International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Opinion Mining Based Fake News Detection in Tweeter Sentiment Analysis Using Lexical Cross Mutation Deep Vectorized Convolution Neural Network

Dr. P. Kavitha[*1]

Submitted: 17/01/2024 **Revised**: 25/02/2024 **Accepted**: 03/03/2024

Abstract: Nowadays, there is a lot of data to process on social media, making it impossible for traditional companies to manage this big data. "Big data" refers to various factual bursts produced by multiple sources. It is categorized into five features or attributes: high volume, different reliability, character and speed. Because this kind of data is beyond the control of conventional systems, this kind of data is beyond relation data, so we can get insights into both structured and unstructured data that had a problematic prediction. To resolve this problem, we propose an Opinion mining-based Fake news detection in tweeter sentiment analysis using Lexical Cross Mutation Deep Vectored Convolution Neural Network (LCM-DVCNN). This process the tweet terms from topic discussion under the dictionary of terms extraction based on Senti Lexicon Demp Score (SLDS), which observes the critical terms to prediction. Then, a unigram evaluation was performed to show the features using the Lexical Uni-Negation Algorithm (LUnA). Based on the feature lexical weights, Topic Cross Mutation Fuzzy Future Selection (TCMFFS) observed the mutual relation features. Then, using the selected features using a Deep Vectorized Conventional Neural Network (DVCNN), these can be predicted based on the gate weights for classifying polar weights. This makes it a better recall in classification accuracy compared to other methods.

Keywords: *Tweet Data Prediction, Fake Content Analysis, Feature Selection and Classification, big data, CNN, and opinion mining.*

1. Introduction

With the rapid growth of data sources on the Internet, it is necessary to analyze millions of posts or comments to understand what users think on websites and microblogs. It is one of the most rapidly emerging platforms for sending and receiving messages and tweets from many people on Twitter. Users typically express their views and ideas concerning associated topics via tweets. The fake news identification from social content forums to improve the prediction accuracy based on big data analysis. This section describes the overview of this research with challenging problem identification factors and the objective of this research in social media.

On social media platforms, Twitter, the public posts their opinions about politics or parties and the political discourse arena. The Twitter intervention has also been shown to impact political activity significantly. The Twitter feed used for political analysis helps us understand the public's response to government issues and political reforms. Some tweets with negative emotions are more widespread than positive ones. In addition, it can use the Sentiment of political tweets to track and influence political opinions, detect consistency between politicians' statements and actual preferences, and predict election outcomes.

[1]*Assistant Professor, Department of Computer Applications, Dhanalakshmi Srinivasan College of Arts & Science for Women (Autonomous), Perambalur- 621212, Tamilnadu, India*
** Corresponding Author Email: kavitha1234ms@gmail.com*

Political Sentiment Analysis (SA) on Twitter is complex because it involves informal language and spelling mistakes. Some tweets are usually marked with unceremonious language, and some short messages show limited emotions. Hashtags, URLs, abbreviations, emoticons, and acronyms are often widely used on Twitter; thus, it is very difficult to analyze. The proposed study is about how positive and negative opinions are disseminated on social media and how vital events affect public opinion. In this regard, the proposed algorithm is focused on analyzing political tweets as positive and negative tweets.

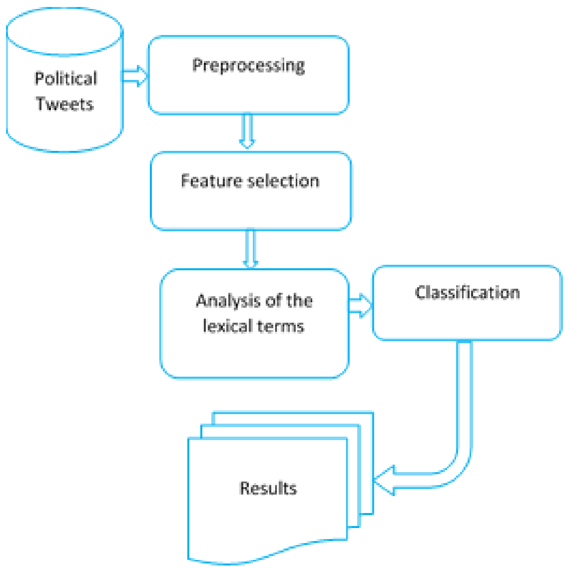

Fig. 1. Process of Political Sentiment Analysis (SA)

The proposed preprocessing removes noise and fills the missing values to make tweets more manageable and adaptable to sequential data processing strategies. Then, feature selection selects syntactic and lexical terms and the polarity of tweet sentences to enhance the classification performance.

