LSTM-BASED MODEL FOR CYBERBULLY DETECTION

Abodunrin O., *Esan A., Sobowale A., Adebiyi T., and Soladoye A.

Department of Computer Engineering, Federal University Oye-Ekiti, Nigeria

Corresponding author's email address: adebimpe.esan@fuoye.edu.ng.

ABSTRACT

Cyberbully has become rampant due to digitalization and conventional cyberbully detection is time-consuming, this led to the development of cyberbully detection systems. Previous cyberbully detection systems yielded low accuracy, hence, this research developed a LSTM-based model for cyberbully detection. The dataset for training the model was obtained from Kaggle and pre-processed by removing punctuation marks and stop words, stemming, tokenization and one hot representation. The system was implemented using Python 3.9 with a hold-out evaluation method, with 150 epochs and 64 batch sizes. The developed system was evaluated using: accuracy, precision, Recall and F1 measure and the results obtained were compared to other machine learning models as well as a hybrid of CNN-LSTM. The result shows The developed model yielded an accuracy of 77.0% with a validation time of 3.024 sec in the detection of cyberbullying while the hybridization of LSTM-CNN gave an accuracy of 74.80% for fake news and cyberbully detection. The developed model was also benchmarked with other machine learning models: SVM, KNN and RF and the system developed outperformed them. The outcome of this research shows that the deep learning approach used outperformed the machine learning models considered in this research for cyberbullying detection. However, future research should employ locally collected datasets for cyberbully detection.

Keywords: *Double exclusive OR, Encryption, Symmetric encryption algorithm, Triple data encryption algorithm, Web server*

INTRODUCTION

The current pace of digitalization with respect to the advancement in technology has brought about the rapid involvement of every individual in different social media platforms. This gave the people some level of independence and extreme freedom of expression and opinion which is a great advantage as it gives no room for intimidation or fear (Ezema and Inyama 2012). However, this privilege is being recently abused by some people as they use this platform to abuse fellow people because they know they cannot be seen at that moment.

Bullying refers to the abusive behavior of individuals physically while cyberbullying is abusive behaviour on the internet in which there is an imbalance of power among peers with the intent of harming others in a prolonged manner (Olweus 1993; Rosa *et al.*, 2018). It is any crime that involves computer and network which is getting more advanced and proves a degree to immorality and insanity. The main factor that separates cyber bullying from traditional bullying is the effect that it has on the victim. Traditional bullying may end in physical damage as well as emotional and psychological damage, as opposed to cyberbullying, where it is all emotional and psychological. Given the consequences of cyberbullying on victims, there is a need for an urgent approach to prevent its menace.

Cyberbully detection is a method of detecting and vetting a tweet or post on any social media platform to safeguard the emotional and mental stability of the netizens from all sorts of bullying ranging from gender, emotional, political, religious and age bullies. Conventional means of cyberbullying detection are done by reporting bully posts or tweets to the platform admin. This method consumed lots of time as the admin may not be presently available to attend to all reports at once. Hence, there is a need for the introduction of cyberbully detection systems which if not eradicating online bullying then will eventually reduce the rate at which it spreads.

Several approaches ranging from traditional machine learning to deep learning have been employed by various researchers for the detection of cyberbullying. Some of the employed approaches include the use of Support Vector Machines (SVM), Logistic Regression (LR) (Muhammad *et.al*, 2019), and Artificial Neural Networks (ANN) (Niklas and Brennan 2022), the limitation of these works includes: low performance or unsuitability for larger dataset during training and testing, inability to solve non-linear problems and high processing time.

Therefore, this research developed a Long Short Term Memory-based model for cyberbully detection. The developed model was chosen because it was able to solve complex sequential data, was better at handling long-term dependency, and was not affected by vanishing gradient problems.

