REGAC: Multi-Class Offensive Content Identification Using Graphical Approach

Sneha Chinivar, Roopa M S, Arunalatha J S, Venugopal K R

***Abstract*—Online offensive behaviour continues to rise with the increasing popularity and use of social media. Various techniques have been used to address this issue. However, most existing studies consider offensive content identification as a binary or ternary problem, disregarding the potential for multi-class classification of derogatory content. In this work, we propose *RoBERTa* Embedding-based Graphical Approach Classifier *(REGAC)*, which aims to designate the identified offensive content into varied classes,like age, gender, ethnicity, religion, and others to understand what exact qualities the bully usually targets in their victim so that pertinent measures can be taken to battle them. Additionally, the models' efficiency is evaluated using an unbalanced dataset to identify types of offensive content. The word embeddings are generated for a balanced dataset using *RoBERTa* embedding model to bring out the best-fitting vectors. These vectors are then input into traditional Machine Learning algorithms *(SVM, KNN, Logistic Regression, Random Forest,* and *XGBoost*) and graph-based algorithms (Graph Convolution Network *(GCN), GraphSAGE, and Graph Attention Network (GAT))* for fine-grained categorization of offensive textual content. The experimental results demonstrate the efficiency of combating social media's offensive content of the proposed work with higher Precision, Recall, F1-Score, and Accuracy compared with most state-of-art approaches.**

***Index Terms*- Graphical Approach, Machine Learning, Online Offensive Behaviour, Social Media.**

I Introduction

Social media is now widely used worldwide for communication, entertainment, marketing, and other online activities. It has become an integral part of daily life. It is one of the most convenient and cost-effective ways to stay connected with people worldwide. In the last couple of years, social media has played a vital role in assisting individuals and communities in desperate requirements. It is one of the right places to start any noble cause, as it helps reach out to many people. It can also serve as a platform to unify people with the same interest worldwide. It is the best place on the internet to be updated with the latest information and news. It contributes literally to the globalization of the world at large.

The increased use of social media and their anonymity encourages for exhibiting online offensive behaviour, like harassment, hate speech, abuse, bullying *etc.* These abusive behaviours are showing a significant impact on society by affecting the mental health of many. An increasing number of people are experiencing mental health disorders, such as anxiety and depression, due to online offensive behaviour [1]. In some cases, online bullying has driven individuals to take the extreme step of self-harm or suicide. This motivated us to develop an approach to efficiently combat offensive behavior on social media.

Social media networks can be represented as graphs, as shown in Fig. 1. Graphs are widely recognized data structures that represent networks or relationships among data elements. They provide a comprehensive representation of relationships among data. Usually, an adjacency matrix is used to represent the network or graph to make them computationally understandable. Still, the graphs can also be expressed using an edge or adjacency list and are often considerably better than one can depict in lists or data frames. Graphs enable us to represent the connection between the data distinctly and help to understand how things are related to one another [2].

In this work, we have used the *RoBERTa* word embedding-based Multi-Class Offensive Content Identification using Graphical Approach.

Manuscript received April 18, 2024; revised November 14, 2024 .

Sneha Chinivar is a Research Scholar in the Department of Computer Science and Engineering, Bangalore University, K R Circle, Bengaluru, Karnataka, India. (e-mail:schinivar@gmail.com)

Roopa M S is an Associate Professor in the Department of Computer Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, Karnataka, India. (email:roopams22@gmail.com)

Arunalatha J S is a Professor in the Department of Computer Science and Engineering, Bangalore University, K R Circle, Bengaluru, Karnataka, India. (e-mail:arunajs99@gmail.com)

Venugopal K R is a Former Vice-Chancellor, Bangalore University, Bengaluru, Karnataka, India. (e-mail:venugopalkr@gmail.com)

Fig. 1. An Instance of Depicting Social Media Network as Graph

.

RoBERTa word embedding technique converts the textual content to vectors. The vectors are given as input to both Machine Learning and Graph-based classifiers to efficiently identify what particular victim qualities the bullies are attacking and to recognize the type of offensive content by performing fine-grained classification. Most existing works have focused on detecting online bullying content using binary or ternary classification, and the fine-grained Multi-class classification of offensive content identification is limited. The fine-grained classification will facilitate in identifying not only the offensive content but also its type and better comprehend the bullies general mindset or psychology and help to take further steps to address them *via.,* cautioning users in advance, the targeted healing process, *etc.,*

The main contributions of this work are summarized as follows.

(i) We propose a generic model for Multi-labelled text classification using Graph Neural Network.
(ii) We devise a *REGAC* Algorithm to achieve Multi-class offensive content identification using a graphical approach.
(iii) We have analyzed the performance of the *RoBERTa* Embedding-based Graphical Approach classifier in terms of Precision, Recall, F1-Score, and Accuracy.

The proposed approach's novelty lies in the combination of embedding technique and three varied graph-based neural network layers applied on the entire dataset to perform multi-class offensive content detection, unlike [3], where it uses a different identification approach and that too only on small part of the dataset.

The following section outlines the literature survey; in Section 3, we describe the methodology and brief the experimental setup and results in Section 4. The results are discussed in Section 5, and finally, Section 6 contains the conclusions.

II Literature Survey

This section summarizes recent studies focused on identifying derogatory content in social media networks. The advancement of Natural Language Processing (NLP) over the years has enhanced embedding models' ability to understand textual content and its context. To appropriately recognize online offensive content by the system, the crucial thing is to understand the content correctly. Various word embedding techniques developed over the years facilitate this process. Based on this understanding, an appropriate classifier should be chosen to categorize the content effectively.

Feature extraction is extracting only essential features instead of giving the entire data to the classification algorithms so that the extracted data will still contain all the critical information necessary for further processing. Using this technique reduces the computation time. Word embedding techniques are employed to convert textual data into vectors, as classification algorithms cannot process textual content directly. Mostly *word2vec* and *TF-IDF* are used to get feature vectors of textual data. But Khan *et.al* [4] has used discrete emotional features along with *word2vec*. Then the extracted features are fed to deep neural network models to classify social media's aggressive and non-aggressive textual content appropriately.

A framework based on a graph convolutional network named *SOSNet* has been proposed in Wang *et.al* [3], which uses the tweet's inherent semantic connection to identify the objectionable content. Further, the identified derogatory content is categorized into an appropriate class of multi-class classification. *SOSNet* classification technique has experimented with varied word embedding techniques *viz., SBERT, BERT, DistilBERT*

etc., and *SOSNet* in combination with *SBERT* gave the best performance comparatively.

Murshed *et.al* [5] have used *word2vec* and *TF-IDF* word embedding techniques to extract features. The information Gain method is used to select the prominent features among them that are necessary to recognize bullying events. Then these features are given to a deep learning-based hybrid classifier to identify tweets containing bullying content accurately.

BoW, TF-IDF and *GloVe* word embedding techniques convert textual content to numeric vectors in Ojo *et al.,* [6]. Then these vectors are given to prominent Machine Learning algorithms *viz., SVM, Naive Bayes, Logistic Regression, Random Forest* and one-dimensional *CNN* in combinations to recognize hate speech and compared their results to identify the best combinations on sentence level annotated English posts of internet forums.

An end-to-end method has been proposed in Miao *et al.,* [7] that has used community structure and features of text to recognize the offensive language. *Graph Attention Network (GAT)* layers are used here to capture the community structure features, fused with text embeddings generated by *BERT* using attention mechanisms. In addition, this method represented users with information about their historical behaviour, indicating their general tendency to use derogatory language in social media. This reduced the computation burden of the model as the number of parameters to be considered by the graphical neural network got reduced and helped to make the model more efficient.

A systematic method to identify online harassment and to analyse intentions behind every comment on social media has been proposed in Abarna *et al.,* [8]. Similarity measures and *Fast Text* models are used to build an efficient conventional model to analyse the text's lexical meaning and the order of the word in the textual comments that contain harassment words. Various feature extraction techniques are used to accurately identify the target groups and understand the intentions behind every textual comment.

Identification of cyberbullying in social networks has been proposed in Azeez *et al.,* [9] and Fang *et al.,* [10] in which the former uses the approach of artificial intelligence where various traditional machine learning algorithms and ensemble models are used to recognize bullying tweet. The latter uses *GloVe* embeddings for vector generation and Bi-Directional Gated Recurrent Unit (*Bi-GRU*) and the self-attention mechanisms to perform binary classification of whether the given text contains bullying content or not.

Anti-cyber bullying system based on artificial intelligence was developed by Ige *et al.,* [11] which uses Multinomial *Naive Bayes* and optimized *SVM* to identify and intercept both incoming and outgoing bullying messages and to take appropriate action.

A neural network based model is developed by Agbaje *et al.,* [12] which in addition uses sentiment analysis on Twitter data to appropriately detect content that contains cyberbullying and aggression text. Chatzakou *et al.,* [13] proposed a robust methodology to identify aggressors and bullies from normal users on Twitter social media using textual content that gets posted, user profile information, and network-based attributes and classify those accounts into an appropriate category based on the detection using prominent machine learning algorithms.

Online bullying content has also been detected across multiple social media platforms *viz.,* Formspring, Twitter, and Wikipedia in Agrawal *et al.,* [14] utilizing various machine learning and deep learning based models and transfer learning.

A Deep Neural Network is combined with a Convolutional and Gated Recurrent Network in Zhang *et al.,* [15] to recognize online hate speech in Twitter social media. This method captured both word sequence and order of information in comparatively shorter texts and efficiently recognized hate speech content.

Rezvani *et al.,* [16] have proposed an attention-based model that combines context and textual features of the text to detect cyberbullying content on social media.

Song [17] suggested a framework that integrates online posted text's semantic, context and structural features of interaction networks to improve the efficiency of recognizing abusive online content. Cecillon [18] and Mishra [19] utilized graph-based techniques to identify the offensive online language, where, [18] used a graphical embedding approach to learn the representations of conversational messages depicted as graphs, whereas [19] used GCN to capture the user's linguistic behaviour along the online platform's structural details.

Ahmed [20] analyzes the performance of varied transformer-based models such as BERT,

DistilBERT, RoBERTa *etc.,* and their ensembles to identify cyberbullying traits with balanced and imbalanced datasets. Maity [21] uses a graph-based framework with cosine similarity to generate a single graph for the entire BullySent corpus to detect online bullying content.

From the above reviewed recent research works, we see that researchers are taking offensive language detection as mostly a binary or ternary problem instead of a multi-class identification problem. Most of them ignored considering the qualities of the victim the bully is targeting, *viz.*, age, religion, gender, *etc*. In addition, using a graphical approach to address the problem is limited. It is necessary to consider the fine-grained classification implementation utilizing a graph-based approach to design a mechanism that efficiently controls offensive language usage in social media since a graph is one of the most efficient ways to represent social media.

III Problem Statement

In this section, we define the problem statement for offensive content identification. Given a set of Online posted texts $T = (t_1, t_2, \ldots, t_N)$, where N is the total number of texts in our input data, our proposed *REGAC* algorithm, which consists of graph-based classifiers aims

1) To efficiently classify identified offensive content into one of five multi-class categories—age, gender, religion, ethnicity, and others—and determine the specific qualities or features targeted by bullies.
2) To reconfirm the models' efficiency on an unbalanced dataset in identifying the type of offensive content.
3) To enhance Precision, Recall, F1-Score, and Accuracy metrics compared to state-of-the-art methods.

In particular, the input dataset is converted to machine-readable format (vectors) using the Word embedding technique, which is capable enough to capture the relevant features of the text. Then these representations are converted into a graph using Approximate Nearest Neighbour Oh Ya (*ANNOY*). The graph data are given to three graph convolution-based classifiers, to classify the Online texts into their appropriate multi-class category.