Nowadays, false messages can be generated from subjective sentences using Machine Learning (ML) algorithms to observe features. Additionally, semantic-based approaches can analyze the meaning of concealed connections. Because social media, particularly its content, consists of brief texts, individuals often employ expressions and abbreviations like emotional, sarcasm, and predominant words, making it challenging to effectively extract their Sentiment using existing Natural Language Processing (NLP) systems. Figure 5 is expanded as a political SA process by using different quantities and features every time to use SA and find sentiments. Emotions can be detected using words, text or queries from social media based on some methods—an ordering procedure to classify reviews as either opinions classification as negative or positive. According to the text granularity, the SA analysis in the Deep Neural Network (DNN) can be divided into three levels: the word, the sentence, and the chapter, as well as the analysis of the word-level SA word. Word SA is a prerequisite for sentence reputation analysis.

The proposed approach categorizes positive and negative tweets about political information. The primary contribution of this paper is that the proposed algorithm delivers high classification performance and reduces time complexity, as depicted in Figure 1. The remaining paper follows Chapter 2, a literature review of previous methods; Chapter 3, which includes the proposed implementation; Chapter 4, which consists of the proposed algorithm's result analysis; and Chapter 5, which provides the conclusion.

2. Related Work

Tweets help to analyze a user's feelings about multiple political parties. In reference [1], the author examined the use of ML methods like SVM (Support Vector Machine) and NB (Naive Bayes) for analyzing the public opinion of various political parties. However, the algorithms could have offered the proper accurate results.

In ref [2], the author focuses on SA on Twitter data using sentiment diffusion called sentiDiff to detect the polarity of textual information on Twitter. That study only observed the textual information of political tweets, and it is difficult for them to work well in the face of short, ambiguous Twitter messages.

In ref [3], the author introduced the DCNN (Deep Convolutional Neural Network) algorithm to find the polarity and lexical terms of the tweets classification. Similarly, in ref [4], the author uses Conv-BiLSTM (CNN (Conv) - Bidirectional Long Short-Term Memory (BiLSTM)) to detect the SA on political tweets. The author considered only the classification process and didn't focus on feature selection, which leads to predicting the wrong classification results.

In reference [5], the author introduced ordinal regression (OR) using ML techniques like support vector regression (SVR), RF, and DT to forecast political Twitter SA. Similarly, in ref [6], the author provided ensemble algorithms, including SentiWordNet (SWN), NB, and Hidden Markov Model (HMM), to forecast political election outcomes based on public sentiment tweets, both positive and negative. Classifiers do not guarantee the best result performance with invisible data.

In ref [7], the author suggested that the majority of votes can be used by various steps to identify the Twitter SA, which can be used by the cluster approach to a single, complete, and average linkage. Nevertheless, the approach could be more time-consuming for a vast dataset.

In ref [8], the author examines the binary and ternary approaches to categorize the text as positive or negative. However, in that method, challenging tasks during the classification process.

In ref [9], the author introduces the Conv and LSTM algorithms to improve the SA of classification performance. That study focused on something other than the feature selection process, which is challenging and time-consuming.

In ref [10], the author introduces the Feature Ensemble Model (FEM) and Fuzzy Sentiment (FS) to analyze political tweet words' polarity and lexical terms. However, that method tries to syntactic information of a word without observing the sentiment context of the tweet word.

In ref [11], the author suggests automatically getting a text sample from Twitter. It also addresses the challenge of training a classifier on the imbalance between positive and negative sentiments in the training and testing dataset samples. Nevertheless, it gives only moderate classification results.

In ref [12], the author introduces the Dynamic Bayesian Network (DBN), which models the Sentiment of political topics. That method primarily focuses on classifying emotions, but the author should have focused on more than preprocessing and feature selection.

In ref [13], the author introduces the Deformable Conv and Bi-LSTM algorithm to extract the sentiment features from political tweets—the problem of inadequate feature extraction capabilities for local and long-distance-dependent features.

In ref [14], the author presented Sparse Self-Attention-

LSTM (SSA-LSTM) to proficiently analyze the intuitive truths and build an essential sentiment on Twitter. Similarly, in ref [15], the author proposes an ensemble classifier and Topic-Adaptive Sentiment Classification (TASC) to select reliable political tweets. The tweet's text could be more sparse, reducing the sentiment classifier's performance.

In ref [16], the author proposes Iterative Opinion Mining using Neural Networks (IOM-NN) algorithm to detect the syntactic and polarity of tweet sentences. This method cannot provide proper classification results.

In ref [17], the author introduces a Sentiment Hashtag Embedding (SHE) model and Conv classifier. The twain methods allow for maintaining both label semantics and sentiment distribution. However, in that method, hashtag embedding often captures the semantic distribution of hashtags and cannot predict sentiment polarity.

