ORIGINAL RESEARCH ARTICLE UHD JOURNAL OF SCIENCE AND TECHNOLOGY

# Comparative Analysis of Word Embeddings for Multiclass Cyberbullying Detection

**Azhi Faraj[1,2] and Semih Utku[1]**

*[1]Department of Computer Engineering, Faculty of Engineering, Dokuz Eylul University, Izmir, Turkey, [2]Department of Information Technology, College of Commerce, Sulaimani University, Sulaymaniyah, Iraq.*

## ABSTRACT

Cyberbullying has emerged as a pervasive concern in modern society, particularly within social media platforms. This phenomenon encompasses employing digital communication to instill fear, threaten, harass, or harm individuals. Given the prevalence of social media in our lives, there is an escalating need for effective methods to detect and combat cyberbullying. This paper aims to explore the utilization of word embeddings and to discern the comparative effectiveness of trainable word embeddings, pre-trained word embeddings, and fine-tuned language models in multiclass cyberbullying detection. Distinguishing from previous binary classification methods, our research delves into nuanced multiclass detection. The exploration of word embeddings holds significant promise due to its ability to transform words into dense numerical vectors within a high-dimensional space. This transformation captures intricate semantic and syntactic relationships inherent in language, enabling machine learning (ML) algorithms to discern patterns that might signify cyberbullying. In contrast to previous research, this work delves beyond primary binary classification and centers on the nuanced realm of multiclass cyberbullying detection. The research employs diverse techniques, including convolutional neural networks and bidirectional long short-term memory, alongside well-known pre-trained models such as word2vec and bidirectional encoder representations from transformers (BERT). Moreover, traditional ML algorithms such as K-nearest neighbors, Random Forest, and Naïve Bayes are integrated to evaluate their performance vis-à-vis deep learning models. The findings underscore the promise of a fine-tuned BERT model on our dataset, yielding the most promising results in multiclass cyberbullying detection, and achieving the best-recorded accuracy of 85% on the dataset.

**Index Terms:** Cyberbullying Detection, Word Embeddings, Deep Learning, Machine Learning, Text Classification

## 1. INTRODUCTION

Cyberbullying involves the consistent use of electronic or digital platforms by individuals or collectives to repeatedly convey harmful or aggressive messages with the intent of causing discomfort or harm to others [1]. This encompasses actions such as sending cruel or menacing messages, sharing humiliating photos or videos, propagating gossip, or crafting fictitious profiles or websites with the purpose of shaming an individual. Cyberbullying can lead to significant repercussions for both the target and the person engaging in it. Victims of cyberbullying may experience a range of negative effects, including feelings of sadness, anxiety, depression, and even thoughts of suicide. They may also have trouble sleeping, lose interest in activities they used to enjoy, and struggle with self-esteem [2]. Individuals engaging in cyberbullying might face serious repercussions, including disciplinary action from educational institutions or legal authorities, as well as potential legal consequences. Such actions could also negatively impact their future relationships and life opportunities. It is crucial to understand that cyberbullying is not confined to a specific age group or demographic; it can affect anyone. Although young people are commonly involved, adults are not immune to being targeted by cyberbullying [3].

Access this article online

**DOI:** 10.21928/uhdjst.v8n1y2024.pp55-63 **E-ISSN:** 2521-4217 **P-ISSN:** 2521-4209



**Corresponding author's e-mail:** azhi.faraj@univsul.edu.iq

Received: 19-11-2023 Accepted: 06-02-2024 Published: 20-02-2024

Automatic cyberbullying detection refers to the use of computational methods and technologies to automatically identify instances of cyberbullying in electronic communications, such as text, images, and videos.

This typically involves the utilization of natural language processing (NLP) techniques, machine learning (ML) algorithms, and other analytical approaches to analyze the content and context of electronic communications [4]. Different techniques have been put forth for automated cyberbullying identification, encompassing rule-based models, conventional ML models, and deep learning models [5]. Rule-based models depend on pre-established rule sets and lexical resources to detect instances of cyberbullying. On the other hand, conventional ML models employ algorithms such as logistic regression, decision trees, K-nearest Neighbors (KNN), and support vector machines to categorize text as either cyberbullying or not. Deep learning models, conversely, use neural networks to identify patterns in the data that indicate cyberbullying. Research on automatic cyberbullying detection has mainly focused on social media platforms such as Twitter, Instagram, and YouTube [5]. These platforms are particularly relevant for cyberbullying detection due to their popularity among young people and the large amount of user-generated content that is available for analysis.

Research in the literature has used the following features to improve the accuracy of detection [6, p. 2].

### 1.1. Textual Features

This includes techniques such as n-grams, skip grams, content character length, number of emoticons used, and the usage of profanity in the text.

### 1.2. Social Features

Refers to the attributes provided in social networks for example number of friends or followers, reactions received from posts, and additional metrics of centrality that may be derived from a graph representation (e.g., betweenness and eigenvector).

### 1.3. User Features

Information about the authors such as age, gender, ethnicity, and religion.