IV Methodology

The generic model for multi-labeled text classification using a Graph Neural Network (GNN) is depicted in Fig. 2. The dataset, consisting of textual content, is input into a word embedding technique to generate numerical representations of the text. Embeddings convert text into vector representations that can be used for further analysis. It spans the comprehensive human language to that of a machine and disseminates text declarations in an n-dimensional space. They are the method of feature extraction of the text so that these features are fed to the machine learning model to work with text data. In simple terms, word embeddings are vector representation of words such that the words with a similar meaning or the same have a similar kind of representation. They are essential for solving most NLP problems, as machine learning models cannot directly interpret text and require it to be converted into numerical vectors. Various embedding models, such as *TF-IDF, Word2Vec, GloVe, FasText,* and *BERT*, are available for this purpose [22].

TABLE I
NOTATIONS USED IN THE ALGORITHM

Notation	Description
t	Online Posted Texts
N	Total number of Online Posted Texts in input data
v	Vector
n	Number of Vector Dimensions
R	Number of Nearest Neighbours
G	Graph
V	Vertex(Node)
E	Edges (Edges of every V to its few nearest neighbour)

The output of the Word Embedding Technique, *i.e.,* the vectors, is given to Graph Creation Algorithm. Most problems are graphs in their true sense, and the data, such as social networks, molecules, research work citation networks represented as graphs [23]. The Graph Creation Algorithm results in a Graph where each node represents the text, and the nodes are connected to their nearest nodes with edges. The output Graph generated by Graph Creation Algorithm is given to Graph Neural Network (GNN) model to represent each node better and classify them into an appropriate class of multi-labelled classification.

A. *REGAC* Algorithm

The *REGAC* (RoBERTa Embedding-based Graphical Approach Classifier) algorithm for fine-grained offensive content detection in online posts consists of three phases:

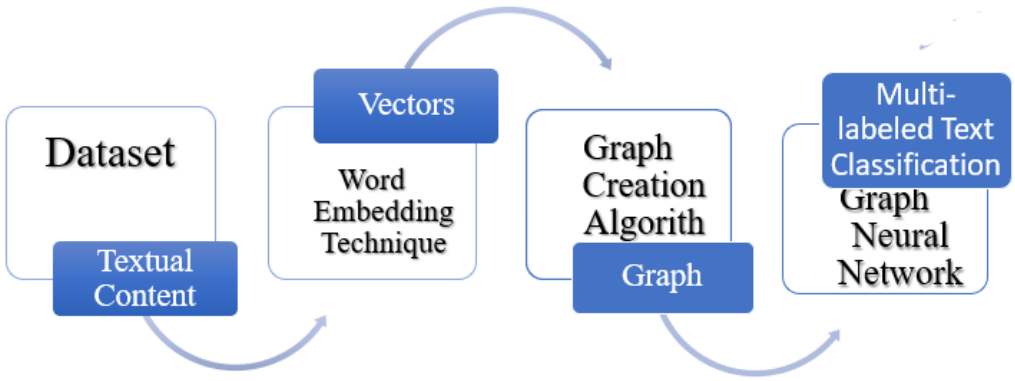


Fig. 2. Generic Model of Multi-labelled Text Classification.

Algorithm 1 : *REGAC*

Input: Online Posted Texts
Output: Multi-class Labelling of Online Posted Texts

Begin
Phase 1:Generation of Vectors from *RoBERTa* Word Embedding Technique
for online posted text $t \leftarrow t_1$ to t_N **do**
Compute Vector $v \leftarrow v_1$ to v_N where, size of each v = n dimensions
end for
Phase 2: Graph Creation
Create *ANN* index for every v embedded with its label using *Annoy*
Create graph *G(V, E)* from the index by identifying R nearest neighbours.
Phase 3: Multi-label Classification using Graph-based Algorithms
(i) ***GCNConv***
Input *G(V,E)* to 2-layers of *GCNConv*
Outputs label for every v
(ii) ***SageConv***
Input *G(V,E)* to 2-layers of *SageConv*
Outputs label for every v
(iii) ***GATConv***
Input *G(V,E)* to 2-layers of *GATConv*
Outputs label for every v
End

1) **Phase 1: Generation of Vectors from *RoBERTa* Word Embedding Technique**

Each piece of textual content is input into the RoBERTa word embedding technique, which is pre-trained on hate speech, to generate n-dimensional vectors as output.

2) **Phase 2: Graph Creation**

The vectors generated in phase 1, embedded with its label, are given as input to Annoy to create the *ANN* index. The index is then utilized to construct the graph *G(V, E)* by taking a few nearest neighbouring vertexes for every node. This phase gave a graph consisting of vertex V equal to the number of texts passed as input and edge E to R nearest neighbouring nodes of every node.

3) **Phase 3: Multi-label Classification using Graph-based Algorithms**

This phase uses three graph-based classifiers *viz GCNConv*, *SageConv* and *GATConv*.

(i) For every vertex V, *GCNConv* initially computes the addition of vector representation of all its neighbouring nodes and itself and then applies the *Mean Aggregator* function over the obtained values. Subsequently, it is passed through two layers of *GCNConv*, which consist of the *ReLu* activation function, and later through softmax to get

an appropriate label for every node.

(ii) The *SageConv* layer selects two hops and randomly samples nodes from these hops to construct the computation graph for each vertex *V*. The representation was acquired for all the nodes on the computation graph and passed them through the Aggregator (*Mean* and *Pooling Aggregator* separately to know the impact of the results upon the usage of varied Aggregators. *LSTM Aggregator* did not work for our datasets as it required sorted indices). Later vertex *V's* already existing representation is concatenated with the new representation and multiplied the same with the weight matrices, followed by passing them through non-linearity. Once after reaching the final representations of every node, they will be passed through the neural network to obtain one of the multi-class labels for each vertex *V*.

(iii) The *GATConv* for every vertex *V*, computes its transformed representation by calculating the attention weight of all its connected neighbours by passing its present representation, and its connected neighbouring node's representation through non-linearity and later concatenates these representations and passes them through a single-layer feed-forward neural network and further through *softmax* to get attention weight for the connected neighbouring node. The exact process will be repeated for every connected neighbour of vertex *V*. Finally, for each neighbouring node of vertex *V*, a linear transformation is performed, multiplied by the respective attention weight, and summed. The result is then passed through a non-linear activation function. After getting the final transformed representation for every vertex *V*, pass those representations or vectors through the neural network to obtain their labels.

B. Implementation of *REGAC* Algorithm

In this section, we discuss the implementation of the *REGAC*algorithm through an illustration for online Twitter posts. The Fig. 3 illustrates the overall approach adopted to recognize the offensive content of social media by carrying out fine-grained classification; they are described as follows:

(i) **Dataset:** A comparatively balanced dataset developed by Wang [3](copyright obtained) using the process of semi-supervised based Dynamic Query Expansion (*DQE*) to extract specific class's natural data points of Twitter to automatically generate a multi-class balanced dataset, which is made publicly available has been used in this work [24]. The dataset consists of 47,689 entries, with 39,744 containing cyberbullying content and 7,945 classified as non-offensive content. Out of those 39,744 cyber bullying data, 7,992 data have age related offensive content, 7,961 data are ethnicity related, 7,973 are gender related, 7,998 are religious related, and finally, the others class contains 7,820 offensive content data and the same is pictured in Fig. 4.

In addition we have used the dataset provided by [25](copyright obtained), which consists of 62,587 data instances in total. This dataset has multiple labellings for many of the data instances, hence with the help of three annotators we relabelled the entire dataset for single labelling by clearly describing each of the labels and distinctions between them to annotators. The class wise distribution of this dataset is depicted in Fig. 5. In this work, we have used 70% of the dataset for training, 10% for validation, and the remaining 20% for testing purposes.

(ii) ***RoBERTa* Word Embedding Technique:** Embedding models play a huge role in better understanding of contents by models; usage of appropriate word embedding models is essential in determining the efficiency of detection models. We experimented with varied embedding models, precisely on *Word2Vec, GloVe, FasText, BERT, SBERT, DistilBERT,* and *RoBERTa* on the same dataset used here in our previous work [26] and found *RoBERTa* performance a cut-above for our goal of fine-grained classification of identified offensive textual content. Therefore, in this work, we have used the *RoBERTa* embedding, a transformer-based model particularly pre-trained on hate speech and retained its default output vector dimension size of 768.

A robustly optimized *BERT* pretraining approach is an extension of *BERT*. The goal of *RoBERTa* is to optimize the pre-training procedure of *BERT* architecture. It shares a similar architecture to *BERT* with a simple modification in its design and training procedure.

RoBERTa architecture removed the *BERT*'s Next Sentence Prediction (NSP) objective as removing NSP loss slightly improves or at the least

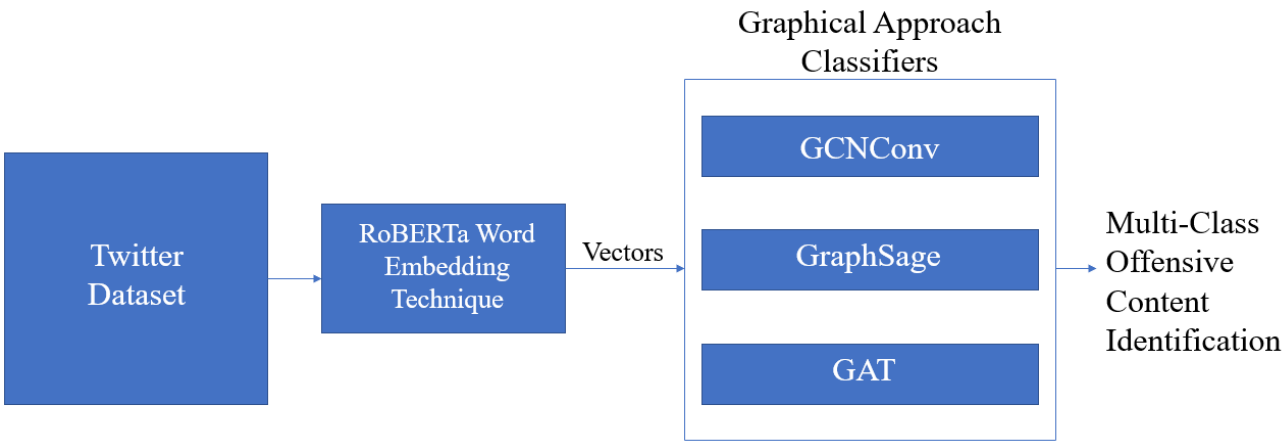


Fig. 3. Architecture of the Proposed Model for Fine-grained Offensive content Detection.

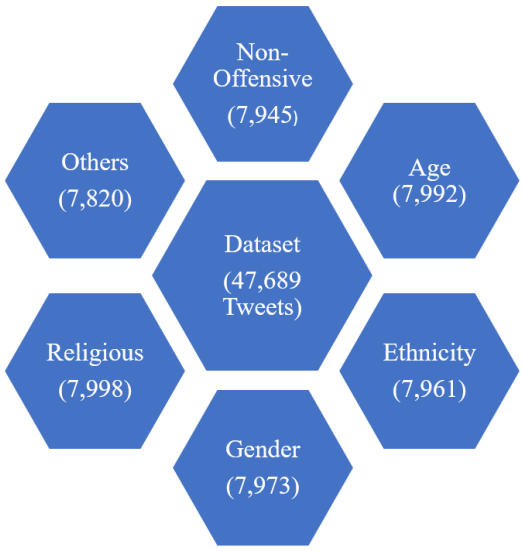


Fig. 4. Multi-Class Distribution of the Dataset Provided by [24].

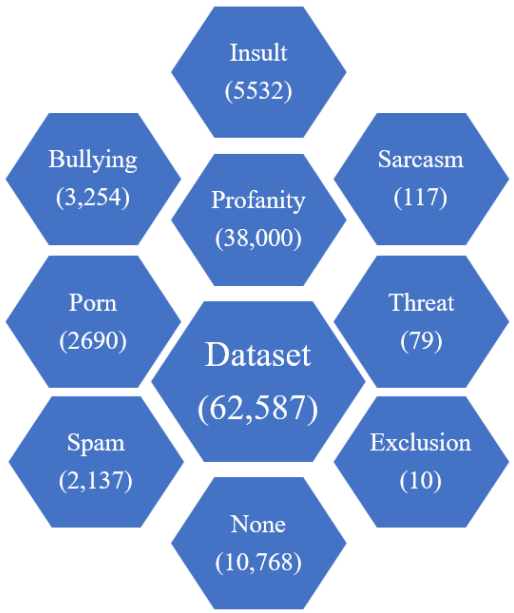


Fig. 5. Multi-Class Distribution of the Dataset Provided by [25].

matches the performance of the downstream task. It also trains with large-size batches and long sequences compared to *BERT*, as the big batch size enhanced the accuracy of the end task and the masked language modeling objective's perplexity. Along with this, parallelizing big batch sizes is easier *via* distributed parallel training. *RoBERTa* uses a dynamic masking pattern to generate different masking each time data is passed to the model against *BERT*'s single static mask.

