Cyberbullying Detection: A Hybrid Models based on Machine Learning and Natural Language Processing Techniques

Chahat Raj

School of Computer Science, FEIT, University of Technology Sydney, Sydney, Australia, chahatraj58@gmail.com

Ayush Agrawal

Department of Computer Science, Delhi Technological University, Delhi, India, ayush286@gmail.com

Gnana Bharathy

School of Computer Science, FEIT, University of Technology Sydney, Sydney, Australia, gnana.bharathy@uts.edu.au

Bhuva Narayan

School of Communication, FASS, University of Technology Sydney, Sydney, Australia, bhuva.narayan@uts.edu.au

Mukesh Prasad

School of Computer Science, FEIT, University of Technology Sydney, Sydney, Australia, mukesh.prasad@uts.edu.au

Abstract: The rise in web and social media interactions has resulted in the easy proliferation of offensive language and hate speech. Such online harassment, insults, and attacks are commonly termed cyberbullying. The sheer volume of user-generated content have made it challenging to identify such illicit content. Machine learning has wide applications in text classification and researchers are shifting towards using deep neural networks in detecting cyberbullying due to their several advantages over traditional machine learning. This paper proposes a novel deep learning framework with parameter optimization and an algorithmic comparative study of eleven classification methods: four machine learning algorithms and seven deep learning methods, on two real-world cyberbullying datasets. In addition, this paper also examines the effect of feature extraction and word-embedding-techniques-based natural language processing on algorithmic performances. It is observed that bidirectional neural networks and attention models provide high classification results. Logistic Regression is observed to be the best among machine learning classifiers used. Term Frequency- Inverse Document Frequency (TF-IDF) demonstrates consistently high accuracies with machine learning techniques. Global Vectors (GloVe) perform thoroughly better with deep learning models. Bi-GRU and Bi-LSTM worked best amongst the deep neural networks used. The extensive experiments validated on the two datasets establish the importance of this work by comparing eleven classification methods and seven feature extraction techniques. The proposed deep learning methods outperform existing state-of-the-art approaches for cyberbullying detection with accuracy and F1-scores as high as ~95% and ~98%, respectively.

Keywords: Cyberbullying, Hate speech, Offensive language, Deep learning, Machine learning

1. Introduction

Social media is an interactive tool that brings people together to share information. The primary function of Online Social Networks (OSNs) is to allow people to communicate virtually using the internet. However, such technologies have also resulted in several additional social issues, one of them being 'cyberbullying.' Although bullying has existed in society before these technologies, the perceived protection of the online interfaces has resulted in increased cyberbullying. Cyberbullying is commonly defined as an intentional violent or aggressive behaviour using electronic media carried out by an individual or a group targeting a victim online [1]. This action involves repeated online insulting, harassing, or attacking a target verbally [2]. Malicious social media users use sexist remarks, offensive language, hate speech, toxic comments, and abusive language to target victims. Such content demeans the integrity of social networks and torments its users' and causes mental health issues [3]. Manual flagging of such illicit content online is neither feasible nor fruitful [4]. The need for an automated detection mechanism for cyberbullying detection is evident.

Natural Language Processing (NLP) has wide applications in this domain, as researchers have employed several feature extraction techniques for textual content. Primary endeavors include supervised classification using bag-of-words at character-level representation [5, 6, 7] by various machine learning algorithms, such as

Random Forest, Naïve Bayes, Linear Regression, Support Vector Machine, and XGBoost for cyberbullying detection [8]. Recurrent Neural Networks (RNNs) are among researchers' primary choices for text classification [4, 9] due to their ability to mine the implicit semantic features in the text. Advancement in approaches have brought a user-level detection mechanism where author profiling is used to check if a user has been previously involved in any acts of cyberbullying [3, 10]. Existing approaches have also focused on the background information and user characteristics, such as their demographic data for characterizing malicious users [11]. With recent evolution, Text-based Convolutional Neural Networks (Text-CNN) have come into focus [12]. Although being primarily employed in visual tasks, CNNs have displayed tremendous performance in one-dimensional text classification [13, 14].