### 1.4. Sentiment Analysis

The polarity of the post is used to determine whether a post is cyberbullying or not.

Cyberbullying detection using word embeddings is a subject of active research in NLP [6]. Word embeddings are a technique used to depict words within a vector space with multiple dimensions. Each dimension corresponds to a distinct semantic or syntactic aspect of the word. By mapping words to vectors, it becomes possible to use ML algorithms to automatically detect patterns in the text that may indicate bullying [7].

This paper explores the use of word embeddings for multiclass cyberbullying detection. It addresses the growing issue of cyberbullying in today's society and the need for effective methods to detect and prevent it. The study compares the performance of different word embedding techniques, including trainable word embeddings, pre-trained word embeddings, and fine-tuning language models. It goes beyond simple binary classification and focuses on multi-class detection of cyberbullying. The research employs various techniques, such as convolutional neural networks (CNN) and bidirectional long short-term memory (biLSTM), as well as popular pre-trained models such as word2vec and bidirectional encoder representations from transformers (BERT). The research contributes to the field of automatic cyberbullying detection and provides insights into the effectiveness of word embeddings for this task that follow.

This research is motivated by the urgent need to address the escalating issue of cyberbullying in the digital age, where social media platforms are intertwined with daily life. Recognizing the limitations of existing binary classification approaches in accurately detecting diverse forms of cyberbullying, this study introduces an advanced methodology that employs a combination of word embeddings and deep learning techniques. The primary contributions include the development of a multiclass detection model, which significantly improves the accuracy of cyberbullying identification compared to traditional methods. This is achieved by leveraging a fine-tuned BERT model alongside other innovative techniques such as CNN and biLSTM. The real-world impact of this work is substantial, offering tools for social media platforms and digital communities to better identify and mitigate cyberbullying, thus fostering safer online environments. Our approach not only enhances cyberbullying detection accuracy but also contributes to the broader field of digital safety and online behavioral analysis. The proposed method is particularly beneficial in social media platforms for monitoring content, in educational settings to safeguard students, and on online forums for community guideline enforcement. It also holds potential for mental health support initiatives, offering insights for targeted interventions, and can assist law enforcement in identifying and addressing severe cases. This technology aims to enhance online safety

and respect across various digital platforms, contributing to a healthier digital environment.

## 2. RELATED WORK

In their research, Yin *et al.* [8] have the first to explore automated recognition of cyberbullying in online environments. They collected three sets of data from three distinct online platforms to detect instances of harassment. One dataset was sourced from the Kongregate platform, whereas the other two were collected from Reddit. To classify the data, the authors used a linear kernel classification model and applied different techniques for feature extraction, including N-grams and term frequency-inverse document frequency (TF-IDF). Despite having uncertain experimental results, the study was a significant step toward further research in this field.

Researchers in Iwendi *et al.* [9] explored the effectiveness of four deep learning models – BLSTM, gated recurrent units (GRU), long short-term memory (LSTM), and recurrent neural network (RNN) – in detecting cyberbullying. They applied rigorous data pre-processing steps, including text cleaning, tokenization, stemming, lemmatization, and removal of stop words, before feeding the data into the models for prediction. Among these models, BLSTM stood out, showing higher accuracy and F1-measure scores compared to RNN, LSTM, and GRU [10] proposed a ML approach for detecting cyberbullying in text-based online communication. The authors collected a large dataset of messages from social networking sites and labeled them as either bullying or non-bullying. Subsequently, they derived a range of linguistic attributes from the messages, encompassing syntactic and semantic aspects, alongside social network characteristics. The authors employed various ML techniques, such as decision trees and support vector machines (SVMs), to categorize messages as either exhibiting bullying behavior or not.

Bozyiğit *et al.* [11] collected a dataset from twitter for Turkish language and used social features to detect cyberbullying, the authors also created a web application that detects cyberbullying live. In Aizawa [12], an approach based on feature engineering was introduced for identifying multi-class cyberbullying on Twitter. The authors utilized a dataset containing 10,000 tweets that were labeled with three levels of cyberbullying severity: low, medium, and high. Initially, they employed the synthetic minority oversampling technique to oversample the underrepresented categories (low, medium, and high) by a factor of 300. Subsequently, they applied a weighted cost for misclassification in the minority categories.

On the other hand, the study outlined in Dinakar *et al.* [13] utilized deep learning architectures to investigate the performance of various deep learning algorithms (LSTM, BiLSTM, RNN, and GRU) in terms of their effectiveness for identifying antisocial behavior. The researchers performed empirical assessments to measure how well the algorithms work in recognizing antisocial behavior. [14] utilized a dataset collected from twitter with 16K records, they applied TF-IDF, task-specific embeddings to detect hate speech that was categorized to sexist, racist, or none comments, the authors reported an F1-score of 0.93.

