is completely statistical and relies on how often terms occur. It is frequently used during knowledge discovery, and information extraction in specific. This statistical measure enables one to assess a term's significance in relation to a corpora or set of terms. The word's weight rises in direct proportion to how frequently it appears in the text. Additionally, it changes based on how frequently a term appears in the corpus. In order to determine if a document is relevant to the user's search requirements, search engines frequently employ variations of the original algorithm.

Term frequency (TF): The frequency of occurrence of a word in the example document is its "raw" frequency. We can choose this frequency to express the frequency of a term.

$$TF(t) = \frac{n_t}{\sum_{k=1}^{K} n_k} \quad (22)$$

Where:
n_t reveals how frequently the word t appears in the text.
$\sum_{k=1}^{m} n_k$ represents the overall count of the term in the document.

Inverse Document Frequency (IDF): IDF is a measurement of the frequency of the word in the whole corpora. The TF-IDF pattern is designed to give greater weight to the less frequent words, deemed to be more discriminatory. It consists in calculating the logarithm (in base 10 or base 21) of the inverse of the percentage of the corpus that includes the word w:

$$IDF(W,P) = log\frac{|P|}{1+|m \in P : w \in m|} \quad (23)$$

P: infers our corpus of documents. It can equally be described as $P = m_1, m_2, ..., m_n$ where *n* is the number of documents.

$|m \in P : w \in m|$: means the total number of repetition of the word *w* in the document *m* (the $m \in P$).

Therefore :

$$TFIDF(w,m,P) = TF(w,m) \times IDF(w,P) \quad (24)$$

Where: *w* : means words or terms; *m*: indicates every document; *P*: denotes the corpus.

3) GloVe

Terms are transferred into a vector space containing digital values when term integration procedures are used. A good term incorporation should ideally map terms so that two different terms with almost the same semantic importance have mappings that are extremely comparable in the vector space [54]. It is possible to keep additional linguistic linkages between terms that are unrelated. As an illustration, the subsequent operations "King - Man + Woman" deliver a rate that closely resembles the word "Queen" vector space representation when we employ these vector space representations.

A widely applied word integration pattern is GloVe (Global Vectors). Which is a model suggested by the NLP

research staff at Stanford University. This method merges the benefits of both local context and global matrix factorization methods. The content is a window of a fixed size of lexical elements that is placed around the word. We try to map every term i and every term j paired in the similar content by the vectors spaces z_i and z_j respectively, of size d like :

$$z_i.z_j + bs_i + bs_j = log(X_{ij}) \tag{25}$$

Where X_{ij} indicates the frequency with which word j occurs in the content of term i. bs_i and bs_j are the biases related to the terms i and j respectively.

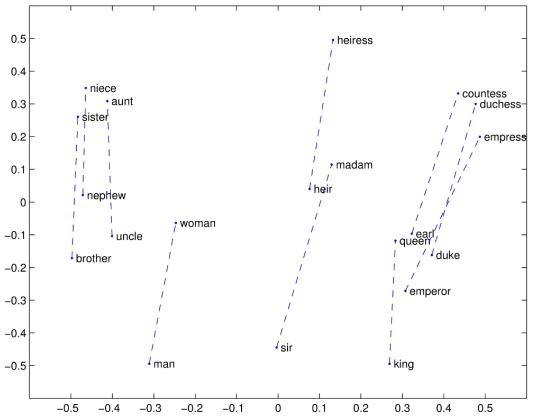

FIGURE 13. **Example of linear sub-structures when working with GloVe.**

E. STAGE OF FEATURE SELECTION

Finding the " relevant " subset of features from the starting collection is the goal of feature selection.The system's goals and criteria are always taken into consideration while determining the significance of a subset of features. In our research, we combined three ways to carry out the feature selection:

1) Mutual information approach

The semantic similarity of two random attributes is a measure of the mutual correlation between the two random attributes in knowledge and likelihood theories.More specifically, it refers to the quantity of knowledge gained about one random attribute through investigation into another random attribute. It can identify non-linear correlations between the two random attributes and is symmetric. This approach is valuable in the topic of feature selection since it allows us to determine the relevance features from a subset of features with regard to the outcome space vector. In formal terms, mutual information approach is described as follows:

$$MI(w_1, w_2) = \sum_{c=1}^{m}\sum_{d=1}^{m} p(w_1(c), w_2(d)).log\Big(\frac{p(w_1(c), w_2(d))}{p(w_1(c)).p(w_2(d))}\Big) \tag{26}$$

Where MI is equal to zero when the random variables w_1 and w_2 are both statistically unrelated,i.e. $p(w_1(c), w_2(d)) = p(w_1(c)).p(w_2(d))$, and $p(w_1(c), w_2(d))$ represents the joint mass probability between both variables w_1 and w_2.

2) Chi-square method

In statistics, the chi-square is typically used to examine the unrelated of 2 features. It is a computational test that determines the divergence from the expected apportionment while accounting for the fact that the variable occurrence is not associated with the choice of variable's value. Chi-square determines whether the appearance of a certain word and the appearance of a particular class are unrelated in feature selection. As a result, every term is assessed, and the terms are ranked according to their scores. A high rating suggests that the term's appearance and the class are correlated, and thus the null hypothesis of unrelated must be discarded. The feature is chosen for categorization process if the term and the class depend on one another. In general, the chi square score is computed from the following parameters such as false positives (fp), false negatives (fn), true negatives (tn) true positives (tp), probability of number of negative instances P_{neg} and probability of number of positive instances and P_{pos}.

$$chi{-}square{-}score = t(fn, (fn{+}tn).P_{pos}){+}t(tn, (fn{+}tn).P_{neg}){+}t(tp, (tp{+}fp).P_{pos}){+}t(fp, (tp{+}fp).P_{neg}) \tag{27}$$

Where t(observed value, expected value) = (observed value– expected value)^2/expected value.

The chi-squared approach involves the following stages:

1) Define the hypothesis
2) Create an assessment plan
3) Investigate the samples of the data
4) Determine the outcomes.

Create an assessment plan : after the hypothesis is declared, the assessment plan specifies how to use the data from the model to reject or accept the hypothesis.

- Importance Range: the researchers select an significance range that equals 0.01, 0.05, or 0.10, although it may be any number between 0 and 1.
- Testing approach: the chi-square test is employed to assess the degree of independence to determine if there is a significant association between both categorical variables.

Investigate the samples of the data: the sample of data must be examined to compute the degrees of liberty, the test value, the expected frequencies, and the P-value that is related to the test value.

- **degrees of liberty**

$$dl = (r-1) * (c-1) \tag{28}$$

Where *r* is the set of the levels of one categorical attribute and *c* is the set of the levels of the second.

- **Test statistic:**

$$x^2(f, c) = \frac{L * (AD - CB)^2}{(A+C)(A+B)(B+D)(C+D)} \tag{29}$$

Where:

A: represents the number of instances where the feature

'f' and category tag 'c' co-occur.
B: Number of cases where the "f" occurs without a "c".
C: Number of cases where the "c" occurs without a "f".
D: Number of instances where neither the "f" nor the "c" occurs.
L: Total number of instances.

3) ANOVA technique

In this proposal, two one-way ANOVA approaches, based on P-value and F-value, are utilized to statistically pick out the relevant features.

In the former one-way ANOVA approach, features were chosen based on the F-values and the specified percentile (p%) of the initial number of features. Only the features with the highest score (p%) were utilized to train the machine learning classifiers.

The second approach is dependent on the p-values of the one-way ANOVA, that identify the appropriate features of the classification process as well as a comparison with the level of significance. If the P-value of a variable is lower than the significance degree, the variable is retained for further processing. If not, it is rejected. The importance degree (α) is usually fixed at 0.05 [55].

The feature selection methodology for the two one-way ANOVA approaches is depicted in the Algorithms 1 and 2.

F. TWEET CLASSIFICATION STEP

A concatenation of multiple constrained Boltzmann machines (RBM) makes up the Deep Belief Network (DBN). Unsupervised training is used to develop an RBM into a feature extractor [56]. This approach can tackle complex data structures and find features that are inaccessible to direct detection. As a result, increasingly complicated structures can be represented by successively stacking RBMs. In fact, the current RBM is trained to recognize traits that were implicit in the prior RBM using the output of the previous RBM, etc. Generally, the layer-by-layer algorithm is used to train the DBN, and finding descriptive features that demonstrate the relationship between the inputs in each layer is one of its benefits [57]. The layer-by-layer learning approach makes it possible to optimize the weights within layers more effectively. Additionally, initializing the DBN probabilities may enhance the outcomes in comparison to using randomized weights. The advantages of DBN learning also include its capacity to lessen the negative impacts of underlearning and overlearning,where both affect large deep learning model and are prevalent issues. These factors led to the DBN being selected as the predictor in this study.

The probabilistic energy-based pattern is a popular approach [58] that is used to set up a joint relationship between the hidden variables *hv* and the observed data *ov*, as indicated below:

Algorithm 1: Pseudocode of one-way ANOVA based on F-values

Input : A pair (F, C), where F denotes the set of features retrieved by one of the used extractors, and C the class label of every feature. Also, %m is the percentage of the chosen features

Output: A chosen subset of relevant features according to the F-value.

Begin

$T_{Classes} \leftarrow$ find(C) // Extract the total number of class labels.

for *each* $F_j \in (F, C)$ **do**

$n_{instance\_per\_class} \leftarrow$ *find*(C_i)

$n_{total\_instance} \leftarrow$ *find(F, C)*

$d_1 \leftarrow T_{Classes} - 1$ *// rate of liberty between the classes*

$d_2 \leftarrow n_{total\_instance} - 1$ *//rate of liberty within the classes*

$sum_{square\_all\_features} \leftarrow \sum(\sqrt{F})$

$sum_{all\_features} \leftarrow \sum(F)$

$s_{of\_s\_all\_features} \leftarrow \sqrt{(sum\_all\_features)}$

$total_{sum\_square} \leftarrow sum_{square\_all\_features} - \frac{s_{of\_s\_all\_features}}{n_{total\_instance}}$

$v \leftarrow 0$

for *each* $c_i \in C$ **do**

$SS \leftarrow \sqrt{\sum(F_{c_i})}$

$v \leftarrow v + \frac{SS}{count(F,c_i)}$

end for

$v \leftarrow v$ - $\frac{s\_of\_s\_all\_features}{n_{total\_instance}}$ *//Somme of squares between categories*

$SS_{sc} \leftarrow total_{sum\_square} - v$ *//Somme of squares inside the categories*

$V_{bc} \leftarrow \frac{v}{d_1}$ *// Inter-class variance*

$V_{wc} \leftarrow \frac{SS_{sc}}{d_2}$ *// Intra-class variance*

F-value $\leftarrow \frac{V_{bc}}{V_{wc}}$

end for

Order ascending (F based on F -value)

BV $\leftarrow$ *Choose (The biggest %m of F according to the F-value)*

return *BV*

$$P(ov, hv_1, ..., hv_n) = \Big(\prod_{i=1}^{n} P(hv^i | hv^{i+1})\Big).P(hv^{n-1}, hv^n) \tag{30}$$

Where $P(hv^i|hv^{i+1})$ is a conditional probability distribution for the hidden-hidden neuron in an RBM connected to the earth layer of the DBN and $P(hv^{n-1}|hv^n)$ represents the joint hidden by hidden probability distribution in top-layer RBM.

At each layer, the computed outcome was taken as input for the subsequent layer.

Algorithm 2: Pseudocode of one-way ANOVA based on P-values

Input : A pair (F, C), where F denotes the set of features retrieved by one of the used extractors, and C the class label of every feature. Also, %m is the percentage of the chosen features

Output: A chosen subset of relevant features according to the F-value.