RoBERTa has trained on Book Corpus and English Wikipedia dataset, which contains 16 GB of text data, CC-News which has 63 million news articles, Openwebtext dataset of size 38 GB, and Stories containing 31 GB of text data [27].

The embedding model is specifically pre-trained on hate speech and implemented using Hugging Face's transformer library [28], [29].

(iii) **Graphical Approach Classifiers:** After obtaining 768 dimension vectors from *RoBERTa* embedding, an Approximate Nearest Neighbor (*ANN*) index is created using Approximate Nearest Neighbor Oh Yeah (*ANNOY*) to construct a graph by applying a distance threshold. Particularly for this work, we have chosen the 50 nearest neighbours to maintain the right balance of computational cost and efficiency.

Graph Convolution Networks (*GCN*) are convolutional neural networks that can be applied straight on graphs and benefit from their structural details. They are very capable neural network architecture on graph data. It is a technique for semi-supervised learning on data that can be structured as a graph. Numerous varied kinds of graph convolution layers are available in the literature. Exploration is the only way to select

the most appropriate layer for a downstream task [30]. In this work, we have explored three known *GCN* layers, namely *GCNConv* presented by [31], *GraphSAGE* given by [32], and *Graph Attention Network* (*GAT*) conferred by [33], which took in vectors generated by the *RoBERTa* embedding model discretely and gave out pertinent node representation which in turn are given to neural network separately to recognize offensive textual content by performing multi-label categorization.

V Experiments

This work was implemented on a Tesla T4 GPU with 32GB RAM. Python was used for implementation, and the Google Colab environment was utilized to execute the code. PyTorch Geometric has been employed for graph implementation and python scikit-learn as a machine learning library.

A. Baseline Models

The performance of *REGAC*, *i.e., REGAC* (*RoBERTa + GCNConv*) and *REGAC* (*RoBERTa + GraphSAGE*), is compared with the two baseline methods, namely, Graph Convolutional Network Approach (*SOSNET*) [3] and Hybrid Deep Learning Model of 1-Dimensional Convolutional Neural Networks and Bidirectional *LSTM* (*Res-CNN-BiLSTM*) [34]. Both baseline methods share a similar objective with *REGAC* to combat offensive content on social media by categorizing bullying content into fine-grained classes, thereby understanding the targeted qualities of victims and controlling objectionable content. *SOSNET* combines *SBERT* and a graph convolution network for the multi-class classification of offensive content. However, this approach was applied to only 10% of the dataset, with statistical methods used to extrapolate results for the entire dataset. Whereas *Res-CNN-BiLSTM* , a hybrid deep learning model, has used GloVe embedding technique in combination with Bi-directional Long Short Term Memory (*Bi-LSTM*) and one-dimensional *CNN* individually and concatenated the results obtained from both and passed them through linear transformation followed by SoftMax. Our approach (*REGAC*) outperformed *SOSNET* in terms of Precision, Recall, F1-Score, and Accuracy, while also achieving relatively better results compared to the *Res-CNN-BiLSTM* implementation.

RoBERTa word embedding generated vectors are given as input to traditional Machine Learning algorithms to check their performance in appropriately classifying the recognized offensive content into multi-label classification *viz.,* age, gender, religion, ethnicity, and others.

The traditional machine learning algorithms that are used are

1) K-Nearest Neighbours(*KNN*) algorithm: *KNN* is a supervised Machine Learning algorithm that uses similarity criteria to classify the given data into its appropriate category based on the already available data [35].

2) Support Vector Machine(*SVM*) algorithm: *SVM* is a very popular supervised Machine Learning algorithm that creates the best possible decision line named *Hyperplane* to separate n-dimensional space into classes. Any new data that comes in is put into a suitable category using this hyperplane [36].

3) Logistic Regression: Logistic Regression is another well-known supervised machine learning algorithm which predicts the dependent variable category given the set of independent variables. It is basically used to foresee the likelihood of occurring a binary event.

The three types of Logistic Regression are

a) Binary Logistic Regression - It is used when there are two possible outcomes such as 0 and 1.
b) Multinomial Logistic Regression - It is used when there are multiple unordered outcomes such as Cat, Dog, Lion *etc.,*
c) Ordinal Logistic Regression - It is used when there will be ordered outcome such as Low, Medium, and High [37].

4) Random Forest: Random Forest is a well-known ensemble based supervised Machine Learning algorithm. Instead of depending on the result of one decision tree, this classifier takes the outcome from multiple decision trees and uses the majority voting method to predict the final outcome. As the number of trees increases, the accuracy of the result also increases and simultaneously prevents the over-fitting problem [38].

5) Extreme Gradient Boosting(XGBoost): XGBoost is a fast, optimal, ensemble tree-based Machine Learning algorithm. It uses the framework of gradient boosting to solve classification, prediction, regression, and ranking problems. It is highly scalable and robust in handling a variety of distribution, data types, and relationships [39].

B. Graph-based Models

1) *GCNConv*: The concept of 'convolutional' originated from images with fixed structures, but it becomes complex when applied to graphs.

The generic idea of *GCN* is that for every node, feature information of all its neighbour nodes along with itself is collected and applied some aggregate function over that information. Then the obtained values are fed to the neural network. An example of the same is represented in Fig. 6, where for the green node, all its neighbour node information along with itself are collected, and used the *Average Aggregator* function over that information. The obtained values are passed through a neural network and got 2-dimensional vectors as output. *GCNConv* scales the number of graph edges linearly and learns the representation of hidden layers, which encodes both the node features and local graph structure. An example of a 2-layer *GCN* is depicted in Fig. 7, where the output of the first layer is given as input to the second layer.

Let's say for an undirected graph *G* = (*V,E*) having *N* nodes, vertices $v_i \in V$, edges$(v_i, v_j) \in E$, Adjacency Matrix $A \in R^{NXN}$ (binary or weighted), Degree Matrix $D_{ii} = \sum_j A_{ij}$, Feature Vector Matrix $X \in R^{NXC}$ (*N* - Number of nodes, *C* - Number of dimensions of feature vector). As an example, let us consider the graph *G* and its associated adjacent matrix, degree matrix, and feature vector, as shown in Fig. 8. To get each node's feature values of neighbours, multiply Adjacency matrix *A* with the Feature vector *X*. But this calculation missed adding the features of the node itself as it is as vital as its neighbour. So to do that, an identity matrix *I* is added to the adjacency matrix *A* and obtained a new adjacency matrix $\tilde{A}$ as shown in Fig. 9.

To pass the information from neighbours to a specific node, first a new degree matrix $\tilde{D}$ should be computed from $\tilde{A}$, and then the inverse of that $\tilde{D}$ denoted as $\tilde{D}^{-1}$. Finally, multiply $\tilde{D}^{-1}$ with $\tilde{A}X$. To deal with the weighted average, replace $\tilde{D}^{-1}(\tilde{A}X)$ with $(\tilde{D}^{-1}\tilde{A})X$ by following the matrix multiplication's associative property so that $\tilde{D}^{-1}$ becomes the scaling factor of $\tilde{A}$. But this method did scaling of $\tilde{A}_{ij}$ only by D_{ii} ignoring the *j* index, *i.e.,* scaling is done only by rows missing the corresponding columns. A new scaling strategy addresses the above issue by using $\tilde{D}^{-1}\tilde{A}\tilde{D}^{-1}X$ instead of $\tilde{D}^{-1}\tilde{A}X$.

The new scalar strategy gave the weighted average by putting more weight on low-degree nodes and reducing the impact of high-degree nodes on their neighbours.

TABLE II
NOTATIONS USED EQUATION 1

Notation	Description
$\hat{A}$	Scaled Adjacency Matrix (*NXN*)
X	Feature Vector Matrix (*NXC*)
$W^{(0)}$	Trainable Weights (*CXH*)
$W^{(1)}$	Trainable Weights (*HXF*)
$ReLU(\hat{A}XW^{(0)})$	Represents First Layer
N	Number of Nodes
C	Number of Dimensions of Feature Vector
H	Number of Nodes in the Hidden Layer
F	Dimensions of Resulting Vector

Using two scalers, normalization happened twice, one time for the row and another time for the column. So performing rebalancing becomes necessary, and that is done by modifying $\tilde{D}_{ii}\tilde{D}_{jj}$ to $\sqrt{\tilde{D}_{ii}\tilde{D}_{jj}}$, *i.e.,* replace $\tilde{D}^{-1}$ with $\tilde{D}^{-1/2}$ making the formula $\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}X$ and the pictorial representation of rebalancing is as shown in Fig. 10.

In totality, for 2-layer *GCN*, the form of the forward model is as shown in Equation 1, and the notations used in it are described in TABLE 1. The loss function is calculated using the cross-entropy error on all labeled examples.

$$Z = f(X, A) = softmax(\hat{A}ReLU(\hat{A}XW^{(0)})W^{(1)}) \quad (1)$$

The 2-layers of *GCNConv* and *Mean Aggregator* function are used for fine-grained classification of offensive content. Initially, the 768 dimension vectors obtained from the embedding generating algorithm (RoBERTa) are given as input to the first layer of *GCNConv* and obtained 128 dimension vectors as output. Then that 128 dimension vectors are again given as input to the second layer of *GCNConv* and got a 128 dimension vectors as output. The obtained vectors are passed through the softmax function for multi-class classification.

The number of layers here is the farthest distance the node feature can proceed. Fig. 11 represents a sample of the information collecting process with two layers for target node *i*. The process of information gathering happens individualistically for all nodes at once [23],[31].

2) *GraphSAGE*: It is an acronym for Sample and Aggregate. It is a framework for learning representations appropriate for dynamic graphs.

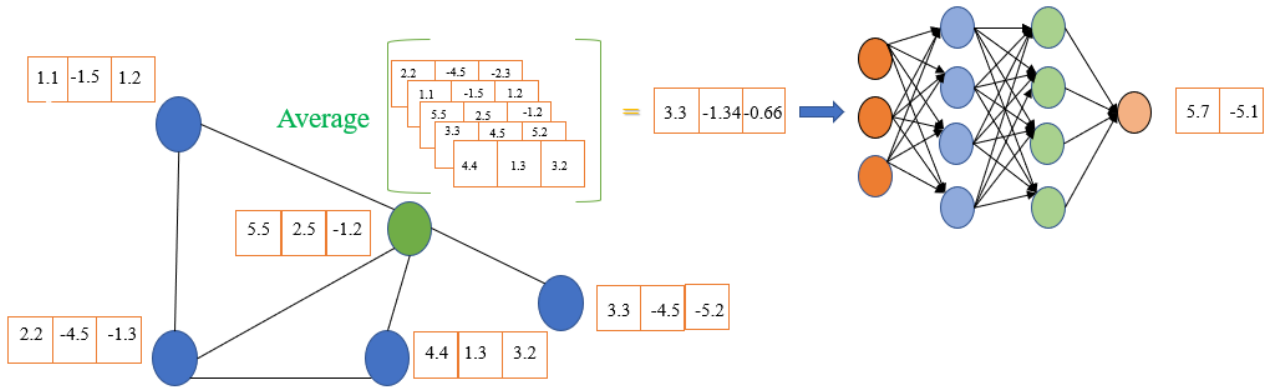


Fig. 6. The Generic Idea of *GCN* Working.

