Rev. Roum. Sci. Techn.– Électrotechn. et Énerg.
Vol. **69**, *2*, pp. 201–206, Bucarest, 2024

EFFECTIVE OFFENSIVE LANGUAGE DEDUCTION USING DEEP LEARNING IN SOCIAL MEDIA

KALAIVANI ADAIKKAN[1], DURAIRAJ THENMOZHI[2]

Keywords: Offensive language detection; Graph-based deep learning (GDL); Red fox optimization (RFO); Term frequency-inverse document frequency (TF-IDF); Lexicon-based feature.

Offensive language detection is the technique of identifying and detecting user-generated offensive comments such as insults, pain, profanity, and racism that are targeted at a specific individual or group on social media. As social media platforms become more prominent, offensive language is used more frequently, becoming a major challenge in modern society. A novel effective offensive language classification (EOLC) technique has been proposed to overcome these challenges. English language tweets from YouTube and X (Twitter) with offensive, mild, swear, and non-offensive tweets are used in this paper. Initially, the tweets and comments are pre-processed, and the features are extracted using different techniques, namely term frequency-inverse document frequency (TF-IDF), WordVec, and lexicon-based features. The extracted features are classified using the graph-based deep learning (GDL) method for numerical representation and decision-making. GDL network is optimized with red fox optimization (RFO) to normalize the weight and biases of the network and achieve better accuracy. The proposed GDL model achieves the highest levels of classification accuracy on the X (Twitter) and YouTube datasets, with 95.5 % and 96.8 %, respectively. The results obtained from GDL are more accurate and of higher quality than those obtained from traditional classifiers. The proposed EOLC method improves the overall accuracy by 5.56 %, 7.4 %, 7.7 %, and 10.2 % better than Text CNN, CNN-LSTM, DRNN, and LogitBoost, respectively.

1. INTRODUCTION

Nowadays, social media is one of the most widely utilized mediums for people to convey their opinions and thoughts online, including text, voice, hate speech, and character images [1]. Users can connect and share their ideas on various topics, including events, videos, and entertainment, using social media platforms like WhatsApp, X (Twitter), and Facebook. The internet has allowed people of many religions, nations, languages, backgrounds, genders, and ethnicities to engage [2].

People can now easily voice their opinions without fear because of the increased availability of laptops, cell phones, tablets, and other devices [3]. Foul comments, such as abuses, disappointments, blackmail, and insults, can create stress and hurt the mental health of social media users [4]. It is important to control offensive language on social media so children and teenagers don't learn offensive language from it [5]. Therefore, it is necessary to identify offensive language on social media.

In recent years, the bag of words (BoW) technique has been widely used for feature extraction [6]. However, there are some drawbacks. If a new sentence contains a new word, the vocabulary would grow, which would cause the vectors to become more complex. To overcome this issue, this research used TF-IDF, lexicon-based features, and word2Vec. In the TF-IDF model, both significant terms and less significant terms are included [7]. Comments and tweets can be categorized according to their meaning or semantics when using lexicon features. Word2Vec retains the semantic meaning of different words in a document. These three techniques are best compared to a bag of words scheme [8].

Due to the rise of social media, offensive information has become more common online [9]. A wide range of unpleasant content can be found on the internet, such as racist and sexist messages, insults, and threats directed at individuals or organizations [10]. It has become a significant issue for online communities because of the rise of online content. Detecting offensive words in content can be challenging. This system has several difficulties, such as a) the informal language used in social media posts, which is typically written in slang and short forms that are hard to comprehend and process semantically, and b) the diversity and variety of English dialects and forms, which makes it more difficult to identify offensive content. As a result, harassment or the use of foul words online has become a major concern among people of all ages. A novel effective offensive language classification (EOLC) technique is proposed to overcome these challenges. The following are the main contributions of this paper:

- The Social media tweets are collected and pre-processed to remove the irrelevant tweets.
- Following pre-processing, Word2Vec, TF-IDF, and lexicon-based feature extraction algorithms extract the features from the data.
- Based on the collected features, a graph-based deep learning technique categorizes tweets into three categories: moderate, offensive, swear, and non-offensive.
- The proposed GDL is further adjusted by red fox optimization (RFO) to improve classification performance, which normalizes the network's weight and biases.
- The efficiency of the proposed method was estimated based on parameters like F1 score, recall, precision, and accuracy.