Begin

$T_{Classes} \leftarrow$ find(C) // Extract the total number of class labels.

for *each* $F_j \in (F, C)$ **do**

$n_{instance\_per\_class} \leftarrow$ *find*(C_i)

$n_{total\_instance} \leftarrow$ *find(F, C)*

$d_1 \leftarrow T_{Classes} - 1$ *// rate of liberty between the classes*

$d_2 \leftarrow n_{total\_instance} - 1$ *//rate of liberty within the classes*

$sum_{square\_all\_features} \leftarrow \sum(\sqrt{F})$

$sum_{all\_features} \leftarrow \sum(F)$

$s_{of\_s\_all\_features} \leftarrow \sqrt{(sum\_all\_features)}$

$total_{sum\_square} \leftarrow sum_{square\_all\_features} - \frac{s_{of\_s\_all\_features}}{n_{total\_instance}}$

$v \leftarrow 0$

for *each* $c_i \in C$ **do**

$SS \leftarrow \sqrt{\sum(F_{c_i})}$

$v \leftarrow v + \frac{SS}{count(F,c_i)}$

end for

$v \leftarrow v$ - $\frac{s\_of\_s\_all\_features}{n_{total\_instance}}$ *//Somme of squares between the categories*

$SS_{sc} \leftarrow total_{sum\_square} - v$ *//Somme of squares inside the categories*

$V_{bc} \leftarrow \frac{v}{d_1}$ *// Inter-class variance*

$V_{wc} \leftarrow \frac{SS_{sc}}{d_2}$ *// Intra-class variance*

F-value $\leftarrow \frac{V_{bc}}{V_{wc}}$

P-value $\leftarrow F\_survival(d_1, d_2, F - value)$ *//Feature' score*

If P-value < 0.05 then

Add a feature into the subset SF

end for

return *SF*

1) Restricted Boltzmann machine

The restricted Boltzmann machine is a type of Boltzmann machine without any inner link between the visible and hidden layers. In this pattern, the joint probability configuration (*k,hv*) is described as follows:

$$Jp(k, hv) = \frac{e^{-(Energy(k,hv))}}{nc} \tag{31}$$

Where $nc = \sum_{i,j} e^{-(Energy(k_i,hv_j))}$ is known as the normalization coefficient with *k* is the number of stacks of the restricted Boltzmann machine.

The probability of a visible neuron is obtained by the sum of all hidden neurons.

$$p(vn) = \frac{1}{nc} \sum_{hv} e^{-(Energy(k,hv))} \tag{32}$$

The derivative of the logarithm of the mentioned probability equation is described as follows:

$$\frac{\partial log(p(vn)}{\partial \theta}) = \frac{\partial \sum_{hv} e^{-(Energy(k,hv))}}{\partial \theta} = \frac{\partial log(nc)}{\partial \theta}) = \varphi^+ - \varphi^- \tag{33}$$

With φ^+ and φ^- being called positive and negative stages, respectively.

The estimation of the positive stage is straightforward due to the lack of inner link between the hidden or visible neurons. The conditional probability for every pair of hidden neurons is obtained by :

$$P(hv_i = 1|k) = \frac{e^{d_i + M_i.k}}{1 + e^{d_i + M_i.k}} = sig\Big(d_i + \sum M_i.k\Big) \tag{34}$$

Where M_i is the i*th* row of the matrix *M* and *sig(x)* represents the sigmoid activation function. The visible neurons can be reconstructed in the same manner as the hidden neurons.

The second negative phase, must be computed for every hidden and visible neuron. One algorithm suggested to approximate the gradient of log-likelihood is the divergence contrastive (DC). This particular algorithm has been employed to upgrade the learning metrics, weights and biases in every RBM. The benefit of this algorithm is obvious when applying parallel computing with Matlab.

By supposing that the hidden neurons are binary, all visible neurons are sorted into different classes according to the batch size set in the former stages. Afterwards, the hidden neurons are computed with the next equation (37).

$$P(hv_i) = sig\Big(d_i + \sum M_{i,j}.k_j\Big) \tag{37}$$

Lastly, the hidden neuron will be activated if the likelihood exceeds the threshold. For updating the visible neurons, it is usual to apply the probability, pi, which is calculated using the following equation (38):

$$P(vn_i) = b_i + \sum_i M_{i,j}.hv_j \tag{38}$$

After having computed the gradient, we can now update the metrics, the weights and the biases. Two major metrics, the learning ratio and the momentum, can enhance the upgraded metrics with respect to the former ones. The learning ratio is multiplied by the matrix ΔM. If this metric is too big, the reconstruction error will increase, and if it is too weak, the running time will be important. The better score for the learning ratio is combined with the averaged weights across multiple upgrades. Momentum is helpful to boost the learning velocity. It is applied after calculating the lot data

Algorithm 3: Stages carried out by the algorithm of contrastive divergence

Stage 1 Initialize n, m, N, W, a, b, and ϵ (learning ratio)
Stage 2 Then, assign a sample s as the initial state v_0 for the visible layer from the training data.
Stage 3 Based on the Equation (35) evaluate $P(hv_{0n} = 1|vv_0)$ and from the conditional distribution $P(hv_{0n} = 1|hv_0)$ extract $hv_{0n} \in \{0, 1\}$, where n=1,2,...,k

$$P(hv_i = 1|vv) = sig\Big(b_i + \sum_{j=1}^{k} M_{j,i}.vv_j\Big) \quad (35)$$

Stage 4 Based on the Equation (36) evaluate $P(vv_{1m} = 1|hv_0)$ and from the conditional distribution $P(vv_{1i} = 1|hv_0)$ extract $vv_{1i} \in \{0, 1\}$, where m=1,2,...,l

$$P(vv_j = 1|hv) = sig\Big(a_j + \sum_{i=1}^{k} M_{i,j}.hv_i\Big) \quad (36)$$

Stage 5 Using Equation (35), evaluate $p(hv_{1n} = 1|vv_1)$
Stage 6 Based on the subsequent equations update the parameter:

- $M = M + \epsilon\Big(P(hv_0 = 1|vv_0)vv_0^T - P(vv_0 = 1|hv_0)hv_0^T\Big)$
- a = a+$\epsilon(vv_0 - vv_1)$
- b = b+$\epsilon\Big(P(hv_0 = 1|vv_0) - P(hv_1 = 1|vv_1)\Big)$

Stage 7 Assign another sample as the initial state v0 for the visible layer from the training data and again iterate from steps 3 to 7. Continue this process, till applying Ntraining data for processing.

and maintaining the metrics, therefore it is multiplied by M_{old}

All the stages of training the RBM approach can be summarized as follows:

1) Identify the necessary metrics:
   a) ϵ : Learning ratio for gradient descent in a stochastic manner
   b) μ :Momentum for the update of the metrics
   c) Hidden and visible biases set to a zero number
   d) The starting weights variables are defined as a random value with a Gaussian probability distribution.
   e) The total number of hidden neurons.
   f) The total number of layers
   g) Lot size for divergence contrastive sampling
2) Calculate φ^+ = hv.$\hat{\text{k}}$
3) Calculate $\tilde{\text{h}}$v = $P(hv_j = 1)$ and $\tilde{\text{k}} = P(k_j = 1)$ for all i,j applying divergence contrastive algorithm.
4) Calculate φ^- = $\tilde{\text{h}}$v.$\tilde{\text{k}}$
5) M_{ij}^{new} = $\mu.M_{ij}^{old}$+ $\Delta M_{i,j}$,$\Delta M_{i,j} = \epsilon(\varphi^+ - \varphi^-)$
6) b_{ij}^{new} = $\mu.b_{ij}^{old}$+ $\Delta b_{i,j}$,$\Delta b_{i,j}$ = ϵ(k - $\tilde{\text{k}}$)
7) c_{ij}^{new} = $\mu.c_{ij}^{old}$+ $\Delta c_{i,j}$,$\Delta c_{i,j}$ = ϵ(hv - $\tilde{\text{h}}$v)

G. PARALLELIZATION USING HADOOP

Behind the so-called "big data" systems, we have a core concept: which is distributing both data and treatments on a set of machines/computers that form a cluster using the framework Hadoop. In this framework, the storage of raw data is most often based on a distributed file system and MapReduce is the first implementation for big data of the principle of parallel processing applied to distributed files [59]. It is based on two main functions, map and reduce, which are sometimes applied multiple times. The former function applies a transformation to the values of a collection of data in the key/value format; the latter applies an operation to all the values of the same key. In this contribution, in order to resolve this runtime issue encountered by our proposed hybrid approach, we have utilized the Hadoop framework. This framework provides us the ability to parallelize our proposed approach across five computational nodes: four slaves nodes and one master node. This framework employs its HDFS with which to stock the **sentiment140** datasets to be evaluated and the classification decision. Furthermore, it uses the programming model MapReduce, which handles and evaluates our fuzzy DBN jobs in a parallel mode through the use of various reducers and mappers as depicted in the figure 14. The implementation of our suggested hybrid model on the MapReduce programming pattern mainly comprises 3 steps: the Map stage, the Combining phase and the Reduce phase, briefly outlined as follows:

Map phase: The mapping stage is composed of 4 mappers. Every mapping function (Mapper) takes one or more pieces of input data from the HDFS as input data under form of different key-value pairs. Each mapper implements the semi-automatic labelling process (Vedar+FuzzyLogic) on every piece of data, then store the labelling process results in the HDFS as a first phase of our proposed approach. In its second phase, each mapping function takes one more pieces of labelled data and applies on them the data preprocessing tasks, then it turns out them into numerical vector space by using one of the presented features extractors previously. Furthermore, the mapper applies one of the outlined feature selectors previously for reducing the dimensional vector spaces of the extracted features. Finally, the mapper applies the deep belief classifier into the treated pieces of data. After processing all pieces of the input data, the results produced employing our proposed approach are converted into a set of key-value intermediate pairs and are written to the local hard drive.

Aggregation by keys: The MapReduce pattern performs this process. Its primary purpose is to gather all resulting intermediate values from the Mapper which have the identical intermediate key into a set of values in the form of an array

and then pass it to the Reducer.

Reduce stage: In this contribution, the reducer stage is composed of 4 reducers. Every reducer gets the array list of intermediate values from all mappers. The reducer operates on a single key at the same time and merges the list of values related to this key into a smaller ensemble. Then finally, all reducers outcomes are merged and combined into a single intermediate outcome, and this combined outcome is then written as an outcome key-value pair on the HDFS, as illustrated in figure 14.

The benefit of Hadoop framework is its capability to prevent the issue of computer nodes crashing by providing redundant storage of information on multiple computational computers, which allows for automatic data backup. This means the same piece of data is saved on different computational computers. If one computational server crashes, its same piece of data is always ready for use on another computational computer. The MapReduce scheduling framework is a software that provides scaling and reliability requirements for handling and running of distributed operations. More specifically, this scheduling scheme decomposes automatically the calculations into several parallelization jobs. For instance, if a single job is unable to complete its workload, it may be reloaded with no negative impact on the other running jobs. MapReduce avoids the network bottleneck issue by placing computational jobs nearer to the data being stored and disallowing data to be copied across the network, thereby reducing the network bottleneck problem and balancing the computational and information load. The MapReduce pattern also gives its adopters a very easy and simple pattern that removes the complexities of all computational jobs associated with its operation.

