Adam Deep Learning with SOM for Human Sentiment Classification

Md. Nawab Yousuf Ali, *East West University, Dhaka, Bangladesh*
Md. Golam Sarowar, *East West University, Dhaka, Bangladesh*
Md. Lizur Rahman, *East West University, Dhaka, Bangladesh*
Jyotismita Chaki, *Vellore Institute of Technology, Vellore, India*
Nilanjan Dey, *Techno India College of Technology, Kolkata, India*
João Manuel R.S. Tavares, *Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial, Departamento de Engenharia Mecânica, Faculdade de Engenharia, Universidade do Porto, Porto, PORTUGAL*

ABSTRACT

Nowadays, with the improvement in communication through social network services, a massive amount of data is being generated from user's perceptions, emotions, posts, comments, reactions, etc., and extracting significant information from those massive data, like sentiment, has become one of the complex and convoluted tasks. On other hand, traditional Natural Language Processing (NLP) approaches are less feasible to be applied and therefore, this research work proposes an approach by integrating unsupervised machine learning (Self-Organizing Map), dimensionality reduction (Principal Component Analysis) and computational classification (Adam Deep Learning) to overcome the problem. Moreover, for further clarification, a comparative study between various well known approaches and the proposed approach was conducted. The proposed approach was also used in different sizes of social network data sets to verify its superior efficient and feasibility, mainly in the case of Big Data. Overall, the experiments and their analysis suggest that the proposed approach is very promissing.

KEYWORDS

Social Networks, Self-Organizing Map, Adam Deep Learning, Principle Component Analysis, TF-IDF information Retrieval, Random Forest.

INTRODUCTION

In this time of technological development, the social media have become the main way of human communication. The basic comfort of the people irrespective of the distance or magnitude of the audience is only allowed in this platform of social media. Recent improvement in technology depicts that most of the data in social media appears with noise. One of the studies regarding social network data [1] has revealed that approximately 80% of the data currently available in social media is fully unstructured. As a result of this, it is more sophisticated for social platforms to analyze these data and obtain viable information from the data without delay. In order to carry out this work, two approaches called sentiment analysis and opinion mining have become the world's most important techniques for gaining the most feasible data insights. Social media sentiment analysis can deal with many problems to be solved and helps to provide many indications including public opinion, advertisement, healthcare, and public satisfaction. Discovering hidden pattern from those complex and complicated datasets can improve overall prediction and classification in social media. For this goals to be atained, this research work is

focused in analyzing the feelings of many posts and comments on social media data so that immediate action or legal reaction can be conducted. Although, the use of mining techniques to analyzing sentiment from unstructured social network data has become truly challenging. Currently, there are many methods available to enhance the ability of sentiment analysis including feature selection, data integration, data cleaning, and crowdsourcing. Realizing the worth of sentiment analysis in social media platform, this task has become a very potential research area for social media platforms. One of the main reason behind this situation can be demonstrated as free and at any time users authorization and evaluation of the account they have, and all the activities carried out by each and every user of social media are stored for further analysis and manipulation, in order to increase user experience and user satisfaction with these platforms. Several recent works on social media platforms have shown that screening of terrorist activities and the perception of users are now a mandatory task for these platforms. Sentiment analysis is a kind of Natural Language Processing (NLP)where all the texts are processed, understood and sentiments are anticipated to predict suggestions to the users for their better experience with that specific platform. Thousands of works have already conducted by researchers in the field of product reviews, political party, tweet updates, brands of products, social media analysis, etc. [2]. All these systems express opinions in two ways: one is positive and other one is negative. But, here, we focused on multi-class classification for this contribution and also included neutral sentiment. Therefore, from the perspective of the proposed classification, there are two classes: negative and positive. We have mainly concentrated on social network posts, comments, and public opinion, and collected data from Facebook, tweeter, and Instagram. However, the main focus was on Facebook data because this platform is one of the most growing platforms nowadays. Moreover, for status update, Facebook allows a total of 5000 characters for users to save huge data in their own database, and the total number of Facebook users is higher than the one any other platform. It would therefore be easy to collect samples from them, and most of them will be based on real life and challenges. According to a survey article [3], a total of 510000 comments are posted on Facebook every 60 seconds and 293000 posts are updated. Realizing the ways how users express their emotions and opinions, the overall analysis can be improved. Therefore, it is mandatory for all the social network platforms to collect the required data and analyze them further to take steps to add additional features. In addition, the user growth rate of twitter from 2010 to 2018 [4] was shown in a study, figure 1.

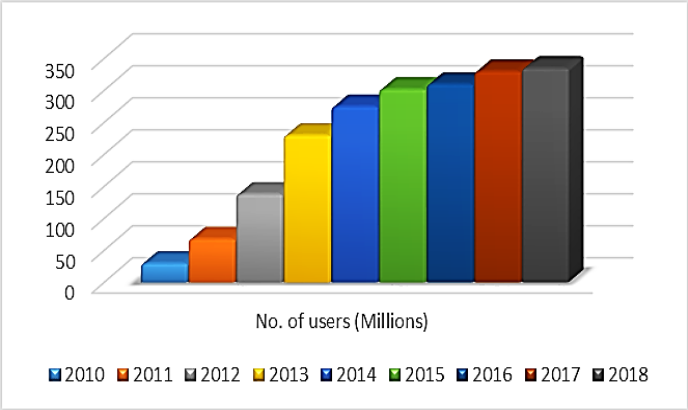


Figure.1. Graphical representation of Twitter's user growth rate

This article is divided into six sections to better clarify the overall idea. Nex section, broadly introduces previous contributions and ideas which have been explored by many academics as well as independent researchers. The used data collection mechanism, resources, required API and sources to collect data, are described in section 3. Section 4 refers the proposed approaches that were used in this work to deal with the problem under study. The experimentation and results are presented and discussed in section 5. Lastly,

section 6 presents the overall conclusions and the future direction of the research in the area of sentiment analysis.