RELATED WORKS

Rosa *et al.* (2019) performed a systematic review o fthe automatic detection of cyberbullying. The authors reviewed a total of 22 papers where most of the studies used textual data for the detection. The study concluded that the automatic detection systems have not been improved due to the model performance obtained during the experiment. Hani *et al.* (2019) conducted a study to detect cyberbullying on social media using machine learning approaches. The study performed tokenization, lowercase conversion, removal of stop words, word correction, and feature extraction using TFIDF. Support Vector Machines and Neural Networks are both employed to classify the extracted input features. The neural Network achieved a better accuracy of 91.76% than the Support Vector Machine. Nandhini*et al.* (2015) proposed a model to detect cyberbullying employing Naïve Bayes as the machine learning classifier. The model's learning data was obtained from MySpace.com. This work achieved a classification accuracy of 91%. Zhao *et al* (2016) also proposed a study to specifically detect cyberbullying. The dataset was preprocessed using some preprocessing techniques like word embedding which makes a bag of pre-defined insulting words and assigns different weights to them for feature extraction to obtain bullying features and for their easy detection in sentences when encountered. Support Vector Machine was employed as the machine learning algorithm for the classification of cyberbullying classification, the model achieved an accuracy of 79.4%. Bayzick *et al* (2011) developed a program called Bully Tracer which gives an average accuracy of 51.9% using the Myspace dataset while Nahar, *et. al.* (2013) employed some machine learning for the detection of cyberbullying using the Twitter dataset, it was discovered that SVM outperforms some other machine learning approach with an average accuracy of 64%. Dinakar *et. al* (2011)aimed to detect bullying language on sex, intelligence, race and culture with a dataset obtained from the YouTube platform by using some of the machine learning i.e SVM, Naïve Bayes, the result shows that SVM gave an average accuracy of 66% while Naïve bayes gave an average accuracy of 63%. Ali and Syed (2020) use some machine learning like LR, RF, SVM, Naïve Bayes and Ensemble for the detection of cyberbullying using the Twitter dataset, result shows that the three models Ensemble, SVM and LR gave a better average accuracy of 73% while

RF and Naïve Bayes which gave an average accuracy of 71% and 72% respectively while Muneer and Fati (2020) employed seven machine learning algorithms for the detection of cyberbully like LR, RF, SVM, LGBM, Adaboost (ADB), NB and SGB using Twitter dataset, result shows that SVM outperform all other machine learning employed. Raj *et. Al* (2021) propose some methods of the neural network plus parameter optimization and study of some traditional machine learning, the result shows that TF-IDF demonstrates the highest accuracy with traditional machine learning. Dani *et. al* (2017) use some sentiment frameworks on various traditional machine learning for the detection of cyberbullying using a dataset from two social media platforms like YouTube. The result shows that KNN model gave a better performance in detection with an average accuracy of 75.39%.

METHODOLOGY

This research developed an LSTM-based model for the detection of cyberbullying. The dataset for training the model was obtained from Kaggle which was a data bank for various types of online bullying. These datasets were text datasets that underwent a series of preprocessing techniques like Stemming, lemmatization, removal of stopwords and punctuations, label transformation, tokenization and vectorization. The next stage thereafter is the feature extraction stage where a word is converted into vector form called word embedding. The embedded word was trained using the LSTM model for the detection of cyberbullying. The system was evaluated using various evaluation metrics like accuracy, Precision, Recall, F1 score and AUC. The developed system was compared with the result from hybridized CNN-LSTM and some other machine learning algorithms. The summary of all the phases involved in the development of this system is represented by a block diagram as shown in Figure 1.