In order to minimize the running time and boost the effectiveness of our proposal, we used the Hadoop framework in this research. Our proposed solution requires for the use of a sizable dataset (Sentiment140). In the initial phase, we employed HDFS to divide and store the massive dataset among all of the Hadoop cluster's computing devices in parallel way. The stage that comes after putting the dataset in HDFS is applying our suggested strategy to the prestored dataset. We have applied the MapReduce scheduler paradigm in this stage 2 to parallelize our strategy among all computational devices in the Hadoop cluster. Each round of the Hadoop Mapreduce begins with a tweet that needs to be classified, and the output is a classified tweet. Every tweet's classification outcome will likewise be stored in the HDFS. Figure 14 provides an outline of all these phases, and the Algorithm 4 presents the MapReduce technique used in our suggestion for categorizing tweets.

H. CROSS-VALIDATION METHOD

Cross-validation strategy is one of the most widely employed techniques for adjusting hyper-parameters (CV), which is addressed in the survey [60]. The 10-fold CV computes ten measurement scores for each hyper-parameter adjustment,

Algorithm 4: Our deep fuzzy belief system based on MapReduce

Input : collection of tweets
Output: the classification decision.
Configure the Hadoop job
Create a Mapper class called ChooseJobMapper.
Create a Reducer class called ChooseJobReducer.
Adjust the HDFS batch size till the corpus of tweets **C** can be subdivided into **D** divisions
$C_l = \{l = 1, 2, ..., D\}$
In the l-th ChooseJobMapper
Input : $C_l = \{c_1, c_2, ..., c_D\}$ is the dataset without labels
Output: (keyM,valueM)=($c_l, ClassTag$)
for $l \leftarrow 1$ **to** D *divisions* **do**
  preprocessing stage for tweets
  employing Vader's lexicon to calculate:
    1) Positivity Score (PS)
    2) Negativity Score (NS)
  Applying the Mamdani fuzzy system, do the following:
    1) Phase of fuzzification
    2) Initial sub-stage, referred to as Application
    3) Second sub-stage, referred to as Implication
    4) Third sub-stage, referred to as Aggregation
    5) Phase of defuzzification
  Mapper-Outcome :
  (keyM,valueM)=($c_l, ClassTag$)
end for
In the l-th ChooseJobReducer
Input : (keyR,valueR)=($c_l, list(ClassTag_i)$)
Output: (keyR,valueR)=($c_{l^*}, ClassTag_i^*$)
for $l \leftarrow 1$ **to** D *Divisions* **do**
  $\text{ClassTag}_i^* = \sum_{j=1}^{D} ClassTag_j.$
end for
In the l-th ChooseJobMapper
Input : $C_l^* = \{c_1^*, c_2^*, ..., c_D^*\}$ is the tagged corpus.
Output: (keyM,valueM)=($c_l^{**}, ClassTag^{**}$)
for $l \leftarrow 1$ **to** D *Divisions* **do**
    1) Stage for tweet vectorization
    2) Phase of selecting the relevant features
    3) Step for classifying tweets
  . **Mapper-Outcome :**
  (keyM,valueM)=($c_l^{**}, ClassTag^{**}$)
end for
In the l-th ChooseJobReducer
Input : (keyR,valueR)=($c_l^{**}, list(ClassTag_i^{**})$)
Output: (keyR,valueR)=($c_l** *, ClassTag_i* * *$
for $l \leftarrow 1$ **to** D *Divisions* **do**
  $\text{ClassTag}_i^{***} = \sum_{j=1}^{D} ClassTag_j^{**}$
end for
return *Decision of the classification: ClassTag*$_i^{***}$

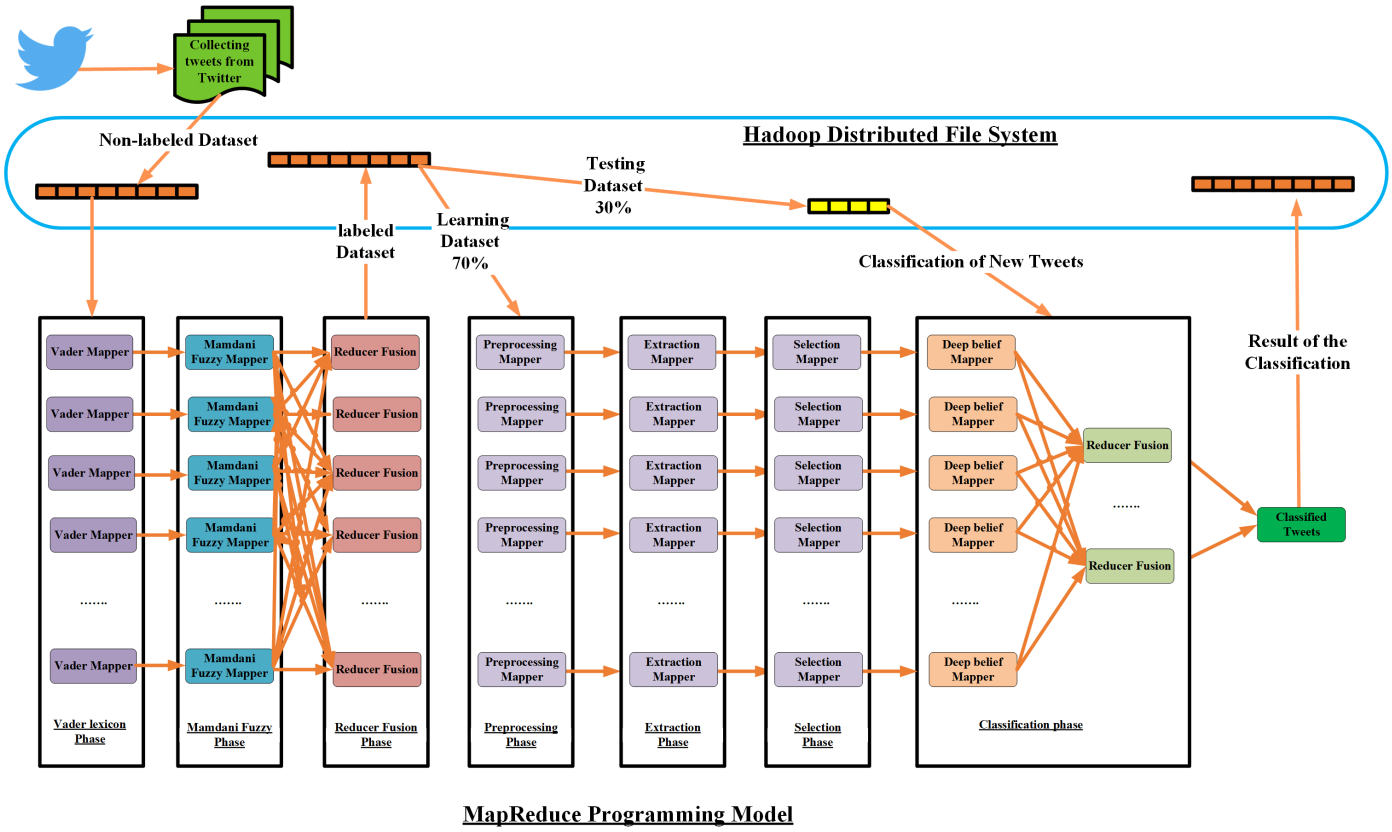

FIGURE 14. Our proposal in Hadoop Framework.

similar to the common CV. The average performance measure is then determined for each hyper-parameter. The final outcome metric for the pattern is the highest median performance measure. The hyper-parameters of our DBN have been modified in this contribution according to the provided values as shown in the table 3.

TABLE 3. Defining the DBN model's parameters.

Metrics	Values
Rate of learning	0.0001
the output layer's activation function	Sigmoid
The hidden layer's activation function	Relu
Number of hidden neurons in each hidden layer	200-225
Number of used layers	6-8
number of back-propagation iterations	125
Batch size	32 and 64
RBM' rate of learning	0.0015
Output	1
number of epochs	50

Choosing the hyper-parameters, like the number of hidden units, RBM and DBN learning rates, the number of epochs, the batch size, the number of hidden and visible layers and the depth of features has a serious influence on the classification's accuracy and the calculation's complexity. The precision of DBN might not be more effective than conventional ML methods. if the depths are wrongly defined. All possible combinations of values should be investigated in order to discover the ideal number of concealed and exposed RBM layers and the ideal number of concealed neurons. For the parameter estimation in this work, we applied the grid search strategy. The grid search methodology is primarily an effective way to select the optimal values for a particular problem's or algorithm's hyper-parameters that result in greater effectiveness. In this proposal, the best hyper-parameter settings for our Proposed deep learning classifier are selected using the CV of 10 folds with a grid search algorithm. However, our experimental results provide some guidance on reliable ranges for hyper-parameters of our DBN classifier, i.e., It seems sufficient to have 6–8 RBM hidden layers with 200–225 hidden neurons per hidden layer, the number of epochs equals 50, the batch size is either 32 or 64,Learning rates for RBM and DBN are 0.0015 and 0.0001, respectively. Additionally, there have been 125 back-propagation rounds.

I. TRAINING AND EVALUATION DATA SET

After the semi automatic labelling stage, we split the labelled dataset into three subsets. The partitioning of the dataset is intended to ensure the representativeness of the training dataset utilized for model building. The three subsets are

presented as follows:

Learning subset: The training subset is a subset of examples employed for learning, which involves adjusting the parameters of a pattern. For instance, a training subset is employed to train the weights of our fuzzy deep belief network. Furthermore, the training set must cover most of the predicted variability of the future example in the data spaces to obtain better models.

Validation subset: The validation subset is a subset of examples taken to adjust the parameters or the structure of a given model. For example, a validation subset is employed to set the number of hidden layers with the number of hidden neurons in the deep belief network.

Testing subset: The testing subset (forecast subset) is a subset of examples utilized purely to evaluate the effectiveness of a fully defined pattern.

To train our system we have taken 60% of the corpus, to validate it we have used 10% and 30% to evaluate it as described in the following table 4.

TABLE 4. Training, validation and evaluation dataset.

Learning Set	Valuation set	Testing set
960000	160000	480000

J. ASSESSMENT CRITERIA

It's crucial to assess a classification system's effectiveness [61]. Numerous metrics are available to evaluate the effectiveness of paradigms, Each of which has distinct qualities and it is often necessary to combine several of them in order to get a full picture of how well our model is performing. Nine assessments [62] were employed in this paper, and they are as follows:

Precision : is used to gauge the accuracy of a classification system. Greater precision implies fewer fake positive cases, while lower precision implies more fake positive cases. This measure is computed employing the next equation (39).

$$Precision = \frac{vp}{vp + fp} \tag{39}$$

Recall : is used to gauge the correctness, or sensibility, of a classification system. Highest recall implies least fake negative cases, while lower recall implies more fake negative cases. This measure is computed employing the next equation (40).

$$Recall = \frac{vp}{vp + fn} \tag{40}$$

F1-Score : represents the weighted harmonic mean of the recall and precision. This metric is computed utilizing the subsequent equation (41).

$$F1 - Score = 2.\frac{Precision.Recall}{Precision + Recall} \tag{41}$$

Classification rate : enables for symmetrical evaluation of the model's performance on both negative and positive items. It gauges the ratio of all items with accurate predictions. The following equation is used to determine this metric (42):

$$ClassificationRate = \frac{tn + tp}{tn + fn + tp + fp} \tag{42}$$

False Positive Rate: represents the ratio used to identify the ineffectiveness of a classification system and to calculate the mis-categorization ratio by computing the number of cases that are in fact negatives but that the classification system has forecasted as positives. The false positive ratio is measured by utilizing the formula (43).

$$FalsePositiveRate = \frac{fp}{fp + tn} \tag{43}$$

Specificity: is used to gauge how effective a classifier is at finding the total number of cases with the class label negative. This measure is calculated according to (44).

$$Specificity = \frac{tn}{fp + tn} \tag{44}$$

False Negative Rate: represents the ratio used to identify the ineffectiveness of a classification system and to calculate the miscategorization ratio by computing the number of cases that are in fact Positives but that the classification system has forecasted as Negatives. The false Negative ratio is measured by utilizing the formula (45).

$$FalseNegativeRate = \frac{fn}{fn + tp} \tag{45}$$

Error ratio: metric is employed to gauge the miscategorization ratio, i.e., this specific measure determines the number of incorrect classification cases over all cases in the utilized corpus. Usually, its purpose is to assess the effectiveness of the classification system in limiting misclassifications. The error ratio is given by the following formula (46)

$$ErrorRatio = \frac{fp + fn}{tp + fp + tn + fn} \tag{46}$$

Kappa statistic: is a measurement of the performance of a classifier that is used to compare an actual accuracy with an estimated accuracy. It is employed not only for evaluating a particular classifier but also for inspecting classifiers against each other. The kappa statistic obtained by calculating the following formula (47)

$$Kappa - Statistic = \frac{P_0 - P_e}{1 - P_e} \tag{47}$$

Where : $P_0 = \frac{tp+tn}{100}$; and $P_e = [\frac{tp+fn}{100} * \frac{tp+fp}{100}] + [\frac{fp+tn}{100} * \frac{fn+tn}{100}]$

Where :

fn : Number of twitter posts which are positive in reality and forecast to be negative.

tp : Number of twitter posts which are positive in reality and forecast to be positive.

fp : Number of twitter posts which are negative in reality and

forecast to be positive.
tn : Number of twitter posts which are negative in reality and forecast to be negative.