LITERATURE REVIEW

Thousands of researchs have already been conducted and more are currently being carried out in the field of social networks sentiment analysis. Machine learning based research has been conducted by many researchers in this field. Some used Naïve Bayes, random forest, and support vector machine to analysis sentiment of Facebook comments [5]. On the other hand, researcher [6] provides an indication of using Naïve Bayes along with decision tree for classification of sentiment from Facebook data. In this work, the authors proposed a machine learning framework based on Apache for dimensionality reduction since it focused on Big Data manipulation. Another study on sentiment analysis [7] incorporates the long short term memory (LSTM) and dynamic convolutional neural network (DCNN). Naïve Bayes along with support vector machine are used in this work. According to M. S. Neethu et al. [8], it can be a prospect way of developing well formulated feature vector using machine learning techniques. In [9], the authors used a convolutional neural network for organizing, preprocessing and classification of sentences. They have achieved high accuracy, but in the case of big data and many layers, the convolutional neural network become very computational expensive. Many works on specific languages have also been conducted by various researchers; for example, in [10], Arabic Facebook texts are collected and analyzed to extract the sentiment associated with those posts, comments, etc. The authors concentrated on several preprocessing strategies like combinations of extraction (n-grams) and have used weighting schemes (TF / TF-IDF) for features construction. According to [11], the proposed dictionary based approach works better than other available approaches, but in case of new generated data which has not been included in the dictionary yet can predicted wrong which will ensure poor performance. With the increasing number of user contributed texts in social networking websites like tweets, product reviews, and blogs, sentiment analysis and classification agenda have appeared to accommodate governments as well as companies to improve their products and services. In contribution [12], the authors have particularly focused on addressing the issues, limitations, contraints on the way of classifying social network sentiments. Moreover, motivational perspective behind sentiment classification in different fields have also been concentrated on. However, for dealing with the exponentially ever gworing datasets new cost efficient mechanism is mandatory to be established. Realizing this scheme, the authors of work [13] have proposed a new approach by incorporating the information of POS based sentiment-rich phrages along with a machine learning approach. Moreover, a feature selection method for extracting relevant bi-tagged phrases and unigram features have been employed for increasing accuracy and efficiency of overall classification task. The obtained overall experimented results ensures better performance of the proposed approach. Some contributions specially focus on combining traditional lexical based approach with machine learning approaches according to contribution [14], it has been carried out by the authors that it is possible to get higher accuracy for facebook sentiment analysis, students sentiments towards a course, for supporting personized learning; the authors have also demonstrated their propsed approach in vekried typesd of fields. One contribution [15] was specifically concentrated on diagnosing human depression by analyzing status of the social networking sites in order to explore the overall idea machine learning approach, and traditional natural language processing approaches have been implemented and applied. Moreover, the authors have also employed emotion theories [16] for accomplishment of their tasks. Therefore, this can be an evolutionalry framework for the automated diagnosis of depression. In another contribution [17], the authors have deeply analyzed the performance of different deep learning architecture for classification of social network sentiment. They have specifically focused on Recursive neural network and convolutional neural network in case of tenegeers sentiment classification of social networking sites. In this emerging sector of sentiment analysis polarity classification of text messages

have been demonstrated in details and effect of supervised, unsupervised and semisupervised approaches have also been analyzed in contribution [18]. Nowadays, many contributions are being conducted by thousands of researchers in the fields of sentiment classification, opinion mining, polarity measures, etc. Consequently, different hybrid machine learning approaches are being proposed, and comparatively to the other available approaches, the hybrid approaches perform much better and faster. To spreading the knowledge of the hybrid mechanisms, the work [19] creates a new vision positively towards the recent advancement and future direction on those fields. Therefore, sentiment classification is extremely effective and obligate in social network monitoring. It usually authorizes one to gain an overall perception towards comprehensive public behavior behind different products. By adopting real-time monitoring capabilities, Bandwatch Analytics [20] have made the sentiment analysis process efficient, easier and quicker. However, the main concern in this field of research is that the data generated from the social networking platforms are increasing in a exponential way. Therefore, to overcome the big data sentiment classification problems more competent, time efficient, and less memory consuming approaches are mandatory in order to successfully process the increasing number of huge datasets. Otherwise, this sentiment classification or opiniom mining agenda will go beyond the limit of human knowledge. This will create a huge problem which are connected to this field like various online based organization, supplier company, government bodies, and people associated with those fields. Consequently, economic growth of a country will be ruined and we will lag behind. Meanwhile, it seems that traditional NLP tools are not much effective and efficient in mamipulating huge datasets that are generated by millions of social network users, millions of product reviews, millions of products tagging, etc.

Therefore, for dealing with exponential growth of social network data, here we have specifically focused on developing a new hybrid approach by incorporating clustering based unsupervised approach with supervised machine learning approach for efficient analysis, as well as immediate classification of sentiments associated with social network comments, posts, etc.

ADAPTED METHODS

In order to accomplish this work, three methods were used: Self Organizing Map (SOM) for clustering, Principal Component Analysis (PCA) for dimentionality reduction and Adam Deep Learning for classification.

Self-Organizing Map

SOM has been used in many works because of its ability to create clusters by incorporating similar data into specified cluster. Initially, after employing the StopWords filtering mechanism on the tokenized word's document, the necessary data are stored in a array [21]. Here, the array of significant words for sentiment analysis is used as the bag of words. Then, after execution of the TF-IDF information retrieval as well as normalization phase, this approach is applied to divide the same types of data. In this case, we have considered three clusters. For attaining highest accuracy as well as efficiency in our approach, SOM is applied to the bag of words so that the performance of Adam convolutional neural network can be improved. The mathematical derivation of SOM is:

$$W = \sum_{i=0}^{n} w_i \text{.....(1)}$$

where w_i is the individual weight in the weight vector and n is the total number of weights. the node's weight vector, and is the number of weights. Initially, equation 1 is used for random initialization of each element of the weight vector.

Them the Euclidean distance (D) is used to determine the similarity between the input vector V and the node's weight vectors $(W_1, W_2, W_3 \text{.....} W_n)$:

$$D = \sqrt{\sum_{i=0}^{n} (V_i - W_i)^2} \quad \text{.....(2)}$$

where V_i is the current input vectors.

After that, the neighborhood radius of the best matching unit (obtained from equation 2) need to be calculated as:

$$\sigma(t) = \sigma_0 e^{(-\frac{t}{\gamma})} \quad \text{.....(3)}$$

where σ_0 is the width of the lattice at time t = 0, γ is the time constant, and t is the current time. The value of directly depends on the width of the lattice σ_0.

Since the new radius has been calculated, all nodes need to be re-scanned to ensure whether they are within the radius or not. This can be conducted as:

$$\sum_{i=0}^{n} w_i(t+1) = w_i(t) + \theta_i(t) L_i(t)(V_i(t) - w_i(t)) \quad \text{.....(4)}$$

where t is the time taken in every iteration, L_i is the learning rate, W_i is the previous weight, $(W_i + 1)$ represents the adjusted new weight, $V_i(t)$ the new input vector in every iteration, and θ_i denotes the node's actual distance from best matched unit. Now, being $\theta_i(t)$ unknown, one can not obtain the actual value and so to negotiate with $\theta_i(t)$ the following calculation should be performed:

$$\sigma_i(t) = e^{(-\frac{d2}{2\sigma 2(t)})} \quad \text{.....(5)}$$

Execution of this phase requires the values of the learning rate (L_i) which can be computed by:

$$L(t) = L_0 e^{(-\frac{t}{\gamma})} \quad \text{.....(6)}$$

For this work we have divided the input vector V into three clusters and stored them in $V_1, V_2 \& V_3$ vectors by executing the steps and equations mentioned above. Further, these $V_1, V_2 \& V_3$ clusters are processed using principle componenet analysis for dimensionality reduction purpose in case of massive datasets.

Principle Component Analysis

To reduce the feature's dimensionality of the clustered data $V_1, V_2 \& V_3$, which have been generated after applying SOM, PCA is used [22, 23, 24]. In this respect, for the inner interpretation of PCA, initially co-variance formula is used for each cluster:

$$COV^{j}_{(x,y)} = \sum_{i=1}^{n} \frac{(x_j^i - \bar{x}_j)(y_j^i - \bar{y}_j)}{n_j - 1} \quad \text{.....(7)}$$

where x and y are the input and output features, x_j is the j[th] element of input vector X, y_j is the j[th] element of output vector Y, $\bar{x}$ & $\bar{y}$ are the mean value of X and Y vector, respectively, and j denotes the number of the cluster to be evaluated. Therefore, for every cluster, co-variance is calculated and stored for further analysis. After getting all the covariance values, a vector S is created for every cluster to find the values of principal components:

$$| S_i - \lambda_i I | = 0 .....(8)$$

whwre S is the covariance vector, I is the identity matrix, λ has the values of the principal components and i is the cluster under evaluation. For simplicity, let's assume we have two dimensional input vector and so there will be two principle component equations. However, these Z_1 *and* Z_2 values are for just one cluster and thus we have three sets of Z_1 and Z_2:

$$Z_1 = a_{11}x_1 + a_{12}x_2 .....(9)$$

$$Z_2 = a_{21}x_1 + a_{22}x_2 .....(10)$$

with , , , are the final singular values of the principle components.