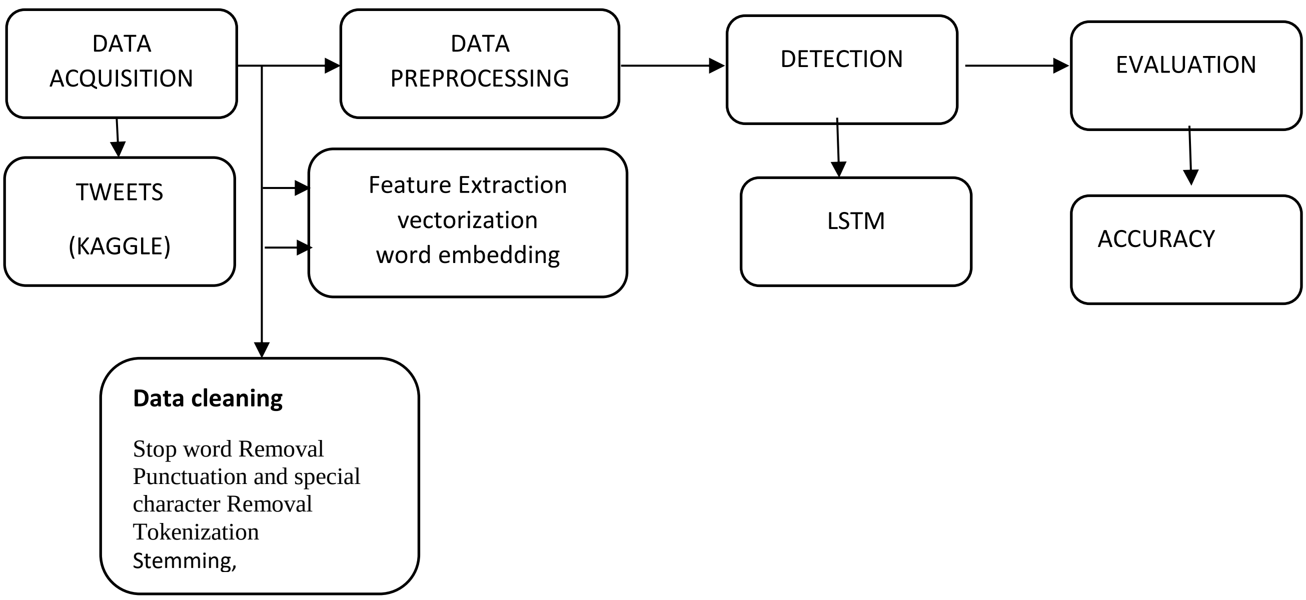

Figure 1 Block Diagram of Cyberbully detection model

Data Acquisition

The dataset for cyberbullying was obtained from Kaggle which contains various tweets from the Twitter page because it was regarded as the most widely acceptable social media platform that is publicly available. The dataset consists of two attributes: tweets text and cyberbullying type. This dataset contains a total of forty-seven thousand six hundred and ninety-two (47692) records. There are different bullying types present in this dataset like ethnicity, age, gender, religion and other bullying. The dataset was manually and randomly selected due to the unbalanced nature of the dataset as 'not bullying' has a total of 7947 instances. Out of the 47692 records, only 5000 instances were used to balance the data ensuring an unbiased outcome.

Data Preprocessing

The acquired dataset was pre-processed using the following techniques; Data cleaning, Lemmatization, Stemming, Removal of stop words and Punctuation marks, Label transformation, Tokenization, Text length uniformity, Vectorization as shown in Figure 3.

Data Cleaning

In this stage, the outliers (unimportant attributes) were removed for effective usage of the dataset. Removal of these attributes helped the system to concentrate on only the useful ones. In this stage, the serial number and title were dropped. This left our dataset with only the text and label columns. Raw news datasets collected from social media platforms for cyberbully detection, and bully data were cleaned of some noise or outliers that are of no relevance for the employed system.

Lemmatization

This is the process of grouping together the inflected forms of a word so that they can be analyzed as a single item, identified by the word's lemma or dictionary form. Lemmatization is the algorithmic process of determining the lemma of a word based on its intended meaning. This is a text pre-processing technique used in natural language processing (NLP) models to break a word down to its root meaning to identify similarities. WordNetLemmatizer was employed in this research for lemmatization due to its good performance as obtained from different journals.

Stemming

This is the process of removing suffixes or affixes that are added to a word. It is the reduction of a word back to its root or stem form after the inflection has been removed. The reduction of words to their stem gives room for the model to focus on the main word for classification and helps in accurate classification. Porter Stemmer was used for stemming in this research work.

Removal of Stopwords and Punctuations

Stopwords are words that do not add meaning to a sentence and the removal of these words will help in a drastic reduction of data size and the system's performance accuracy. Mostly when working with natural language processing, punctuation marks, special characters, and emoji usually don't have relevance with the content of bully words, so these marks and symbols are mostly discarded to reduce the size of data and increase computational time.