V. RESULTS

The findings of the numerical experiments will be discussed in this section. To evaluate the effect of each preprocessing technique on the corpus, a first experiment was conducted. The objective of the second experiment is to identify the most effective feature extractor. Finding the higher efficiency characteristic selector is the goal of the third experiment. And the final one displays the outcomes of various combinations used in our strategy.

A. INFLUENCE OF DATA PRE-PROCESSING

In this experiment, we study the effect of data pre-treatment. We confirm that the use of various pre-treatment techniques yields varying classification results. In addition, not all methods of data pre-treatment are required, and a error rate is computed to determine if every data pre-treatment technique is required and which one is most efficient. In order to handle complex datasets, there are six principal approaches of data pre-treatment that can be applied: field-delete technique, normalized technique, exponent change technique,PCA technique, global ratio change technique, and local ratio change technique.

1) Field-delete technique

If more than 99 percent of the dimensions' data values are equal to 0. This feature (dimension) requires removal because the feature values do not provide sufficient information for the purpose of designing the classifier. This method decreases the features' size of the initial models and simplifies the design of the classifier.

2) Normalized technique

If the average or the co-variance of the values of data in a feature is extremely large, we need to standardize the values of data in that feature so that the center is 0. The basic calculation of this technique is given by the following formula (48)

$$y_{ij} = \frac{y_{ij} - y_{iaverage}}{covar_i} \tag{48}$$

Where y_{ij} represents the j-th value of i-th feature, $y_{iaverage}$ is the average of the values of the i-th feature data, $covar_i$ denotes the co-variance of the i-th feature data.

3) Exponent change technique

If the standardized approach fails to condense the feature data, the exponent change methodology is a suitable option to minimize all feature values. Its data value range is [0,1] and the j-th value of the i-th feature (y_{ij}) is computed by the Equation (49).

$$y_{ij} = e^{y_{ij}} \tag{49}$$

4) Local ratio change technique

If certain feature values are extremely large or extremely small in certain features, but in other features the values of the data are approximately the same, then a local ratio change approach can be a useful option. The basic calculation of this approach is described by the next Equation (51)

$$y_{ij} = \frac{y_{ij} - y_{imin}}{y_{imax} - y_{imin}} \tag{50}$$

Where y_{imax} indicates the maximum values of i-th feature and y_{imin} represents the minimum values of the i-th feature.

5) PCA technique:

PCA is in close relationship with the factor analysis; it utilizes a characteristic matrix to map the models into a novel space. PCA allows models to be projected from a high-dimensional space into a low-dimensional space, where the models can be representative of the original models. The stages of the PCA are the following:

1) Calculate the model dispersion matrix.
2) Calculate the values and the vector spaces of each feature.
3) Order the values of the characteristics from the largest to the smallest.
4) select over 86% of the major features and then aggregate the respective feature vectors in the form of a projection matrix.
5) Utilize the projection matrix to project the models in the initial area into a new area, where the size of the area is given by the dimensions of the matching feature vectors in this mapping matrix.

6) Global ratio change technique

This approach has a similar functionality as the local approach, the only difference is that the local approach modifies every feature by different minimum and maximum data values in every feature, whereas the global approach modifies every feature by the minimum and maximum values of all the dataset values. Its computing result is given as follows:

$$y_{ij} = \frac{y_{ij} - y_{min}}{y_{max} - y_{min}} \tag{51}$$

Where y_{max} indicates the maximum values and y_{min} represents the minimum values of the whole dataset.

Table 5 displays the findings in terms of error rate (ER), accuracy (AC) and runtime (RT with Hadoop) after applying each explained data pretreatment previously on the Sentiment140 dataset.

From table 5, we deduce that our first experiment is divided into several parts. The former part aims the application of every data pretreatment separately (PP). In this part of the experiment, we notice that the Global ratio change method attains a high accuracy equal to 75.02 % , a minimal error rate equal to 24.98% and a less runtime equal to 2.13s.

Furthermore, we remark that the exponent change data preprocessing method outperforms the normalized method with an accuracy equals 74.07%, and an error rate equals 25.93%. Therefore, in the next experimental part, we will apply only the exponent change method. Because that the exponent change and normalized methods aim to carry out the same functionality and the former method outperforms the latter one in terms of error rate and accuracy.

In the second experimental part (SP), we notice that the aggregation **Exponent change+Global ratio change** gives high accuracy equal to 89.47%, with minimal error rate equal to 10.53% and less execution time equal to 5.63s. Therefore, in the third experimental part (TP), we will keep this combination and we will vary the other methods.

In the third experimental part, we remark that the combination **Exponent change+Global ratio change+Field-delete** outperforms all other combinations since it reaches an accuracy equal to 98.56%, an error rate equal to 1.44% and a minimal runtime equal to 8.52s. Moreover in the fourth experimental part (FP), we will keep the combination **Exponent change+Global ratio change+Field-delete** and we will change the other approaches.

In the fourth experimental part, we notice that the aggregation **Exponent change+Global ratio change+Field-delete+PCA** perform better than the combination **Exponent change+Global ratio change+Field-delete+Local ratio change** , since it gives a high accuracy equal to 98.96%, a less error rate equal to 1.04% and a minimal runtime equal to 11.32s. Consequently, in the rest of this contribution, we will only apply the combination **Exponent change+Global ratio change+Field-delete+PCA** as a data preprocessing technique.

B. DATASET AFTER LABELING

As was previously stated, the sentiment140 corpus's tweets were incorrectly labeled since its designers believed that every tweet with positive emoticons, like :), is positive, and that tweets using emoticons denoting sadness, like :(, are negative. As a result, we made the decision to re-label by fusing the Vader lexicon with Mamdani's fuzzy model, which is founded on manually made rules. Our re-labelling process aims to label each tweet by computing the semantic orientation of their composed words and by employing the Mamdani fuzzy system to deal with uncertain and vague tweets. As we said previously, the Mamdani fuzzy system consists of fuzzification and defuzzification process. Also, The fuzzification/defuzzification process is carried out by applying different fuzzification/defuzzification approaches. Therefore, a comparative study is performed in order to determine the fuzzification/defuzzification combination with the highest performance in terms of classification and error rates as depicted in the table 6.

From the table 6, we notice that the combination Gaussian function as fuzzification method and Centre of Largest Area as defuzzification approach is the best fuzzification/defuzzification combination with the highest classification rate (98.96%) and lower error rate (1.04%).

Figure 15 describes the datasets before the re-labeling process and the figure 16 shows the dataset after re-labeling.

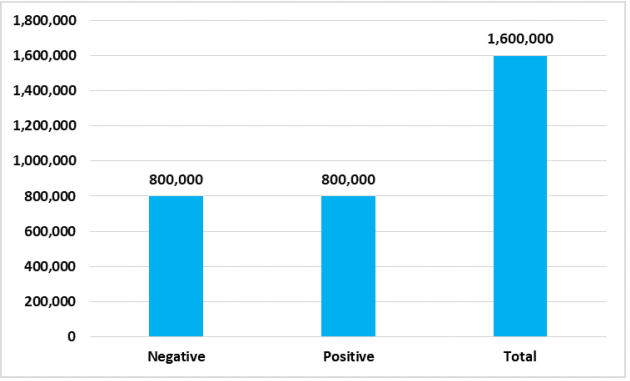

FIGURE 15. data set before labeling.

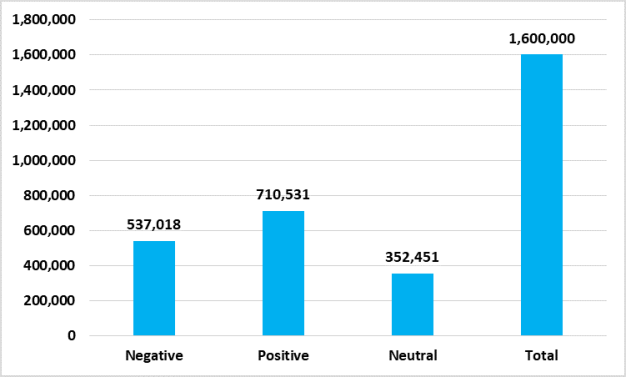

FIGURE 16. data set after labeling.

From the figures 15 and 16, we remark that our hybrid process re-labeled the dataset into three class label: Negative label represents 34% , Positive label represent 44%, and Neutral represents 22% of the whole dataset instead of both class label Negative and Positive in the original dataset and every class label represents 50% of the whole dataset. According to this re-labelling process, we deduce that the original data set is mislabelled with 22% as error rate.

C. A FEATURE EXTRACTORS EVALUATION

The aim of this step is to generate a collection of digital vectors from the input tweets. Based on error rate and accuracy, with regard to the feature extractors employed in this contribution, we conducted a second experiment to identify the most effective feature extractor: GloVe, TF-IDF Trigram, TFIDF Bigram, and TF-IDF Unigram. The accuracy and error rate achieved by implementing GloVe, TF-IDF Trigram, TFIDF Bigram, and TF-IDF Unigram are shown in figure 17.

As depicted in figure 17, TF-IDF Trigram surpasses the performance of other extractors based on the accuracy and

TABLE 5. ER, AC and RT of different data pretreatment techniques.

Experiment	Name	ER(%)	AC(%)	RT(s)
PP	No pretreatment	45.91	54.09	0.00
	Normalized	39.32	60.68	3.27
	Field-delete	37.56	62.44	2.84
	Exponent change	25.93	74.07	4.96
	Local ratio change	40.62	59.38	2.13
	PCA	30.09	69.91	2.48
	Global ratio change	24.98	75.02	3.67
SP	Exponent change+Field-delete	14.04	85.96	7.35
	Exponent change+Local ratio change	20.15	79.85	7.09s
	Exponent change+PCA	17.41	82.59	7.97
	Exponent change+Global ratio change	10.53	89.47	5.63
TP	Exponent change+Global ratio change+Field-delete	1.44	98.56	8.52
	Exponent change+Global ratio change+Local ratio change	9.33	90.67	12.94
	Exponent change+Global ratio change+PCA	3.8	96.20	10.28
FP	Exponent change+Global ratio change+Field-delete+Local ratio change	10.96	89.04	15.36
	Exponent change+Global ratio change+Field-delete+PCA	1.04	98.96	11.32

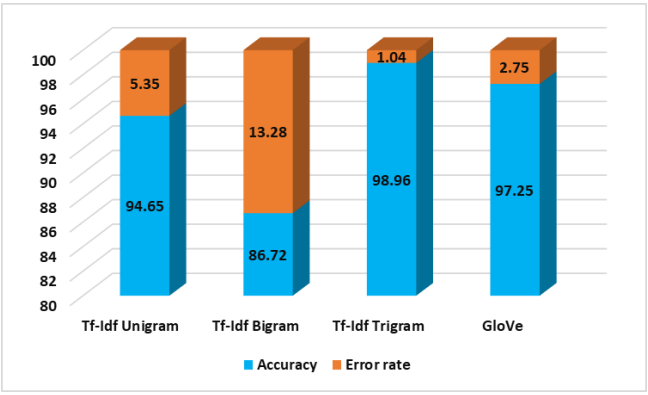

FIGURE 17. Evaluation of feature extractors.