Since we have assumed two dimensional input vector, we will get two values which will be used to calculate the principle components: one is λ_1 and another one is λ_2. Now using λ_1, we can find the values of a_{11} and the values of a_{12}. Similarly, using λ_2, we can calculate values of a_{21} and a_{22}:

$$(a^{11})^2 + \left(a^{12}\right)^2 = 1 .....(11)$$

Finally, all the values of *a* are assigned into equations 9 and 10 to get the principle components. After employing those equation on each clusters, the dimensionality of those clusters is reduced to two dimension which helps the most while negotiating with the massive featured datasets.

Adam Deep Learning

The main aspect of the Deep Convolutional network is the presence of a convolution layer which usually embed a convolution operation into the input data, passing the data towards next layer [25-30]. We have also evaluated the performance of Adam Deep Learning (ADL) with the available neural network based approaches. Moreover, we have applied Adam optimizer technique to update the parameters. Since our datasets are massive and higher dimensional, we have embedded PCA just before the execution of Adam Deep Learning. Technically, PCA creates a new space by conducting linear transformation on the old data. Therefore, incorporating this approach with ADL increases overall accuracy as well as efficiency. As the number of weights and parameters increases in ADL, the number of data required to be able to determined becomes mandatory and increases quite rapidly and, for this phenomena, over-fitting becomes the main concern. The infrastructure for the convolutional neural network used in this work is illustrated in figure. 2.

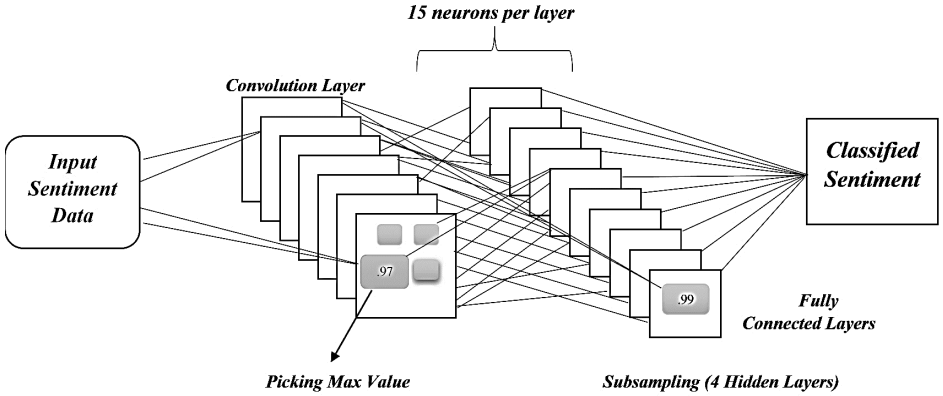


Figure.2: Overall architecture of the used convolutional neural network

The convolutional neural network's input matrix always covers the size of the input image or dataset. Now, the clusters, which were contructed and processed using SOM as well as PCA, are trained using

Adam Deep Learning architecture. The calculation inside ADL is straightforward. Three models are extracted from this phase because of the three clusters that were previously constructed. Initially, forwad calculation mechanism for neural network is employed on the input clusters for ADL (V_1, V_2 & V_3):

$$x_{ij}^{l} = \sum_{a=0}^{N-1}\sum_{b=0}^{N-1} \omega_{ab} V_{(i+a)(j+b)}^{k} \text{.....(12)}$$

where k is the cluster's number, w is the weight vector, V is the input vector. This equation is used for all the neurons of the hidden layer, whereas the convolutional neural layer is mathematically denoted as:

$$y_{ij}^{l} = \sigma(x_{ij}^{l}) \text{.....(13)}$$

The error (E) calculation mechanism is also needed for the upcoming layer with respect to partial derivative each neuron's output of current layer, $\frac{\delta E}{\delta y_{ij}^{l}}$, where we must sum all the output of each neuron as:

$$\frac{\partial E}{\partial \omega_{ab}} = \sum_{i=0}^{M-N}\sum_{j=0}^{M-N} \frac{\partial E}{\partial x_{ij}^{l}} \frac{\partial x_{ij}^{l}}{\partial \omega_{ab}} = \sum_{i=0}^{M-N}\sum_{j=0}^{M-N} \frac{\partial E}{\partial x_{ij}^{l}} V_{(i+a)(j+b)}^{l-1} \text{.....(14)}$$

Now, for further movement towards our desired results, we need to compute the gradient for which we have to know the value of $\frac{\partial E}{\partial x_{ij}^{l}}$, which can be simply calculated and optimized using Adam Optimizer.

Initially, the gradients stochastic objective at time step t need to be computed using:

$$m_t = \beta_1 . m_{t-1} + (1-\beta_1) . g_t \text{.....(15)}$$

Now, biased first moment estimate (β_1) needs to be updated:

$$g_t = \nabla_t f_t(\theta_{t-1}) \text{.....(16)}$$

Then, after updating the 1st moment, the 2nd raw moment should be estimated at time t (V_t):

$$v_t = \beta_2 . v_{t-1} + (1-\beta_2) . g_t^2 \text{.....(17)}$$

Afterwards, bias corrected 1st estimate and 2nd raw moment estimate can be calculated:

$$\overline{m_t} = m_t / (1-\beta_2^t) \text{.....(18)}$$

$$\overline{v_t} = v_t / (1-\beta_2^t) \text{.....(19)}$$

Now, parameters θ are updated using.

$$\theta_t = \theta_{t-1} - \alpha . \overline{m_t} (\sqrt{\overline{v_t}} + \varepsilon) \text{.....(20)}$$

In order to compute the weights of each convolutional layer, the following equation can be used:

$$\frac{\partial E}{\partial V_{ij}^{l-1}} = \sum_{a=0}^{N-1}\sum_{b=0}^{N-1} \frac{\partial E}{\partial x_{(i-1)(j-b)}^{l}} \frac{\partial x_{(i-1)(j-b)}^{l}}{\partial V_{ij}^{l-1}} = \sum_{a=0}^{N-1}\sum_{b=0}^{N-1} \frac{\partial E}{\partial x_{(i-1)(j-b)}^{l}} \omega_{ab} \text{.....(21)}$$

PROPOSED METHODOLOGY

A combination of unsupervised clustering, dimensionality reduction and classification based approach is considered so that sentiment classification can be attained with high accuracy with efficiency using a large number of features. The overall framework of the adopted approach is illustrated in figure 3.