The paper's content is arranged as follows: section 2 reviews previous research in the literature, section 3 provides a thorough analysis of the proposed work, section 4 has the results section, and section 5 contains the conclusion.

2. LITERATURE SURVEY

Many scholars have concentrated on offensive language detection in recent years, although it has some limitations regarding words, sentences, hate speech, and swear words. This section provides a quick summary of some of the most recent studies.

Mishra et al. [11] suggested the TF-IDF approach to identifying hate speech and inappropriate content in Indo-European languages. The suggested models are built on the n-gram and BoW features. One way to find offensive language word patterns is to use character n-grams. Content that could be detrimental to the welfare of society and the community must be removed as quickly as possible.

Four different neural network architectures were developed in [12] to detect incorrect language on Arabic social media. The four approaches are the Bidirectional Long

[1,2] Sri Sivasubramaniya Nadar College of Engineering, Tamil Nadu, India. Emails: kalaivania@ssn.edu.in, theni_d@ssn.edu.in

DOI: 10.59277/RRST-EE.2024.69.2.14

Short-Term Memory (Bi-LSTM), the CNN, the Bi-LSTM with attention mechanism, and the CNNLSTM architecture.

In 2020, [13] proposed identifying hate speech and text on social media with harmful content as contributed to the datasets for the experiment (HASOC). The analyses show the efficiency of the TF-IDF method. In the inter-model area, nasty and abusive messages on social networking sites that combine image and text are assessed.

In 2020, [14] offered the machine learning strategy for our initial data pre-processing experiment. They used the NB and Linear SVM algorithms to identify abusive language in tweets, employing several methods to determine the characteristics. Compared to previous methods, the proposed naive Bayes strategy achieved 90 % accuracy, as per the results.

In 2020, [15] tested four alternative models: two classical machine learning models and two deep neural network models. The results of each classifier are calculated using the F1-score index based on the same training dataset. A deep neural network outperforms traditional models in categorizing hate speech texts, and Text-CNN has the highest F1-macro score of 83.04 %. Compared to Logistic regression, the SVM model is 65.10 percent more accurate.

In 2021, [16] proposed applying deep learning and natural language processing (NLP) to detect meme toxicity. Because of the growing number of memes on the internet, this model will be implemented more effectively and in a more advanced way. So, People are actively communicating memes, as it has become the norm, as many memes are being screened across Facebook and X (Twitter).

[17] published a study examining an automated system detecting abusive language in Roman Urdu and Urdu comments on YouTube. They use individual and combination techniques based on n-grams to extract character and word properties. Using seventeen classifiers from seven machine-learning techniques, they identify abusive language in Roman Urdu and Urdu text comments.

In 2022, [18] developed deep recurrent neural networks (RNNs) for categorizing and detecting offensive language. This suggested that the RNN architecture, DRNN-2, had 10 layers, 32 batches, and 50 iterations for the classification challenge. Based on the proposed models, 99.73 % of binary comments were recognized, 95.38 % of Arabic comments in three classes, and 84.14 % for seven classes.

In [19], it was suggested that the many Romanian basic language resource kits (BLARK) that include already-trained models be evaluated. We retrained the models using the Universal Dependencies version 2.7 of the RRT corpus to ensure a fair comparison, and they evaluated the models using data from both the same domain (the RRT-Test portion of the corpus) and the cross-domain (the SiMoNERo corpus). Recent neural models perform better than older techniques, as expected.

2.1 DIFFERENCES BETWEEN THE EXISTING AND PROPOSED WORK

The key research findings and how the suggested study and previous research differ are listed below.

- Several algorithms overlooked the pre-processing stage and the four-class categorization for offensive language, in contrast to the proposed methodology.
- The proposed work is different and important from previous methods since it presents a new optimization procedure that has not been included in any other method up to this point.
- The proposed deep learning architecture with squeeze and excitation blocks integrated improves their efficiency and allows them to concentrate on more relevant data.