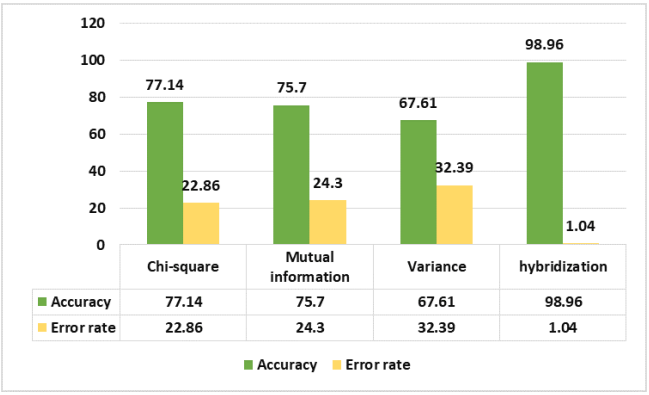

FIGURE 18. Analysis of feature selectors.

error rate because it attained an accuracy equal to 98.96% and an error rate of 1.04%.

D. FEATURE SELECTORS ANALYSIS

The feature selection process, as previously mentioned, comes after the feature extraction step. We use a variety of approaches in this phase, including analysis of variance, mutual information and Chi-square, and after that, we combine these 3 methods. Consequently, the goal of this experiment is to compare the accuracy and error rate of all available feature selectors in order to identify the optimal one. Figure 18 outlines the accuracy and error rate that were obtained using various feature selection strategies.

As illustrated in figure 18, We see that, in terms of accuracy and error rate, the suggested hybridized selector exceeds the performances of other selectors, because it got an accuracy rate of 98.96% with a rate of error equal to 1.04%

E. THE OVERALL EFFECTS OF THE SUGGESTED MODEL

To underline the significance of the proposed model and make it clear how it affects the classification outcomes of tweets; We conducted this experiment to compute the accuracy rate, error rate, runtime, kappa-statistic, specificity, precision, false positive rate, F1-score, recall and false negative rate for every hybridization (Extractor+Selector+Classifier) and then examine it to see which model performs best.

From the results shown in tables (7, 8 and 9), we note that

TABLE 6. Classification rate, Error rate of different combinations Fuzzification/Defuzzification.

Extractor	Selector	Classification rate(%)	Error rate (%)
Triangular function	Max-Membership Principle	80.69	19.31
	Centroid Method	85.51	14.49
	Weighted Average Method	82.65	17.35
	Mean-Max Membership	73.54	26.46
	Centre of Sums	90.37	9.63
	Centre of Largest Area	91.86	8.14
	First of maxima	89.34	10.66
	Last of maxima	87.92	12.08
Trapezoidal function	Max-Membership Principle	80.91	19.09
	Centroid Method	85.79	14.21
	Weighted Average Method	81.67	18.33
	Mean-Max Membership	79.96	20.04
	Centre of Sums	92.02	7.98
	Centre of Largest Area	93.46	6.54
	First of maxima	90.10	9.9
	Last of maxima	92.03	7.97
Monotonically increasing function	Max-Membership Principle	79.53	20.47
	Centroid Method	70.85	29.15
	Weighted Average Method	78.41	21.59
	Mean-Max Membership	79.68	20.32
	Centre of Sums	80.17	19.83
	Centre of Largest Area	76.13	23.87
	First of maxima	74.59	25.41
	Last of maxima	75.46	24.54
Monotonically decreasing function	Max-Membership Principle	68.71	31.29
	Centroid Method	69.24	30.76
	Weighted Average Method	70.34	29.66
	Mean-Max Membership	69.48	30.52
	Centre of Sums	66.98	33.02
	Centre of Largest Area	71.69	28.31
	First of maxima	70.14	29.86
	Last of maxima	67.99	32.01
Gaussian function	Max-Membership Principle	84.79	15.21
	Centroid Method	89.56	10.44
	Weighted Average Method	96.15	3.85
	Mean-Max Membership	89.97	10.03
	Centre of Sums	94.58	5.42
	Centre of Largest Area	98.96	1.04
	First of maxima	90.43	9.57
	Last of maxima	92.75	7.25

TABLE 7. Classification rate, Error rate, Run-time and Kappa-Static of different combinations.

Extractor	Selector	Classification rate(%)	Error rate (%)	Run-time (s)	KappaStatistic (%)
TF-IDF Unigram	Chi-square	67.14	32.86	35.12	69.32
	Mutual information	75.70	24.3	30.61	76.51
	Variance (ANOVA)	57.61	42.39	26.54	56.01
	Hybridization	83.73	16.27	35.98	81.26
TF-IDF Bigram	Chi-square	58.89	41.11	37.87	60.43
	Mutual information	72.01	27.99	40.19	75.19
	Variance (ANOVA)	50.69	49.31	28.32	51.58
	Hybridization	80.12	19.88	37.89	79.82
TF-IDF Trigram	Chi-square	61.05	38.95	33.54	65.34
	Mutual information	69.75	30.25	29.67	70.26
	Variance (ANOVA)	52.14	47.86	30.04	50.84
	Hybridization	98.96	1.04	32.95	99.10
GloVe	Chi-square	72.17	27.83	45.62	71.92
	Mutual information	50.42	49.58	47.68	53.07
	Variance (ANOVA)	66.10	33.9	39.24	67.89
	Hybridization	79	21	36.49	80.64

TABLE 8. Precision, Recall, F1-Score and Specificity of different combinations.

Extractor	Selector	Precision(%)	Recall (%)	F1 Score (%)	Specificity (%)
TF-IDF Unigram	Chi-square	67.14	69.54	68.32	70.50
	Mutual information	76.18	74.64	75.40	73.91
	Variance (ANOVA)	58.44	57.62	58.03	56.09
	Hybridization	84.13	85.71	84.91	84.59
TF-IDF Bigram	Chi-square	57.98	58.65	58.31	60.73
	Mutual information	71.15	72.88	72.00	73.21
	Variance (ANOVA)	51.96	50.13	51.03	52.64
	Hybridization	81.22	83.55	82.36	82.36
TF-IDF Trigram	Chi-square	60.50	62.59	61.53	63.75
	Mutual information	68.47	69.32	68.89	67.49
	Variance (ANOVA)	53.40	56.36	54.84	55.06
	Hybridization	98.96	99.75	99.35	98.67
GloVe	Chi-square	71.28	70.39	70.83	72.34
	Mutual information	51.26	52.81	52.02	50.97
	Variance (ANOVA)	67.13	68.92	68.01	67.58
	Hybridization	79	78	78.49	77.59

TABLE 9. FalsePositiveRate and FalseNegativeRate of different combinations.

Extractor	Selector	FalsePositiveRate(%)	FalseNegativeRate (%)
TF-IDF Unigram	Chi-square	30.46	29.5
	Mutual information	25.36	26.09
	Variance (ANOVA)	42.38	43.91
	Hybridization	14.29	15.41
TF-IDF Bigram	Chi-square	41.35	39.27
	Mutual information	27.12	72.88
	Variance (ANOVA)	49.87	47.36
	Hybridization	16.45	17.64
TF-IDF Trigram	Chi-square	37.41	36.25
	Mutual information	30.68	32.51
	Variance (ANOVA)	43.64	44.94
	Hybridization	0.25	1.33
GloVe	Chi-square	29.61	27.66
	Mutual information	47.19	49.03
	Variance (ANOVA)	47.19	32.42
	Hybridization	22	22.41

the hybridization (TF-IDF Trigram+Hybridization+DBN) delivered the best performance in terms of classification rate (98.96%), error rate (1.04%), runtime (32.95s), kappa-static (99.10%), precision (98.96), recall(99.75%), F1-score (99.53%), specificity (98.67%), false positive rate (0.25%) and false negative rate (1.33%).

F. COMPARATIVE STUDY BETWEEN OUR APPROACH AND BASELINE ALGORITHMS

Inside this subsection, we compare the outcomes of the suggested technique with those of Sentiment140, which is a Twitter project that automatically classifies sentiment. In this project, they classified tweets through the use of seven distinct ML techniques, including naive Bayes (NB), Support vector machines (SVM), Maximum Entropy (MaxEnt), Iterative Dichotomiser 3 (ID3), C4.5, random forest (RF), k-nearest neighbors (KNN) and our approach. In figure 19 We describe all the findings from this project and those of the suggested model according to the assessment metrics : Accuracy, Recall, F-score and Precision.

We were able to draw the conclusion from the comparative analysis that the outcomes produced by our approach significantly outperform those of the seven algorithms, This demonstrates the benefit of employing a deep learning technique for tasks like these involving sentiment analysis and the strong results we obtain when dealing with large amounts of data as opposed to using conventional ML algorithms.

G. COMPARATIVE STUDY BETWEEN OUR APPROACH AND DEEP LEARNING ALGORITHMS

In this subsection we discuss the experimental findings obtained by the implementation of our fuzzy deep belief model and the deep learning algorithms such as Convolutional neural network (CNN), Feedforward neural network (FFNN), Recurrent Neural Network (RNN), Long short-term memory (LSTM). The obtained results in terms of recall, precision, F1-score, and classification rate for this experiment are illustrated in figure 20.

From the figure 20, we remark that our fuzzy deep belief network outperforms all other deep learning models in terms

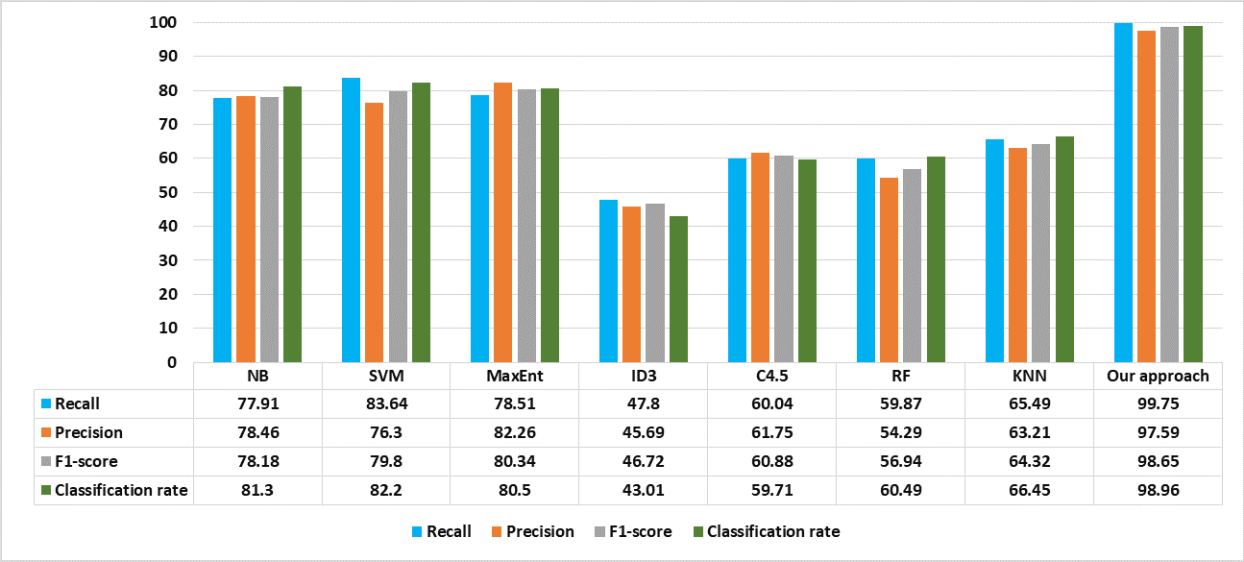

FIGURE 19. Comparative analysis of our method and baseline algorithms.

of four evaluation criteria. Since it achieves a recall equal to 99.75%, a precision equal to 97.59%, a F1-score equal to 98.65% and a classification rate equal to 98.96%. These obtained results shown that the fuzzy deep belief network has the ability to overcome the overfitting and underfitting issues.