Label Transformation

The dataset labels are in the form of categorical data type (bully and not bully), this type of data type cannot be inputted into the model for processing. Therefore, there is a need for the transformation of these labels into their corresponding binary equivalent. Label encoding and one-hot encoding are the most common types of encoding techniques used by various researchers. However, manual encoding is also used as well for the correct encoding, this is the approach used for assigning these categorical labels to their relevant binary values of (0 and 1).

Tokenization

This is the process of breaking a textual dataset into smaller pieces like words, sentences, terms and any other syllabic elements, these smaller pieces are known as tokens. This is sometimes the first stage in natural language processing techniques. Tokenizer breaks the stream of unstructured textual data into discretized elements. Tokenizer was imported differently from the text preprocessing library.

Vectorization

This is the process of converting text into vectors as models will not understand text as input. To achieve this, one hot representation was used.


```
def pre_process_news(news):
    """
     The function applies the WordNetLemmatization , PotterStamming, stop word removal, and punctuation removal on the string for preprocessing.

      Parameters:
          news: string that needs to be pre-processed

      Returns:
          processed string
    """

    lemm = WordNetLemmatizer()
    ps = PorterStemmer()
    stop_words = set(stopwords.words('english'))
    news = news.lower()
    news = re.sub("[^0-9a-z]", ' ' , news) # removes punctuations
    tokenized = news.split(" ")
    news = [ ps.stem(word) for word in tokenized if word not in stop_words] # apply stemming and drop stopwords
    news = " ".join(news)  # join the tokens into a string
    news = lemm.lemmatize(news)  # apply lemmatization
    return news
```

```
# encode categorical data to numerical
import pickle, os, json, re, requests
df.loc[df['label']=='FAKE', 'label'] = 0    # FAKE = 0
df.loc[df['label']=='REAL', 'label'] = 1    # REAL = 1
```

Figure 2: Screenshot of some preprocessing technique

Implementation of the Designed Model For Cyberbullying Detection

The implementation is done using LSTM algorithm with Python 3.9 programing language on Google Colab: a virtual machine for Jupyter notebook developed by Google mainly for research purposes. The detection using a deep learning approach involves a series of steps after the needed Libraries like Pandas, Numpy, Sklearn, Tensorflow, Keras, and Nltk have been imported. The following steps were employed: Reading the dataset, preprocessing the dataset, splitting of dataset into training and testing, Building the Long-Short Term Memory (LSTM), Performing detection of cyberbullying using the developed LSTM and evaluating the system.


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf

import pickle, os, json, re, requests

#import sklearn packsges
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.utils import shuffle
from sklearn.model_selection import GridSearchCV

#import nltk packages
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.stem import PorterStemmer
nltk.download('stopwords')
nltk.download('wordnet')