In Agrawal and Awekar [15], the authors used YouTube comments data that they used TF-IDF and a collection of profane words then applied Naïve bayes, SVM, J48, and JRip to detect three types of cyberbullying, namely, sexuality, race, and intelligence that they achieved a maximum accuracy of 80.20% on sexuality cyberbullying. [16] used char-n grams, word n-grams as features on Wikipedia dataset that they applied Logistic Regression and Multilayer Perceptron to categorize whether the comments included personal attacks or not.

In Wulczyn *et al.* [17], the authors examined various content characteristics, such as the use of first- and second-person pronouns, as well as the presence of vulgar language. They found that these factors indeed served as markers for cyberbullying.

The study [18] involves the utilization of a deep learning algorithm in conjunction with fuzzy logic to analyze 47,733 comments from Twitter, obtained from Kaggle. The methodology includes processing these comments using Keras embeddings and classifying them with a four-layer LSTM network. The application of fuzzy logic then aids in determining the severity of the flagged cyberbullying comments.

A comparable method was employed in Agrawal and Awekar [15], where researchers mainly focused on content attributes such as racist language and profanity. Another interdisciplinary study, discussed in Mikolov *et al.* [19], approached the issue from both computer science and human behavior perspectives. In their proposed technique, the researchers extensively analyzed content features, including URLs and hashtags. Surprisingly, they observed that 64 of the tweets contained external links, and 74.2% included hashtags, but these two aspects were not indicative of bullying. Similarly, [20] continued to explore cyberbullying detection through a content-based lens, introducing new elements like emoticons and a hieroglyph dictionary. They evaluated their approach using various ML algorithms, including SVM and J48, with SVM achieving the highest accuracy rate of 81%.

Another approach in the literature about tackling cyberbullying issue has been using transformer-based models such as BERT [21], BERT-m [21], [22], DistilmBERT, and IndicBERT [23]. Hybridizing various deep learning models, including Simple RNN [14], LSTM, BiLSTM [14], GRU [24], and CNN [25].

Dinakar *et al.* [13] Deep neural networks were utilized on three different datasets (Formspring, Twitter, Wikipedia), the authors in Badjatiya *et al.* [16] used basic word embeddings and language models including RoBERTa, XLNET, and ALBERT to detect cyberbullying. They evaluated their approach on a dataset of 10,000 tweets and found that the best-performing word embedding method is RoBERTa, which achieves an accuracy of 93.2%.

## 3. MATERIALS AND METHODS

Fig. 1 shows the proposed methodology of this research; in the following subsections, we discuss each part of the methodology.

### 3.1. The Dataset

The dataset has been prepared by [26] that contain 47,692 records distributed approximately equally amongst six classes as shown in figure 2, namely, not cyberbullying, religion, gender, ethnicity, age, and other. The dataset has been collected from the twitter social network. The dataset offers the opportunity to establish a multiclassification model for forecasting cyberbullying types, develop a binary classification model to identify potentially harmful tweets, or investigate the words and patterns linked with each form of cyberbullying. Fig. 1 shows sample distribution per type.

### 3.2. Text Preprocessing

In the preprocessing phase, the following actions shown in Fig. 3 were performed to clean and transform the text before further analysis or modeling.

#### *3.2.1. Remove links*

This is the initial cleaning step where any hyperlinks contained in the text are removed. Links may not contribute to the analysis of the text's content, especially when the goal is to understand linguistic patterns or sentiment.

#### *3.2.2. Clean non-alphabetical characters*

At this stage, the text is further cleaned by removing any characters that are not part of the alphabet. This typically includes punctuation, numerical digits, symbols, and any other non-letter characters. This step helps in focusing the analysis on words only.

#### *3.2.3. Convert text to lowercase*

Converting all text to lowercase is a standard normalization technique that helps in maintaining consistency throughout the dataset. It ensures that the same word in different cases (e.g., "Word" vs. "word") is treated as identical during analysis.