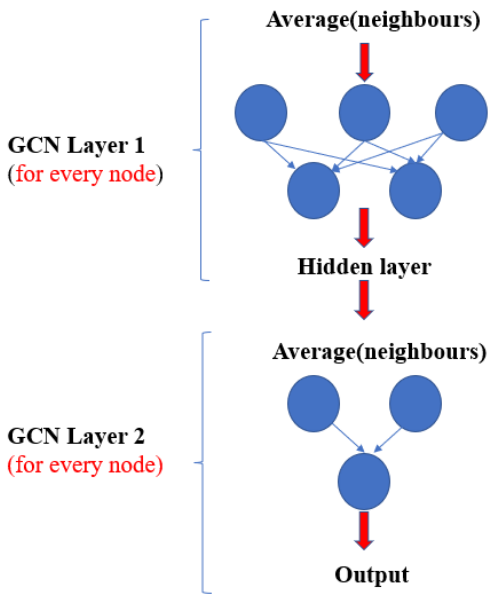


Fig. 7. A Sample of 2-Layer GCN.

It overcame the problem of transductive learning (*GCN*) by learning the aggregator functions that can generate representations of a new node given the node's features and neighbourhood. This method is called inductive learning.

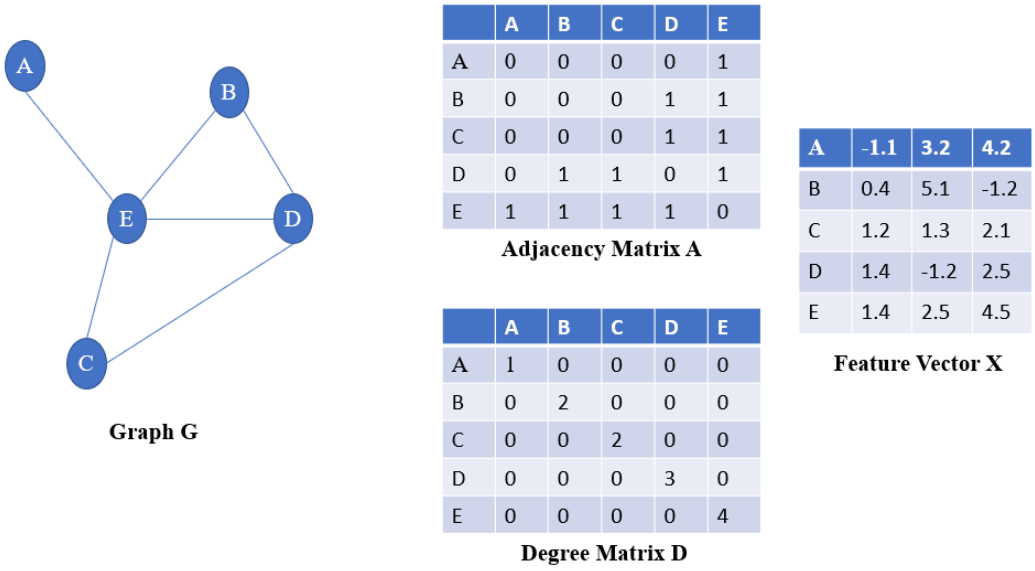
The working principle of *GraphSAGE* is shown in Fig. 12.

The first step of *GraphSAGE* is to sample the neighbourhood nodes for any concerned node. Let's consider the center node colored in "red" in Fig. 11 as our node of interest and take a sample of nodes from 1^{st} hop (represented as k) and 2^{nd} hop neighbours of that node, and that becomes the computation graph which that particular node that we have selected carries and the same is shown in the first step of Fig. 11 by colouring the nodes of computation graph of our node of interest. Similarly, any node chosen will have its computational graph based on the desired number of hops and sample size selected at every hop.

GraphSAGE's second step is to propagate the message that each neighbourhood node has so far. As represented in the second step of Fig. 11, all blue-coloured nodes accept the information from their neighbouring green-coloured nodes. Once all blue nodes have their representation, they propagate them to the red node, and then the red node will generate its final representation. These blue and red nodes have an aggregator function attached, aggregating how to merge the collected information. There are various aggregator functions like *Mean Aggregator Function*, *LSTM Aggregator Function*, *Pooling Aggregator Function*, *etc.,* which are order invariant. For our downstream task of multi-labelled offensive content recognition, we experimented with *Mean* and *Pooling Aggregator* functions. The exact process is repeated for all the nodes of the graph twice and some sample mini batches are used to train the networks.

The third step of *GraphSAGE* starts with the model being trained and having weights freeze. Now one can generate embeddings for a new node that comes in by defining the computation graph for that new node. As the model already has embeddings for the nodes connected to the new node as they were already present during the training process, pass them through the aggregator function of the k^{th} layer to get the representation of the new node.

Getting the embeddings in the first place for the red node depicted in the third step of Fig. 11 can be done in a supervised or unsupervised fashion. The supervised method uses regular cross-entropy loss to perform node classification tasks, which has been used in this work, whereas the



	A	B	C	D	E
A	0	0	0	0	1
B	0	0	0	1	1
C	0	0	0	1	1
D	0	1	1	0	1
E	1	1	1	1	0

Adjacency Matrix A

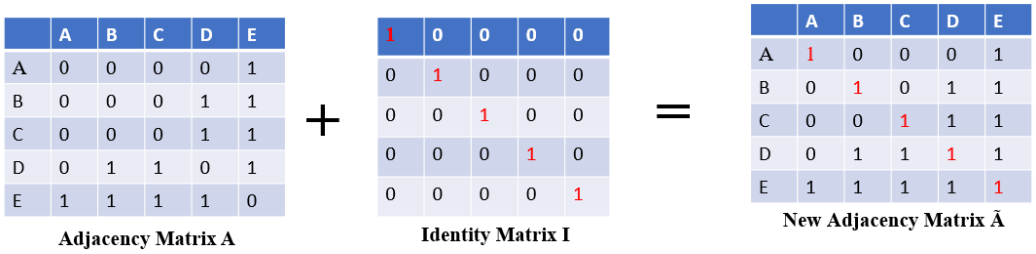
	A	B	C	D	E
A	1	0	0	0	0
B	0	2	0	0	0
C	0	0	2	0	0
D	0	0	0	3	0
E	0	0	0	0	4

Degree Matrix D

A	-1.1	3.2	4.2
B	0.4	5.1	-1.2
C	1.2	1.3	2.1
D	1.4	-1.2	2.5
E	1.4	2.5	4.5

Feature Vector X

Fig. 8. Adjacency Matrix A, Degree Matrix D and Feature Vector X of Graph G.



	A	B	C	D	E
A	0	0	0	0	1
B	0	0	0	1	1
C	0	0	0	1	1
D	0	1	1	0	1
E	1	1	1	1	0

Adjacency Matrix A

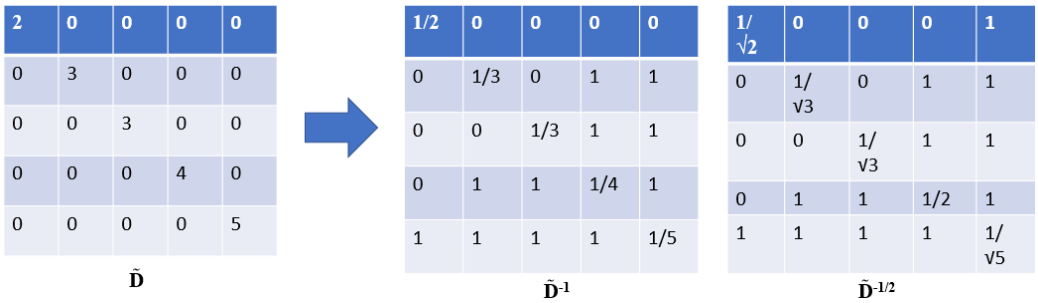
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

Identity Matrix I

	A	B	C	D	E
A	1	0	0	0	1
B	0	1	0	1	1
C	0	0	1	1	1
D	0	1	1	1	1
E	1	1	1	1	1

New Adjacency Matrix $\tilde{A}$

Fig. 9. New Adjacency Matrix Obtained by Adding Self-loop to Every Node



2	0	0	0	0
0	3	0	0	0
0	0	3	0	0
0	0	0	4	0
0	0	0	0	5

$\tilde{D}$

1/2	0	0	0	0
0	1/3	0	1	1
0	0	1/3	1	1
0	1	1	1/4	1
1	1	1	1	1/5

$\tilde{D}^{-1}$

1/√2	0	0	0	1
0	1/√3	0	1	1
0	0	1/√3	1	1
0	1	1	1/2	1
1	1	1	1	1/√5

$\tilde{D}^{-1/2}$

Fig. 10. Pictorial Representation of Rebalancing New Degree Matrix.

unsupervised method uses a property that states if two nodes are neighbours of each other, then in high dimensional space also, they will be close to each other and can optimize on that loss [32], [40].

3) *Graph Attention Network* (*GAT*): *GAT* is a neural network architecture that operates on structured graph data. It is inspired by the work of attention and its success in Natural Language Processing (*NLP*) with recurrent networks. It is an extension of the prior methods based on graph convolution and the related works by incorporating the concepts of self-attention for learning the node embeddings for graph-structured data. It simultaneously addresses numerous challenges of spectral-based *GNN*s and makes the model applicable directly to both inductive and transductive problems. The critical difference between *GCN* and *GAT* are depicted in Fig. 13, where *GCN* exclusively allots non-parametric weight α_{12} through the normalization function during neighbourhood aggregation. But *GAT* implicitly catches the weight α_{12} through the attention mechanism to give higher weights to the more

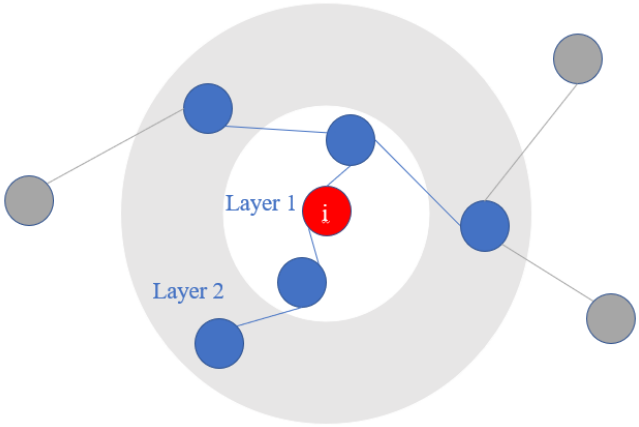


Fig. 11. A Sample of Information Gathering Process with 2-Layers.

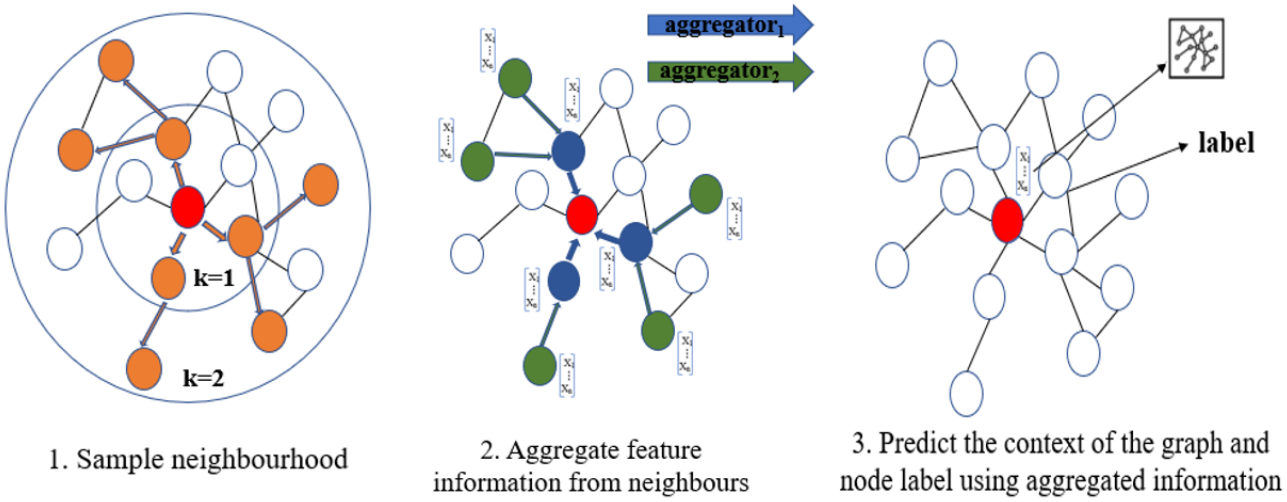


Fig. 12. Working Principle of *GraphSAGE*.