# import keras packages
```

Figure 3: Screenshot of some preprocessing technique

Feature Extraction

It is an approach for representing words and documents. Word Embedding or Word Vector is a numeric vector input that represents a word in a lower-dimensional space. It allows words with similar meanings to have a similar representation and can also approximate meaning. This research used a word vector feature of 300, this is to give room for wider capturing of unique features.

Data Balancing

There are different bullying types present in the cyberbullying dataset, they include ethnicity, age, gender, religion and other bullying. Out of the 47692 records, only 5000 instances were used to balance the data to ensure an unbiased outcome.

Dataset Splitting

The dataset for this model was divided into training and testing; 80% for training, and 20% for testing. The reason for 80% for training is to enable us to have enough datasets for training our model.

Performance Evaluation of the Employed Model

One of the objectives of this research work is the evaluation of this employed model using a confusion matrix. A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions was summarized with count values and broken down by each class. Four different metrics i.e. accuracy, precision, recall and F1-measures were used to evaluate the performance of this model. These metrics are used to express the detection of cyberbullying. Therefore, we prefer the different factors to get more accurate results. The confusion matrix provides the details of the following values: True Positives (TP); which is the total number of positive cases classified as positive, True Negatives (TN); which is the total number of negative cases classified as negative, False Positives (FP); which is the total negative cases classified as positive and False Negatives (FN); which is the total positive cases classified as negative. From the values of this Confusion matrix values, we evaluated our model Accuracy, Precision, Recall and F1 score deduced from the equation below:

Accuracy

It is defined as the ratio of correctly identified cyberbullying to the total number of test cyberbullying which is represented by:

$$ACCURACY = \frac{TP+TN}{TP+FP+TN+FN} \quad (3.6)$$

Precision

This was used to measure the percentage of the truly positive out of all the positive predicted results. It is represented by:

$$PRECISION = \frac{TP}{TP+FP} \quad (3.7)$$

Recall

This was used to measure what percentage is predicted positive out of the total positive.

$$RECALL = \frac{TP}{TP+FN} \quad (3.8)$$

F1 score

It is the harmonic mean of precision and recall. It takes both false positives and false negatives into account.

$$F1\ SCORE = \frac{2 \times PRECISION \times RECALL}{PRECISION + RECALL} \quad (3.9)$$

Design of LSTM-Based Model for Cyberbullying Detection

The LSTM model contains three gates and one cell state, this cell state serves as a memory for the LSTM model for remembering the past, the three gates are forget gate (f), input gate(i) and output gate(o) (Thakur, 2018). Gate in LSTM is a sigmoid activation function that produces a value between "0 or 1", many times it is either "0 or 1". We made use of the sigmoid activation function because we want the gate to produce a positive value of "1". In this model value "0" means the gate will block data from passing through the gate while value "1" means the gate will allow data to pass through the gate. The processes that occurred inside the LSTM model during the implementation of the cyberbully detection are mathematically expressed in equations 3.10 to 3.15; Equations 3.10, 3.11 and 3.12 represent the Equation of LSTM gates (Thakur 2018) while equations 3.13, 3.14, and 3.15 represent LSTM Cell States (Takur, 2018).