VI. DISCUSSION.

For further assessment of our newly suggested fuzzy deep belief classifier, we conducted one more experiment which aims to compare our classifier with the other chosen classifiers from literature which are Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65]. However, in this instance, the assessment measures employed will be false negative rate, recall, runtime, convergence, stability, F1 score, accuracy, error rate, kappa-Static, complexity, false positive rate, precision rate and specificity rate, as discussed in the subsection on evaluation criterias. This comparison is done utilizing the dataset Sentiment140. Its empirical findings are displayed in the figure 21.

From the results shown in the figure 21, we remark that our approach obtained the strongest performances in terms of classification rate (98.96%), error rate (1.04%), runtime (32.95s), kappa-static (99.10%), precision (98.96), recall(99.75%), F1-score (99.53%), specificity (98.67%), false positive rate (0.25%) and false negative rate (1.33%) compared to other chosen classifiers from the literature which are Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65].

A. COMPLEXITY,CONVERGENCE AND STABILITY

In this experience, we have evaluated the effectiveness of our developed hybrid model and the Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65] methods chosen from the literature to train the Sentiment140 dataset by measuring the space and time complexity.

Table 10 reports the complexity's experimental results in terms of space after calculating the memory space seized by the allocation of the parameters and the implementation of the instructions of our developed hybrid model, Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65].

As depicted in table 10, we observe that our suggested hybrid model has operated on numerous instructions that seize a memory space equal to 11.59 M when training the Sentiment140 corpus. Besides, the memory space allocated by our approach's parameters equal to 7.31 M when training the Sentiment140 corpus. As the empirical results reported, our innovative hybrid model occupied much lower memory space in comparison to Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65] approaches.

Table 11 displays the obtained experimental results concerning the complexity in terms of time after calculating the training and testing time spent by our developed hybrid approach and the other evaluated approaches.

As shown in table 11, we notice that our developed hybrid pattern has consumed a training time equal to 11.92 s, in the case of the Sentiment140 corpus. In addition, our proposed hybrid model has expended a testing time of 3.56 s in the case of Sentiment140 corpus. As the acquired practical results reported, our offered hybrid model has a much lower time-complexity in comparison to Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65]. This accurate performance in terms of time-complexity attained by our hybrid pattern is a result of using the Hadoop cluster, which comprises of twelve computational nodes: eleven subordinate nodes and one supervisor node.

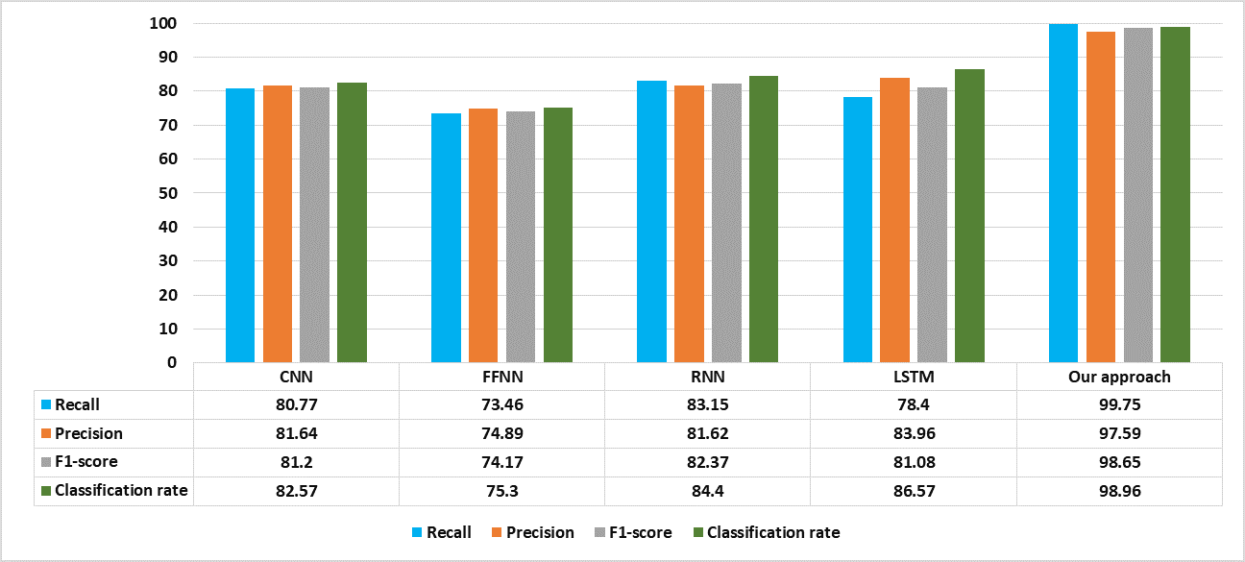

FIGURE 20. Comparative analysis of our method and deep learning algorithms.

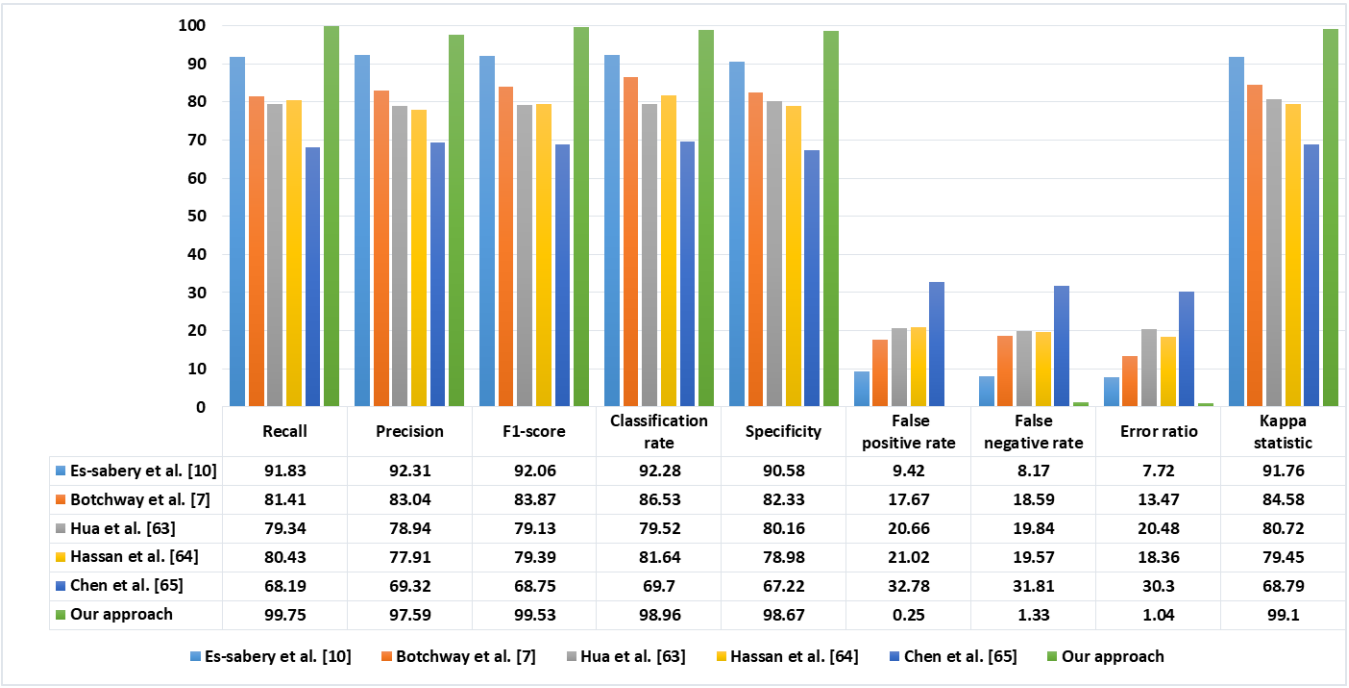

FIGURE 21. Comparison of our model with other approaches selected from the literature.

TABLE 10. Complexity in term of space of our developed hybrid model, Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65].

Corpus	Approaches	No. instructions (M)	No. parameters (M)
Sentiment140	Proposed approach	11.59	7.31
	Botchway et al. [7]	39.80	25.06
	Es-Sabery et al. [10]	12.60	9.57
	Hua et al. [63]	21.37	16.57
	Hassan et al. [64]	29.10	20.19
	Chen et al. [65]	30.45	25.82

TABLE 11. Complexity in terms of time of our developed hybrid model, Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65].

Corpus	Approaches	Training time (s)	Testing time (s)
Sentiment140	Our approach	11.92	3.56
	Botchway et al. [7]	16.21	4.023
	Es-Sabery et al. [10]	33.90	11.30
	Hua et al. [63]	542.98	61.04
	Hassan et al. [64]	330.72	70.65
	Chen et al. [65]	425.64	84.65

In the fifth experiment, we have assessed the efficiency of our proposed hybrid method and the Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65] methods chosen from the literature to train the Sentiment140 dataset by demonstrating the convergence of each evaluated approach using the equation 52 in order to determine the iteration number when the analyzed method verified the condition described below in the following equation 52.

$$\mathrm{Error}_{\mathrm{ratep}} - \mathrm{Error}_{\mathrm{ratec}} \geq \mathrm{T}_{\mathrm{value}} \tag{52}$$

Where $\mathrm{Error}_{\mathrm{ratep}}$ represents the average error rate achieved by the evaluated approach at the previous iteration of the learning process, $\mathrm{Error}_{\mathrm{ratec}}$ measures the average error rate of the assessed method at the current iteration of the learning process, and $\mathrm{T}_{\mathrm{value}}$ represents the threshold rate that initiated the convergence rate. After we performed numerous analyzed experiences, The threshold value was set at 0.0001. Hence, the average error rate of every analyzed method is estimated by use the subsequent equation:

$$\mathrm{E} = \frac{1}{2} \times \frac{\sum_{\mathrm{j}=1}^{\mathrm{I}} \sum_{\mathrm{i}=1}^{\mathrm{D}} (\mathrm{z} - \mathrm{z}_{\mathrm{label}})^2}{\mathrm{I}} \tag{53}$$

Where I signifies the total count of stored instances in the trained corpus, D represents the total count of decision feature labels in the used corpus, z is the expected and required decision feature label at the output of the classification process, and z_{label} represents the obtained label of the decision attribute at the output of the classification process. Suppose the formula defined in the equation (52) is met. In that case, we say that the trained method is converged, and the algorithm is executed till the trained method's average error rate reaches the condition. Otherwise, we would say that the trained approach fails to converge.

Figure 22 presents our proposed hybrid model's convergence rate when it was executed over the Sentiment140 dataset.

As displayed in figure 22, we noticed that our developed hybrid model converged towards the threshold rate value of 0.0001 after our suggested hybrid model algorithm arrived at 254 iterations when it was practiced over the Sentiment140 corpus.