From the literature review, various ML and DL models were focused on classifying the comments in two class classifications without any optimization technique. The proposed effective offensive language classification (EOLC) is focused on four class classifications while improving the accuracy rate. The proposed network is optimized with Red Fox optimization to improve its classification efficiency.

3. EFFECTIVE OFFENSIVE LANGUAGE CLASSIFICATION (EOLC)

In this section, an EOLC model has been proposed for detecting offensive language. This proposed system has three stages: pre-processing, feature extraction, and classification, as depicted in Fig. 1.

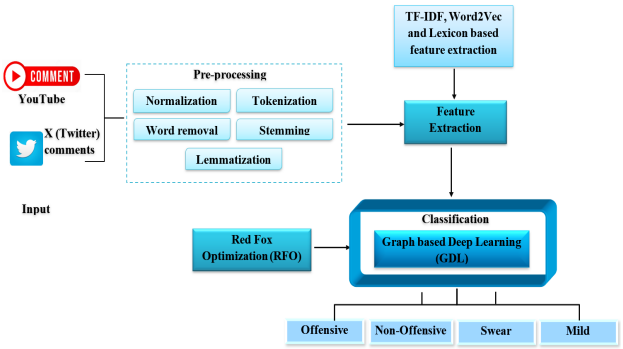


Fig. 1 – Schematic representation of effective offensive language classification (EOLC) model.

The task of EOLC is carried out in the following phases: pre-processing the datasets using tokenization, word removal, stemming, lemmatization, and normalization. Features are extracted using TF-IDF, Word2Vec, and lexicon-based features, and classification is done using GDL. The GDL is optimized with Red Fox optimization to improve the network's classification efficiency.

3.1. DATA PRE-PROCESSING

It is an essential component of many NLP exercises and uses. Currently, YouTube and X (Twitter) are the input data sources. This phase involves tokenization, word removal, stemming, lemmatization, and normalization [20].

3.1.1 NORMALIZATION

Several actions are done at once to accomplish normalization. All text will be converted to upper- or lowercase, punctuation will be removed, and numerals will be replaced with words. Consequently, each text will undergo more uniform pre-processing.

3.1.2 TOKENIZATION

Tokenization is a process that breaks down text into valuable information while preserving its meaning. This step divides long paragraphs, called text or sentences, into tokens. Moreover, these sentences are decomposable into individual words.

3.1.3 WORD REMOVAL

Repeated words are eliminated from the text throughout this phase. Many stop words are used, including "are," "of," "the," and "at." As a result, these must be taken out of the text again.

3.1.4 STEMMING

By returning words from numerous tenses to their most fundamental forms, stemming removes unnecessary computations.

3.1.5 LEMMATIZATION

Combining two or more words into one is called lemmatization (Table 1). Based on the word's morphology, this method reduces ends like shocked to shock, caught to catch, etc.

Table 1

Examples of pre-processing through stages.

Original text	I am just trying to get my car Keys, I accidently locked in my restroom.
Normalization	i am just trying to get my car keys i accidently locked in my restroom
Tokenization	'i' 'am' 'just' 'trying' 'to' 'get' 'my' 'car' 'keys' 'i' 'accidently' 'locked' 'in' 'my' 'restroom'
Word removal	just trying get car keys accidently locked restroom
Stemming	just try get car key accident locked restroom
Lemmatization	just try get car key accident lock bathroom

3.2. FEATURE EXTRACTION

In the feature extraction phase, redundant and irrelevant data are eliminated from the pre-processed data. The features are extracted using TF-IDF, Word2Vec, and lexicon-based features [21]. Combining all three feature extraction techniques will enhance the model's overall performance. A new feature set is created when the features from these approaches are compared with the starting dimension of the input.

3.2.1 TF-IDF

It is represented by the row in this technique, which is used to extract characters, and the words are represented in the column. The data is computed using:

$$TF - IDF = tf(t,d) * \log\left(\frac{S}{DF+1}\right) . \quad (1)$$

3.2.2 LEXICON-BASED FEATURE EXTRACTION METHOD

Four separate features are taken out of the tweets using this method. Connotation counts come in four flavors: positive word count (PC), negative connotation count (NCC), positive connotation count (PCC), and negative connotation count (NC). A positive and negative word dictionary finds positive and negative words in each review. Word connotations are meanings that are unclear in some contexts. In the positive lexicon, the word "avoid" has a good connotation, while it does not have a negative connotation in the negative lexicon. PCCs and NCCs have connotation lexicons that are both positive and negative in addition to their usual positive and negative terminology.