In ref [18], the author uses ML techniques for automated tweet classification. Similarly, in ref [19], the author proposes the Latent Dirichlet Allocation (LDA) method to classify the geotags on various topics with positive Sentiment. However, that method offers misclassification results.

In ref [20], the author presents the Genetic Algorithm (GA), and the ML algorithm is used for feature reduction and classification. However, this method can't provide proper results with a substantial political dataset.

3. Proposed Modelling

The proposed mining-based fake news detection sentiment acts on the impression of the impression of the unidentified tweet data accurately and automatically to create a valuable assortment in this proposed tweeter SA using lexical cross-mutation DVCNN. The ensuing considerations should be evaluated during the implementation to develop specialization. To improve the tweet classification feature selection to make an efficient approach in preprocessing to point the data points based on the feature selection, which will include category as the label of classes. To deploy a neural classifier for classification using DCNN based on the optimal feature selection model. To improve the classification accuracy using multi-attribute case prediction for categorization using the feature selection.

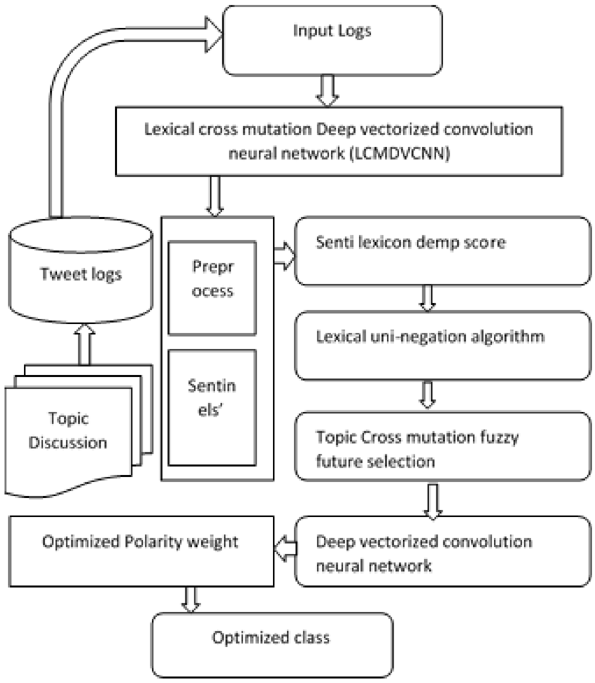

Fig 2. Architecture Diagram LCM-DVCNN

The above Fig 2 demonstrates the Gradient-boosted DT classification utilizing attribute-based subset feature selection. We use subset lexical hypothesis approach networks for straightforwardness to allude to the DVCNN developed with the proposed attribute feature learning calculation. As for the sentiment score calculation plan, the principal refinement of the proposed feature calculation is the novel extraordinary feature calculation intended to develop fake news analysis effectively.

3.1. Senti lexicon demp score

The case for this sentence is to concentrate on the lexical terms to reduce the length of the viewer sentence. This indicates that the tweet counting term was extracted from the record to determine the frequency relationship. Therefore, the Dempster Shafer Theory (DST) method has been applied in this proposed work and predicts the scores of the Sentiment. They detect the sensitive polarity. DST calculates individual emotions and the overall weight of the sentence to derive the extent of the concept of a query from the subjective energy of the relevant query. It calculates Dempster's rules for incorporating layers of such ideas while simultaneously being primarily based on reasonable evidence.

Algorithm:

Input: collective Tweet dataset Crmds

Output: Noise-free Tweet dataset Ps

Step 1 Initialize the preprocessed dataset Tw

Process Start

Read all the datasets Rds →RTw1, Tw2.

Step 2 Extract the subjective term on topic forums Tp→Twn

Process all The tweet terms

For Tw to N on subjective terms

Rteun the lexical ters from Each tweet

Ls←St

Step 3: estimate all the relevant subject score

Check the topic relation ters on each hashtag

Return all the selective lexical terms Ss = $(\sum_{n=1}^{size(Ds)}$ Text€Di) × Splitby(.)

Return all the terms Rs← particular word (SL)

Step 3 Estimate the relative for polarity terms

For all Selective terms, Sl

$$\sum_{i=1}^{n}(Ne_i + N_i) \tag{1}$$

Execute the objective terms of extraction N(i)

Compute N, P, Ne terms

$$PS \leftarrow PScore = \frac{\sum_{i=1}^{n}(Ne_i+N_i)}{T_i} \tag{2}$$

The P score point is used as the compute term for each computation.