Table 12 illustrates the convergence round of our suggested hybrid pattern, Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65].

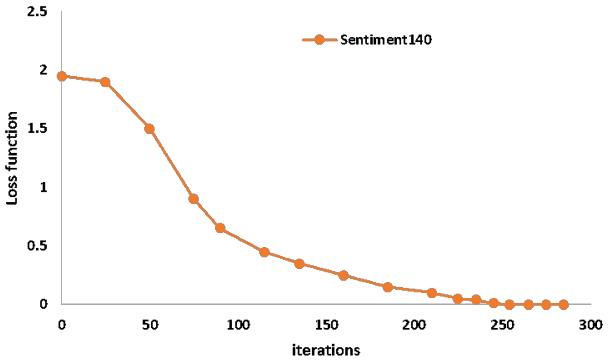

FIGURE 22. The convergence rate of our proposed hybrid model when it was performed over the Sentiment140 dataset.

As described in table 12, we remark that our proposed hybrid pattern converges faster than others because it has a lower misclassification rate in comparison to other evaluated approaches.

TABLE 12. Convergence round of our innovative hybrid approach, Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65].

corpus	Methods	Iterations
Sentiment140	Proposed approach	254
	Botchway et al. [7]	261
	Es-Sabery et al. [10]	270
	Hua et al. [63]	1775
	Hassan et al. [64]	2754
	Chen et al. [65]	3004

In the final experiment, we measured the mean standard deviation (MSD) of each approach in comparison to the various 5 cross-validations of the given corpus to examine the effectiveness of our developed hybrid pattern, Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65]. The primary objective of this experiment is to find the more stable approaches amongst all the evaluated approaches.

Table 13 depicts the obtained MSD and average accuracy (AVA) of the suggested hybrid model in comparison to Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65] over the various 5 cross-validations of used corpus in this proposal.

As reported in table 13, we deduce that our suggested hybrid model is more stable compared to Botchway et al.

TABLE 13. Stability of our innovative hybrid approach, Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65] in comparison to the various 5 cross-validations.

Corpus	Approaches	AVA (%)	MSD (%)
Sentiment140	Proposed approach	92.27	0.18
	Botchway et al. [7]	89.26	0.23
	Es-Sabery et al. [10]	86.53	0.26
	Hua et al. [63]	45.97	2.03
	Hassan et al. [64]	35.64	3.41
	Chen et al. [65]	50.72	1.98

[7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65] in comparison to the various 5 cross-validations, because it reached a higher AVA rate equal to 92.27 % with a lower MSD equal to 0.18 % when it was practiced over the Sentiment140 datasets.

VII. CONCLUSIONS

In this proposal, we develop a novel hybrid paradigm to categorize tweets into three categories: positive, neutral, and negative. Six steps make up the suggested hybrid approach: Phase of data collecting, in which we have chosen the Sentiment140 dataset to evaluate our contribution. Phase of data pre-treatment by performing all required pre-treatment operations. Semi-automatic tagging over the corpus using two methods vocabulary of Vader and the fuzzy system of Mamdani. Data representation step by utilizing four different extraction approaches, including: GloVe,TFIDF (Trigram), TFIDF (Bigram), and TFIDF (Unigram) in order to convert any twitter post into a digital vector. Data selection stage by utilizing 3 feature selection strategies, including : the ANOVA technique,the chi-square method and The mutual information approach. Data classification utilizing a deep belief network as a classification model to assign one of three labels—negative, neutral, or positive—to each tweet that is input. To solve the problem of a long runtime for large data sets, our hybrid approach is finally executed through utilizing Hadoop platform in a parallel configuration.

We conducted numerous simulations to assess the productivity and effectiveness of our proposed hybrid approach compared to the Botchway et al. [7], Es-Sabery et al. [10], Hua et al. [63], Hassan et al. [64] and Chen et al. [65]. The empirical results demonstrated that our proposed model surpasses the performance of all other evaluated approaches in terms of in terms of false negative rate, recall, runtime, convergence, stability, F1 score, accuracy, error rate, kappa-Static, complexity, false positive rate, precision rate and specificity rate.

Our planned scientific endeavors are the utilisation of the deep learning models as opposed to the traditional methods of feature extraction and selection for identifying the pertinent features, and searching for more classifiers to compare their effectiveness and our deep belief network that aims to classify the tweets in this contribution. Utilization of the fuzz rule-based model for handling uncertainty and vagueness data instead of the Mamdani fuzzy system used in this work.

REFERENCES

[1] A. Ligthart, C. Catal, and B. Tekinerdogan, "Systematic reviews in sentiment analysis: a tertiary study," *Artif. Intell. Rev.*, vol. 54, no. 7, pp. 4997–5053, Oct. 2021, doi:10.1007/s10462-021-09973-3.

[2] K. Dashtipour, M. Gogate, E. Cambria, and A. Hussain, "A novel context-aware multimodal framework for persian sentiment analysis," *Neurocomputing*, vol. 457, pp. 377–388, Oct. 2021, doi: 10.1016/j.neucom.2021.02.020.

[3] F. Es-sabery, K. Es-sabery, B. El Akraoui, and A. Hair, "Optimization Focused on Parallel Fuzzy Deep Belief Neural Network for Opinion Mining," *in Business Intelligence. CBI 2022. Lecture Notes in Business Information Processing*, R. Fakir, M., Baslam, M., El Ayachi, Ed. Springer, Cham, 2022, pp. 3–28. doi: 10.1007/978-3-031-06458-6_1.

[4] U. Naseem, I. Razzak, K. Musial, and M. Imran, "Transformer based Deep Intelligent Contextual Embedding for Twitter sentiment analysis," *Futur. Gener. Comput. Syst.*, vol. 113, pp. 58–69, Dec. 2020, doi: 10.1016/j.future.2020.06.050.

[5] J. Carvalho and A. Plastino, "On the evaluation and combination of state-of-the-art features in Twitter sentiment analysis," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 1887–1936, Mar. 2021, doi: 10.1007/s10462-020-09895-6.

[6] F. Es-sabery, K. Es-sabery, and A. Hair, "A MapReduce Improved ID3 Decision Tree for Classifying Twitter Data," *in Lecture Notes in Business Information Processing*, vol. 416 LNBIP, 2021, pp. 160–182. doi: 10.1007/978-3-030-76508-8_13.

[7] R. K. Botchway, A. B. Jibril, Z. K. Oplatková, and M. Chovancová, "Deductions from a SubSaharan African Bank's Tweets: A sentiment analysis approach," *Cogent Econ. Financ.*, vol. 8, no. 1, p. 1776006, Jan. 2020, doi: 10.1080/23322039.2020.1776006.

[8] F. Es-sabery, K. Es-sabery, H. Garmani, and A. Hair, "Sentiment Analysis of Covid19 Tweets Using A MapReduce Fuzzified Hybrid Classifier Based On C4.5 Decision Tree and Convolutional Neural Network," *E3S Web Conf.*, vol. 297, p. 01052, Sep. 2021, doi: 10.1051/e3sconf/202129701052.

[9] S. Yi and X. Liu, "Machine learning based customer sentiment analysis for recommending shoppers, shops based on customers' review," *Complex Intell. Syst.*, vol. 6, no. 3, pp. 621–634, Oct. 2020, doi: 10.1007/s40747-020-00155-2.

[10] F. Es-Sabery et al., "A MapReduce Opinion Mining for COVID-19-Related Tweets Classification Using Enhanced ID3 Decision Tree Classifier," *IEEE Access*, vol. 9, pp. 58706–58739, 2021, doi: 10.1109/ACCESS.2021.3073215.

[11] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Inf.Fusion*, vol. 42, pp. 146–157, Jul. 2018, doi:10.1016/j.inffus.2017.10.006.

[12] S. Nosratabadi, A. Mosavi, R. Keivani, S. Ardabili, and F. Aram, "State of the Art Survey of Deep Learning and Machine Learning Models for Smart Cities and Urban Sustainability," *in Lecture Notes in Networks and Systems*, vol. 101, 2020, pp. 228–238. doi: 10.1007/978-3-030-36841-8_22.

[13] D. N. de Oliveira and L. H. de C. Merschmann, "Joint evaluation of preprocessing tasks with classifiers for sentiment analysis in Brazilian Portuguese language," *Multimed. Tools Appl.*, vol. 80, no. 10, pp. 15391–15412, Apr. 2021, doi: 10.1007/s11042-020-10323-8.

[14] N. Chintalapudi, G. Battineni, M. Di Canio, G. G. Sagaro, and F. Amenta, "Text mining with sentiment analysis on seafarers' medical documents," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 1, p. 100005, Apr. 2021, doi: 10.1016/j.jjimei.2020.100005.

[15] H. Aljuaid, R. Iftikhar, S. Ahmad, M. Asif, and M. Tanvir Afzal, "Important citation identification using sentiment analysis of in-text citations," *Telemat. Informatics*, vol. 56, p. 101492, Jan. 2021, doi: 10.1016/j.tele.2020.101492.

[16] 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.

[17] K. V. Pramod and S. Soumya, "Hybrid Deep Learning Approach for Sentiment Classification of Malayalam Tweets," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 4, pp. 891–899, 2022.

[18] R. K. Behera, M. Jena, S. K. Rath, and S. Misra, "Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data," *Inf. Process. Manag.*, vol. 58, no. 1, p. 102435, Jan. 2021, doi: 10.1016/j.ipm.2020.102435.

[19] S. Bodapati, H. Bandarupally, R. N. Shaw, and A. Ghosh, "Comparison and Analysis of RNN-LSTMs and CNNs for Social Reviews Classifica-

tion," *in Advances in Intelligent Systems and Computing*, vol. 1319, 2021, pp. 49–59. doi: 10.1007/978-981-33-6919-1_4.

[20] M. Goswami and P. Sajwan, "A Comparative Analysis of Sentiment Analysis Using RNNLSTM and Logistic Regression," *in Lecture Notes in Electrical Engineering*, vol. 740 LNEE, 2021, pp. 165–174. doi: 10.1007/978-981-33-6393-9_18.

[21] A. Onan, "Bidirectional convolutional recurrent neural network architecture with groupwise enhancement mechanism for text sentiment classification," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 5, pp. 2098–2117, May 2022, doi: 10.1016/j.jksuci.2022.02.025.

[22] L. D. C. S. Subhashini, Y. Li, J. Zhang, and A. S. Atukorale, "Integration of fuzzy logic and a convolutional neural network in three-way decision-making," *Expert Syst. Appl.*, vol. 202, p. 117103, Sep. 2022, doi: 10.1016/j.eswa.2022.117103.

[23] N. M. Alharbi, N. S. Alghamdi, E. H. Alkhammash, and J. F. Al Amri, "Evaluation of Sentiment Analysis via Word Embedding and RNN Variants for Amazon Online Reviews," *Math. Probl. Eng.*, vol. 2021, pp. 1–10, May 2021, doi: 10.1155/2021/5536560.

[24] A. U. Rehman, A. K. Malik, B. Raza, and W. Ali, "A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis," *Multimed. Tools Appl.*, vol. 78, no. 18, pp. 26597–26613, Sep. 2019, doi: 10.1007/s11042-019-07788-7.

[25] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "A Combined CNN and LSTM Model for Arabic Sentiment Analysis," *in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11015 LNCS, 2018, pp. 179–191. doi: 10.1007/978-3-319-99740-7_12.

[26] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert Syst. Appl.*, vol. 72, pp. 221–230, Apr. 2017, doi: 10.1016/j.eswa.2016.10.065.

