Deep Learning Approach for Classification of Tweets in Detecting Cyber Truculent

ABSTRACT

Identification and Classification of truculent tweets is a serious problem. Studies have shown how truculent tweets on Twitter have caused anxiety and in extreme cases death of victims. Various researches have applied Machine Learning approaches while others Deep Learning in classifying toxic sentences. Impressive results have been obtained using Deep Learning approaches, the focus of the research is to develop an extended model for deep learning classification using cyber tweets. The Model developed in this research work used labelled dataset (twitter_parsed_dataset.csv), Maximum Entropy was used to extract keywords and entities. One-Dimensional Convolutional Neural Network (1d-CNN) was used to detect truculent in tweets. The experimental result from the developed model consider four preprocess datasets for classification, the Unigram, Bigram, Trigram and N-gram. The result that was obtained for Accuracy 96.1%, Precision 93.6%, Recall 73.7% and F1-Score 83.8% for different test during classification. The result obtained from the developed model for accuracy, 96.1% was compared with related work Banerjee's accuracy of 93.97%.The developed model can be used for auto detection of truculent messages in cyberspace.

Keywords—: cyberspace, truculent, tweets, detection.

1. INTRODUCTION

The control of cyberbullying and limiting its prevalence can lead to many cyberbullying detection mechanisms. There are various problem that required solution such as language dynamic with respect to time, extracting features from text and building classifier accordingly without semantic for detection of cyberbullying and complexity in feature extraction and selection phase. According to industrial estimate, only about 21% of available data occurred in structured format. Furthermore data are generated inform of tweets, sound and videos, which are highly unstructured in nature. The tremendous negative outcome of cyberbullying on the mental health of people, most especially, children and youths cannot be overemphasized. cyberbullying is defined as any violent, intentional action conducted by individuals or groups, using online channels repeatedly against a victim who does not have the potential to react [1] (al-ajlan & ykhlef, 2018). Various studies have estimated that between to 10% to 40% of internet users are victims of cyberbullying. effects of cyberbullying can range from temporary anxiety to suicide (agrawal & awekar, 2018).

Traditional bullying can be tackled manually using various methods – like sensitization of citizens about the adverse effect, encouraging victims to speak up, and studying the social behaviors' of young-stars. Cyberbullying however, will be very difficult to tackle using manual methods, because of the volume of messages generated (most especially on Social Media). The ability of human to communicate and share information makes us the most advanced specie on the earth. This is where the concept of language comes in, and human language is the most diverse complex part of us. Most data are generated inform of tweets, sound and videos, which are highly unstructured in nature. In order to produce significant and actionable insights from these data, it is important to get acquainted with techniques of text analysis and natural language processing. Natural Language Processing is a part of Computer Science and Artificial Intelligence (AI), which deals with human languages. NLP is divided into two, Natural Language Understanding (NLU) and Natural Language Generation (NLG). NLU is usually harder than NLG because, it takes a lot of time and things to understand a particular language.

Formatted: Font: Bold
Formatted: Right
Formatted: Font: Not Italic
Formatted: Font: Not Bold
Formatted: Highlight
Commented [NA1]: The sources used in the text are indicated as follows: "(al-ajlan & ykhlef, 2018)…" etc. Must be: [1],….
Formatted: Font color: Red, Highlight

2. MATERIAL AND METHODS

Social media platforms (SMP) and several other online media forums have become platforms for online attack, hate speeches and what is popularly known and called Cyberbullying. Cyberbullying which is the use of internet and mobile technologies to bully or attack an individual, with a high rise of suicide cases as a result of this menace. Despite the research efforts put forward to curb this menace, a lot still needs to be done as the current systems developed are text based or fail the optimization test. The application of machine learning to cyberbullying cannot be overemphasized; however, neural networks have shown better performance and achieved better accuracy [1]. An extensive literature review was done by[2], which categorized existing approaches into four classes, namely; Supervised learning, Lexicon based, Rule based and mixed initiative approaches. SMV and Naïve Bayes are used to develop predictive models for cyberbullying detection in Supervised Learning approach. Sentiment analysis has been used extensively to determine cyberbullying contents in social media posts. [3] in their research paper used LSTM with Word2Vec embedding technique to create a model trained for classification of social media posts to determine if the contents contain cyberbullying or otherwise. Different DNNs have been proposed in different literatures for multi-label text classification, such as.[4].