If Negative terms of access on keyterm

$$NS \leftarrow NScore = \frac{\sum_{i=1}^{n}(P_i+Ne_i)}{T_i} \tag{3}$$

Return Ns score

$$S(\emptyset) = 0 \sum_{i=1}^{n} t' \sum_{j=1}^{n} Se = 1 \tag{4}$$

Else

Segregate the Se terms with objective lexical X, Y.

$$Se_{1,2}(S) = \frac{\sum X \cap Y = S\, Se_1(X) S_e(Y)}{1 - \sum X \cap Y = \emptyset Se_1(X) Se_2(Y)} \tag{6}$$

Return the lexical demp as S.

End

End

Step 4: Estimate the lexical redundant for integration approach Ia

Ia←Se

$$Se_{1,2,\dots,n}(S) = \frac{\sum \cap_{i=1}^{n} X_i = S\left(\prod_{j=1}^{n} Se_j(X_i)\right)}{1 - \sum \cap_{i=1}^{n} Se = \emptyset\left(\prod_{j=1}^{n} Se_j(X_i)\right)} \tag{7}$$

End

End

Stop

The above algorithm prepares tweet data for further related relationship link identification processes. Next, the Senti_Temp_Score is activated, focusing on how to separate emotions, acronyms and emoticons. The Senti_temp_score algorithm collects preprocessed data, storing all tweets in Ts. The feature selection is then used as a unigram, and the blank sign separates the words. This reduces the dimension of unwanted tweet datasets and simplifies the process. This process reduces the time-consuming hassle of tweet method surveys by selecting fewer features for efficient tweet analysis. It finds that the behavior of the data comes from crucial keywords.

3.2. Lexical uni-negation algorithm

In text classification, unigrams are single words, and they are used as a standard feature. These words will only be used if they occur more than once in a tweet data set. We extract an unigram to count its frequency, and if there are multiple frequencies, we add the word as a function and set its value to the frequency of the population. The opinions could be handled with negations because the negation can change the sentence meaning in the sentiment phase, which is very complex to determine the polarity. Therefore, the sentiments could be isolated into different classes, like positive, negative, and neutral. The Senti_Negation approach detects the polarity and classifies the Sentiment, respectively. Negations are computational linguistics because they inversely affect the polarity of the Sentiment.

The proposed senti_negation approach evaluates extensive data sensory knowledge using a lexicon-based approach. Tweets showing negative or positive emotions are labeled accordingly. The tweet will be marked as neutral if they do not see the emotion.

$$total\ words\ in\ T'(T'Words) \leftarrow \sum_{i=0}^{n} unigram(T'_i) \tag{8}$$

$$unigram\ (T'_i) \leftarrow \sum_{i=1}^{n}\left(\sum_{j=1}^{n} t'_{\cup}\right) \tag{9}$$

To find the polarity of lexical term difference,

$$T'_{word}\ polarity \leftarrow T'_p + T'_{Ne} \tag{10}$$

$$T'_P \leftarrow \sum_{i=1}^{n} T'_{pi} \tag{11}$$

Positive negative class

$$T'_{Ne} \leftarrow \sum_{i=1}^{n} T'_{Ne(i)} \tag{12}$$

refined neutral be featured

$$T'_{Ne} \leftarrow \sum_{i=1}^{n} T'_{Ne(i)} \tag{13}$$

observed as

$$\sum_{i=1}^{n} t'_{polarity} \leftarrow sum\left(\sum_{i=1}^{n} t'_{p(i),} \sum_{i=1}^{n} t'_{Ne(i)}\right) \tag{14}$$

for each t'

if $t'_{P(i)} > t'_{Ne(i)} then\ T'_{class(i)} \leftarrow Positive$

else if $t'_{Ne(i)} > t'_{P(i)} then\ T'_{class(i)} \leftarrow Negative$

else

$T'_{class(i)} \leftarrow Neutral$

end for

This algorithm works well for emotional classification and polarization measurements. In the process, all tweets are taken for sorting. Senti_nation then focuses on categorizing emotions, abbreviations, and emoticons with denial.

3.3. Topic Cross mutation fuzzy feature selection

Fuzzy-based approach lectures to arrangement learning and assistances to control fitness estimate comparing for appropriate pronouncement base for dissimilar ranges. Classify the identified terms using dictionary-based techniques in positive and negative sentiments. Generate merged Data Frame of classified terms and cumulative frequency to get Input Parameters: Sentiment Category, Frequency, and Share/like/retweet Count. Applying fuzzy clustering means getting cluster groups of similarity from available data. The nearest mean weight can be determined by categorizing the neural network using the membership function if the categories of two attributes closely match the "classic note for classic note by Fazzy. The particle condition fluctuates due to the persistent disparity in simplicity.

$$\vec{\mathcal{P}}_i = \vec{p}_i + \vec{\mathcal{V}}_i \quad (15)$$

After updating, $\vec{P}_i$ is verified and should be within the acceptable range.