3.2.3 WORD2VEC

The Word2vec neural network uses two layers to create feature vectors from a text corpus. It is an unsupervised model based on word embedding that identifies meaning and semantic relationships between words by analyzing the co-occurrence of terms in a specific corpus of documents. The primary purpose of Word2Vec is to capture the context of words using machine learning techniques such as recurrent neural networks. This is achieved using skip-grams (SGs) or continuous bag-of-words (CBOWs).

3.3. CLASSIFICATION USING GDL

The proposed GDL model identifies offensive language on both YouTube and X (Twitter) datasets. The GDL can be considered an adaptation of a typical CNN for encoding local information about unstructured data. The structure of the proposed GDL model is illustrated in Fig. 2. The proposed model includes two phases: classification and optimization.

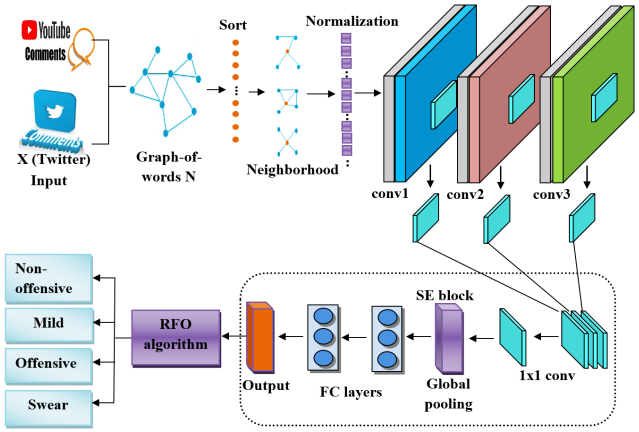


Fig. 2 – Graph-based deep learning architecture.

3.3.1 GRAPH-OF-WORDS

Graph-of-word is an alternative approach that capitalizes on a graph representation of comments. For textual comments, vertices represent singular words, and edges represent co-occurrences between words. This graph comprises vertices and edges, where V represents the vertices and E represents the edges. Graphs of co-occurrences are undirected and result from the association between words within a fixed-size sliding window. In the graph, each node is linked to a specific master, which obscures its structure during message transit. The following are two methods for creating normalized sub-graphs from a word graph.

Generating sub-graphs

Each node in a graph is ordered according to its degrees (number of neighbors). Nodes are arranged according to their frequency in the document if the degrees are equal. If the occurrences are similar in more ways than one, we sort them by their connection with neighbors, specifically the quantity of co-occurrence with the neighboring neighborhoods. This action will result in N subgraphs with g or more nodes each.

Normalization

When a convolution mask is used to convolve a subgraph, the order of the nodes is to be convoluted. Based on the labeling of nodes, a uniform convolution should be achieved across all subgraphs and texts.

3.3.2 CONVOLUTIONAL LAYER

The first convolutional layer input feature space is N*g*D, where N denotes the number of selected and normalized subgraphs, g is the receptive field of the subgraphs, and D is the dimension of word embedding. Convolution is performed using a g*D kernel on the input tensor N*g*D. This kernel combines sub-semantics graphs to provide higher-level semantic information. Convolutional layers are activated with ReLU to speed up training and prevent overfitting.

3.3.3 SQUEEZE AND EXCITATION MECHANISM

The SE block enriches the output volume of a transformation process by calibrating the extracted features. The sub-feature mappings in the SE block are reduced to 512 channels via a 1x1 convolutional layer, which is then pooled using global average pooling to produce a 512-D vector. After that, the vectors are encrypted and decrypted by two fully linked layers. The excitation vectors' scores are organized, and only the highest K values are kept. The excitation vectors are used to recalibrate the sub-feature maps to provide the output for the second classifier. Additionally, the RFO method is employed to generate the best classification outcome.