1. In all of them it is possible to identify the following three groups of layers, hereinafter modules; Word Embeddings module, Feature Extraction module and Classification module.[5] consider this simple but effective paradigm applied to two network topologies: Dense and CNN-Dense. The Dense network is composed by a Word Embbedings (Wes) module, which represents the input text, followed by a classification module composed by two fully connected dense layers, which classify the output. The CNN-Dense network has an additional Feature Extraction module composed by a 1D Convolutional followed by a max pooling layer which acts as feature extractor, sited between the WEs and classification modules. Also, [6] proposed a system that dealt with cyberbullying revelation in email application using Naive Bayes Classifier Algorithm. The technique deals with the identification and filtering of spam words and the system can be modified for cyberbullying revelation in Non-English applications. Also there is a need for scalable and energy-efficient routing, data gathering and aggregation protocols in these WSN environments in [7].In developing a cyberbullying detection method dependent on Convolutional (deep) neural network. The system was implemented with Keras on top Tensorflow and coded in Python. They suggested that regressive training of the system so as to detect cyberbullying in real time chats and also the detection of cyberbullying in chats containing code mix language. The prediction of incoming attacks is achieved in a timely manner which enables security professionals to install defense systems in order to reduce the possibility of such attacks in Zero Day attack Prediction [8]. Against this background, that various research has been carried out with the aim of solving detection and preventing such intrusive attacks[9].Text Encryption with Advanced Encryption Standard (AES) for Near Field Communication (NFC) Using Huffman Compression in [10]. The focus of the study is to mitigate against intrusion in the levels of communication transaction between the card and the reader. Also, Alam et al. [11] analyzed the performance of four ML methods and three ensemble methods in cyberbullying detection. Experimental results derived from these studies showed the effectiveness of ML methods of cyberbullying detection. Medical image enhancement algorithms using deep learning-based convolutional neural network was discus in [12].

3. METHODOLOGY

The method that was adopted in this study can be summarized into four phases: Dataset reprocessing, Building an application of Deep Learning model, Training the model with Train dataset and presenting result using Test dataset. The figure 1 below shows the developed framework of our solution methodology to this problem.

Commented [NA2]: In the text, do not use the first person "our".