Step 4: Updating of memory – Update $\vec{P}_{i,best}$ and $\vec{\mathcal{P}}_{gi,best}$ using the formula.

$$\vec{\mathcal{P}}_{i,best} = \vec{p}_i \ if\ f(\vec{p}_i) > f(\vec{p}_{i,best}) \quad (16)$$

$$\vec{\mathcal{G}}_{i,best} = \vec{g}_i \ if\ f(\vec{g}_i) > f(\vec{g}_{i,best}) \quad (17)$$

Where,($\vec{x}$)-point function to expand

The spider foraging model prioritizes the selection of maximum fitness weights by considering the strong correlation between conflicting traits, such as facial expressions and textual features, similar to strict words. It groups these features and selects the relational feature using the search function. The corresponding features are then marginalized to extract the closest maximum weight. This process creates an inner corner to predict the connected features, as the same class is closest to the centroid of the class.

3.4. Deep vectorized convolution neural network

In tweet tags, DVCNN determines the emotions of attitudes, and it can be widely used for text cognitive analysis. In addition, the proposed DVCNN algorithm can analyze positive, negative and neutral tweets to classify valuable data. The probability of activating the polynomial process can be predicted using a definitive release layer in the Softmax process. In addition, various classifications provide the operation of three or more label types using problems.

$\sigma\left(\underset{X}{\rightarrow}\right)_n = \frac{a^{x_i}}{\sum_{i=1}^{s} a^{X_j}}$, for N=1….S, and x= $(X_1 \ldots X_s)$ (18)

Softmax is very valuable in converting the score of the average probability into a score, which may provide the input of other computers prescribed to the user.

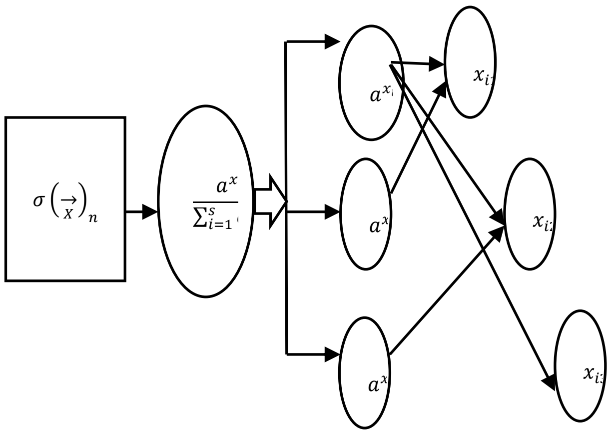

Fig 3: Sof-tmax training vector in hidden neurons

Here, $\underset{X}{\rightarrow}$ describes the actual input vector of the multivariate classifier. x_i -an input vector that can have positive or negative values. Each element of the input function, a^{x_i}s -is essential for the probability function. $\sum_{i=1}^{s} a^{X_j}$- is used to verify if the output function is a probability function.

Algorithm Steps

Input: A feature selection dataset

Output: Tweets classification

Step 1: Computation the feature evaluation

Step 2: Computing the $S_s^R - \mathcal{S}_s = s_1, s_2 \ldots s_n$

For each tweet dataset, Dr_i from Drs

For each dataset, Dr_i from Drs.

Feature T = Extracted node from Di.

Subset s_s= $(\sum_{n=1}^{size(Ds)} T_t(V) \in D_i) \times S_t(C)$

For each node, scales point to the sentimental term D_{si} from D_r

End for

Repeat

Step 3: $C_w = (\frac{\text{In tag}}{Nr} \times \frac{\text{Ext tag}}{Nr}) + NIL$

Step 4 Return the polarity weight of CW by class by reference

Step 5: End

The algorithm above is specifically designed for sentiment analysis. The DVCNN algorithm categorizes most unstructured review text into social discussion comments with positive, negative, or neutral sentiments. WhereS_s^R is the syntactic-senti rule, $T_t(V)$ −tweet text value, $S_t(C)$ −separate tag class, C_w −Class weightages

4. Result and Discussion

In this category, the proposed results can be compared to previous methods to define the performance of the analysis. Furthermore, comparing previous algorithms such as DCNN and TASC, the proposed LCM-DVCNN can be made by the algorithm.

Table 1: Simulation parameters

Simulation	Variable
Tool	Anaconda
Language	Python
Dataset Name	Political Twitter SA
Total number of dataset	1000
Training data	700
Testing data	300

The details of the simulation parameters about comparing the proposed LCM-DVCNN approach to existing approaches are described in Table 1, and their confusion matrix calculates all parameters. The precision is calculated using the following Equation:

Precision (P) = $\frac{AP}{AP+FP}$x100

Let's assume AP is actual positive, AP is true positive, correctly identifying the actual value, and FP is false positive, incorrectly identifying actual values.