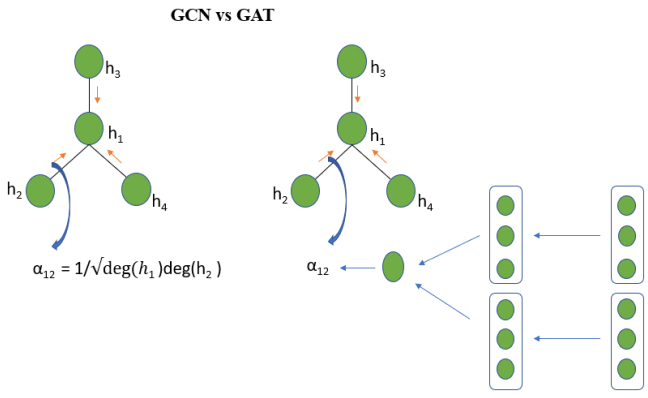


Fig. 13. An Illustration of Key Differences Between *GCN* and *GAT*.

significant nodes during neighbourhood aggregation [41].

GAT uses a sole layer throughout called the graph attentional layer and input to this layer is the set of node features or can also be defined as the initial hidden representation of every node, and this is represented as $h = \{\overrightarrow{h}_1, \overrightarrow{h}_2, ..., \overrightarrow{h}_N\}$ and the dimension for each of them as F. After passing through one hidden layer, the new representation it gets is represented as $h' = \{\overrightarrow{h}'_1, \overrightarrow{h}'_2, ... \overrightarrow{h}'_N\}$ and the transform length for each of the feature representations for every node becomes F' and this F' could be greater or lesser than F depending on whether the network is a wider or a narrow one.

To obtain the attention weight in terms of how much the neighbouring node puts on to the current node, the Equation 2 is performed.

$$e_{ij} = a(W\overrightarrow{h}_i, W\overrightarrow{h}_j) \tag{2}$$

Where, i is considered to be the central node, j to be one of the neighbours and there is an edge between i and j. By default, *GAT* calculates attention weights only if there is an edge between

any two neighbours, and the rest of everything can be assumed to be zero.

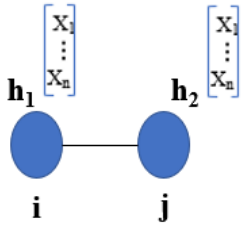


Fig. 14. Example Node Representation

Let the hidden representation of node *i* for the example depicted in Fig. 14 be h_1 and of node *j* be h_2. First, they will be passed through a non-linearity which is $W\overrightarrow{h}_i$ and similarly for the neighbouring node, and that is $W\overrightarrow{h}_j$. This gives F' length representation for every node. Since there is a pair of these embeddings, they are passed through a function called *a* so ended up getting unnormalized attention weight between node *i* and *j*. The softmax function depicted in Equation 3

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k \in N_i} exp(e_{ik})} \quad (3)$$

is applied on the output to make these values comparable, where the numerator has the value between node *i* and *j* and the denominator kind of normalises based on the number of neighbours contained by node *i* and then summation is done over the importance over those neighbours. The α_{ij} values obtained after applying the softmax function are called the attention scores. Attention score implies at what weightage the j^{th} node impacts its representation for getting the final representation of *i*.

The attention mechanism *a* that is used in Equation 2 is nothing but a single-layer feed-forward neural network that takes in a length of $2F'$ as the transformed representation of h_i and h_j, and both are concatenated. The concatenated $(||)$ transformed h_i and h_j ($W\overrightarrow{h}_i$ and $W\overrightarrow{h}_j$) are passed through a linear layer ($\overrightarrow{a}^T$) followed by non-linearity (Leaky ReLU) and then this entire thing is passed through softmax to get the normalized weights and in turn ends up getting attention weights and the same is represented mathematically in Equation 4.

$$\alpha_{ij} = \frac{exp(LeakyReLU(\overrightarrow{a}^T[W\overrightarrow{h}_i || W\overrightarrow{h}_j]))}{\sum_{k \in N_i} exp(LeakyReLU(\overrightarrow{a}^T[W\overrightarrow{h}_i || W\overrightarrow{h}_k]))} \quad (4)$$

Where, $.^T$ represents transposition.

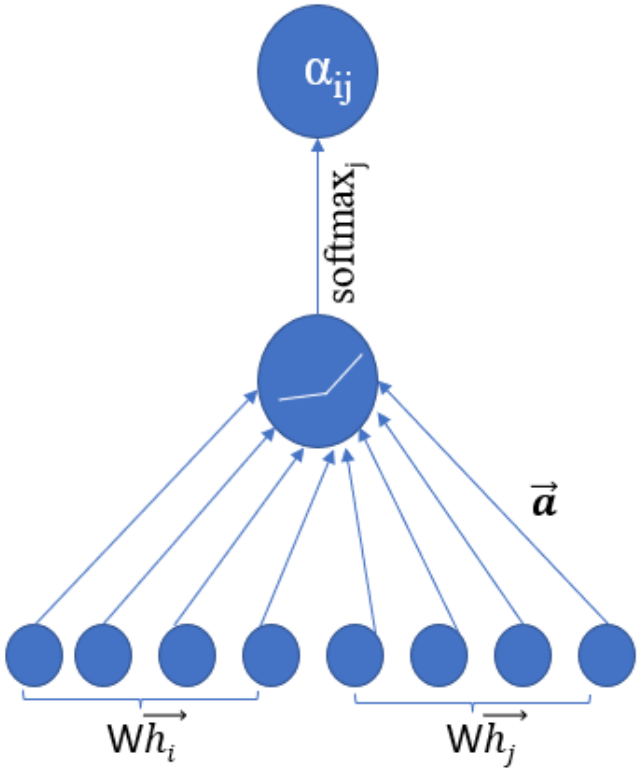


Fig. 15. Pictorial Representation of the Calculation of Attention Weights.

Fig. 15 summarises the overall process of computing attention weight, where $W\overrightarrow{h}_i$ is the transformed representation of center node *i* and $W\overrightarrow{h}_j$ is the transformed representation of a neighbouring node. Concatenate and pass them through a linear layer, have a non-linearity followed by softmax to get attention weights.

$$\overrightarrow{h}'_i = \sigma\left(\sum_{j \in N_i} \alpha_{ij} W\overrightarrow{h}_j\right) \quad (5)$$

The transformed equation for the forward pass is shown in Equation 5, where each of the neighbouring nodes of the center node *i* has some early hidden representation. For those representations, a linear transformation is done, multiplying each of them with their respective attention weights, and then summation is performed, followed by non-linearity [33], [42].

C. Results and Discussion

The experiment is initially accomplished with 47,689 tweets, and the performance of *REGAC* is validated by *RoBERTa* embedding with varied Machine Learning and Graphical Approach Classifiers.

The confusion matrix for the baseline Machine Learning algorithms like *KNN*, *SVM*, *Logistic Regression, Random Forest*, and *XGBoost* with *RoBERTa* are represented in Fig. 16 - Fig. 20, respectively.

The confusion matrices for graphical approach algorithms—*GCNConv*, *GraphSAGE*, and *GAT*—using *RoBERTa* embedding are shown in Fig. 21, Fig. 22, and Fig. 23, respectively. These matrices clearly show religion and age category are the most targeted quality of victims and the rest of the class's resultant metric scores are also significant.

The primary aim of this work is to combat offensive content on social media by categorizing identified bullying content into fine-grained classes—such as age, gender, religion, ethnicity, and others—to better understand the qualities targeted by bullies and take appropriate preventive measures. To achieve this, we experimented with various embedding models in combination with baseline Machine Learning classifiers, and *RoBERTa* embedding generated vectors performed better for our goal of bullying content detection as it is pre-trained particularly on hate speech. So, we used *RoBERTa* embedding technique to generate vectors, and those vectors are given as input to both traditional Machine Learning and Graph-based algorithms. The results of the same are shown in Table III and IV, respectively.

The evaluation metrics that are mainly selected are accuracy and F1-Score. Accuracy is the most intuitive one, and F1-Score is the harmonious mean of precision and recall where precision is the rate at which the classification precisely predicts the appropriate category of the tweet out of all the tweets that it has expected that belongs to this particular category and recall is the rate at which the classifier performs accurate classification of the tweets into its appropriate category when a tweet of that category is given.

Table III clearly shows that *SVM* outperformed other methods across all four metrics. While Random Forest performed well in the precision metric, its performance in the other three metrics was suboptimal. The rest of the algorithms *i.e KNN*, Logistic Regression, and XGBoost, showed average performance with respect to all four metrics.

The confusion matrices of all five Machine Learning algorithms show that religion, age, and ethnicity are the major categories based on which victims are being targeted by bullies. The resulting numbers obtained on the rest of the categories are also no less.

Table IV demonstrates that applying graph-based algorithms significantly improved the efficiency of the textual offensive content detection system across all four metrics: Accuracy, Precision, Recall, and F1-Score. Graph-based algorithms exceed the performance of baseline machine-learning algorithms.

The confusion matrices of graphical approach algorithms (Fig. 18–Fig. 21) indicate that age, religion, and ethnicity are the primary targets of online offenders. But the resulting numbers of other categories show that they are not any less being targeted but comparatively less.

Table V compares various methods on the same dataset to effectively classify offensive content and explicitly shows that the results obtained exceed the performance of most of the previous approaches. From our previous work [26], we observed RoBERTa's ability to generate appropriate embeddings. Similarly, the embeddings generated by SBERT were also quite effective. In fact, in instances where RoBERTa's performance dropped, SBERT performed exceptionally well and the same can be observed in Table VI and Table VII. Therefore, we explored SBERT in combination with three graphical algorithms (GCNConv, GraphSage, and GAT), and the results are listed in the Table V. Although these combinations performed well, they did not surpass the performance of RoBERTa embeddings when combined with graph-based algorithms.

All three variations of our proposed approach *REGAC i.e., RoBERTa + GCN*, *RoBERTa + GraphSAGE* and *RoBERTa + GAT* measured the efficiency of the model from all four metrics perspective *viz.,* Accuracy, Precision, Recall, and F1-Score unlike [3], [34], and [43].