3.3.4 RED FOX OPTIMIZATION (RFO) ALGORITHM

The fitness function is crucial because in RFO [22], it converges the method to find optimal groups by analysing the effectiveness of network partitions during the optimization process. Red fox hunting behavior serves as the inspiration for RFO, a new metaheuristic optimization method. Here is an example in which random individual generation can represent RFO initialization.

$$Z = \{z_0, z_1, z_2, \dots., z_{n-1}\}. \quad (2)$$

When j stands for problem sizes in the exploring space, i is the total number of groups, and $(Z_j^i)^t$ characterizes the z_i in iteration *t*. Given that f is the function in the R_n condition and that n is an attribute in the interval [x, y] ^{n},

$$(Z)^i = \{(z_0)^i, (z_1)^i, (z_2)^i, \dots, (z_{n-1})^i\}. \quad (3)$$

where x, y ∈ R. Therefore, $f((Z)^i)$ suggests the global ideal outcomes while the optimal solution is reached. Each individual is supposed to help the exploration team in a particular way. To increase their chances of catching prey, animals who do not find enough prey in one location will relocate to another. If a more suitable area is found, the location is shared with the other members. The red fox looks at its prey and then moves closer to it. The RFO procedure, which represents a random value rv in the interval [0,1], is applied in this instance:

$$\begin{cases} move\ closer\ \ if\ rv > \frac{3}{4} \\ stay\ and\ hide\ \ if\ rv \le \frac{3}{4} \end{cases}. \quad (4)$$

The member's moment is then determined using an improved cochleoid formula. Classification is a critical component of all medical imaging. CNN uses the backpropagation process, as was previously explained, to facilitate learning. The RFO technique for the best system selection was established by this study by minimizing the mean square error. The MSE has the following numerical expression:

$$mse = \frac{1}{T}\sum_{j=1}^{q}\sum_{i=1}^{p}(x_j^i - y_j^i)^2. \quad (5)$$

where x_j^i and y_j^i indicate the attainable and appropriate magnitudes for the j-th unit in the network's output layer in time T, respectively, and $p\ and\ q$ denote the values of the output layers and the facts, respectively.

4. RESULT AND DISCUSSION

This section details various experiments to investigate the effectiveness of the Python programming language-based EOLC approach for identifying aggression in social media tweets. The proposed method performs experiments in Anaconda using an Intel Core i7 processor running at 3.40 GHz and 8 GB of RAM.

4.1 DATA DESCRIPTION

4.1.1 YOUTUBE

We collected text comments about 18 top videos from YouTube comment boards. The videos were categorized into 13 categories: music, cars, comedies, education, film, gaming, fashion, news, non-profits, animals, sciences, and sports. The dataset includes 2,175,474 comments from different users. 1815462 comments were used for training, and 360012 comments were used for testing. Table 2 shows an example of YouTube comments and their classification.

Table 2
Example of YouTube comments and their classification

Comments	Classification
Dang, never knew my neighbor's sofa feels that comfortable!	Mild
Why has YouTube not killing your channel	Offensive
Dude, I don't even like her so chill out. No wonder you attempt are falling, you are a creep.	Offensive
Why do guys think women owe them for being nice	Offensive
Ur 80 lbs over weigh not healthy "papa"	Non-offensive
This music is very irritating	Offensive
white bitches in san junipero 24/7	Offensive
0:57 that laughing Monkey was probably thinking: "Do they really think I am that stupid?"	Swear
He is so funny	Non offensive
Thank you so much I shall use this	Mild
Why other peoples don't like this, they have no reason to dislike	Non-offensive
You are looking WILD, Mahyer!!	Swear
Be awesome! Love this little kid!	Mild

4.1.2 X (TWITTER)

This research uses OLID, the official dataset for OffensEval 2019, to classify offensive language. OLID is a hierarchical dataset used to find offensive texts on social media. We collected X (Twitter) data and made it publicly accessible. Of 14,100 tweets, 13,240 are utilized for training, while 860 are used for testing. Table 3 shows a few instances of text classification using the suggested approach.