The input gate is represented as $i_t=\sigma\ (w_i(\ [h_t\text{-}1,x_t]+b_i$……………………..…(3.10)

The forget gate is represented as $f_t=\sigma(w_f(\ [h_t\text{-}1,x_t]+b_f$…………………..……(3.11)

The output gate is represented as $o_t=\sigma\ (w_o([h_t\text{-}1,x_t]+b_o)$……………….…… (3.12)

The cell state is represented by equations 4 to 6

$c_t'=\tanh(w_c(\ [h_t\text{-}1,x_t]+b_c)$………………….. (3.13)

$c_t = f_t c_{t-1} + i_t c_t'$ …………….…………… (3.14)

$h_t=o_t \tanh(c_t)$ ……………….……….……(3.15)

Where,

i_t*=Input Gate*

f_t*=Forget Gate*

o_t=Output Gate

σ *= Sigmoid function*

w=weight of the respective gate (neuron)

h_t*-1=output of the previous LSTM block at (tims stamp t-1)*

x_t*=input at current timestamp*

b=Bias for the respective gates

c_t*=cell State (memory)at timestamp (t)*

c_t'*=candidate for cell State at timestamp (t)*

RESULT AND DISCUSSION

Determination of Optimal Parameters

Due to the hybridization of CNN and LSTM, the determination of the parameters is done using the varying input layer units. The input units' values are to the power of 2, to have the optimal parameter, 128 input units are used for both CNN and LSTM as this gave the best performance accuracy for the developed system. The 128 units for CNN and LSTM gave a total trainable parameter of 1,823,841.

Due to our diverse dataset, the performance accuracy obtained from hybridized CNN-LSTM and ordinary LSTM varies. CNN-LSTM gave the best performance with the Kaggle dataset at 128-8 hidden layers, while LSTM with the same number of hidden layers with epoch of 150 and batch size of 64 gave the best performance, likewise for cyberbullying dataset, LSTM with the same parameters and training arguments gave the best performance accuracy.

Result from Evaluation of the Developed Cyberbully Detection System

The optimal parameter used was 2 LSTM hidden layers with 128 and 64 neurons respectively. This amounted to a total of 1769,121 total and trainable parameters. The developed cyberbully system when run with 150 epochs and batch size of 8 gave us the below parameters in Table 1 shows the embedding layer, LSTM layers and the Dense layer with their output shapes and number of their parameters with the total trainable parameter of the developed LSTM model.

Table 1 gives the summary of the number of LSTM hidden layers designed and their respective output shape and parameters used for the implementation of the cyberbullying detection system.

Table 1: Summary of the developed LSTM model for Cyberbullying

Layer (Type)	Output Shape	Param #
embedding (Embedding)	(None, 100, 300)	1500000
lstm (LSTM)	(None, 100, 128)	219648
lstm_1 (LSTM)	(None, 64)	49408
dense (Dense)	(None, 1)	65

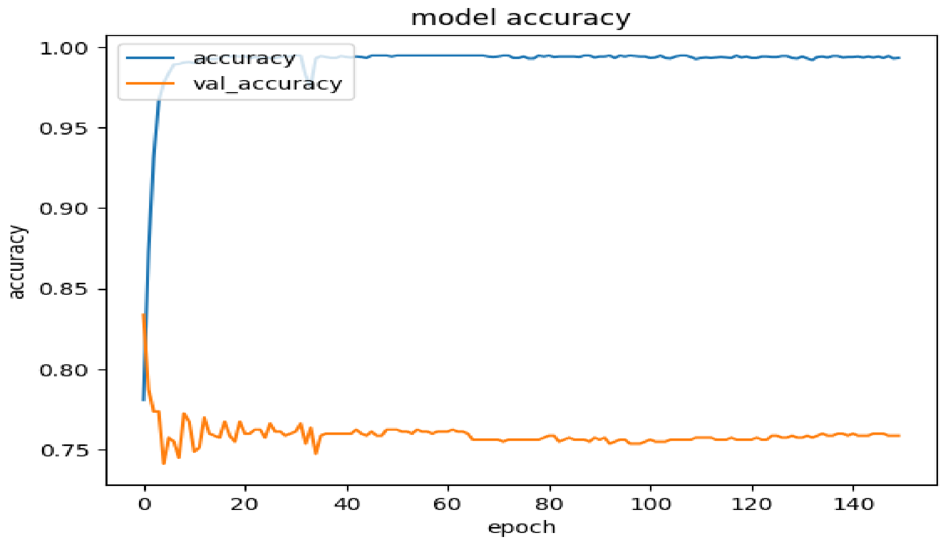

Figure 4: Cyberbullying Detection Training and Validation Accuracy

From Figure 5 above, it was discovered that our training accuracy is higher than our validation accuracy, and the validation accuracy was not that low, this shows that our developed cyberbullying detection system did not overfit. From Figure 1, it will be observed that the training accuracy increased drastically when the system reached around epoch 20, that was when the accuracy was 100% and it maintained this till the end of the 150 epoch, just that it dropped a bit to something close to 98% when the system was approaching epoch 39 and later returned to its 100% training accuracy from 40-150 epoch**.**