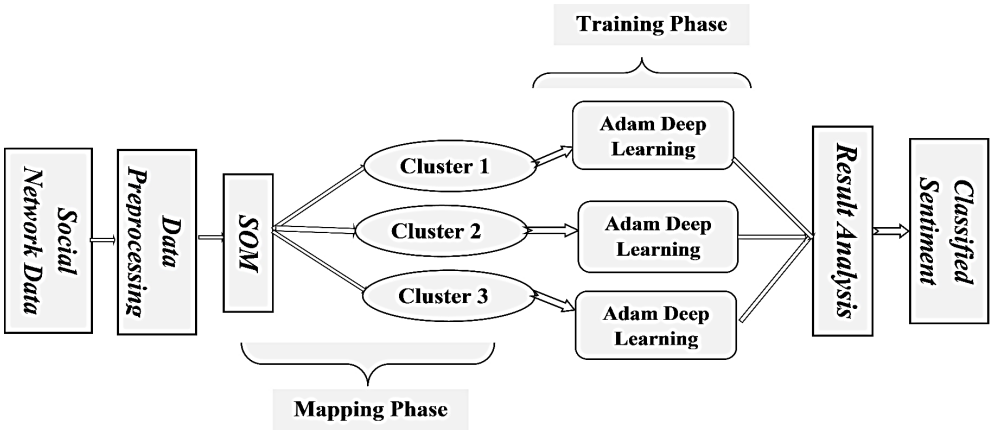


Figure 3: Overall framework of the proposed approach

Prior to feature extraction, the collected data is preprocessed. Majority of the data collect from social networking sites are unstructured and contains noisy data, irrelevant data, special characters, etc., which restricts machine learning approaches to work efficiently. Moreover, for training purpose, all the data need to be converted into corresponding numerical values. Therefore, in case of achieving better performance from the machine learning approaches, the representation of the data need to be in a proper manner. Some of the machine learning approaches only works with specified formats. For an example, Random Forest approach can not process data having Null value. Therefore, data has to be cleared before using Random Forest. Additionally, another aspect is that the dataset has to be formatted in such a way that various number of approaches can be employed without further edition. For accomplishment of this task initially, our datasets are tokenized and filtered using StopWords in Natural Language Toolkit (NLTK). Moreover, special characters are also omitted. After that, the data are converted into lower case to avoid case contradictions. Now, the term frequency-inverse document frequency (TF-IDF) information retrieval mechanism is applied to convert data into numerical corresponding values so that the data can be used to train machine learning based approaches. TF-IDF is basically one of the numerical statistic that is actually intended to find how important a word is to a document or corpus. Most of the time, it is indicated as a weighting factor in searches of information retrieval. Here, the term frequency refers to the total number of times a term occurs in each document, and an inverse document frequency factor is used to diminish the weight of terms that occur very frequently in the document set, and increase the weight of terms that occur rarely. Nowadays, in machine learning, it is one of the most popular term-weighting scheme [31]. Moreover, to incorporate with different types of datasets, normalization is conducted in the preprocessing phase in order to assemble the whole data within a specific range, which ensures remarkable performance. Almost in all cases, it is complex to use high-dimensional data for creating a model. Increasing the number of features can lead to impoverished and inconsistent accuracy and efficiency. To overcome this problem, in this study, principle component analysis and self-organizing map are used for feature reduction and clustering. Since different sizes of big datasets are considered on this work, in order to make the training algorithm faster, the input dataset is divided into three clusters using Self-Organizing-Map (SOM). In addition, this work is also focused on in the data visualization mapping phase of the dimensionality reduction mechanism within SOM. Then, each cluster is trained using Adam deep learning architecture, which ensures good accuracy and efficiency. The main goal behind adopting this scheme is to reducing the complexity of the training using Neural Network. After analyzing the results, the sentiments are classified. Because data are similar in pattern and small cluster size, the training process is much faster and the error lower.

EXPERIMENTATION AND RESULTS

In this work, seven datasets with different sizes were studied. The majority of the data used was collected by using Facebook Graph API and Twitter API from Facebook and Twitter social network users aged between 18 and 30 years. Table 1 indicates the datasets used with the proposed approach for classification and analysis.

Table 1. Used Datasets

Dataset Number	No. of instances
Dataset1	15k
Dataset2	30k
Dataset3	45k
Dataset4	60k
Dataset5	75k
Dataset6	90k
Dataset7	100k

Initially, it was conducted tokenization of words on the datasets using Natural Language Toolkit (NLTK). This built-in toolkit helps to tokenize every sentences into words. The following demonstrates word tokenization by NLTK:

Love my kindle too Also enjoyed the New York Times article thought provoking as well kindle loving

⇩

"Love" "my" "kindle" "too" "Also" "enjoyed" "the" "New" "York" "Times" "article" "thought" "provoking" "as" "well" "kindle" "loving"

With the execution of this phase on the used datasets, results filtered tokenized words based on the NLTK.corpus library. A stop word is basically that kind of word that a search engine is programmed to ignore and those stop words have no meaning for sentiment classification. Therefore, these words were ignored before the training phase. The following tokenized example resulted after the NLTK stopwords filtering performing:

"Love" "kindle" "enjoyed" "New" "York" "Times" "article" "thought" "provoking" "kindle" "loving"

Since, machine learning approaches always works on numerical data, the input words are converted into the corresponding numerical values using the term frequency–inverse document frequency (TF-IDF) information retrieval mechanism. After that, Self-organizing map is implemented on those converted data to cluster them into similar groups. After clustering, it was performed the PCA based ADL. A comparative study on the findings of both the approaches in case of time requirement, space requirement and testing error, has been conducted on the preprocessed data and the obtained results are indicated in Table 2.

Table 2: Comparison between findings of ADL and proposed approach

Input Data	Results	Testing error (%)	Space requirment for training (MB)	Time requirement for training

								(ms)	
	ADL	**Proposed approach**	**Actual**	**ADL**	**Proposed approach**	**ADL**	**Poposed Approach**	**ADL**	**Proposed Approach**
[array(['will', 'never', 'go', 'ever', 'back'])]	0	0	0	Testing Error: 18.91% Accuracy: 81.09%	Testing Error: 11.66% Accuracy: 88.34%	1365 while training	729.34 while training	7645.32 ± 2 for training	7360.5 ± 2 ms for training
[array(['wonderful', 'servers', 'menu', 'loved', 'it', 'great', 'friendly', 'food', 'and'])]	1	1	1						
[array(['was','not', 'the', 'presentation', 'of', 'food', 'awful'])]	0	1	1						
[array(['very', 'service', 'is', 'excellent', 'everyone', 'customer', 'attractive'])]	1	1	1						
[array(['was', 'waiter', 'the', 'real', 'our', 'disappointment'])]	0	0	0						
[array(['you', 'that', 'can', 'beat'])]	1	1	1						
. . Toal instances 100k	. . .	. . .	. . .						

All the input data studied here were basically preprocessed by tokenizing into words from sentences using NLTK, then filtered using stopwords. After that, the TF-IDF mechanism was adopted to convert the data into corresponding numerical values. The following step employes SOM as well as PCA and then is obtained the data representation after execution of the transverse form of TF-IDF mechanism. In Table 2, "0" represents a negative sentiment, and "1" represents a positive sentiment. Moreover, the accuracy of the proposed approach was computed for the used datasets, Table 3. Hence, Table 3 let's one conclude that when we applied our approach on 15000 test cases, it obtained 13251 true classifications and 1749 false classifications, which corresponds to an accuracy of 88.34%. Most remarkable outcome we have found using the proposed approach is that it performs better with the increasing size of the data in the datasets, which is almost rare in most of the machine learning approaches.