Table 3
Example of tweets and their classification

Tweets	Classification
Will you come tomorrow	Non offensive
Learning English becomes fun and easy when you learn with movie traders.	Non-offensive
Slap on your face.	Offensive
You are stupid, Getlost, ashamed me, you are foolish.	Offensive
Kick you	Offensive
I don't like tea	Non offensive
Don't underestimate me	Swear
Beat you	Offensive
I don't bother about anything or any one	Swear
It's ok!	Mild

4.2 PERFORMANCE METRICS

Accuracy, precision, F1-score, and recall scores were used to analyze the experiment's outcomes. A statistical analysis of the parameters is presented below. Performance analysis of

proposed techniques is shown in Table 4.

$$\text{Accuracy} = \frac{TP+TN}{total\ no.of\ samples}. \quad (6)$$

$$recall = \frac{TP}{TP+FN}. \quad (7)$$

$$f1\ score = 2\left(\frac{precision*recall}{precision+recall}\right). \quad (8)$$

$$precision = \frac{TP}{TP+FP}. \quad (9)$$

Table 4

Performance analysis of proposed techniques.

Features Measures	Offensive language	Non-offensive language	Mild language	Swear language
Accuracy	98.6	97.5	92.5	95.8
Recall	95.2	96.5	93.4	93.5
Precision	96.5	95.8	91.5	94.2
F1 score	97.3	94.6	92.7	93.8

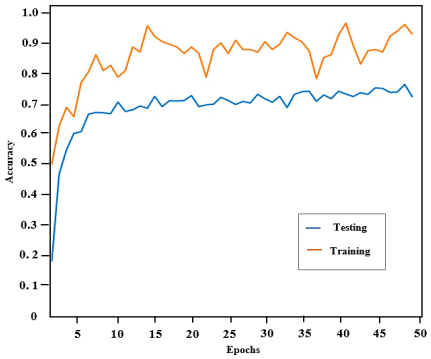
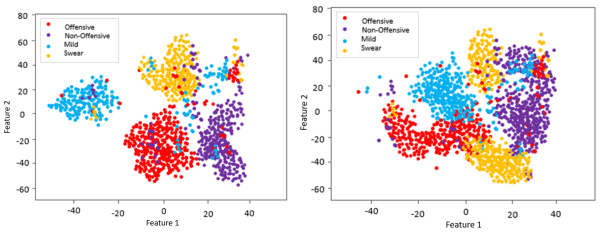


Fig. 3 – Training and testing accuracy of the proposed method.



(a) (b)

Fig. 4 – Visualization of GDL classification for (a) YouTube comments (b) X (Twitter) tweets.

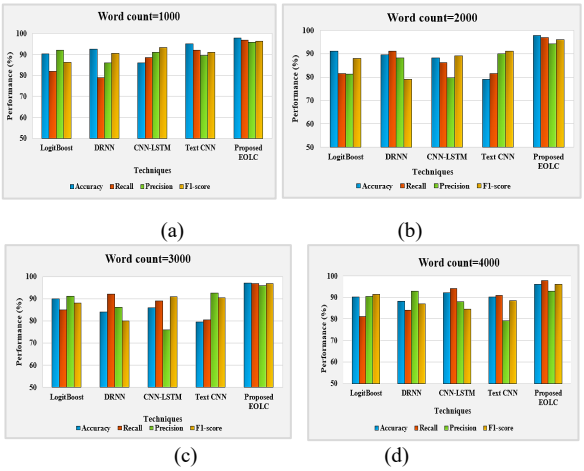
Figure 3 illustrates the excellent accuracy the suggested model attained in training and testing. Performance is measured by F1-score, specificity, recall, accuracy, and precision. The proposed approach achieves an accuracy of 98.58 %.

The tweets and comments from YouTube and Twitter are displayed in Fig. 4(a) for YouTube comments and Fig. 4(b) for X (Twitter) tweets. The classification results of the suggested GDL method are displayed in these figures. The tweets and comments are classified as mild, offensive, non-offensive, and swearing. Languages deemed objectionable are prohibited based on the classification results.

4.3 COMPARATIVE ANALYSIS

To demonstrate that the proposed approach is more effective, its performance was compared to that of the existing strategies. Fig. 4 shows a comparison of the proposed EOLC model to other existing techniques, such as LogitBoost [17], DRNN [18], CNN-LSTM [12], and Text CNN [15]. According to the figures, offensive language detection was evaluated using multiple approaches at varying word counts.