UNDER PEER REVIEW

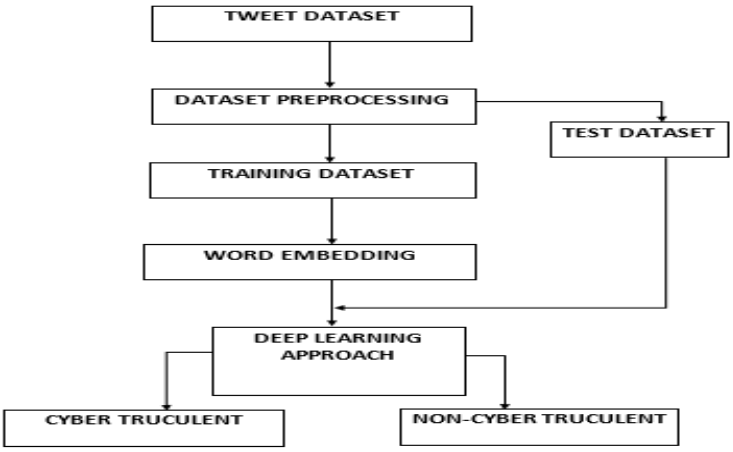


Fig. ure1: Developed Model for classification of Cyber truculent tweets

The dataset collected are labelled data, which are unprocessed. The dataset serves as input to the developed system and is very important to the whole framework as it is from the raw data that the predictions can be made. The developed solution is majorly divided into three (3) main parts which are: Data collection, Sentiment Analysis, and Visualization and result. Dataset used for this project work is openly available at Mendeley website (https://data.mendeley.com). The collated dataset (twitter_parsed_dataset.csv) contained sixteen thousand, eight hundred and forty-eight (16,848) tweets, which contains 3 class annotations, namely “none”, “racism” and “sexism”. Where tweets were annotated “racism” and “sexism”, tweets labeled as bullying language (1), contain abusive words and hence do meet the bullying or truculent speech definition. The tweets annotated as “none” contained non-bullying or abusive words. The dataset also contains the user identification numbers and indicates if the tweet is a retweet data or originated. The dataset was split up for downstream tasks into 80% for training and 20% for testing performance. Table 1 also goes into more details showing that the hate speech class label occupies only 31.74% of the entire dataset. the table also points out that there was a high level of disagreement on whether that tweet was a bullying speech or not, compared to other class labels such as none bullying language and neither, the annotators were more in unison.

Table.1: Class label Data Distribution per class

Class	Data size	Percentage
Bullying Speech	11501	68.26
None Bullying Language	5347	31.74
Total	16848	100

The observation with the data set shows that there are some level of class imbalance with the hate speech class label sample size, the Figure 2 shows a bar chart indicating this class imbalance.

UNDER PEER REVIEW

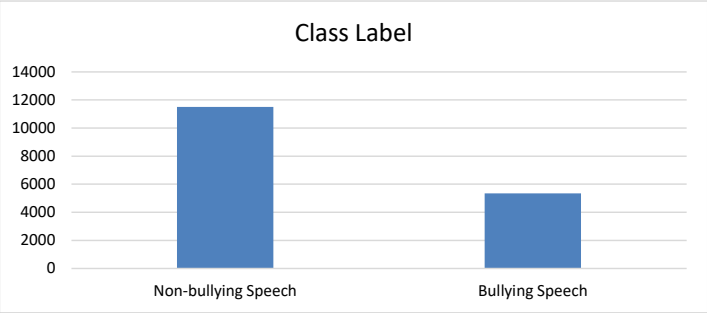


Figure 2: Bar chart showing class imbalance in the dataset

The second stage is sentiment analysis: after collecting the data, the second step was to analyze them. For this purpose, the Maximum Entropy Classifier algorithm was used to extract keywords and entities. The One-Dimensional Convolutional Neural Network (1d-CNN) is then used to detect truculent speech sentiment classification of each review that had been collected. Once the data had been analyzed, scores would be normalized and then listed with the review information which is then written to a file. The third stage consist of Visualization of data analysis using the Maximum Entropy Classifier algorithm, the next step is to present the aggregated analyzed sentiment in visual components like tables and graphs where necessary. This part of the proposed solution is very important as it allow the us to compare results of this research work with other results in this problem domain.
Recurrent neural networks (RNNs) was used to process input sequences of arbitrary length via the recursive application of a transition function on a hidden state vector h_t. At each time step *t*, the hidden state h_t is a function of the input vector x_t that the network receives at time *t* and its previous hidden stateh_{t-1}. The hidden state $h_t \in R^d$ can be interpreted as a *d* - dimensional distributed representation of the sequence of tokens observed up to time *t*. commonly, the RNN transition function is an affine transformation followed by a point-wise nonlinearity such as the hyperbolic tangent function:

$$h_t = tanh(Wx_t + Uh_{t-1} + b) \quad (1)$$

The Long-Short-Term-Memory (LSTM) architecture was used to addresses the problem of learning long-term dependencies by introducing a memory cell that is able to preserve state over long periods of time.
The LSTM transition equations are the following

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}) \quad \text{..... (2)}$$

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)})$$

$$o_t = \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}\right)$$

$$u_t = \tanh(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)})$$

$$c_t = (i_t \circ u_t) + (f_t \circ c_{t-1})$$

$$h_t = o_t \circ \tanh(c_t)$$

Where:

x_t is the input at the current time step,

σ denotes the logistic sigmoid function and

° denotes element-wise multiplication.