Table 3: Classification results using the proposed appraoch

No. of Instances	**Instances In test Case (15%)**	**True Classification**	**False Classification**	**Accuracy (%)**
15k	2250	1905	345	84.67
30k	4500	3830	670	85.12
45k	6750	5797	953	85.89
60k	9000	7810	1190	86.78
75k	11250	9811	1439	87.21
90k	13500	11831	1669	87.63
100k	15000	13251	1749	88.34

Figure 4 shows the accuracy rate obtained for SOM based ADL, where x-axis is associated to the accuracy level, and the y-axis to the number of instances.

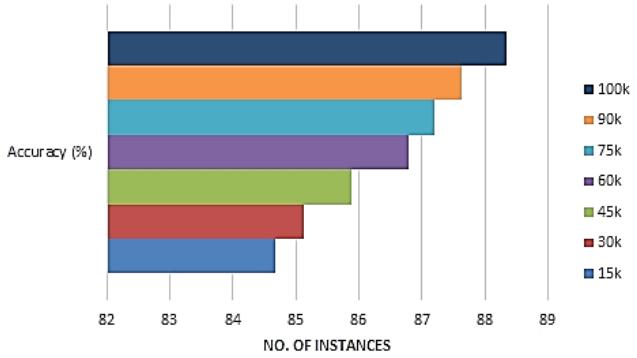


Figure 4: Accuracy rate of the classification results obtained by the proposed approach

The percentage error rate for 15,000 instances was: $\frac{|84.67-100|}{100}\times 100\% = 15.33\%$; and for 100,000 instances was: $\frac{|88.34-100|}{100}\times 100\% = 11.66\%$, which is very small to the limit that can be processed. The difference rate for 15,000 and 100,000 was: $\frac{|84.67-88.34|}{\left(\frac{84.67+88.34}{2}\right)}\times 100\% = 4.24\%$. When the number of instances in the dataset was 60,000, and the number of instances in the test case was 9000, the confusion matrix presented in Table 4 was obtained. A confusion matrix is basically an error matrix and a specific table layout that permits the visualization of the performance of a machine learning approach. The used performance measurement metrics were:

$$Accuracy = \frac{True\ Positive + True\ Negative}{Positive + Negative}\times 100\% \quad \text{.....(22)}$$

$$Specificity = \frac{True\ \ Negative}{False\ Positive + True\ Negative}\times 100\% \quad \text{.....(23)}$$

$$Sensitivity = \frac{True\ \ Positive}{True\ Positive + False\ Negative}\times 100\% \quad \text{.....(24)}$$

$$\Pr ecision = \frac{True\ Positive}{True\ Positive + False\ \ Positive}\times 100\% \quad \text{.....(25)}$$

$$\mathrm{Re}\, call = \frac{True\ Positive}{True\ Positive + False\ \ Negative}\times 100\% \quad \text{.....(26)}$$

Table 4: Confusion matrix obtained for the proposed approach

		Actual	
		Positive	**Negative**
Predicted	**Positive**	3830	340
	Negative	850	3980

From the confusion matrix in Table 4, one can confirm that the total number of true classifications was (3830+3980) = 7810 and of false classifications was (340+850) = 1190. By using equations 22 to 26, one get an accuracy of 86.78%, which is much better than the ones obtained by the other algorithms considered here. The obtained sensitivity, specificity, precision and recall values are presented in Table 5.

Table 5: Performance measurement values obtained for the proposed approach

Metric	Value (%)
Accuracy	86.78
Sensitivity	81.83
Specificity	92.12
Precision	91.84
Recall	81.83

The values in confusion matrix Table 5 confirm that the proposed approach achived high values of accuracy, sensitivity, specificity, precision and recall. On the other hand, the similarity between the values in Table 3 and 4 and the ones obtained for 60k instances can be noticeabled. Similarly, Table 6 indicates the classification accuracy using different sizes of datasets and a logistic regression classifier. The findings indicate that when the data set size is small, accuracy of predicting true classifications is higher than for a massive dataset. For example, when the number of instances are 60,000, the logistic regression predicted 7020 true classifications and 1980 false classifications from 9000 test cases, leading to an accuracy level of 78%, but when the number of instances was 100,000, the logistic regression predicted 10540 true classifications and 4460 false classifications in 15,000 test cases, leading the accuracy level to 70.26%.

Table 6: Classification results obtained applying Logistic Regression

No. of Instances	Instances In test Case (15%)	True Classification	False Classification	Accuracy (%)
15k	2250	1830	420	81.33
30k	4500	3630	870	80.67
45k	6750	5359	1391	79.39
60k	9000	7020	1980	78.00
75k	11250	8670	2580	77.06
90k	13500	9814	3686	72.69
100k	15000	10540	4460	70.26

Figure 5 shows the accuracy rate obtained for logistic regression, where x-axis represents the number of instances and y-axis the accuracy rate in percentage. Therefore, the overall findings demonstrated that the accuracy was quite meaningful for small sized datasets and also negligible as well. But, the performance of logistic regression diminishes over increasing the number of instances in the datasets.

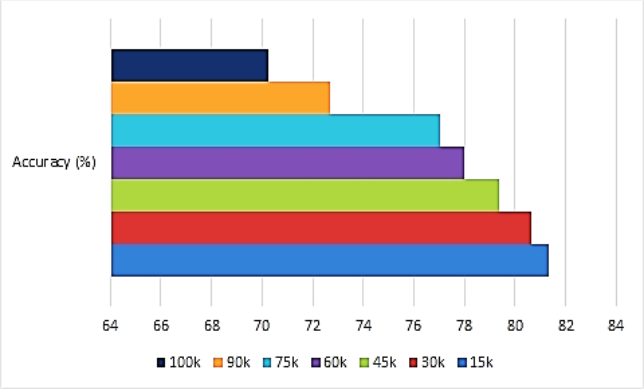


Figure 5: Accuracy rate obtained for the classification results by Logistic Regression

Moreover, percentage error, which is sometimes referred to as fractional difference, was also analized. If E refers to the experimental value, and A is defined as the accepted value, then the percentage error rate can be computed as:

$$\%Error\ rate = \frac{|E-A|}{A} \quad \text{.....(27)}$$

Thus, the percentage error rate for 15,000 instances was: $\frac{|81.33-100|}{100}\times 100\% = 18.67\%$, and for 90,000, was: is $\frac{|72.69-99|}{100}\times 100\% = 27.31\%$. Moreover, the percentage difference rate can be computed as:

$$\%Difference\ rate = \frac{|E_1 - E_2|}{\left(\frac{E_1+E_2}{2}\right)} \quad \text{.....(28)}$$

where E1 and E2 are the experimental accuracy for the two different datasets. For 15,000 and 90,000 number of instances, the percentage difference rate was: $\frac{|81.33-72.369|}{\left(\frac{81.33+72.69}{2}\right)}\times 100\% = 11.21\%$.