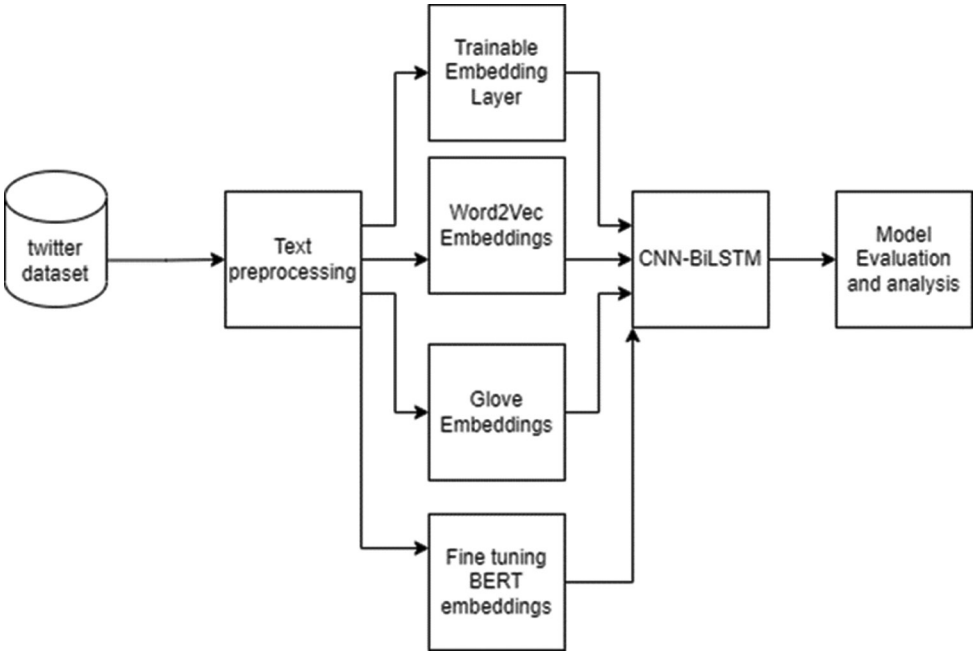


**Fig. 1.** Proposed methodology.

#### 3.2.4. Tokenize

Tokenization is the process of splitting the text into individual terms or words. This is a crucial step because most language processing methods work at the word level.

#### 3.2.5. Remove stop words

Common words such as "the," "is," and "in," which appear frequently and have little lexical content, are removed in this step. By eliminating these, the data are distilled, focusing on the more meaningful content words.

#### 3.2.6. Stemming

Stemming algorithms work by cutting off the end or the beginning of the word, considering a list of common prefixes and suffixes that can be found in an inflected word. This process reduces words to their root form, which helps in generalizing different forms of the same word (e.g., "running," "ran," "runs" all become "run").

Although text preprocessing steps can impact the original meaning of tweets to some extent yet their importance in reducing noise and shifting focus on the most meaningful words remains crucial. While stop words (such as "the" and "is") are often grammatically essential, they usually carry little semantic weight. The subsequent word embedding stage effectively captures the deeper semantic and syntactic relationships between the remaining words. Therefore, although some nuances may be lost, this trade-off is a necessary part of optimizing the model for efficiently detecting patterns indicative of cyberbullying.

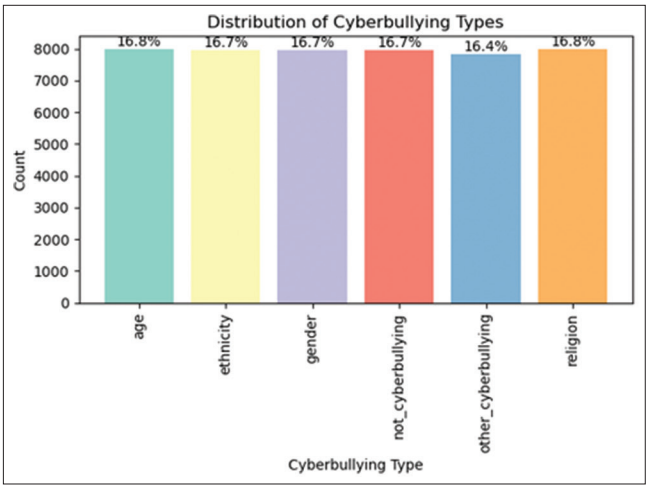


**Fig. 2.** Sample distribution amongst classes.

### 3.3. Word Embeddings

It represents a method employed within the field of (NLP) and ML. They involve representing words as dense numerical vectors within a multi-dimensional space. These vectors grasp both the semantic (meaning-related) and syntactic (grammar-related) associations amongst words, which are learned from how the words are used together in a large collection of text [19]. Word embeddings allow machines to understand and process language more effectively by capturing the meaning and associations of words in a numerical format which can be effortlessly utilized as input for various ML algorithms. In recent years, word embeddings have gained extensive usage in a variety of text classification and information retrieval tasks [19]. In this research, several different embeddings have been compared.

#### 3.3.1. Trainable embedding

To establish a baseline, we create an embedding layer with the vocabulary size (100,000), the dimensionality of the dense vector representation (8), and the length of the input sequences (input-length=50), which is randomly initialized and trainable. This implies that the baseline does not utilize pre-existing embeddings but instead learns them from scratch alongside other model parameters. Consequently, this approach allows us to examine how performance is affected by either training word embeddings from scratch or using pre-trained ones.

#### 3.3.2. Word2Vec

Mikolov *et al.* [27] is a neural network-based approach for learning word embeddings. It employs a neural network consisting of two layers to predict a target word, given the context of neighboring words (Referred to as the "continuous bag-of-words" model) or to forecast a context of neighboring words given a target word (referred to as the "skip-gram" model). The embeddings learned by the model are the weights of the input layer, which can be used as a dense representation

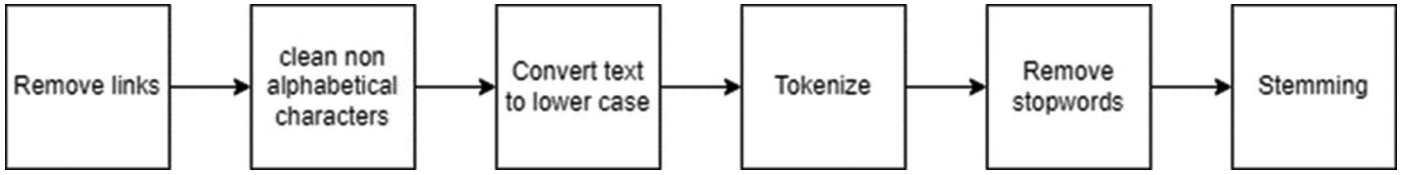


**Fig. 3.** Preprocessing steps taken.

of words in a high-dimensional space. The model contains 300-dimensional vectors for 3 million words and phrases.