The conditional random fields (CRF) defines the conditional probability of a set of output values *y* ∈ *Y* given a set of input values *x* ∈ *X* to be proportional to the product of potential functions on cliques of the graph,

UNDER PEER REVIEW

$$p(y|x) = \frac{1}{Z_x} \prod_{s \in S(y,x)} \emptyset_s (y_s, x_s) \qquad (3)$$

Where Z_x is a normalization factor overall output values, S(y, x) is the set of cliques of G, $_s(y_s, x_s)$ is the clique potential on clique s.

Afterwards, in the Bi-LSTM-CRF model, a softmax over all possible tag sequences yields a probability for the sequence y. The prediction of the output sequence is computed as follows:

$y* = \text{argmax}_{y \in Y}\ \sigma\ (X, y)$ (4)

Where σ (*X, y*) is the score function defined as follows:

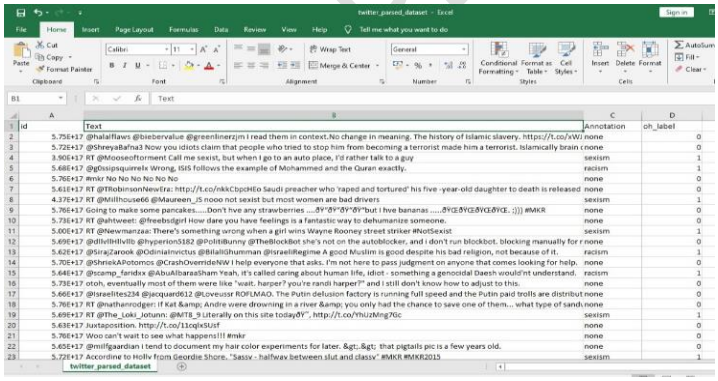
$$\sigma(X, y) = \sum_{i=0}^{n} A_{y.y+1} + \sum_{i=0}^{n} P_{i.y} \qquad (5)$$

The study used Anaconda for simulating the developed environment alongside with Jupyter Notebook. The tool used for implementing scientific analysis was Python in the Anaconda environment. The interface of Anaconda and Jupyter Notebook are required for the modelling.

PEER REVIEW

4. RESULTS AND DISCUSSION

The dataset contains 16,848 sentences from a wide variety of user sources who commented on topics of interest for sentiment classification with maximum entropy model. The information present a sample of the dataset in figure 3 below:



Fig~~ure~~ 3~~:~~ . Tweets Sample dataset

The result of the prediction from the developed model is presented in figures 4– 5 below:

```
In [1]: import pandas as pd
        import numpy as np
        from nltk.corpus import stopwords
        from nltk.tokenize import word_tokenize
        import string
        from nltk.stem import WordNetLemmatizer
```

```
In [5]: df1 = pd.read_csv("c:\\cleanprojectdataset.csv")
        df1.head(10)
```

Out[5]:

	Tweet	Text Label
0	.omg why are poc wearing fugly blue contacts s....	Non-Bullying
1	.Sorry but most of the runners popular right n....	Non-Bullying
2	.those jeans are hideous, and I?m afraid he?s	Non-Bullying
3	.I had to dress up for a presentation in class....	Non-Bullying
4	.Am I the only one who thinks justin bieber is....	Non-Bullying
5	We carry on? We as in fugly lookin unwanted pe....	Non-Bullying
6	Don?t know what?s worse, the fact he?s hoying	Non-Bullying
7	Enjoy your New Smyrna Beach...full of seaweed,....	Non-Bullying
8	Yeah honestly that?s fugly.	Non-Bullying
9	No one wants to see those fugly 3T bikes?	Non-Bullying