From Figure 5, our validation loss is more than our training loss, and it is known that the lower the loss value of a system the better performed the system is. Figure 2 shows that the developed cyberbullying detection system gave an increasing validation loss from epoch 40 and maintained the increment till 150 epochs, but the training loss from form 0.5 loss value and maintained the low loss value below 0.5 from epoch value 3 till the final 150 epochs. The employed system was trained with 150 epochs and 64 batch sizes with a validation split of 0,2, the system gave an impressive accuracy of 77% with a weighted average for precision, recall and F1 score are 77% respectively and detection time was 3.024secs. the experimentation result of the developed system is shown in Table 2.

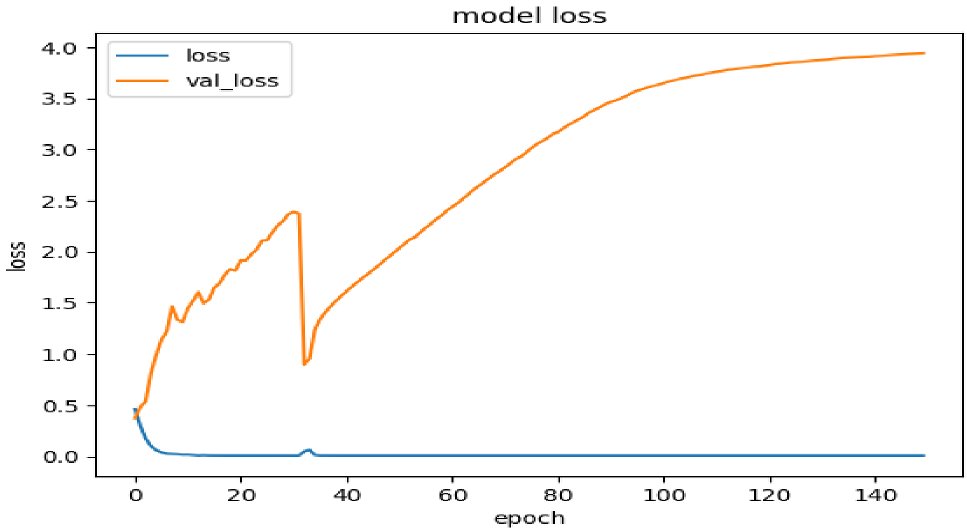

Figure 5: Cyberbullying Detection Training and Validation Loss

Table 2: Performance Evaluation Result of the Cyberbullying Detection System

Dataset	Avg. accuracy (%)	Avg. Precision (%)	Avg. F1Score (%)	Detection Time (sec)
Kaggle Cyberbullying dataset	77	77	77	3.024

Table 3: Comparison of the developed Cyberbullying Detection System with other ML algorithms using the Kaggle Dataset

S/N	Algorithm	Avg. Accuracy (%)	Avg. Precision (%)	Avg. Recall (%)	Avg. F1Score (%)	Detection Time
1	LSTM	77.00	77.00	77.00	77.00	3.02
2	CNN-LSTM	74.80	75.00	75.00	75.00	5.289
3	Logistic Regression	69.80	71.00	70.00	69.00	0.40
4	KNN k=3	63.20	63.00	63.00	63.00	0.41
5	SVM	60.00	60.00	60.00	60.00	0.55

Comparison of the Developed Cyberbullying Detection System with Other Machine Learning Algorithms Using Twitter Dataset.

To achieve part of the fourth objective of the study, the developed system for cyberbullying was compared with other traditional machine learning algorithms like Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and CNN-LSTM. This comparison was experimented to ensure that the deep learning algorithm (LSTM) employed as the detection model for the developed system gives the best performance than other commonly used and related algorithms. Table 3 shows the experimental results obtained from using the same dataset with varying traditional machine learning and deep learning algorithms concerning different evaluation metrics like Accuracy, Precision, Recall, F1Score and how fast they can detect maybe a tweet is cyberbullying or not.

From the Cyberbullying detection result presented in Table 4.4, LSTM also gave the best performance accuracy of 77%, followed by CNN-LSTM. But Logistic Regression gave the fastest detection time of 0.40 seconds

Comparison of the Developed Cyberbullying Detection Systems with other Existing systems.

The developed system uses the foreign dataset from Kaggle that consists of tweets classified as cyberbullying and not cyberbullying. The developed system was compared with other existing systems for the detection of cyberbullying. The developed system gave a better performance accuracy than all the existing systems. The study carried out by Dinakar *et al* (2011) was the only system that gave a performance accuracy that was closer to that of the developed system. Table 4 shows the comparison result obtained when the developed system was compared with other existing cyberbullying detection system

Table 4: Comparison of the Developed System Using Kaggle Dataset with Existing Systems for Cyberbullying Detection.

S/N	Author	Algorithm	System	Accuracy (%)
1	Nahar, *et. al* (2014)	SVM	Cyberbullying	64.00
2	Dinakar*et.al* (2011)	SVM	Cyberbullying	66.00
3	Bayzick*et al* (2011)	Buller tracer	Cyberbullying	51.90
4	Developed system	LSTM	Cyberbullying	77.00

CONCLUSIONS

This research work developed a Cyberbully detection system using the Long Short Term Memory Model. The dataset for training the model for cyberbullying detection was obtained from Kaggle.