#### 3.3.3. Global vectors for word representation (GloVe)

Pennington *et al.* [28] have a model that learns word embeddings through a matrix factorization technique. It operates on the co-occurrence matrix of words in a corpus, which is a symmetric matrix where each element (i, j) represents the frequency at which word i and word j co-occur within the same context. GloVe factorizes this matrix into two lower-dimensional matrices, which are used as the word embeddings. This method also enables keeping track of the relationship between the words. We used glove-twitter-100, which is trained on 2B tweets, 27B tokens, 1.2M vocab, uncased.

#### 3.3.4. BERT

Devlin *et al.* [29] constitutes a model rooted in the transformer architecture, which acquires profound bidirectional representations by jointly considering both preceding and succeeding context across all layers. Through pre-training on an extensive text corpus, it employs masked language modeling, wherein certain input words are replaced with a (MASK) token. Subsequently, the model is trained to predict the original words, enabling it to discern contextual associations between terms. BERT's effectiveness spans a diverse array of natural language understanding tasks, and it presently enjoys widespread utilization in NLP endeavors. In this study, we opted for the bert-base-uncased version, featuring 12 layers and 12 attention heads.

Each embedding method has its own unique characteristics, and it's crucial to understand the feature vectors they provide, whether used individually or in combination. In the text classification workflow, the embedding layer applies one or more pre-trained embeddings to create word representations for the downstream encoder, denoted as e. In our approach, we use CNN-BilSTM to transform the embedded vectors ⟨x0, x1, x2., xn⟩ into a single sequence representation, summarized as *O*, that is, O = $e^{(X0, X1. X2)}$ Once encoded, *O* is then passed to a fully connected layer denoted as f to produce logits across all labels: g = f (O).

In the case of multi-label classification, we calculate the probability using the sigmoid function:

$$P(li \mid s) = sigmoid\ (gi)$$

Here, the label li is assigned to the training example s if the estimated probability exceeds 0.5.

### 3.4. Classification (CNN-BiLSTM)

Following the embedding layer is a bidirectional layer encapsulating an LSTM layer which is depicted in Fig. 4. The LSTM is a type of RNN designed for sequential data processing. The bidirectional wrapper enables the LSTM to process input in the forward and backward directions simultaneously. Moreover, the LSTM layer is configured to return the full sequence.

Subsequently, a convolution1D layer is added, primarily used for one-dimensional convolution applied to sequential data, such as text. The layer requires three parameters: The number of filters (32), the kernel size (8), and the activation function employed (ReLU). To perform global max pooling on temporal data, the model incorporates a GlobalMaxPool1D layer, thereby extracting the maximum value across all time steps. To flatten the input, a Flatten layer is introduced, which reshapes the multidimensional output into a one-dimensional vector.

Next, a Dense layer with 128 neurons and a ReLU activation function is appended. A dense layer represents a regular layer of neurons within the neural network, with each neuron receiving input from all neurons in the preceding layer, establishing dense connections.

To combat overfitting, a Dropout layer with a dropout rate of 0.2 is included. Dropout serves as a regularization technique by randomly deactivating a fraction of input units during the forward pass. The final layer added is a dense layer consisting of 6 neurons and employing a SoftMax activation function. This layer is tasked with producing the probabilities linked to the input's membership within each of the six classes. The model is ultimately compiled with the sparse categorical cross-entropy loss function, the Adamax optimizer, and accuracy as the evaluation metric.

## 4. RESULTS AND DISCUSSION

This section presents the findings of our experiments, where we utilized deep learning models to create a cyberbullying detection system (CDS). The purpose of the CDS was to detect and categorize instances of linguistic cyberbullying into multiple classes. Regarding the choice of the word embeddings used in our experiments, pre-trained word embeddings performed better than trainable word embeddings.