This paper provides a wide comparative study on various machine learning and deep learning algorithms. We experiment on two real-world datasets and compare the performances of eleven classification algorithms. We examine the classification efficiencies of XGBoost, SVM, Naïve Bayes, and Logistic Regression under the machine learning category. We implement four types of feature extraction techniques for these methods: count vectorization, (Term Frequency- Inverse Document Frequency) TF-IDF word unigram, TF-IDF word bigram & trigram, and TF-IDF character bigram & trigram. In the deep learning category, we propose a novel framework accommodating CNN, (Long Short-Term Memory) LSTM, (Bidirectional Long Short-Term Memory) Bi-LSTM, (Gated Recurrent Unit) GRU, (Bidirectional Gated Recurrent Unit) Bi-GRU, CNN-BiLSTM, and Attention-BiLSTM. The embedding techniques experimented with deep learning algorithms are (Global Vectors) GloVe, FastText, and Paragram. This comparative study examines the performances of algorithms and their feature extraction techniques. The results, compared using four evaluation metrics: accuracy, precision, recall, and F1-score are observed to outperform several state-of-the-art cyberbullying detection mechanisms.

The rest of the paper is organized as follows: Section 2 explains the existing literature and mechanisms developed for efficient cyberbullying detection. Section 3 describes the methodology of the proposed work, the mathematical background of algorithms used, and the novel framework to accommodate all deep learning algorithms. Section 4 lists the experimental results obtained and compares the performances, and finally, Section 5 concludes the contributions and future prospects.

2. Related Works

Classification frameworks of cyberbullying posts primarily utilize Natural Language Processing (NLP) methods [15]. Term Frequency- Inverse Document Frequency (TF-IDF) is an established method to extract textual features from the data [16]. It measures a word's importance in a given document using its frequency and inverse frequency count. Several NLP techniques like TF-IDF, Vector Space Model (VSM), Linear Discriminant Analysis (LDA), Latent Semantic Analysis (LSA) have been designed for such feature extraction [17, 18, 19]. The disadvantage lies in considering the word order, which is overcome by robust deep neural networks. There is a considerable availability of real-world cyberbullying datasets in the present scenario such as the Twitter dataset [11], the Chinese Sina Weibo dataset [13] and the Kaggle dataset [20]. These labeled datasets allow the use of machine learning and deep learning algorithms by aiding supervised and semi-supervised training. Ample availability of annotated training data aids in building efficient supervised frameworks. However, the task of online cyberbullying detection holds certain limitations which are to be regarded. The drawbacks lie in the low availability of positively labeled cyberbullying posts because the datasets available are highly imbalanced. Wulczyn et al. [21] crowdsourced and aggregated a vast corpus of annotated Wikipedia articles with over 100K items extracted from talk pages. Another annotated dataset was presented by Warner and Hirschberg [22], collecting commonly used hate-speech terms from Twitter. They identified that hate speech targets specific groups of a particular ethnicity, race, caste or creed, and found that a correlation exists between hate speech and stereotypical words. They performed hate speech classification using Support Vector Machines (SVM) linked with word sense disambiguation, using a lexicon of stereotypical words as features. A Naïve Bayes approach was implemented by Kwok and Wang [23] on a Twitter dataset comprising of racist and non-racist comments, that demonstrated average classification performance using the unigram bag-of-words model for feature extraction. A combined approach utilizing linguistics, n-grams, syntactic, and distributed syntactic features was designed by Nobata et al. [24] to detect online hate-speech. Yin et al. [25] proposed a supervised approach with TF-IDF for cyberbullying detection that uses content, context, and sentiment as textual features. A systematic review of existing research in this domain is compiled by Tokunaga [26], discussing the cyberbullying typologies, detection frameworks, and potential directions.

With the advent of deep learning algorithms for cyberbullying detection, Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) approaches have primarily been employed. Bu and Cho proposed a novel ensemble framework that uses two deep learning models for knowledge transferring: a CNN for capturing