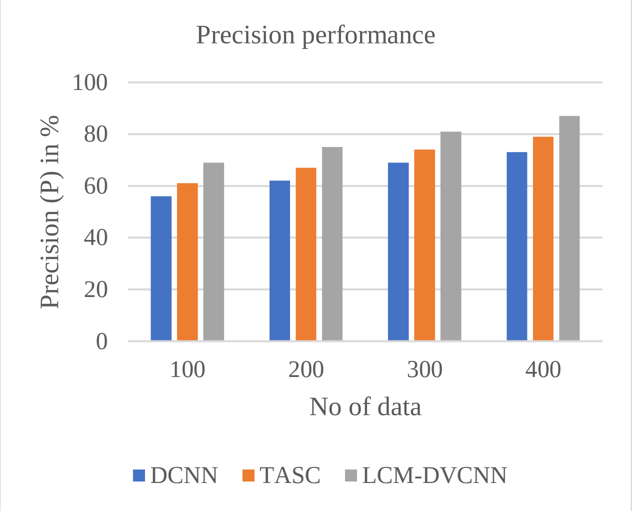

Fig 4: Analysis of Precision (P) performance

Fig 3 depicts the precision performance; the proposed LCM-DVCNN algorithm precision is 87% for 400 data. Similarly, the previous algorithm results show that the TASC algorithm precision is 79% and the DCNN algorithm precision is 73% for 400 data.

Recall (R) =$\frac{\boldsymbol{AP}}{\boldsymbol{AP+FN}}$**x100**

The above Equation is calculated as recall performance. Let's assume FN False-negative, which correctly identifies the negative values.

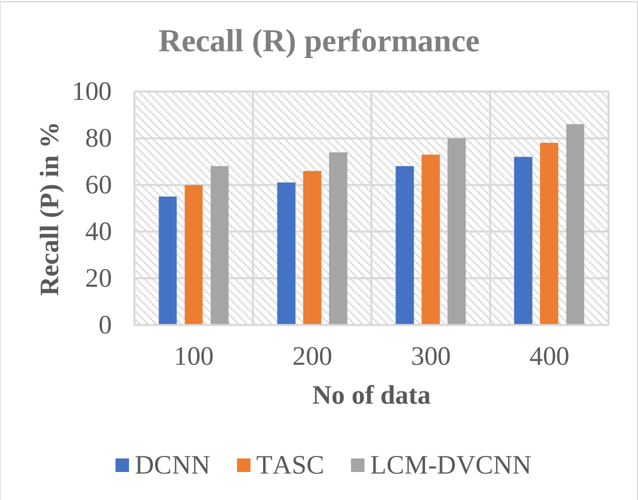

Fig 5: Analysis Recall (R) performance

As illustrated in Figure 5, the proposed algorithm can be analyzed by recalling and comparing the previous algorithms. The proposed LCM-DVCNN algorithm recall is 86% for 400 data; similarly, the previous algorithm results are that the TASC algorithm precision is 78%, and the DCNN algorithm precision is 72% for 400 data. The Equation is used to calculate the F1 score performance,

F1 score=2* $\frac{\boldsymbol{P*R}}{\boldsymbol{P+R}}$

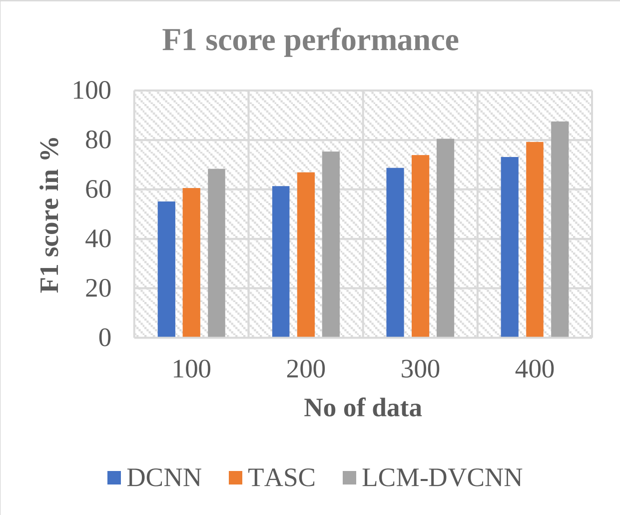

Fig 6: Analysis of F1 score performance

As described in Figure 6, the F1 score offers comparative results between the description analysis and the previous algorithm. The suggested LCM-DVCNN algorithm score is 87.43% for 400 data; likewise, the previous algorithm results show that the TASC f1 score is 79.21, and the DCNN algorithm f1 score is 73.1% for 400 data.

$$\text{Accuracy (Ac)} = \frac{AP+AN}{AP+FP+AN+FN} \text{x} 100$$

The above Equation calculates the correct value for the political tweet dataset. Let's assume an is negative, which correctly identifies the negative values.