In this investigation, we assessed the efficacy of a proposed

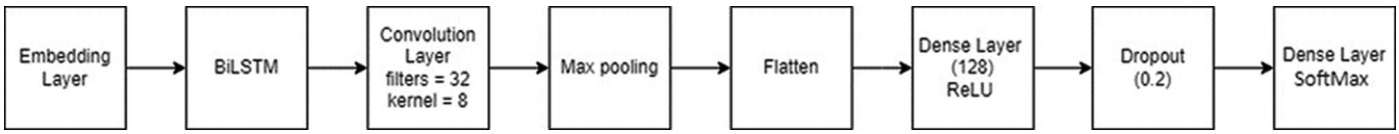


**Fig. 4.** Convolutional neural networks-bidirectional long short-term memory model.

model by utilizing various evaluation metrics to gauge its capacity for distinguishing the six classes of cyberbullying. The following commonly used evaluation criteria are employed to gauge the effectiveness of cyberbullying classifiers for social media networks:

- Accuracy (equation 1): This measures the ratio of correctly identified instances to all cases and is frequently employed to evaluate the performance of cyberbullying prediction models.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \# \quad (1)$$

- Precision (equation 2): Precision determines the percentage of relevant tweets among tweets classified as both true positives and false positives for a given category.

$$Precision = \frac{TP}{TP + FP} \# \quad (2)$$

- Recall (equation 3): This metric assesses the proportion of relevant tweets that are correctly identified out of all relevant tweets.

$$Recall = \frac{TP}{TP + FN} \# \quad (3)$$

- F-measure (equation 4): The F-measure offers a unified metric that considers both recall and precision, providing a balanced assessment of both aspects.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \# \quad (4)$$

However, since the dataset is balanced these metrics produced similar numbers; therefore, we report the accuracy of each embedding in (Fig. 5). The bar chart visualization of the results showed that fine-tuning BERT achieved the highest accuracy of 85%, indicating its effectiveness in identifying instances of cyberbullying. Glove and Word2Vec followed closely behind with accuracies of 82% and 81.31%, respectively, showcasing their strong performance as well. The Trainable model, which represents a non-trained approach to word embeddings, achieved an accuracy of 80.2%. We used a batch size of 32 and ran the model for six epochs.

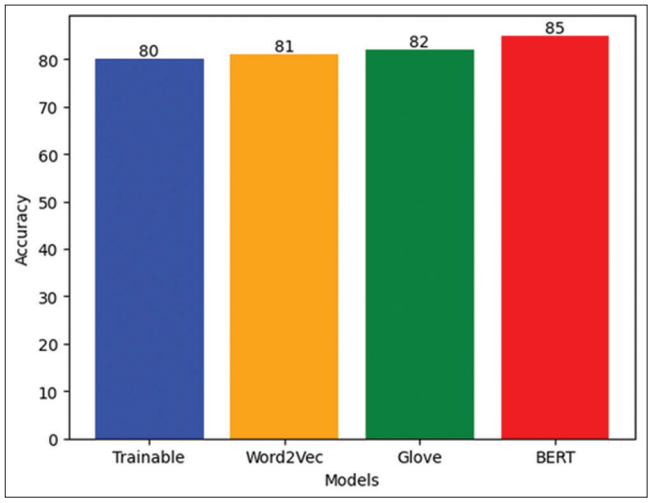


**Fig. 5.** Accuracy of different word embeddings for the convolutional neural networks-bidirectional long short-term memory model.

Regarding the individual classes of cyberbullying, the two classes of other cyberbullying and non-cyberbullying recorded worse results in all the experiments (Fig. 6) show the accuracy, precision, recall, and F1-score of BERT model. In an attempt to further increase model performance, [30] have used a modified version of this dataset where the other cyberbullying class has been removed from the dataset.

To further investigate the results achieved with our proposed method, we conducted tests using several classical ML algorithms, as shown in Fig. 7: Logistic Regression, KNN, Random Forest, and Naïve Bayes. The data underwent the same preprocessing steps, as described in Fig. 3. Subsequently, we applied the TF-IDF vectorizer to extract features and reduce the dimensionality of the vocabulary.

TF-IDF stands as a numerical metric utilized in information retrieval and text mining. Its purpose is to gauge the significance of a term within a specific document or an entire collection [12]. For our experiments, we allocated 20% of the dataset for testing and used the remaining 80% for training.

For logistic regression, we set the maximum iteration to 300 and chose lbfgs as the solver. As for KNN, we selected