Table 7: Confusion matrix obtained for Logistic Regression

		Actual	
		Positive	**Negative**
Predicted	**Positive**	3340	820
	Negative	1160	3680

Table 7 presents the values as to the performance of Logistic Regression in our input data. The true classifications was computed as: (3340+3680) = 7020, and the false classifications as: (1160+820) = 1980, which are similar to the values in Table 6.

Table 8: Performance measurements of the logistic regression

Metric	Value (%)
Accuracy	78.00
Sensitivity	74.22
Specificity	81.77
Precision	80.29
Recall	74.22

Now, applying equations 22 to 26, the values presented in Table 8 were obtained for sensitivity, specificity, accuracy, precision and recall. These values are the measurements as to the performance of logistic regression in our input data. However, in case of comparison with our proposed approach, overall performance of logistic regression depending on these metrices are less significant.

Table 9: classification results obtained applying Random Forest

No. of Instances	Instances In test Case (15%)	True Classification	False Classification	Accuracy (%)
15k	2250	1575	675	70.00
30k	4500	3010	1490	66.89
45k	6750	4310	2440	63.85
60k	9000	5610	3390	62.34
75k	11250	6789	4461	60.34
90k	13500	7915	5585	58.63
100k	15000	8579	6421	57.19

Similarly, Table 9 presents the accuracy level of classification on the used datasets by applying Random Forest. From these findings, it is possible to conclude that, like Logistic Regression, Random Forest accuracy rate decreased as per the number of instances increased. For example, for 60,000 instances, the accuracy was 62.34%, while for 100,000 instances, the accuracy was 57.19%. Figure 6 illustrates the accuracy rate for the different datasets when Random Forest was used.

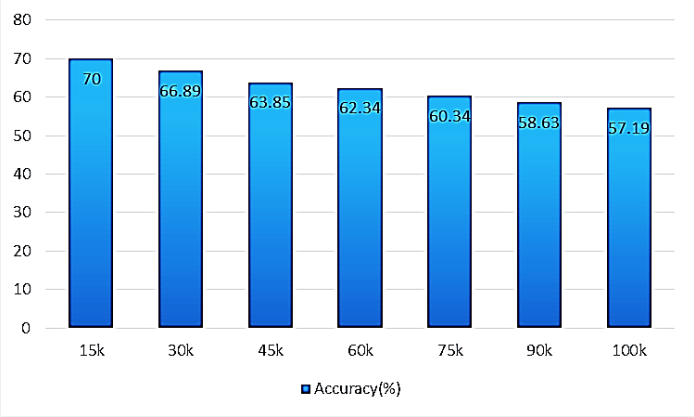


Figure 6: Classification accuracy rate obtained by applying Random Forest

In Figure 6, x-axis represents the number of instances, and y-axis represents the obtained accuracy. The percentage error rate for 15,000 instances was: $\frac{|70.00-100|}{100}\times 100\% = 30\%$, and for a number of 90,000 instances was: $\frac{|58.063-100|}{100}\times 100\% = 41.37\%$. For 15,000 and 90,000 instances, the percentage difference rate was $\frac{|70.00-58.63|}{\left(\frac{70.00+58.63}{2}\right)}\times 100\% = 17.67\%$. Now, the confusion matrix was computed for 60k instances, Table 10.

Table 10: Confusion matrix obtained for Random Forest

		Actual	
		Positive	**Negative**
Predicted	**Positive**	3100	1650
	Negative	1740	2510

Further evaluation on the performance achivied by random forest can be possible from the data in Table 10. The number of true classifications was 5610, and of false classifications was 3390, which was identical to the ones of Table 9. By using equations 22 to 26, the performance values presented in Table 11 can be abtained, which indicates the overall performance of this approach in the used datasets.

Table 11: Performance measurements obtained for the proposed approach

Metric name	**Value (%)**
Accuracy	62.34
Sensitivity	64.04
Specificity	60.33
Precision	65.26
Recall	64.04

Compared to logistic regression and to the proposed approach, random forest obtained poor performance in case of huge number of instances in the datasets. It obtained 62.34% of accuracy while the number of instances was 60k, while logistic regression had 78% and the proposed approach 86.78% of overall accuracy. Therefore, random forest can not be a good consideration in such cases of real life circumstances.

Table 12 presents the accuracy levels obtained by applying Polynomial Regression on the studied seven datasets. Also in this case, it was found that the smaller datasets had a higher accuracy level than the larger datasets. When the number of instances was 15k, polynomial regression demonstrated a quite better

accuracy (76%), but when the number of instances increased, it reacted by diminishiung the accuracy level. For example, when the number of instances was 100k, this approach obtained 65.34% of accuracy, which can be considered as low when compared to the ones of the other approaches considered in this work. Figure 7 illustrates the the accuracy rates against the number of instances, where x-axis holds the number of instances and y-axis the accuracy in percentage.

Table 12: classification results obtained by applying Polynomial Regression

No. of Instances	Instances In test Case (15%)	True Classification	False Classification	Accuracy (%)
15k	2250	1710	540	76.00
30k	4500	3310	1190	73.56
45k	6750	4897	1853	72.54
60k	9000	6251	2749	69.45
75k	11250	7689	3561	68.34
90k	13500	8932	4568	66.16
100k	15000	9801	5199	65.34

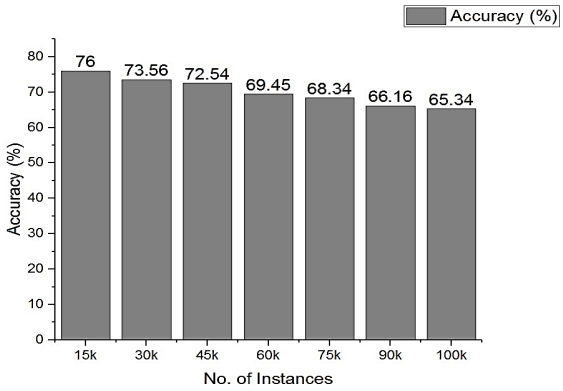


Figure 7: Classification accuracy rate obtained by applying Polynomial Regression

The percentage error rate for the number of instances equal to 15,000 was: $\frac{|76.00-100|}{100}\times 100\% = 26.00\%$, and for a number of instances of 90,000, was: $\frac{|66.16-100|}{100}\times 100\% = 33.84\%$. For 15,000 and 90,000 instances, the percentage difference rate was: $\frac{|76.00-66.16|}{\left(\frac{76.00+66.16}{2}\right)}\times 100\% = 13.84\%$.

Table 13: Confusion matrix obtained for Polynomial Regression

		Actual	
		Positive	**Negative**
Predicted	**Positive**	3031	1424
	Negative	1325	3220

When the number of instances in the dataset was 60,000 and the number of instances in the test case was 9000, the data in Table 13 was obtained.