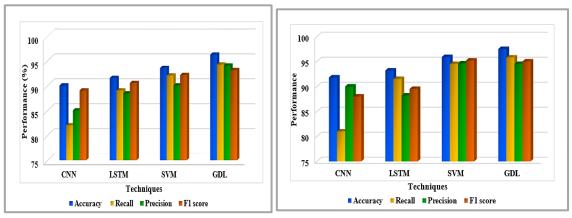
Comparing the observations from Fig. (5a) to the current methods, the suggested EOLC model has a maximum accuracy of 97.2%. Figures (5b, 5c, and 5d) demonstrated that accuracy decreases as word count rises. However, when the word count reaches 1000, the proposed technique outperforms the traditional approach with the highest accuracy of 97.2 %. In addition, results were compared with traditional models based on YouTube and X (Twitter) comments.



(a) (b)

(c) (d)

Fig. 5 - Result of offensive detection for various word count



(a) (b)

Fig. 6 – Comparison of GDL with traditional classifiers based on (a) YouTube dataset, (b) X (Twitter) dataset.

Furthermore, Fig. 6 shows the proposed and existing models' accuracy, F1-score, precision, and recall. Based on the results, our proposed deep learning strategy overrides the existing methods. The proposed GDL model achieves the highest levels of classification accuracy on the X (Twitter) and YouTube datasets, with 95.5 % and 96.8 %, respectively.

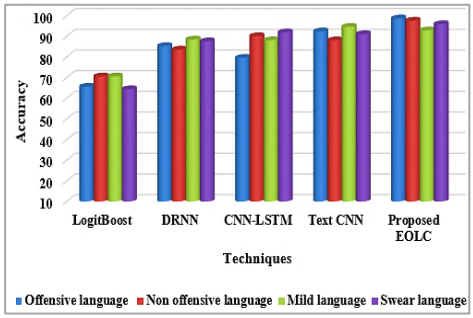


Fig. 7 – Category of word detection using the proposed method.

Figure 7 shows how offensive language is detected using numerous algorithms for various category words, sentences, and hate speech types. Compared to existing approaches such as LogitBoost, DRNN, CNN-LSTM, and Text CNN the

proposed EOLC method recognized the category offensive language with 98.6 % and non-offensive language with 97.5 %, respectively.

5. CONCLUSIONS

This work proposes a novel EOLC technique. English language tweets and comments from YouTube and X (Twitter) with both offensive, mild, swear, and non-offensive tweets are utilized in this paper. Initially, the tweets and comments are pre-processed, and the features are extracted using different techniques, namely WordVec, TF-IDF, and Lexicon-based features. The extracted features are classified using the Graph-based deep learning (GDL) method for numerical representation and decision-making. In the GDL, texts are first converted into graphs of words, and then the word graphs are convolved using graph convolution procedures. It is possible to capture non-consecutive and distant semantics when representing texts as graphs of words. GDL network is optimized with RFO to normalize the weight and biases of the network and achieve better classification results. The proposed GDL model achieves the highest levels of classification accuracy on the Twitter and YouTube datasets, with 95.5 % and 96.8 %, respectively. Based on the experimental data, the proposed technique is more accurate by 98.05 % than other methods. In the future, the advanced deep learning method will detect the type of individual or a specific group to whom the offensive comments generated by users are directed in social networks, and the proposed model can be elaborated on in other languages.