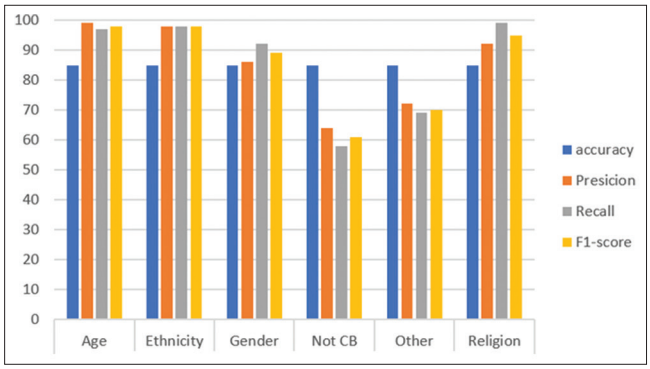


**Fig. 6.** Bidirectional encoder representations from transformers metrics for the six classes of the dataset.

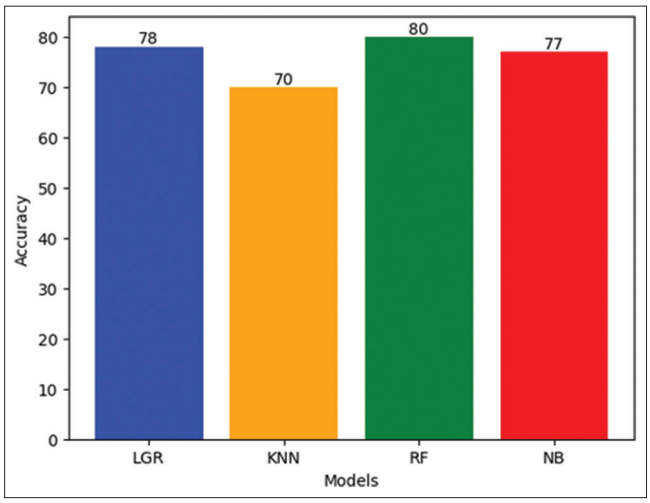


**Fig. 7.** Accuracy of baseline models.

k=3 and used uniform weights. In the case of random forest, we specified a maximum depth of 100. All other hyperparameters were set to the default values provided by the sklearn library [31].

## 5. CONCLUSION AND FUTURE WORK

This research article focuses on the application of word embeddings for multiclass cyberbullying detection. The study compares the performance of trainable word embeddings, pre-trained word embeddings, and fine-tuning language models using various techniques and models such as CNN and biLSTM, as well as word2vec and BERT.

The research goes beyond simple binary classification and instead focuses on multi-class detection of cyberbullying. By employing a dataset collected from Twitter, the study explores the effectiveness of different word embedding techniques in detecting cyberbullying types such as religion, gender, ethnicity, age, and others. The preprocessing phase includes actions such as removing links, cleaning text, tokenization, removing stop words, and stemming.

The results indicate that fine-tuning the BERT model on the dataset yields the most promising results. The proposed CNN-BiLSTM model, incorporating word embeddings and deep learning techniques, demonstrates effectiveness in classifying and detecting cyberbullying instances.

In the future, it is important to focus on domain adaptation and transfer learning for cyberbullying detection. Researchers should investigate techniques for adapting models to different social media platforms or domains. Exploring transfer learning approaches that leverage knowledge from related tasks or domains can enhance the performance of cyberbullying detection models. Creating cross-domain datasets will enable the evaluation of domain adaptation and transfer learning techniques in realistic settings. In addition, developing knowledge distillation methods can facilitate the transfer of knowledge to smaller, more efficient models. Real-world evaluations are crucial to measure the effectiveness of domain adaptation and transfer learning techniques, considering variations in data distribution, biases, and user behaviors. By enhancing the generalizability of cyberbullying detection models, we can better address the challenges across diverse platforms and user contexts.

The findings can aid in the development of more accurate and efficient methods for identifying and preventing cyberbullying, particularly in the context of social media platforms.

## REFERENCES

[1] R. S. Tokunaga. "Following you home from school: A critical review and synthesis of research on cyberbullying victimization". *Computers in Human Behavior*, vol. 26, no. 3, pp. 277-287, 2010.

[2] K. Hellfeldt, L. López-Romero and H. Andershed. "Cyberbullying and psychological well-being in young adolescence: The potential protective mediation effects of social support from family, friends, and teachers". *International Journal of Environmental Research and Public Health*, vol. 17, no. 1, p. 45, 2020.

[3] K. Rudnicki, H. Vandebosch, P. Voué and K. Poels. "Systematic review of determinants and consequences of bystander interventions in online hate and cyberbullying among adults". *Behaviour and Information Technology*, vol. 42, no. 5, pp. 527-544, 2023.

[4] H. Rosa, N. Pereira, R. Ribeiro, P. Ferreira, J. Carvalho, S.

Oliveira, L. Coheur, P. Paulino, A. V. Simão and I. Trancoso. "Automatic cyberbullying detection: A systematic review". *Computers in Human Behavior*, vol. 93, pp. 333-345, 2019.

[5] F. Elsafoury, S. Katsigiannis, Z. Pervez and N. Ramzan. "When the timeline meets the pipeline: A survey on automated cyberbullying detection". *IEEE Access*, vol. 9, pp. 103541-103563, 2021.

[6] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis and L. Edwards. "Detection of harassment on web 2.0". *Proceedings of the Content Analysis in the Web*, vol. 2, pp. 1-7, 2009.

[7] F. Almeida and G. Xexéo. "*Word Embeddings: A Survey*". arXiv, 2023. Available from: https://arxiv.org/abs/1901.09069 [Last accessed on 2023 Nov 13].

[8] D. Yin, Z. Xue and L. Hong. *"Detection of Harassment on Web 2.0. In: Proceedings of the Content Analysis in the WEB,* vol. 2, pp. 1-7, 2009.

[9] C. Iwendi, G. Srivastava, S. Khan and P. K. R. Maddikunta. "Cyberbullying detection solutions based on deep learning architectures". *Multimedia Systems*, vol. 29, no. 3, pp. 1839-1852, 2023.