The developed systems were implemented on Google Colab with Python 3.9. the LSTM model used on Cyberbullying datasets, 128 neurons at the input layer and one hidden layer with 64 neurons with *Tanh* as the activation function at the input and hidden layers while Sigmoid was used as the activation function at the dense layer, the same design and hyper-parameters were used for the system design of CNN-LSTM. The cyberbullying dataset acquired from Kaggle was also used to develop a system with the following models LSTM, CNN-LSTM, LR, KNN, and SVM with average accuracies of 77.00%, 74.80%, 69.80%, 63.20%, and 60.00% respectively, this shows that LSTM outperforms all other algorithms with an average accuracy of 77%.

Cyberbullying detection was implemented with LSTM as the model gave the best performance accuracy. With the above result, it shows that Long Short-Term Memory (LSTM) model is the best detection algorithm for cyberbully detection using an Open Access dataset obtained from Kaggle.

REFERENCES

Ali, A. and Syed, A. (2020) "Cyberbullying detection using machine learning", *a Pakistan Journal of Engineering and Technology (PekJET) 5,(01), Pp 45-50*

Bayzick, J., Kontostathis, A., and Edwards, L. (2011). "Detecting the presence of cyberbullying using computer software". *Proceedings of the 3rd annual ACM Web Science Conference (web. Sci 11) pp.63*

Dani Harsh, Li Jundong, and Liu Huan,(2017) Sentiment informed cyberbullying detection in social media. *In Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Pp 52– 67.

DinakarKarthik, ReicartRoi, and Lieberman Henry (2011)" Modelling of Textual Cyberbullying-Conference": *The Social Mobile Web, Papers from the 2011 ICWSM Vol. II.*

Ezema, M.E, and Inyama, H.C, (2012): "An assessment of Internet Abuse in Nigeria". *fake news detection- Department Computer Science, University Nigeria Nsukka, and Department of Computer and Electronics Engineering, Nnamdi Azikiwe University Awka*

Hani, N., Ahmed, A., Amer, E., and Mohammed (2019) "Social MediaCyberbullying detection using machine learning": International *Journal of Advanced Computer Science and Applications 10(5):703-707 DOI:10.14569/IJACSA.2019.0100587*

Muhammad S. M, Yusmadi J, Novia A, and Noraini C. P (2019)" Fakebuster: Fake News Detection System Using Logistic Regression Technique In Machine Learning" *International Journal of Engineering and Advanced Technology 9(1):2407-2410 DOI:10.35940 /ijeat.A2633.109119*

Muneer Amgad and Suliman Mohamed Fati (2020)- "A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter"- *Future Internet, MDPI, 12(11), Pp 1-20, 7202*

Nandhini, B, and Sheeba, J .(2015) "Cyberbullying detection and classification using Information Retrieval Algorithm", *Conference: the 2015 International Conference, DOI:10.1145/2743065.2743085.*

Nahar, V., Li, X., & Pang, C. (2013). "An effective approach for cyberbullying detection".*Communications in Information Science and Management Engineering, 3(5), 238–247.*

Niklas, D. and Brennan(2022)- "4 Disadvantages of Neural Networks and Deep Learning"

Olweus, D. (1993) Bullying at school: tackling the problem: *Article paper, Pp. 24. Gale Academic, One file, link gale.com/apps/doc/A75022135*

Raj, C., Ayush, A., Gnana, B., Bhuya, N. and Mukesh, P. (2021)-"Cyberbullying Detection: Hybrid Models Based on Machine Learning and Natural Language Processing Techniques" *– 10th Anniversary of Electronics: Recent Advances in Computer Science & Engineering) https://doi.org/10.3390/electronics10222810*

Rosa, H., Pereira, N., Ribeirom R., Ferreira, P.C., Carvalho J.P., Oliveira, S., Coheur, L., Paulino, P.,VeigaSimão, A.M., and Trancoso, I. (2018) "Automatic cyberbullying detection: A systematic review" full-length a*rticle: Computers in Human Behavior 93, Pp 333-345, https://doi.org/10.1016/j.chb.2018.12.021*

Thakur, D. (2018) LSTM and its equation, Google Summer Code 2019

Zhao, R., Zhou, A., and Mao, K. (2016.)" Automatic Detection of Cyberbullying on Social Networks based on Bullying Features". *Conference DOI:10.1145/2833312.2849567*