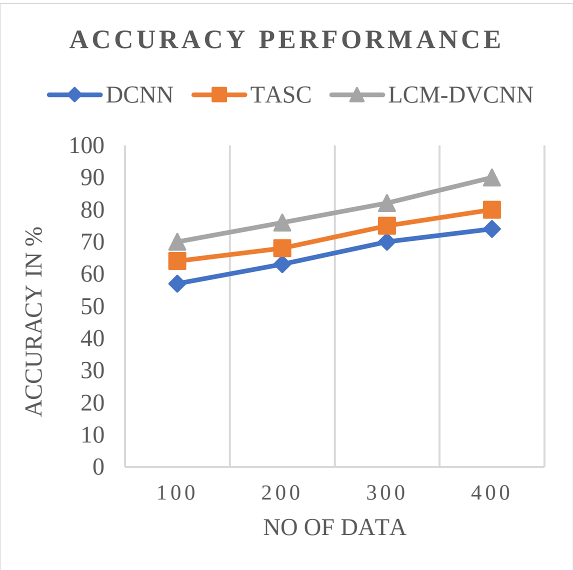

Fig 7: Exploration of Classification of Accuracy (Ac) performance

With an accuracy of 90%, Figure 7 presents the accuracy performance analysis of the suggested LCM-DVCNN method. Similarly, on 400 data sets, the results of the current approaches were 74% accurate for the DCNN algorithm and 80% correct for TASC.

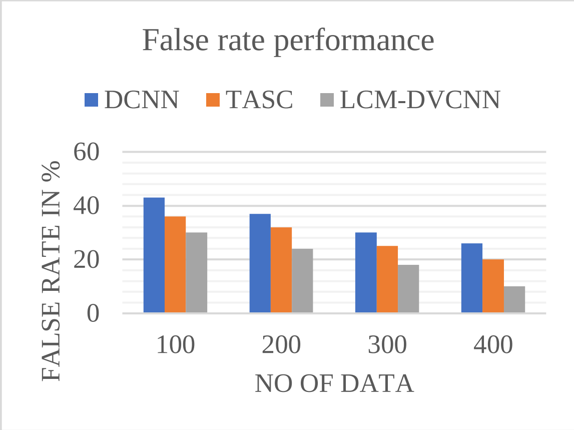

Fig 8: Analysis of False rate performance

An analysis of the false positive rate performance is shown in Figure 8. Compared to earlier methods, the suggested approach performs better regarding errors. The suggested LCM-DVCNN has an error rate efficiency of 10%. Similarly, current techniques yield a DCNN algorithm accuracy of 26% for 400 data sets and a TASC accuracy of 20%.

5. Conclusion

This paper presents the proposed algorithm for political tweet classification results, whether positive or negative. The proposed first step, preprocessing, fills in the missing values, reduces noise, and removes irrelevant values. After that, feature selection is analyzed for political tweet sentences' lexical terms, syntactic, and polarity. Finally, the proposed algorithm effectively classifies the political tweets' positive or negative sentences. The results of the suggested method are as follows: 87% for precision, 86% for recall, 87.43% for the F1-score, 90% for accuracy, and 10% for false rate performance. The proposed LCM-DVCNN algorithm gives better performance results than previous methods.

References

[1] Wu, Y., Khisti, A., Xiao, C., Caire, G., Wong, K. K., & Gao, X. (2018). V. K. K. M, P. Sunagar and A. Kanavalli, "Analysis of sentiments in political-based tweets using machine learning techniques," 2019 Global Conference for Advancement in Technology (GCAT), 2019, pp. 1-5, doi: 10.1109/GCAT47503.2019.8978440.

[2] L. Wang, J. Niu and S. Yu, "SentiDiff: Combining Textual Information and Sentiment Diffusion Patterns for Twitter Sentiment Analysis," in IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 10, pp. 2026-2039, 1 Oct. 2020, doi: 10.1109/TKDE.2019.2913641.

[3] Z. Jianqiang, G. Xiaolin and Z. Xuejun, "Deep Convolution Neural Networks for Twitter Sentiment

Analysis," in IEEE Access, vol. 6, pp. 23253-23260, 2018, doi: 10.1109/ACCESS.2017.2776930.

[4] S. Tam, R. B. Said and Ö. Ö. Tanriöver, "A ConvBiLSTM Deep Learning Model-Based Approach for Twitter Sentiment Classification," in IEEE Access, vol. 9, pp. 41283-41293, 2021, doi: 10.1109/ACCESS.2021.3064830.