Received on 20 February 2023

REFERENCES

1. A. H. Razavi, D. Inkpen, S. Uritsky, S. Matwin, *Offensive language detection using multi-level classification*, Canadian Conference on Artificial Intelligence Springer, Berlin, Heidelberg, pp. 16–27 (2010).
2. F.Z. El-Alami, S.O. El Alaoui, N.E. Nahnahi, *A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model,* Journal of King Saud University-Computer and Information Sciences, pp. 1–9 (2021).
3. S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, J. Gao, *Deep learning–based text classification: a Comprehensive Review*, ACM Computing Surveys (CSUR) **54**, 3, pp. 1–40 (2021).
4. R. Jenke, A. Peer, M. Buss, *Feature extraction and selection for emotion recognition from EEG*, IEEE Transactions on Affective Computing **5***, 3*, pp. 327–339 (2014).
5. O. Sharif, M.M. Hoque, A.S.M. Kayes, R. Nowrozy, I.H. Sarker, *Detecting suspicious texts using machine learning techniques*, Appl. Sci, **10,** *18*, pp. 6527 (2020).
6. H. Gupta, P. Kumar, S. Saurabh, S.K. Mishra, B. Appasani, A. Pati, C. Ravariu A. Srinivasulu, Category boosting machine learning algorithm for breast cancer prediction, Rev. Roum. Sci. Tech. – Électrotechn. Et Énerg., **66**, *3*, pp. 201–206 (2021).
7. H. Razavi, D. Inkpen, S. Uritsky, S. Matwin, *Offensive language detection using multi-level classification*, Canadian Conference on Artificial Intelligence, Springer, Berlin, Heidelberg, pp. 16–27 (2010).
8. W. Zhang, T. Yoshida, X. Tang, *A comparative study of TF* IDF, LSI and multi-words for text classification*, Expert Syst. Appl. **38,** 3, pp. 2758–2765 (2011).
9. S. Abro, Z.S. Shaikh, S. Khan, G. Mujtaba, Z.H. Khand, *Automatic hate speech detection using machine learning: a comparative study*, Mach. Learn., **10***, 6* (2020).
10. W. Zhang, T. Yoshida, X. Tang, *TFIDF, LSI and multi-word in information retrieval and text categorization*, IEEE International Conference on Systems, Man and Cybernetics, Singapore, pp. 108–113 (2008).
11. P. Mishra, V. Varadharajan, U. Tupakula, E.S. Pilli, *A detailed investigation and analysis of using machine learning techniques for intrusion detection*, IEEE Commun. Surv. Tutorials, **21***, 1*, pp. 686–728 (2018).
12. H. Mohaouchane, A. Mourhir, N.S. Nikolov, *Detecting offensive language on arabic social media using deep learning*, Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), Granada, Spain, pp. 466–471 (2019).
13. N.D. Srivastava, Y. Sharma, *Combating online hate: a comparative study on identification of hate speech and offensive content in social media text*, IEEE Recent Advances in Intelligent Computational Systems (RAICS), Thiruvananthapuram, India, pp. 47–52 (2020).
14. G.A. De Souza, M. Da Costa-Abreu*, Automatic offensive language detection from Twitter data using machine learning and feature selection of metadata,* International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, pp. 1–6 (2020).
15. S.T. Luu, H.P. Nguyen, K. Van Nguyen, N.L.T. Nguyen, *Comparison between traditional machine learning models and neural network models for Vietnamese hate speech detection*, International Conference on Computing and Communication Technologies (RIVF), pp. 1–6 (2020).
16. R.K. Giri, S.C. Gupta, U.K. Gupta, *An approach to detect offense in memes using natural language processing (NLP) and deep learning*, International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, pp. 1–5 (2021).
17. M.P. Akhter, Z. Jiangbin, I.R. Naqvi, M. Abdelmajeed, M.T. Sadiq, *Automatic detection of offensive language for Urdu and Roman Urdu*, IEEE Access **8**, pp. 91213-91226 (2020).
18. F.Y.A. Anezi, *Arabic hate speech detection using deep recurrent neural networks*, Appl. Sci, **12,** *12*, pp. 6010 (2022).
19. V. Pais, R. Ion, A.M. Avram, M. Mitrofan, D. Tufis*, In-depth evaluation of Romanian natural language processing pipelines*, Romanian Journal of Information Science and Technology, **24**, *4*, pp. 384–401 (2021).
20. R. Alqaisi, W. Ghanem, A. Qaroush, *Extractive multi-document Arabic text summarization using evolutionary multi-objective optimization with K-medoid clustering*, IEEE Access, **8**, pp. 228206–228224 (2020).
21. R. Ahuja, A. Chug, S. Kohli, S. Gupta, P. Ahuja, *The impact of features extraction on the sentiment analysis*, Procedia Computer Science, **152**, pp. 341–348 (2019).
22. D. Połap, and M. Woźniak, *Red fox optimization algorithm*, Expert Systems with Applications, **166**, pp. 114107 (2021).