character-level syntactic features of the text and a Long-term Recurrent Convolutional Network (LRCN) for extracting semantic features [2]. Agrawal and Awekar [27] experimented on deep learning models by domain transferring the knowledge using CNN, LSTM, Bi-LSTM, and Bi-LSTM with attention using random, GloVe, and SSWE (Sentiment-Specific Word Embedding). Aroyehun and Gelbukh established the efficacy of deep neural networks by using seven different combinations of CNN, LSTM, and Bi-LSTM models and comparing the results with traditional machine learning algorithms for aggression detection [28] incorporating data augmentation and pseudo labeling for the same. Mishra et al. proposed a novel method of Twitter user profiling for cyberbullying detection [3] using authors' community-based data in addition to textual information. Rawat et al. also rely on user information for abusive content detection [10] employing web scraping and exploratory data analysis to analyze the characteristics of users involved in spreading hate speech by combining machine learning algorithms, sentiment analysis, and topic modeling for malicious user detection. The offensive tweet detection model by Aglionby et al. [29] proposes a multi-layer RNN and Gradient Boost Decision Tree (GBDT) classifier framework with a self-attention mechanism that enhances text classification. Chen et al. [30] have analyzed embedding methods for words and sequences experimenting with word-level and sentence-level embedding techniques. Chu et al. also explored deep learning models with word embeddings for abuse detection [31] by developing an RNN with LSTM and two CNNs with word and character-level embeddings. Anand and Eswari developed an LSTM and a CNN network and analyzed their effect in the presence and absence of GloVe embeddings for cyberbullying detection [32]. Badjatiya et al. explored the performances of CNN and LSTM with various text embedding models observing GDBT combined with LSTM as their best performing model [9]. Pavlopoulos et al. established that their proposed RNN with attention-mechanism outperforms Logistic Regression, a Multi-Layer Perceptron (MLP) and a vanilla-CNN model for cyberbullying detection [33]. Banerjee et al. propose a simple convolutional network with GloVe embeddings performing better than RNN GloVe and several machine learning techniques [34]. A Bi-LSTM network with attention mechanism was proposed by Agarwal et al. [35] to classify cyberbullying posts using under sampling and class weighting for avoiding class imbalance in the dataset.

3. Methodology

Online cyberbullying detection frameworks primarily rely on machine learning algorithms. Machine learning poses a disadvantage due to the inability to process vast volumes of data. Existing studies are advancing towards utilizing deep neural networks that overcome this limitation and provide higher results and robust mechanisms. This section covers various popular machine learning and deep learning algorithms. We discuss the proposed methodology of our classification frameworks and the architectures of all the proposed networks.

3.1 Pre-processing and Feature Extraction

Algorithms for text classification cannot process raw data due to the inability to understand high-level human language directly. The text undergoes conversion into vector notation to be processed by classification algorithms. Prior to this step, raw textual data undergoes several pre-processing steps, often referred to as data cleaning. Figure 1 illustrates the workflow of the proposed methodology. Data is pre-processed by removing empty rows, removing punctuation, special characters, numerical values, lowercasing text, stopword removal, tokenization, and stemming. To create vector notations of input text, we experiment with several methods. For machine learning models, we use four methods: Count Vectorization, TF-IDF word unigram, TF-IDF word bigram & trigram, and TF-IDF character bigram & trigram. For deep learning approaches, we use GloVe, FastText, and Paragram as the embedding representations. Transformer-based deep-learning models use their transformer word embeddings. Post embedding, we implement the stated machine learning and deep learning algorithms. To deal with the class imbalance in the available real-world datasets, we employ stratified 5-fold cross validation technique that splits the dataset into five sets of training and testing, addressing the imbalance by assigning samples to each fold in a proportionate manner.

Count Vectorization [36]**:** This is a simple statistical method to generate embedded vectors of input text. The algorithm uses the frequency of the occurrence of a term in a document to generate its embedding vector. A matrix is created for the entire document set where rows contain each document and columns represent each word. The cells contain the values of occurrence frequency of a term in a document.

TF-IDF [37]**:** Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical approach that uses the occurrences of words as a measure to extract textual features. A word's importance is directly proportional to its frequency in the document and inversely proportional to its frequency in the entire document set. For a term, w_i in a document x_j, where its occurrence is $n_{i,j}$ in x_j, the term-frequency, $TF_{i,j}$ is calculated by Eq. 1.