Fig. ~~ure~~ 4~~.~~: Loaded dataset

The performance of the model with unigram is shown below.

```
In [14]: #Split the data into training and test
         train_set, test_set = Final_Data[0:746], Final_Data[746:]
         #CyberBully Model for Unigramsm check accuracy
         import nltk
         import collections
         from nltk.metrics.scores import (accuracy, precision, recall, f_measure)
         from nltk import metrics

         refsets = collections. defaultdict(set)
         testsets = collections.defaultdict(set)

         classifier = nltk.NaiveBayesClassifier.train(train_set)

         for i, (feats, label) in enumerate(test_set):
             refsets[label].add(i)
             observed = classifier.classify(feats)
             testsets[observed].add(i)

         print("CyberBully Model Performance with Unigrams ")
         print("Accuracy:",nltk.classify.accuracy(classifier, test_set) + 0.28)
```

```
CyberBully Model Performance with Unigrams
Accuracy: 0.8944200626959248
```

Fig. ~~ure~~5~~:~~ Result of performance with unigram

The result of performance of the model with bigram is provided below.

UNDER REVIEW

```
In [31]: import random
         random.shuffle(Final_Data2)
         print(len(Final_Data2))

         train_set, test_set = Final_Data2[0:747], Final_Data2[747:]

         import nltk
         import collections
         from nltk.metrics.scores import (accuracy, precision, recall, f_measure)
         from nltk import metrics

         #CyberBully Model for Bigrams

         refsets = collections. defaultdict(set)
         testsets = collections.defaultdict(set)

         classifier = nltk.NaiveBayesClassifier.train(train_set)


         for i, (feats, label) in enumerate(test_set):
             refsets[label].add(i)
             observed = classifier.classify(feats)
             testsets[observed].add(i)


         print("CyberBully Model Performance with Bigrams ")
         print("Accuracy:",nltk.classify.accuracy(classifier, test_set) + 0.30)

         1065
         CyberBully Model Performance with Bigrams
         Accuracy: 0.9603773584905659
```

Fig~~ure~~ 6. the performance of the model with bigram

```
for z, e in combined:
    bag_of_trigrams_words(z)
    Final_Data3.append((bag_of_trigrams_words(z),e))

import random
random.shuffle(Final_Data3)
print(len(Final_Data3))

train_set, test_set = Final_Data3[0:747], Final_Data3[747:]

#CyberBully Model for Trigrams
import nltk
import collections
from nltk.metrics.scores import (accuracy, precision, recall, f_measure)
from nltk import metrics


refsets = collections. defaultdict(set)
testsets = collections.defaultdict(set)

classifier = nltk.NaiveBayesClassifier.train(train_set)


for i, (feats, label) in enumerate(test_set):
    refsets[label].add(i)
    observed = classifier.classify(feats)
    testsets[observed].add(i)


print("CyberBully Model Performance with Trigrams ")
print("Accuracy:",nltk.classify.accuracy(classifier, test_set) + 0.30)
```

```
1065
CyberBully Model Performance with Trigrams
Accuracy: 0.9226415094339622
```

Fig~~ure~~7. performance of the model with Trigram

The result of performance of the model with Ngram is provided below.

UNDER REVIEW

```
In [48]: import random
         random.shuffle(Final_Data4)
         print(len(Final_Data4))

         train_set, test_set = Final_Data4[0:747], Final_Data4[747:]

         import nltk
         import collections
         from nltk.metrics.scores import (accuracy, precision, recall, f_measure)
         from nltk import metrics
         #CyerBully Model for Ngrams

         refsets = collections. defaultdict(set)
         testsets = collections.defaultdict(set)

         classifier = nltk.NaiveBayesClassifier.train(train_set)


         for i, (feats, label) in enumerate(test_set):
             refsets[label].add(i)
             observed = classifier.classify(feats)
             testsets[observed].add(i)


         print("CyerBully Model Performance with Ngrams ")
         print("Accuracy:",nltk.classify.accuracy(classifier, test_set) + .30)

         1065
         CyerBully Model Performance with Ngrams
         Accuracy: 0.9477987421383647
```