[10] B. A. Talpur and D. O'Sullivan. "Multi-class imbalance in text classification: A feature engineering approach to detect cyberbullying in twitter". *Informatics*, vol. 7, p. 52, 2020.

[11] A. Bozyiğit, S. Utku and E. Nasibov. "Cyberbullying detection: Utilizing social media features". *Expert Systems with Applications*, vol. 179, p. 115001, 2021.

[12] A. Aizawa. "An information-theoretic perspective of tf-idf measures". *Information Processing and Management*, vol. 39, no. 1, pp. 45-65, 2003.

[13] K. Dinakar, R. Reichart and H. Lieberman. "*Modeling the Detection of Textual Cyberbullying. In: Proceedings of the International AAAI Conference on Web and Social Media*", pp. 11-17, 2011. Available from: https://ojs.aaai.org/index.php/icwsm/article/view/14209 [Last accessed on 2023 Nov 13].

[14] A. Dewani, M. A. Memon and S. Bhatti. "Cyberbullying detection: Advanced preprocessing techniques and deep learning architecture for Roman Urdu data". *Journal of Big Data*, vol. 8, no. 1, p. 160, 2021.

[15] S. Agrawal and A. Awekar. Deep learning for detecting cyberbullying across multiple social media platforms. In: G. Pasi, B. Piwowarski, L. Azzopardi and A. Hanbury, Eds. "*Advances in Information Retrieval. Lecture Notes in Computer Science*". vol. 10772. Springer International Publishing, Cham, pp. 141-153, 2018.

[16] P. Badjatiya, S. Gupta, M. Gupta and V. Varma. "*Deep Learning for Hate Speech Detection in Tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion-WWW '17 Companion*". ACM Press, Perth, Australia, 2017, pp. 759-760.

[17] E. Wulczyn, N. Thain and L. Dixon. "*Ex Machina: Personal Attacks Seen at Scale. In: Proceedings of the 26th International Conference on World Wide Web*". International World Wide Web Conferences Steering Committee, Perth Australia, pp. 1391-1399, 2017.

[18] M. H. Obaid, S. K. Guirguis and S. M. Elkaffas. "Cyberbullying detection and severity determination model". *IEEE Access*, vol. 11, pp. 97391-97399, 2023.

[19] T. Mikolov, W. Yih and G. Zweig. "*Linguistic Regularities in Continuous Space Word Representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*", pp. 746-751, 2013. Available from: https://aclanthology.org/n13-1090.pdf [Last accessed on 2024 Jan 10].

[20] C. Wang, P. Nulty and D. Lillis. "*A Comparative Study on Word Embeddings in Deep Learning for Text Classification. In: Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*". ACM, Seoul Republic of Korea, pp. 37-46, 2020.

[21] M. Das, S. Banerjee and P. Saha. "*Abusive and Threatening Language Detection in Urdu Using Boosting based and BERT based Models: A Comparative Approach*". arXiv, 2021. Available from: https://arxiv.org/abs/2111.14830 [Last accessed on 2024 Jan 10].

[22] S. Gaikwad, T. Ranasinghe, M. Zampieri and C. M. Homan. "*Cross-lingual Offensive Language Identification for Low Resource Languages: The Case of Marathi*". arXiv, 2021. Available from: https://arxiv.org/abs/2109.03552 [Last accessed on 2024 Jan 10].

[23] D. Saha, N. Paharia, D. Chakraborty, P. Saha and A. Mukherjee, "*Hate-Alert@DravidianLangTech-EACL2021: Ensembling Strategies for Transformer-based Offensive Language Detection*". arXiv, 2021.

[24] A. M. Ishmam and S. Sharmin. "*Hateful Speech Detection in Public Facebook Pages for the Bengali Language. In: 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA)*", IEEE, pp. 555-560, 2019.

[25] R. Cao, R. K. W. Lee and T. A. Hoang. "*DeepHate: Hate Speech Detection Via Multi-Faceted Text Representations. In: 12th ACM Conference on Web Science*". ACM, Southampton United Kingdom, pp. 11-20, 2020.

[26] 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, pp. 1699-1708, 2020.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "*Distributed Representations of Words and Phrases and Their Compositionality. In: Advances in Neural Information Processing Systems*". vol. 26, 2013. Available from: [Last accessed on 2024 Jan 10].

[28] J. Pennington, R. Socher and C. D. Manning. "*Glove: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*", pp. 1532-1543, 2014.

[29] J. Devlin, M. W. Chang, K. Lee and K. Toutanova. "*BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*". arXiv, 2019. Available from: https://arxiv.org/abs/1810.04805 [Last accessed on 2024 Jan 10].

[30] T. H. Aldhyani, M. H. Al-Adhaileh and S. N. Alsubari. "Cyberbullying identification system based deep learning algorithms". *Electronics*, vol. 11, no. 20, p. 3273, 2022.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas,… and E. Duchesnay. "Scikit-learn: Machine learning in python". *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.