In totality, *RoBERTa*, in combination with graph-based algorithms, particularly *RoBERTa + GCNConv*, performs best concerning all metrics calculated in the fine-grained classification of bullying content of toxic content detection system. But the training time of *SageConv* was comparatively very less as it took a sample of nodes instead of all nodes, unlike *GCNConv*. Even its results in comparison with *GCNConv* are less in negligible amount, it was a minute tradeoff of impact for saving time and computation spent on training. Expectations for *GATConv* were high compared to the other two convolution layers

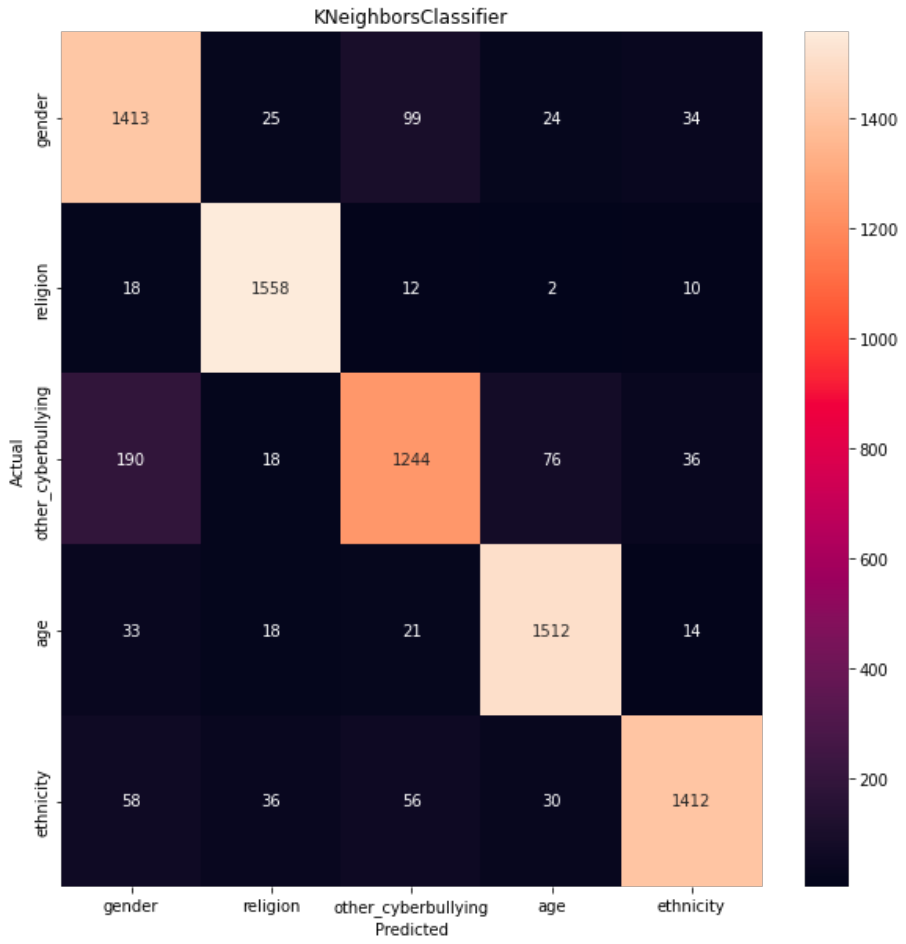


Fig. 16. Confusion Matrix for *RoBERTa + KNN*.

TABLE III
RESULTS OF *RoBERTa* EMBEDDING WITH VARIED MACHINE LEARNING CLASSIFIERS ON WANG [24] DATASET

Machine Learning Model	Accuracy	Recall	Precision	F1-Score
KNN	0.891807	0.827063	0.893197	0.858832
SVM	0.922252	0.830668	0.901541	0.864629
Logistic Regression	0.922755	0.772655	0.897697	0.830462
Random Forest	0.813084	0.182502	0.974286	0.307351
XGBoost	0.892782	0.778615	0.893913	0.832256

TABLE IV
RESULTS OF *RoBERTa* EMBEDDING WITH VARIED GRAPHICAL APPROACH CLASSIFIERS ON WANG [24] DATASET

Graphical Approach Classifier	Accuracy	Recall	Precision	F1-Score
GCNConv	0.9350	0.9350	0.9350	0.9350
GraphSAGE + Mean	0.9326	0.9325	0.9329	0.9327
GraphSAGE + Pooling	0.9285	0.9285	0.9304	0.9287
GAT	0.9265	0.9265	0.9304	0.9273

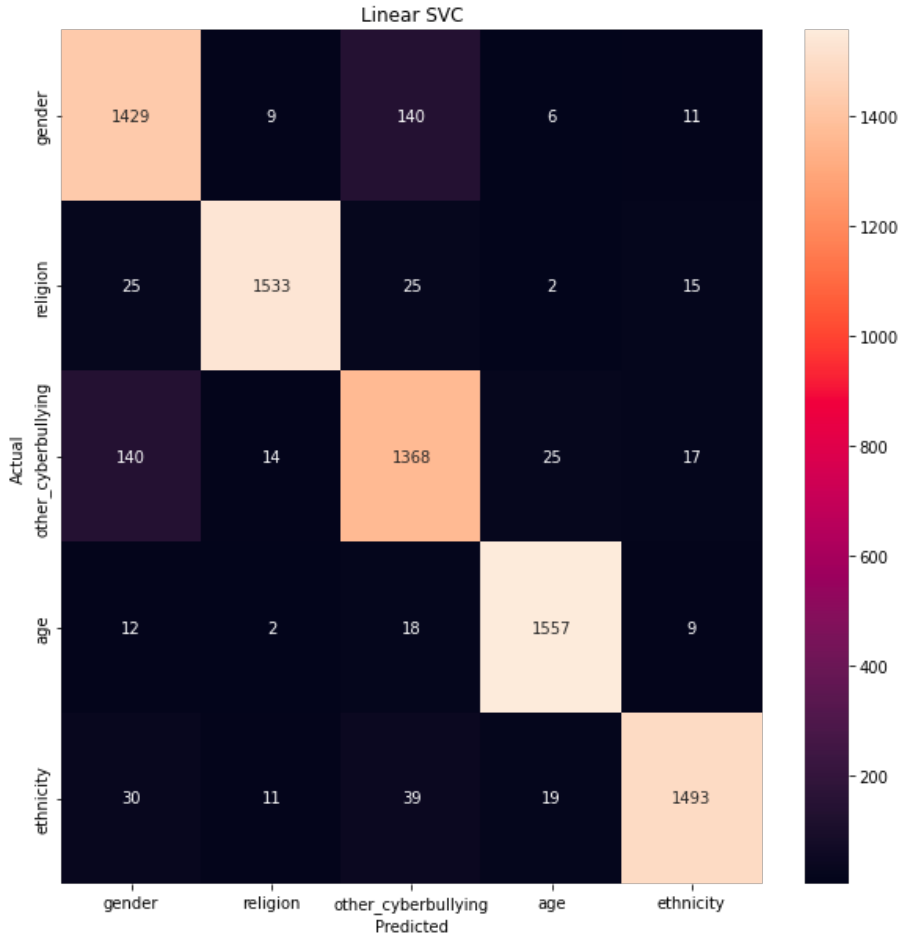


Fig. 17. Confusion Matrix for *RoBERTa + SVM*.

used here as it is capable enough to implicitly capture the weights of all edge connecting nodes so that during the aggregation process, more significant nodes are given higher importance, in turn providing an efficient classification of nodes. It might be because of overfitting or the kind of data used here that the results of *GAT* are comparatively less.

In order to reconfirm the efficiency of the proposed REGAC model, we experimented with another dataset provided by [25]. The results of all three graph-based approaches on both the datasets utilized and their comparisons with the state-of-the-art methods are listed in Table V. Since the number of data instances are relatively more in the [25] given dataset , GAT performs relatively better in comparison to GraphSAGE, but GCNConv still works the best even with this dataset. With this we evaluated our models with both a balanced and an unbalanced dataset

VI Conclusions

Widespread internet access has led to an increase in offensive online behaviour. As smartphones become more affordable with technological advancements, some individuals misuse them for harmful purposes. Social media has become a tool for some to harass others online. In extreme cases, this harassment leads to severe mental health issues such as depression and anxiety, significantly disrupting victims' social and personal lives. Addressing this issue is crucial to creating a healthy, safe, and inclusive online environment, which is essential for fostering a healthier society.

This work aims to identify objectionable online textual content and determine the qualities most frequently targeted by online bullies. Understanding these patterns will enable the development of more effective solutions to combat online offensive behaviour. To root out offensive online content, we first used different embedding techniques to make the model better understand the exact context and content of the post and found *RoBERTa* works best for our purpose. Then various machine learning algorithms are fed with these generated vectors to classify

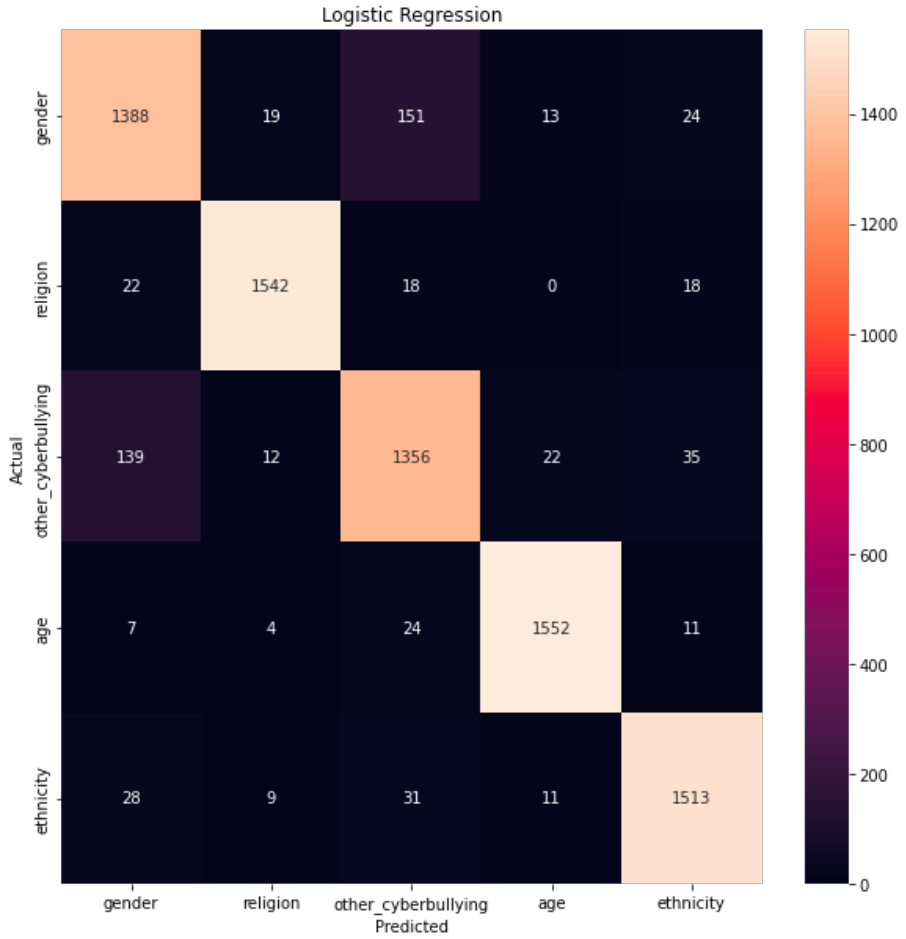


Fig. 18. Confusion Matrix for *RoBERTa* + Logistic regression.

them into the multi-class category. Additionally, three graph-based algorithms—*GCNConv*, *GraphSAGE*, and *GAT*—are used to enhance the classification of online offensive content into fine-grained categories and identify the specific qualities targeted by offenders. The proposed model is again evaluated on the dataset provided by [25], where it demonstrated its efficiency in recognizing the type of offensive behaviour.

This work put forward a step not only to identify the online offensive content posted on social media but to understand what qualities the bullies generally target in their victims. This, we believe, will give a boost in combating online bullying behaviour and create a safer online environment. This study focused on three specific convolution layers. Future work could explore additional convolution layers to assess their effectiveness in representing nodes and improving multi-class classification.

References

[1] Sneha Chinivar and Roopa M S and Arunalatha J S and Venugopal K R, "Online Offensive Behaviour in Social-media: Detection Approaches, Comprehensive Review and Future Directions," *Entertainment Computing*, p. 100544, 2022.

[2] S. Arcidiacono, "Why Graph Theory Is Cooler Than You Thought," https://www.topbots.com/why-graph-theory-cooler-than-you-thought/, 2021.

[3] J. Wang, K. Fu, and C.-T. Lu, "Sosnet: A Graph Convolutional Network Approach to Fine-grained Cyberbullying Detection," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 1699–1708.

[4] "Aggression Detection in Social Media from Textual Data Using Deep Learning Models."

[5] B. A. H. Murshed, J. Abawajy, S. Mallappa, M. A. N. Saif, and H. D. E. Al-Ariki, "DEA-RNN: A Hybrid Deep Learning Approach for Cyberbullying Detection in Twitter Social Media Platform," *IEEE Access*, vol. 10, pp. 25 857–25 871, 2022.

[6] O. E. Ojo, T. H. Ta, A. Gelbukh, H. Calvo, G. Sidorov, O. O. Adebanji, and L. Dong, "Automatic Hate Speech Detection using Deep Neural Networks and Word Embedding," *Computacion y Sistemas*, vol. 26, no. 2, pp. 1007–1013, 2022.