[5] S. E. Saad and J. Yang, "Twitter Sentiment Analysis Based on Ordinal Regression," in IEEE Access, vol. 7, pp. 163677-163685, 2019, doi: 10.1109/ACCESS.2019.2952127.

[6] R. Jose and V. S. Chooralil, "Prediction of election result by enhanced sentiment analysis on Twitter data using classifier ensemble Approach," 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), 2016, pp. 64-67, doi: 10.1109/SAPIENCE.2016.7684133.

[7] M. Bibi, W. Aziz, M. Almaraashi, I. H. Khan, M. S. A. Nadeem and N. Habib, "A Cooperative Binary-Clustering Framework Based on Majority Voting for Twitter Sentiment Analysis," in IEEE Access, vol. 8, pp. 68580-68592, 2020, doi: 10.1109/ACCESS.2020.2983859.

[8] M. Bouazizi and T. Ohtsuki, "A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter," in IEEE Access, vol. 5, pp. 20617-20639, 2017, doi: 10.1109/ACCESS.2017.2740982.

[9] A. Feizollah, S. Ainin, N. B. Anuar, N. A. B. Abdullah and M. Hazim, "Halal Products on Twitter: Data Extraction and Sentiment Analysis Using Stack of Deep Learning Algorithms," in IEEE Access, vol. 7, pp. 83354-83362, 2019, doi: 10.1109/ACCESS.2019.2923275.

[10] H. T. Phan, V. C. Tran, N. T. Nguyen and D. Hwang, "Improving the Performance of Sentiment Analysis of Tweets Containing Fuzzy Sentiment Using the Feature Ensemble Model," in IEEE Access, vol. 8, pp. 14630-14641, 2020, doi: 10.1109/ACCESS.2019.2963702.

[11] S. Martin-Gutierrez, J. C. Losada and R. M. Benito, "Semi-Automatic Training Set Construction for Supervised Sentiment Analysis in Political Contexts," 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2018, pp. 715-720, doi: 10.1109/ASONAM.2018.8508386.

[12] H. Liang, U. Ganeshbabu and T. Thorne, "A Dynamic Bayesian Network Approach for Analysing Topic-Sentiment Evolution," in IEEE Access, vol. 8, pp. 54164-54174, 2020, doi: 10.1109/ACCESS.2020.2979012.

[13] J. Zhou, S. Jin and X. Huang, "ADeCNN: An Improved Model for Aspect-Level Sentiment Analysis Based on Deformable CNN and Attention," in IEEE Access, vol. 8, pp. 132970-132979, 2020, doi: 10.1109/ACCESS.2020.3010802.

[14] D. Deng, L. Jing, J. Yu and S. Sun, "Sparse Self-Attention LSTM for Sentiment Lexicon Construction," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 27, no. 11, pp. 1777-1790, Nov. 2019, doi: 10.1109/TASLP.2019.2933326.

[15] S. Liu, X. Cheng, F. Li and F. Li, "TASC: Topic-Adaptive Sentiment Classification on Dynamic Tweets," in IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 6, pp. 1696-1709, 1 June 2015, doi: 10.1109/TKDE.2014.2382600.

[16] L. Belcastro, R. Cantini, F. Marozzo, D. Talia and P. Trunfio, "Learning Political Polarization on Social Media Using Neural Networks," in IEEE Access, vol. 8, pp. 47177-47187, 2020, doi: 10.1109/ACCESS.2020.2978950.

[17] L. G. Singh, A. Anil and S. R. Singh, "SHE: Sentiment Hashtag Embedding Through Multitask Learning," in IEEE Transactions on Computational Social Systems, vol. 7, no. 2, pp. 417-424, April 2020, doi: 10.1109/TCSS.2019.2962718.

[18] P. Vyas, M. Reisslein, B. P. Rimal, G. Vyas, G. P. Basyal and P. Muzumdar, "Automated Classification of Societal Sentiments on Twitter with Machine Learning," in IEEE Transactions on Technology and Society, doi: 10.1109/TTS.2021.3108963.

[19] T. Hu, B. She, L. Duan, H. Yue and J. Clunis, "A Systematic Spatial and Temporal Sentiment Analysis on Geo-Tweets," in IEEE Access, vol. 8, pp. 8658-8667, 2020, doi: 10.1109/ACCESS.2019.2961100.

[20] F. Iqbal et al., "A Hybrid Framework for Sentiment Analysis Using Genetic Algorithm Based Feature Reduction," in IEEE Access, vol. 7, pp. 14637-14652, 2019, doi: 10.1109/ACCESS.2019.2892852.