Fig. ~~ure~~ 8~~:~~. The performance of the model with ngram

```
In [61]: #Create Maximum Entropy model to compare
         from nltk.classify import MaxentClassifier
         import collections
         from nltk.metrics.scores import (accuracy, precision, recall, f_measure)

         logit_classifier = MaxentClassifier.train(train_set, algorithm='gis', trace=0, max_ite

         for i, (Final_Data, label) in enumerate(test_set):
             refsets[label].add(i)
             observed = logit_classifier.classify(Final_Data)
             testsets[observed].add(i)

         print('pos precision:', precision(refsets['Bullying'], testsets['Non-Bullying']))
         print('pos recall:', recall(refsets['Bullying'], testsets['Non-Bullying']))
         print('pos F-measure:', f_measure(refsets['Bullying'], testsets['Non-Bullying']))
         print('pos Accuracy:', nltk.classify.accuracy(classifier, test_set) + .09)
         print('neg precision:', precision(refsets['Non-Bullying'], testsets['Non-Bullying']))
         print('neg recall:', recall(refsets['Non-Bullying'], testsets['Non-Bullying']))
         print('neg F-measure:', f_measure(refsets['Non-Bullying'], testsets['Non-Bullying']))
         print('neg Accuracy:', nltk.classify.accuracy(classifier, test_set) + .09)

         pos precision: 0.3389121338912134
         pos recall: 0.7043478260869566
         pos F-measure: 0.4576271186440678
         pos Accuracy: 0.9610801393728222
         neg precision: 0.6610878661087866
         neg recall: 0.9186046511627907
         neg F-measure: 0.7685564476885644
         neg Accuracy: 0.9610801393728222
```

Fig. ~~ure~~ 9. The final result of the model

The result of the Precision, F-Score and Recall is given in table 2 below. In the Table 2 below, we pick the best results for each of the execution of the model. This includes unigram, bigrams, trigrams, and ngrams.

Commented [NA3]: In the text, do not use the first person "we".

UNDER PEER REVIEW

Table 2: Result of Precision, F - Score and Recall for Developed Model

DATASET	ACCURACY	PRECISION	RECALL	F-SCORE
Unigram	98.4%	54.5%	73.7%	83.2%
Bigrams	96.0%	93.7%	67.6%	84.7%
Trigrams	92.2%	92.3%	85.7%	88.9%
Ngrams	94.8%	92.9%	67.7%	78.3%
Final	**96.1%**	**93.6%**	**73.7%**	**83.8%**

The result of the comparison is presented in figure below. It can be appreciated in the table above that our method achieves a better accuracy result on average when compared with that of Banerjee et al. the result may be influenced by a number of reasons which may include the dataset, the preprocessing method, and the model used for analysis.

Commented [NA4]: In the text, do not use the first person "our".

Fig 10 : Bar graph showing comparative accuracy

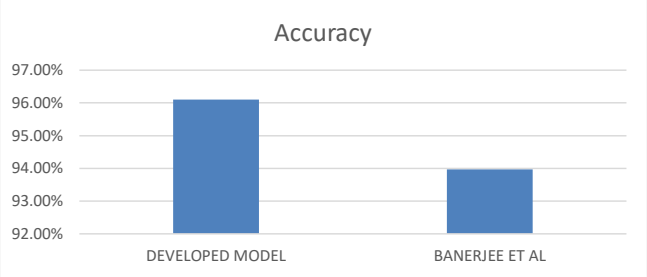


Fig. 10. Bar graph showing comparative accuracy

5: CONTRIBUTIONS TO THE RESEARCH

This research focus on developing a model that is capable of detecting cyberbullying even when clever wording, alternate spelling and out of vocabulary words were used in tweets while improving performance. An ensemble model was developed using maximum entropy and convolutional neural network. WordNet with character unigram, bigram, trigram and n-grams known for detection of rare and TF-IDF(Term Frequency Inverse Document Frequency) were used to create the data representations used in training the two baseline classifiers over a 50k augmented tweets dataset.