Table 14: Performance measurements obtained by polynomial regression

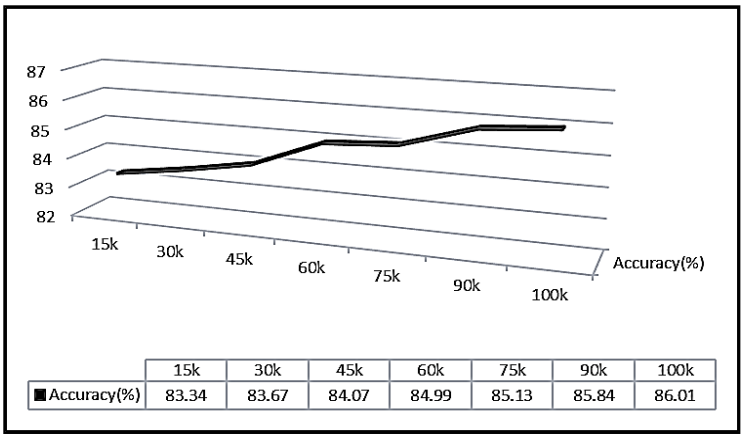
Metric name	Value (%)
Accuracy	69.45
Sensitivity	69.57
Specificity	69.32
Precision	68.01
Recall	69.57

From the confusion matrix in Table 13, we can notice that there were a total of 6251 true classifications obtained by applying polynomial regression on the used datasets. Meanwhile, 2749 false or miss classifications were obtaioned by polynomial regression. Therefore, for further evaluation the data in Table 14 was compute. Hence, table 14 presents the overall performance of the polynomial regression approach on the used datasets. Table 14 let's one confirm the better outcome of polynomial regression compared to random forest, but the poorer performance relatively to the proposed approach. Table 15 presents the accuracy classification rate obtained for the seven used datasets by applying PCA based ADL. The findings show that higher accuracy levels of true classifications result in bigger number of instantes than in lower number of instances.

Table 15: Classification results obtained by applying PCA based ADL

No. of Instances	Instances In test Case (15%)	True Classification	False Classification	Accuracy (%)
15k	2250	1875	375	83.34
30k	4500	3765	735	83.67
45k	6750	5675	1075	84.07
60k	9000	7649	1351	84.99
75k	11250	9578	1672	85.13
90k	13500	11589	1911	85.84
100k	15000	12901	2099	86.01

For example, the number of instances equal to 15,000, had an accuracy rate of 83.34%, while the number of instances equal to 100,000 had an accuracy rate of 86.01%. This indicates that when the dataset is big, the obtained accuracy is also high.



	15k	30k	45k	60k	75k	90k	100k
Accuracy(%)	83.34	83.67	84.07	84.99	85.13	85.84	86.01

Figure 8: Classification accuracy rate obtained by applying PCA based ADL

Figure 8 depicts the accuracy level obtained by PCA based ADL on different sized datasets. In this figure, the horizontal x-axis holds the number of Instances, and the vertical y-axis holds the accuracy rate in percentage. The percentage error rate for PCA based ADL when the number of instances was 15,000, was: $\frac{|83.34-100|}{100}\times 100\% = 16.66\%$, and when the number of instances was 90,000, was: $\frac{|85.84-100|}{100}\times 100\% = 14.16\%$. Moreover, the difference rate for 15,000 and 90,000 instances, was: $\frac{|83.34-85.84|}{\left(\frac{83.34+85.84}{2}\right)}\times 100\% = 2.95\%$.

Table 16: Confusion matrix built for PCA based ADL

		ACTUAL	
		Positive	**Negative**
PREDICTED	**Positive**	3389	640
	Negative	711	4260

Meanwhile, Table 16 presents the confusion matrix nuilt for PCA based CNN. From this table, one can conlude that when the number of instances was 60k, a total of 7649 true classifications along with 1351 of false or miss classifications were obtained. Using the performance metric equations, the corresponding values could be computed, Table 17.

Table 17: Performance measurements obtained for polynomial regression

Metric name	**Value (%)**
Accuracy	84.99

Sensitivity	86.93
Specificity	82.68
Precision	84.11
Recall	82.68

The experimental findings suggested that the proposed classifier led to better accuracy than the other approaches considered in this work. Moreover, our experiments surprisingly let's one conlude that our approach bt integrating competent preprocessing led to even better performances in cases of big as well as higher dimensional data. This feature is mandatory nowadays because all the data associated with sentiment are now unstructured and massive in nature.

Performance Evaluation

It has been previously proved that accuracy can be largely contributed by a large number of True Negatives. However, F-Score might be a better measure to use for balancing between precision and recall:

$$F - Score = 2 * \frac{\Pr ecision * \Re call}{\Pr ecision + \Re call} \quad \text{.....(27)}$$

By using the Precision and Recall values obtained by all studied algorithms, the F-Score values were computed to determine the best performing approach, Table 18.

Table 18: F-score value obtained for each of the studied algorithms

Algorithm Name	Precision	Recall	F-Score
Polynomial Regression	68.01	69.57	68.78
Logistic Regression	83.09	78.66	77.13
Random Forest	77.46	65.47	64.65
PCA based CNN	81.15	82.35	83.39
Proposed Approach	87.67	82.05	86.55

Table 18 presents the F-measure values obtained for all the studied approaches. Analyzing F-score values, one can conclude that the proposed approach led to much higher score comparetively to PCA based CNN and to all the other approaches available in the literature. The F-score value obtained for SOM based ADL was 86.55. Therefore, the proposed approach processed efficiently even on higher dimensional data as well as on big data. On the other hand, when the total number of instances in the dataset was 100k, the accuracy values obtained for each of the studied algorithms are the ones presented in Table 19.

Table 19: Accuracy values obtained for all the studied algorithms

Algorithm Name	Accuracy (%)
Polynomial	65.34%

Regression	
Logistic Regression	70.24%
Random Forest	57.19%
PCA based CNN	86.01%
Proposed Approach	88.34%

Table 19 allows a comparative study between all the studied algorithms performing on the used social networks sentiment datasets. It is quite remarkable to notice that SOM based ADL obtained higher accuracy than polynomial regression, logistic regression, random forest and even PCA based CNN.

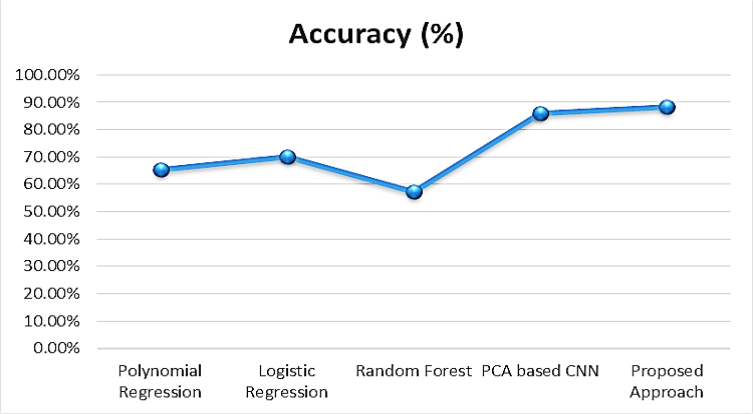


Figure 9: Accuracies obtained by each studied approach

Figure 9 presents the accuracy level obtained by each algorithm considered in this work. In this figure, X-axis represents the used algorithms and Y-axis has the accuracy (%). From the figure, we can determine that the proposed approach has the highest accuracy level compared to all the approaches. It obtained 88.34% of accuracy, while PCA based CNN obtained 86.01%, in a dataset with 100k of instances. Therefore, this confirms the superior performance along with the best efficiency of the proposed approach compared to well-known approaches available in the literature.