[27] D. Gibney and S. V. Thankachan, "Text Indexing for Regular Expression Matching," *Algorithms*, vol. 14, no. 5, p. 133, Apr. 2021, doi: 10.3390/a14050133.

[28] M. Kapoor, G. Fuchs, and J. Quance, "RExACtor: Automatic Regular Expression SignatureGeneration for Stateless Packet Inspection," *in 2021 IEEE 20th International Symposium on Network Computing and Applications* (NCA), Nov. 2021, pp. 1–9. doi: 10.1109/NCA53618.2021.9685959.

[29] H. Chouikhi, H. Chniter, and F. Jarray, "Arabic Sentiment Analysis Using BERT Model," *in Communications in Computer and Information Science*, vol. 1463, 2021, pp. 621–632. doi: 10.1007/978-3-030-88113-9_50.

[30] W. Bourequat and H. Mourad, "Sentiment Analysis Approach for Analyzing iPhone Release using Support Vector Machine," *Int. J. Adv. Data Inf. Syst.*, vol. 2, no. 1, pp. 36– 44, Apr. 2021, doi: 10.25008/ijadis.v2i1.1216.

[31] L. S. Meetei, T. D. Singh, S. K. Borgohain, and S. Bandyopadhyay, "Low resource language specific pre-processing and features for sentiment analysis task," *Lang. Resour. Eval.*, vol. 55, no. 4, pp. 947–969, Dec. 2021, doi: 10.1007/s10579-021-09541-9.

[32] M. Pota, M. Ventura, H. Fujita, and M. Esposito, "Multilingual evaluation of preprocessing for BERT-based sentiment analysis of tweets," *Expert Syst. Appl.*, vol. 181, p.115119, Nov. 2021, doi: 10.1016/j.eswa.2021.115119.

[33] R. Štrimaitis, P. Stefanovič, S. Ramanauskaitė, and A. Slotkienė, "Financial Context News Sentiment Analysis for the Lithuanian Language," *Appl. Sci.*, vol. 11, no. 10, p. 4443, May 2021, doi: 10.3390/app11104443.

[34] H.-J. Kwon, H.-J. Ban, J.-K. Jun, and H.-S. Kim, "Topic Modeling and Sentiment Analysis of Online Review for Airlines," *Information*, vol. 12, no. 2, p. 78, Feb. 2021, doi: 10.3390/info12020078.

[35] Y. Sun, E. Lank, and M. Terry, "Label-and-Learn," *in Proceedings of the 22nd International Conference on Intelligent User Interfaces*, Mar. 2017, pp. 523–534. doi: 10.1145/3025171.3025208.

[36] V. Bonta, N. Kumaresh, and N. Janardhan, "A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis," *Asian J. Comput. Sci. Technol.*, vol. 8, no. S2, pp. 1– 6, Mar. 2019, doi: 10.51983/ajcst-2019.8.S2.2037.

[37] A. Amin, I. Hossain, A. Akther, and K. M. Alam, "Bengali VADER: A Sentiment Analysis Approach Using Modified VADER," *in 2019 International Conference on Electrical, Computer and Communication Engineering* (ECCE), Feb. 2019, pp. 1–6. doi:10.1109/ECACE.2019.8679144.

[38] A. Dey, M. Jenamani, and J. J. Thakkar, "Senti-N-Gram: An n -gram lexicon for sentiment analysis," *Expert Syst. Appl.*, vol. 103, pp. 92–105, Aug. 2018, doi:10.1016/j.eswa.2018.03.004.

[39] Tung, C. Quek, A mamdani-takagi-sugeno based linguistic neural-fuzzy inference system for improved interpretability-accuracy representation, in: *2009 IEEE Int. Conf. Fuzzy Syst.*, IEEE, 2009: pp. 367-372. doi: 10.1109/FUZZY.2009.5277194.

[40] E. Pourjavad, R. V. Mayorga, A comparative study and measuring performance of manufacturing systems with Mamdani fuzzy inference system, *J. Intell. Manuf.*, 30 (2019) 1085-1097. doi: 10.1007/s10845-017-1307-5.

[41] S. Mohurle, M. Devare, A study of triangular membership function and multiple regressions to calculate the MSW compost index, *J. Eng. Res.*, (2021). doi: 10.36909/jer.14183.

[42] H. Zheng, W.-B. Xie, H.-K. Lam, L. Wang, Membership-function-dependent stability analysis and local controller design for T-S fuzzy systems: A space-enveloping approach, *Inf. Sci. (Ny).*, 548 (2021) 233-253. doi: 10.1016/j.ins.2020.09.043.

[43] M.O. Kalinin, Application of Neuro-Fuzzy Inference to Detect Network Scanning, *Autom. Control Comput. Sci.*, 55 (2021) 908-917. doi: 10.3103/S0146411621080150.

[44] A. Golzio, M. Puerta-Díaz, D. Martínez-Ávila, A Fuzzy Logic Model for the Analysis of Social Corporate Responsibility, *ICST Trans. Scalable Inf. Syst.*, 8 (2018) 170556. doi: 10.4108/eai.23-7-2021.170556.

[45] P. Aimtongkham, P. Horkaew, C. So-In, Multistage fuzzy logic congestion-aware routing using dual-stage notification and the relative barring distance in wireless sensor networks,*Wirel. Networks.*, 27 (2021) 1287-1308. doi: 10.1007/s11276-020-02513-x.

[46] F. Fanian, M. Kuchaki Rafsanjani, Memetic fuzzy clustering protocol for wireless sensor networks: Shuffed frog leaping algorithm, *Appl. Soft Comput.*, 71 (2018) 568-590. doi: 10.1016/j.asoc.2018.07.012.

[47] W.B. Zulfikar, Jumadi, P.K. Prasetyo, M.A. Ramdhani, Implementation of Mamdani FuzzyMethod in Employee Promotion System, *IOP Conf. Ser. Mater. Sci. Eng.*, 288 (2018) 012147. doi: 10.1088/1757-899X/288/1/012147.

[48] C.-H. Cheng, C.-H. Chen, Fuzzy time series model based on weighted association rule for financial market forecasting, ExpertSyst., 35 (2018) e12271. doi: 10.1111/exsy.12271.

[49] K. Orkphol, W. Yang, Sentiment Analysis on Microblogging with K-Means Clustering and Artificial Bee Colony, *Int. J. Comput. Intell. Appl.*, 18 (2019) 1950017. doi: 10.1142/S1469026819500172.

[50] M.J. Vilela, G.F. Oluyemi, Fuzzy Logic, in: 2022: pp. 163-199. doi: 10.1007/978-3-030-86989-2_6.

[51] M. Praharaj, B.M. Mohan, Modeling and Analysis of Mamdani Two-Term Controllers Using Non-Uniformly Distributed Multiple Fuzzy Sets and CoA/CoG Defuzzication, *IETE Tech. Rev.*, (2021) 1-22. doi: 10.1080/02564602.2021.1933628.

[52] M. Dhokare, S. Teje, S. Jambukar, V. Wangikar, Evaluation of Academic Performance of Students Using Fuzzy Logic, in: *2021 Int. Conf. Adv. Electr. Electron. Commun. Comput. Autom.*, IEEE, 2021: pp. 1-5. doi: 10.1109/ICAECA52838.2021.9675557.

[53] K.S. Gilda, S.L. Satarkar, Maxima-Based Defuzzication Methods Proposed for Fuzzy Systems, in: *R.G. Singh Mer, K.K., Semwal, V.B., Bijalwan, V., Crespo (Ed.), Proc. Integr. Intell. Enable Networks Comput.Algorithms Intell. Syst.*, Singapore, 2021: pp. 629-639. doi: 10.1007/978-981-33-6307-6_64.

[54] P. Atandoh, Z. Fengli, D. Adu-Gyamfi, H.L. Leka, P.H. Atandoh, A Glove CNN-Bilstm Sentiment Classiffication, in: *2021 18th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process.*, IEEE, 2021: pp. 245-249. doi: 10.1109/ICCWAMTIP53232.2021.9674171.

[55] U. Moorthy, U.D. Gandhi, A novel optimal feature selection technique for medical data classiffication using ANOVA based whale optimization, *J. Ambient Intell. Humaniz. Comput.*, 12 (2021) 3527-3538. doi: 10.1007/s12652-020-02592-w.

[56] I. Sohn, "Deep belief network based intrusion detection techniques: A survey," *Expert Syst. Appl.*, vol. 167, p. 114170, Apr. 2021, doi: 10.1016/j.eswa.2020.114170.

[57] Z. Fang, K. Roy, J. Mares, C.-W. Sham, B. Chen, and J. B. P. Lim, "Deep learning-based axial capacity prediction for cold-formed steel channel sections using Deep Belief Network," Structures, vol. 33, pp. 2792–2802, Oct. 2021, doi: 10.1016/j.istruc.2021.05.096.

[58] N. Zeng, H. Li, and Y. Peng, "A new deep belief network-based multi-task learning for diagnosis of Alzheimer's disease," *Neural Comput. Appl.*, Jun. 2021, doi: 10.1007/s00521-021-06149-6.

[59] 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.

[60] B. Hamzi and H. Owhadi, "Learning dynamical systems from data: A simple cross-validation perspective, part I: Parametric kernel flows," *Phys.*

D Nonlinear Phenom., vol. 421, p. 132817, Jul. 2021, doi: 10.1016/j.physd.2020.132817.
[61] 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.
[62] F. Es-Sabery and A. Hair, "An improved ID3 classification algorithm basedon 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.
[63] Yuming Hua, Junhai Guo, and Hua Zhao, "Deep Belief Networks and deep learning," in *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things*, Jan. 2015, pp. 1–4. doi: 10.1109/ICAIOT.2015.7111524.
[64] M. M. Hassan, M. G. R. Alam, M. Z. Uddin, S. Huda, A. Almogren, and G. Fortino, "Human emotion recognition using deep belief network architecture," *Inf. Fusion*, vol. 51, pp. 10– 18, Nov. 2019, doi: 10.1016/j.inffus.2018.10.009.
[65] X. Chen, T.-H. Li, Y. Zhao, C.-C. Wang, and C.-C. Zhu, "Deep-belief network for predicting potential miRNA-disease associations," *Brief. Bioinform.*, vol. 22, no. 3, May 2021, doi: 10.1093/bib/bbaa186.

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, Morocco, in 2014, and the master's degree in business intelligence from the Department of Computer Sciences, Sultan Moulay Sliman's University, Beni Mellal, Morocco, in 2016. She has published several research papers in many international conferences and journals, i.e., *International Journal of Fuzzy Information and Engineering*, *International Journal of Informatics and Communication Technology*, The 3rd International Conference on Networking, Information Systems Security, and 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.

IBRAHIM ES-SABERY received the master's degree in business intelligence from the Department of Computer Sciences, Sultan Moulay Sliman's University, Beni Mellal, Morocco, in 2019. His general interests span include Data Mining area, Big Data field, Wireless Sensor Networks, Fuzzy systems, Machine learning, Deep learning, and Internet of Things.

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).

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 are the development and evaluation of e-Health systems, m-Health, medicine 2.0., cloud computing, among others. 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.

BEGONA GARCIA ZAPIRAIN (Member, IEEE) graduated in telecommunications engineering and specialized in telematics at the University of the Basque Country (UPV/EHU), Leioa, Spain. She received the Ph.D. degree (summa cum laude) in the pathological speech digital processing field in 2003, and the Executive MBA degree from the University of the Basque Country, in 2011, with the Best Student Award. In 2012, she graduated from the Advanced Program in Health Management at the Deusto Business School, Deusto University, Bilboa, Spain. After spending five years at ZIV Company, she joined the Engineering Faculty at the University of Deusto, in 1997, as a Lecturer in Signal Theory and Electronics, where she led the Telecommunications Department 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.