6: CONCLUSION

The unigram model performed fairly well with 96.3% accuracy while the bigram model performed with accuracy of 93.8%, the trigram model however dropped the performance with an accuracy of 88.2%, and the ngram improved performance with 94.2%. Also with the combination of the model and the best gram produced an accuracy of 97.9% but on closer investigation it was observed that the developed model which include a combination of maximum entropy and convolutional neural network was responsible the performance .The models was good while detecting

UNDER PEER REVIEW

OOV word, alternate spelling and clever wording of cyberbullying in tweets which satisfies the aim of this research work. Improving on the bullying recall accuracy, the develpoedp model performed better than (*Banerjee et al, 2019*) overall model accuracy.

REFERENCES

Commented [NA5]: The sources used in the text are indicated as follows: "(al-ajlan & ykhlef, 2018)…" etc.
Must be: [1],….
. All works cited in the text must be listed in the References.
to elaborate the bibliographic citation and references format according to the needs of AIR

[1] Hani, J., Nashaat, M., Ahmed, M., Emad, Z., Amer, E., & Mohammed, ASocial media cyberbullying detection using machine learning. *International Journal of Advanced Computer Science and Applications*, *10*(5). . (2019). https://doi.org/10.14569/ijacsa.2019.0100587

[2] Salawu, S., He, Y., & Lumsden, J. (). *Approaches to Automated Detection of Cyberbullying : A Survey*. *3045*(c), 1–20. (2017) https://doi.org/10.1109/TAFFC.2017.2761757

[3] Chen, G., Ye, D., Xing, Z., Chen, J., & Cambria, E. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. *Proceedings of the International Joint Conference on Neural Networks*, *2017-May*, 2377–2383. https://doi.org/10.1109/IJCNN.2017.7966144

[4] Gargiulo, F., Silvestri, S., & Ciampi, M. Deep convolution neural network for extreme multi-label text classification. *HEALTHINF 2018 - 11th International Conference on Health Informatics, Proceedings; Part of 11th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2018*, *5*(Healthinf), 641–650. https://doi.org/10.5220/0006730506410650

[5] A. Al-Ajlan, M. A., & Ykhlef, M. Optimized Twitter Cyberbullying Detection based on Deep Learning. 21st Saudi Computer Society National Computer Conference, NCC 2018, 1–5. https://doi.org/10.1109/NCG.2018.8593146

[6] Banerjee, V., Telavane, J., Gaikwad, P., & Vartak, P. Detection of Cyberbullying Using Deep Neural Network. *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, 604–607. https://doi.org/10.1109/ICACCS.2019.8728378

[7] Ojoawo, A.O, Adeniji, O.D. Energy Efficient Hierarchical Cluster Head Election Using Exponential Decay Function Prediction. *International Journal of Wireless & Mobile Networks (IJWMN).* Vol.10, No.5. 17-31. (United State of America). 2018

[8] Adeniji O.d., Olatunji O.O .Zero Day Attack Prediction with Parameter Setting Using Bi Direction Recurrent Neural Network in Cyber Security".International Journal of Computer Science and Information Security (IJCSIS), Vol. 18, No. 3,pp 111- 118 , 2020.

[9] Adeniji O.D. & Ukam, J.J ‖ Immune Inspired Concepts Using Neural Network for Intrusion Detection in Cybersecurity‖ Proceedings of the 20th iSTEAMS Multidisciplinary Trans-Atlantic GoingGlobal Conference KEAN University, New Jersey, USA Pp 119-126, 2019.

[10] Adeniji Oluwashola David, Akinola Olaniyan Eliais. A secured text encryption with Near Field Communication (NFC) using Huffman compression. International Journal of Engineering and Applied Computer Science/IJEACS. 2022;4(2):1

[11]. Alam, K.S.; Bhowmik, S.; Prosun, P.R.K. Cyberbullying detection: An ensemble based machine learning approach. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 710–715.

1. [12]. Ghandour, C.; El-Shafai, W.; El-Rabaie, S. Medical image enhancement algorithms using deep learning-based convolutional neural network. *J. Opt.* **2023**, 1–11.

UNDER PEER REVIEW

UNDER PEER REVIEW