CONCLUSION

Sentiment analysis, more specifically, opinion mining has been appearing to be most effective predicting user's attitude and perceptions towards any topic by analyzing big social data. Therefore, in this work, we concentrated on unsupervised clustering and dimensionality reduction based classification approach (SOM based ADL) for classification as well as analysis of social network sentiments. For better accuracy and efficiency, we focused on Adam optimization technique to update convolutional neural networks parameters which has been proved to be a better optimizer compared to all the available optimization techniques. Moreover, for clear specification, we used the PCA approach for dimensionality reduction. Still, there are some cases where run time performance is comparatively high. To overcome this issue, our future work will be making the proposed approach parallel by combining parallel approaches available within machine learning techniques. Hopefully, embedding parallel approach will make a positive huge difference in maintaining highest performance.

REFERENCES

1) Ko, Y., & Seo, J.(2000). Automatic text categorization by unsupervised learning. In *Proceedings of the 18th conference on Computational linguistics* (Vol 1, pp. 453-459), Saarbrücken, Germany.
2) Agarwal, A., Xie, B., Vovsha, I., Rambow, O. and Passonneau, R.(2011), Sentiment Analysis of Twitter Data. In *Proceedings of the Workshop on Languages in Social Media*(pp. 30-38), , PA, USA.
3) Mohammadi, E., Thelwall, M., Kwasny, M., & Holmes, K. L. (2018). Academic information on Twitter: A user survey. *PloS one*, 13(5).
4) Bruns, A., & Weller, K.(2014). "Twitter data analytics – or: the pleasures and perils of studying Twitter", *Aslib Journal of Information Management*, 66(3).
5) Elouardighi, A., Maghfour, M., Hammia, H., & Aazi, F. Z. (2017). "A machine Learning approach for sentiment analysis in the standard or dialectal Arabic Facebook comments,". In *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)* (pp. 1-8), Rabat, Morocco.
6) Jain, A., P. and Dandannavar P.(2016), "Application of machine learning techniques to sentiment analysis*,". In 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*(pp. 628-632)*, Bangalore*, India.
7) Li, D., and Qian, J., "Text sentiment analysis based on long short-term memory,".In *2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)*(pp. 471-475)*, Wuhan, China*.
8) Neethu, M. S. and Rajasree R.(2013), "Sentiment analysis in twitter using machine learning techniques,". In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*(pp. 1-5), Tiruchengode, India.
9) Chachra, A., Mehndiratta, P., & Gupta, M.(2017), "Sentiment analysis of text using deep convolution neural networks,". In *2017 Tenth International Conference on Contemporary Computing (IC3)*(pp. 1-6), Noida, India.
10) Sankar, H., Subramaniyaswamy, V.,(2017). "Investigating sentiment analysis using machine learning approach,". In *2017 International Conference on Intelligent Sustainable Systems (ICISS)*(pp. 87-92), Palladam, India.
11) Duwairi, M. R., & Qarqaz, I.,(2014). "Arabic Sentiment Analysis Using Supervised Classification,". In *2014 International Conference on Future Internet of Things and Cloud* (pp. 579-583), Barcelona, Spain.
12) Gelbukh, A.(2017). "Sentiment analysis and opinion mining: Keynote address,". In *2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO*)(pp. 41-47)*,* Noida, India.
13) Agarwal, B., Mittal, N., & Cambria, E.(2013). "Enhancing Sentiment Classification Performance Using Bi-Tagged Phrases,". In *2013 IEEE 13th International Conference on Data Mining Workshops* (pp. 892-895)*, Dallas, USA*.
14) Ortigosa, A., Martín, J. M., & Carro, R. M.(2014). Sentiment analysis in Facebook and its application to e-learning. *Computers in human behavior*, 31 ,Pp. 527-541.
15) Hassan, A. U., Hussain, J., Hussain, M., Sadiq, M., & Lee, S.(2017). "Sentiment analysis of social networking sites (SNS) data using machine learning approach for the measurement of depression,". In *2017 International Conference on Information and Communication Technology Convergence (ICTC)(*pp. 138-140*),* Jeju, South Korea.

16) Jiang, P.(2012). "Comprehensive information emotional theory — An assumption of cognitive-emotional interaction mechanism,". In *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*(pp. 1386-1392), Hangzhou, China.
17) Rahman, L., Sarowar, G., & Kamal, S.(2018). Teenagers Sentiment Analysis from Social Network Data in *Social Networks Science: Design, Implementation, Security, and Challenges* (pp. 3-23). Springer, Cham.
18) Fersini, E.,(2017), Sentiment Analysis in Social Networks: A Machine Learning Perspective in Pozzi, F., A., Fersini., E., Messina & E., Liu, B.(Ed), *Sentiment Analysis in Social Networks*, pp. 91-111, Morgan Kaufmann.
19) Eshak, M. I., Ahmad, R., & Sarlan, A.(2017). "A preliminary study on hybrid sentiment model for customer purchase intention analysis in socialcommerce,". In *2017 IEEE Conference on Big Data and Analytics (ICBDA)*(pp. 61-66), Kuching, Malaysia.
20) Chen, S., Lin, L., & Yuan, X.(2017). Social media visual analytics. *Computer Graphics Forum*, 36(3), pp. 563-587.
21) Kamal, M. S., Sarowar, M. G., Dey, N., Ashour, A. S., Ripon, S. H., Panigrahi, B. K., & Tavares, J. M. R.(2017). Self-organizing mapping based swarm intelligence for secondary and tertiary proteins classification. *International Journal of Machine Learning and Cybernetics*, pp. 1-24.
22) Nandi, D., Ashour, A. S., Samanta, S., Chakraborty, S., Salem, M. A., & Dey, N. (2015). Principal component analysis in medical image processing: a study. *International Journal of Image Mining*, 1(1), pp. 65-86.
23) Virmani, J., Dey, N., & Kumar, V. (2016). PCA-PNN and PCA-SVM based CAD systems for breast density classification. In *Applications of intelligent optimization in biology and medicine*(pp. 159-180). Springer, Cham.
24) Rajinikanth, V., Satapathy, S. C., Dey, N., & Vijayarajan, R. (2018). DWT-PCA image fusion technique to improve segmentation accuracy in brain tumor analysis. In *Microelectronics, Electromagnetics and Telecommunications*(pp. 453-462). Springer, Singapore.
25) Lan, K., Wang, D. T., Fong, S., Liu, L. S., Wong, K. K., & Dey, N. (2018). A survey of data mining and deep learning in bioinformatics. *Journal of medical systems*, 42(8), 139.
26) Hodge, V. J., O'Keefe, S., & Austin, J. (2016). Hadoop neural network for parallel and distributed feature selection. *Neural Networks*, 78,pp. 24-35.
27) Guo, T., Dong, J., Li, H., & Gao, Y. (2017). "Simple convolutional neural network on image classification,". In *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA*)(pp. 721-724), Beijing, China.
28) Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM, 60(6)*,pp. 84-90.
29) Jaderberg, M., Simonyan, K., & Zisserman, A. (2015). Spatial transformer networks. In *Advances in neural information processing system*s (pp. 2017-2025), arXiv:1506.02025v3 [cs.CV].
30) Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization, In *the 3rd International Conference for Learning Representations, San Diego*, arXiv:1412.6980v9 [cs.LG].

31) Mishra, A., & Vishwakarma, S. (2015). "Analysis of TF-IDF Model and its Variant for Document Retrieval," In *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*(pp. 772-776), Jabalpur, India.