[7] Z. Miao, X. Chen, H. Wang, R. Tang, Z. Yang, and W. Tang, "Detecting Offensive Language on Social Networks: An End-to-end Detection Method based on Graph Attention Networks," *arXiv preprint arXiv:2203.02123*, 2022.

[8] S. Abarna, J. Sheeba, S. Jayasrilakshmi, and S. P. Devaneyan, "Identification of Cyber Harassment and Intention of Target Users on Social Media Platforms," *Engineering applications of artificial intelligence*, vol. 115, p. 105283, 2022.

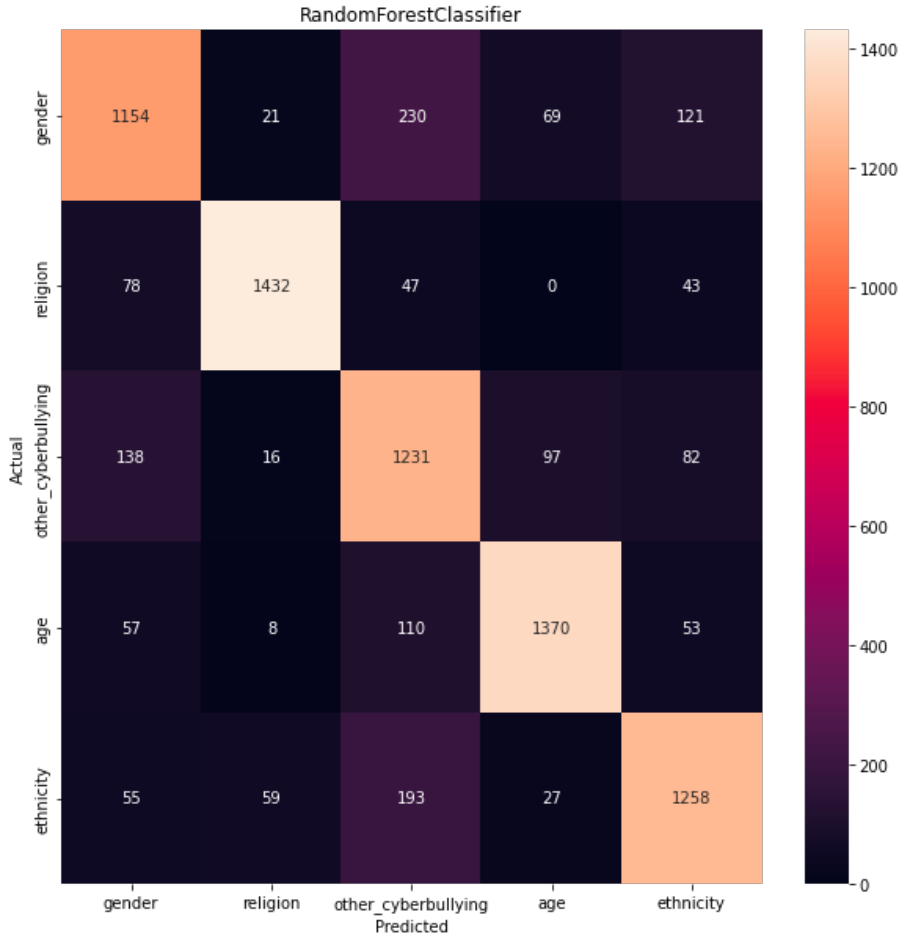


Fig. 19. Confusion Matrix for *RoBERTa* + Random Forest.

[9] N. A. Azeez, S. O. Idiakose, C. J. Onyema, and C. Van Der Vyver, "Cyberbullying Detection in Social Networks: Artificial Intelligence Approach," *Journal of Cyber Security and Mobility*, pp. 745–774, 2021.

[10] Y. Fang, S. Yang, B. Zhao, and C. Huang, "Cyberbullying Detection in Social Networks using Bi-gru with Self-Attention Mechanism," *Information*, vol. 12, no. 4, p. 171, 2021.

[11] T. Ige and S. Adewale, "AI Powered Anti-cyber Bullying System using Machine Learning Algorithm of Multinomial Naive Bayes and Optimized Linear Support Vector Machine," *arXiv preprint arXiv:2207.11897*, 2022.

[12] M. AGBAJE and O. Afolabi, "Neural Network-Based Cyber-Bullying and Cyber-Aggression Detection Using Twitter Text," 2022.

[13] D. Chatzakou, I. Leontiadis, J. Blackburn, E. D. Cristofaro, G. Stringhini, A. Vakali, and N. Kourtellis, "Detecting Cyberbullying and Cyberaggression in Social Media," *ACM Transactions on the Web (TWEB)*, vol. 13, no. 3, pp. 1–51, 2019.

[14] S. Agrawal and A. Awekar, "Deep Learning for Detecting Cyberbullying across Multiple Social Media Platforms," in *European Conference on Information Retrieval*. Springer, 2018, pp. 141–153.

[15] Z. Zhang, D. Robinson, and J. Tepper, "Detecting Hate Speech on Twitter using a Convolution-GRU based Deep Neural Network," in *European Semantic Web Conference*. Springer, 2018, pp. 745–760.

[16] N. Rezvani and A. Beheshti, "Towards Attention-Based Context-Boosted Cyberbullying Detection in Social Media," *Journal of Data Intelligence*, vol. 2, no. 4, pp. 418–433, 2021.

[17] R. Song, F. Giunchiglia, Q. Shen, N. Li, and H. Xu, "Improving abusive language detection with online interaction network," *Information Processing & Management*, vol. 59, no. 5, p. 103009, 2022.

[18] N. Cécillon, V. Labatut, R. Dufour, and G. Linares, "Graph embeddings for abusive language detection," *SN Computer Science*, vol. 2, pp. 1–15, 2021.

[19] P. Mishra, M. Del Tredici, H. Yannakoudakis, and E. Shutova, "Abusive language detection with graph convolutional networks," *arXiv preprint arXiv:1904.04073*, 2019.

[20] T. Ahmed, S. Ivan, M. Kabir, H. Mahmud, and K. Hasan, "Performance analysis of transformer-based architectures and their ensembles to detect trait-based cyberbullying," *Social Network Analysis and Mining*, vol. 12, no. 1, p. 99, 2022.

[21] K. Maity, T. Sen, S. Saha, and P. Bhattacharyya, "Mtbullygnn: A graph neural network-based multitask framework for cyberbullying detection," *IEEE Transactions on Computational Social Systems*, 2022.

[22] P. Shashank Gupta, "Word Embeddinds," https:https://www.kdnuggets.com/2019/02/word-embeddings-nlp-applications.html#:~:text=Word%20embeddings%20are%20basically%20a,for%20solving%20most%20NLP%20problems., 2022.

[23] C. Pham, "Graph Convolutional Networks (GCN)," https://www.topbots.com/graph-convolutional-networks/, 2020.

[24] L. Wang, Fu, "Dataset," https://drive.google.com/drive/folders/1oB2fan6GVGG83Eog66Ad4wK2ZoOjwu3F?usp=sharing, 2020.

[25] S. Salawu, J. Lumsden, and Y. He, "A large-scale english

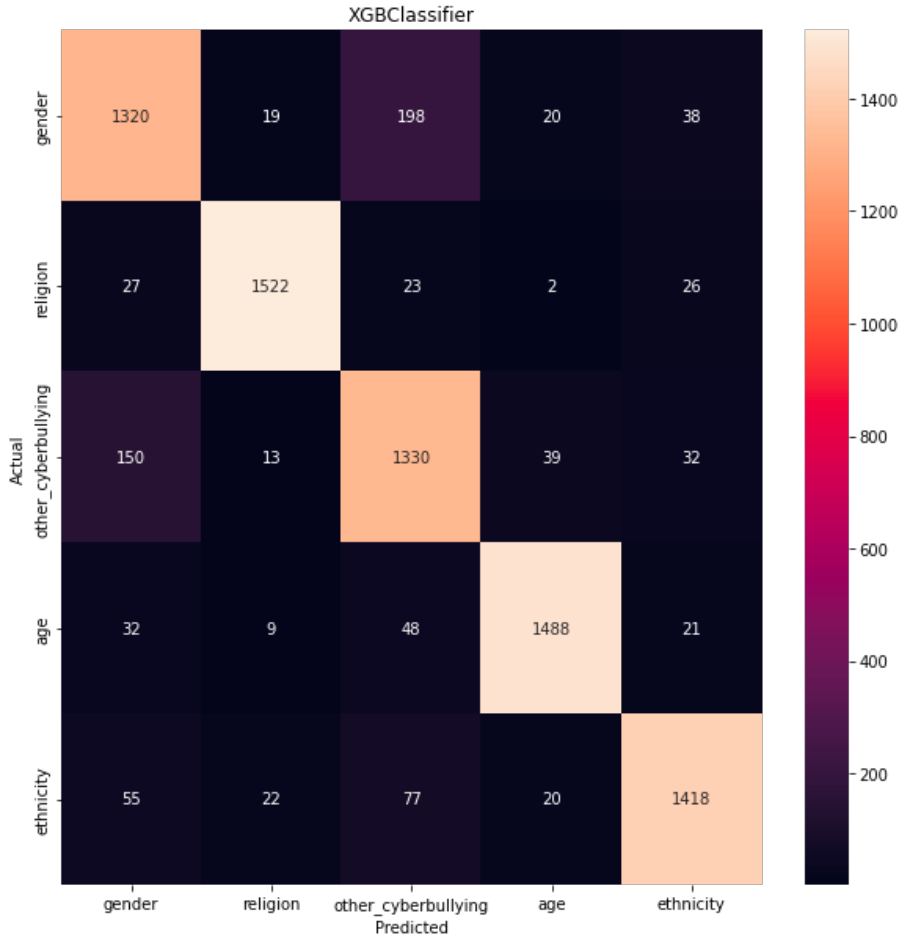


Fig. 20. Confusion Matrix for *RoBERTa* + XGBoost.

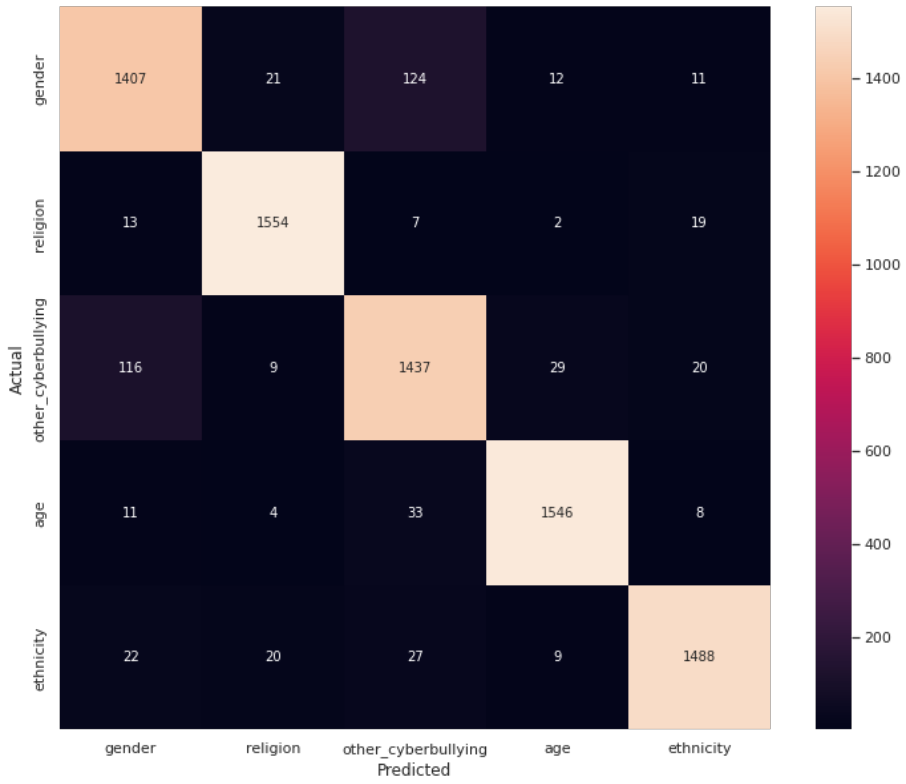


Fig. 21. Confusion Matrix for *RoBERTa* + *GCNConv*.

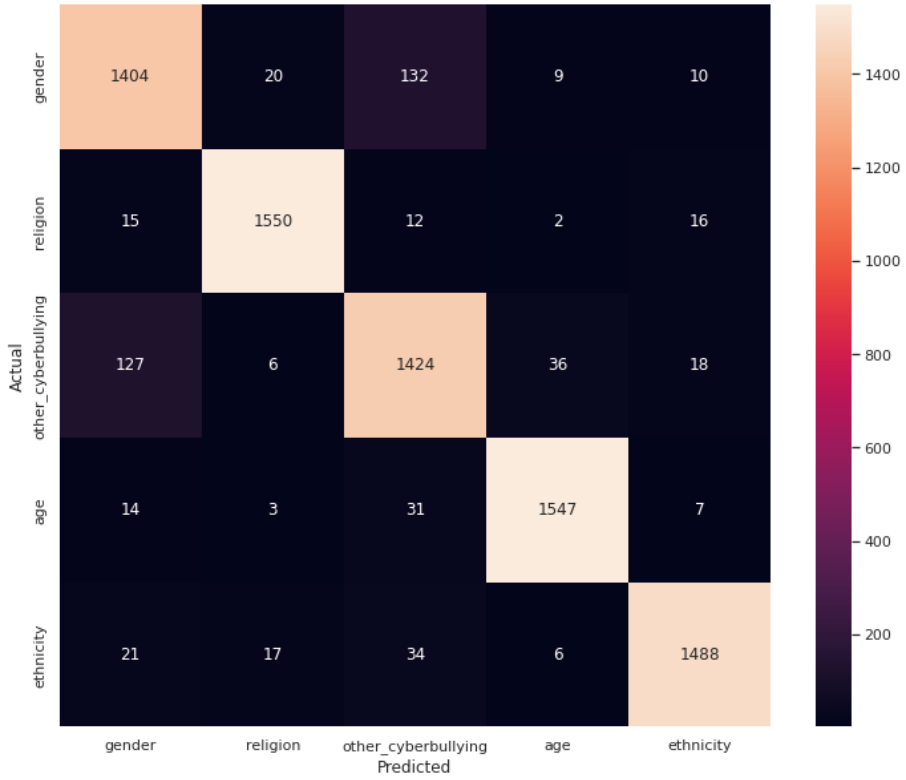


Fig. 22. Confusion Matrix for *RoBERTa + GraphSAGE*.

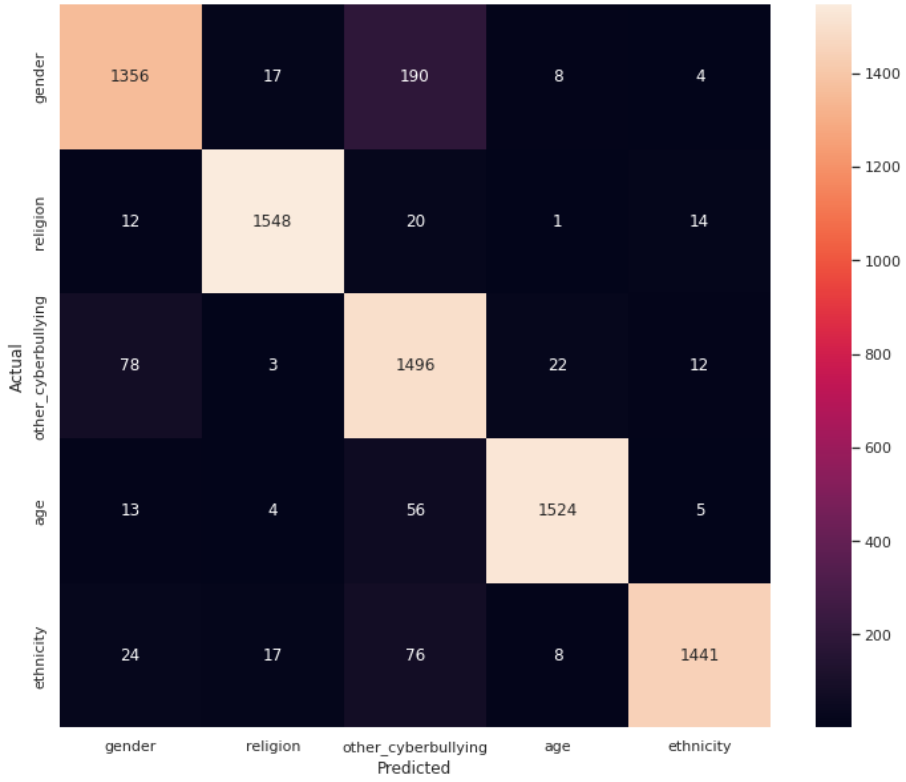


Fig. 23. Confusion Matrix for *RoBERTa + GAT*.

TABLE V
COMPARISON OF VARIOUS APPROACHES IN EFFECTIVE OFFENSIVE CONTENT CLASSIFICATION

Dataset	Model	Accuracy	F1-Score	Precision	Recall
Wang [24]	***REGAC (RoBERTa + GCNConv)***	**0.9350**	**0.9350**	**0.9350**	**0.9350**
	REGAC (RoBERTa + GraphSAGE)	**0.9326**	**0.9327**	**0.9329**	**0.9325**
	REGAC (RoBERTa + GAT)	**0.9265**	**0.9273**	**0.9304**	**0.9265**
	SBERT + GraphSAGE	0.9268	0.9256	0.9258	0.9254
	SBERT + GAT	0.9135	0.9128	0.9160	0.9100
	SBERT + GCNConv	0.9290	0.9290	0.9290	0.9290
	SBERT + SOSNet [3]	0.9270	0.9258	-	-
	Res-CNN-BiLSTM [34]	-	0.9200	0.9200	0.9200
	Probabilty Averaging(Ensemble) [44]	0.9076	0.9065	0.9057	0.9076
	LightGBM [45]	0.8550	0.8449	0.8400	0.8500
	GRU [46]	0.9200	0.9200	0.9200	0.9200
	CNN [43]	0.8310	-	-	-
	Stacking Classifier [47]	0.9265	0.9273	0.9304	0.9265
[25]	***REGAC (RoBERTa + GCNConv)***	**0.8290**	**0.8290**	**0.8290**	**0.8290**
	REGAC (RoBERTa + GraphSAGE)	**0.8136**	**0.8137**	**0.8139**	**0.8135**
	REGAC (RoBERTa + GAT)	**0.8146**	**0.8146**	**0.8146**	**0.8146**
	SBERT + GraphSAGE	0.8098	0.8104	0.8108	0.8106
	SBERT + GAT	0.8110	0.8112	0.8115	0.8113
	SBERT + GCNConv	0.8102	0.8113	0.8115	0.8108
	[25]	0.5834	0.8081	-	-

TABLE VI
ACCURACY RESULT OF VARIOUS EMBEDDING MODELS WITH VARIED MACHINE LEARNING ALGORITHMS ON WANG [24] DATASET [26]

Embedding Technique	KNN	SVM	Logistic Regression	XGBoost
Word2Vec	0.767	0.921	0.880	0.921
GloVe	0.772	0.922	0.880	0.916
FastText	0.703	0.907	0.867	0.900
BERT	0.775	0.897	0.879	0.878
SBERT	0.855	0.926	0.898	0.915
DistilBERT	0.831	0.918	0.903	0.905
RoBERTa	**0.891**	**0.922**	**0.922**	**0.892**

multi-label twitter dataset for cyberbullying and online abuse detection," in *The 5th Workshop on Online Abuse and Harms*. Association for Computational Linguistics, 2021, pp. 146–156.

[26] S. Chinivar, M. Roopa, J. Arunalatha, and K. Venugopal, "Comparision of varied embedding and machine learning classifiers for fine grained offensive content identification," in *2022 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*. IEEE, 2022, pp. 1–6.

[27] GeeksforGeeks, "RoBERTa," https://www.geeksforgeeks.org/overview-of-roberta-model/, Jun. 2022.

[28] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Huggingface's Transformers: State-of-the-art Natural Language Processing," *arXiv preprint arXiv:1910.03771*, 2019.

[29] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A Robustly Optimized BERT Pretraining Approach," *arXiv preprint arXiv:1907.11692*, 2019.

[30] CarloLucibello, "Convolutional Layers," https://carlolucibello.github.io/GraphNeuralNetworks.jl/dev/api/conv/.

[31] T. N. Kipf and M. Welling, "Semi-supervised Classification with Graph Convolutional Networks," *arXiv preprint arXiv:1609.02907*, 2016.

[32] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[33] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph Attention Networks," *arXiv preprint arXiv:1710.10903*, 2017.

[34] R. Joshi, A. Gupta, and N. Kanvinde, "Res-CNN-BiLSTM Network for overcoming Mental Health Disturbances caused due to Cyberbullying through Social Media," *arXiv preprint arXiv:2204.09738*, 2022.

[35] Javatpoint, "KNN," https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning.

[36] Javatpoint, "SVM," https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm.

[37] Javatpoint, "Logistic Regression," https://www.javatpoint.com/logistic-regression-in-machine-learning.

[38] Javatpoint, "Random Forest," https://www.javatpoint.com/machine-learning-random-forest-algorithm.

[39] K. Ghatak, "XGBoost," https://www.naukri.com/learning/articles/xgboost-algorithm-in-machine-learning/, Jun 2022.

[40] R. Özçelik, "An Intuitive Explanation of GraphSAGE," https://towardsdatascience.com/an-intuitive-explanation-of-graphsage-6df9437ee64f, 2019.

[41] Yugesh, "All You Need to Know About Graph Attention Networks," https://analyticsindiamag.com/all-you-need-to-know-about-graph-attention-networks/, May 2022.

[42] T. T. D. S. Guy, "GAT: Graph Attention Networks (Graph

TABLE VII
F1-SCORE RESULT OF VARIOUS EMBEDDING MODELS WITH VARIED MACHINE LEARNING ALGORITHMS ON WANG [24] DATASET [26]

Embedding Technique	KNN	SVM	Logistic Regression	XGBoost
Word2Vec	0.732	0.921	0.880	0.921
GloVe	0.742	0.923	0.879	0.917
FastText	0.679	0.908	0.867	0.900
BERT	0.774	0.898	0.879	0.879
SBERT	0.848	0.927	0.898	0.915
DistilBERT	0.830	0.919	0.903	0.906
RoBERTa	**0.858**	**0.864**	**0.830**	**0.832**

ML Research Paper Walkthrough)),” https://www.youtube.com/watch?v=v2P1yZhP8cs.

[43] V. A. Joseph, B. R. Prathap, and K. P. Kumar, “Detecting cyberbullying in twitter: A multi-model approach,” in *2024 4th International Conference on Data Engineering and Communication Systems (ICDECS)*. IEEE, 2024, pp. 1–6.

[44] T. Ahmed, M. Kabir, S. Ivan, H. Mahmud, and K. Hasan, “Am i being bullied on social media? an ensemble approach to categorize cyberbullying,” in *2021 IEEE international conference on big data (Big data)*. IEEE, 2021, pp. 2442–2453.

[45] M. I. Mahmud, M. Mamun, and A. Abdelgawad, “A deep analysis of textual features based cyberbullying detection using machine learning,” in *2022 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*. IEEE, 2022, pp. 166–170.

[46] N. K. Singh, P. Singh, and S. Chand, “Deep learning based methods for cyberbullying detection on social media,” in *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. IEEE, 2022, pp. 521–525.

[47] A. F. Alqahtani and M. Ilyas, “An ensemble-based multi-classification machine learning classifiers approach to detect multiple classes of cyberbullying,” *Machine Learning and Knowledge Extraction*, vol. 6, no. 1, pp. 156–170, 2024.