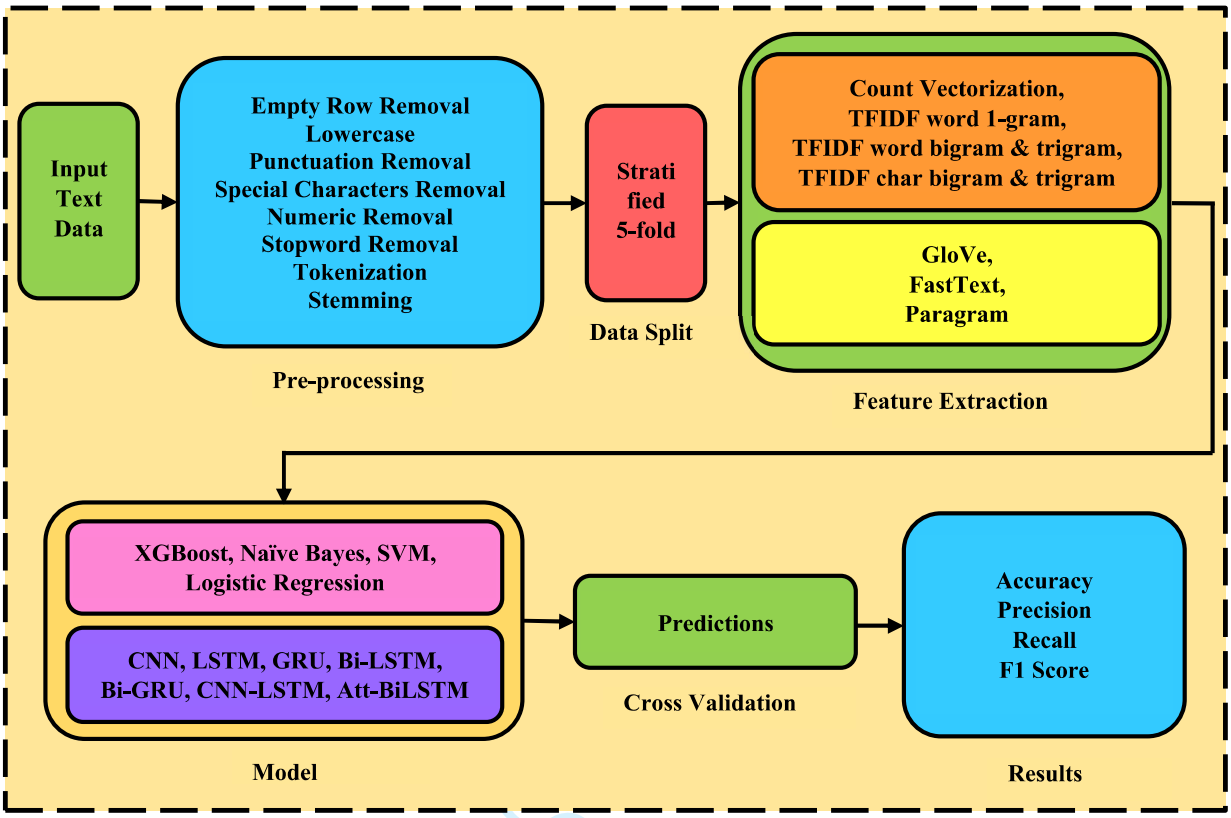

Figure 1: Workflow of the proposed framework

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (1)$$

Here, $\sum_k n_{k,j}$ denotes the sum of occurrences of a term w_i in the entire document set. The inverse document frequency (IDF) is computed by taking the logarithm of the total number of documents divided by the number of documents with the term w_iAs in Eq. 2

$$IDF_i = log\ (\frac{|D|}{|\{j:w_i \in x_j\}|+1}) \quad (2)$$

Here, $|D|$ denotes the total number of documents and $|\{j: w_i \in x_j\}|$ is the number of documents with the term w_i. The TF-IDF of the term w_i is given by Eq. 3

$$(TF - IDF)_{w_i} = TF_{i,j} \times IDF_i \quad (3)$$

We use TF-IDF with word unigram, word bigram & trigram, and character bigram & trigram.

GloVe [38]**:** Global Vectors (GloVe) for word representations is an unsupervised technique to derive word embeddings from text input. The approach utilizes an A×A term-based co-occurrence matrix to obtain representations. The method relies on examining the semantic relationship between the terms. For instance, high cosine similarity is demonstrated between words like 'queen' and 'king' or 'mother' and 'woman'. The technique learns from a large Wikipedia and Gigaword corpus in an unsupervised fashion. For a word i with its vector representation w_i, the objective function is denoted by Eq. 4.

$$f(w_i - w_j, \widetilde{w}_k) = \frac{P_{ik}}{P_{jk}} \quad (4)$$

where i and k are words with a similar context and P_{ik} is the probability of them occurring together.

FastText [39, 40]**:** Introduced by Facebook's AI Research Lab (FAIR), FastText is a skip-gram-based model for enhanced word representations. The effectiveness of this technique lies in its consideration of the morphology of words in a language. Other embedding techniques denote each word as a distinct vector. In contrast, FastText is specially designed to handle words of the same root using character n-grams. Naturally, it contains the sub-word information for every word by dividing each word into a bag of its n-gram combinations. For example, for a word

'language' with n as 3, the bag of character n-grams will contain 'la', 'lan', 'ang', 'ngu', 'gua', 'uag', 'age' and 'ge'. FastText enables understanding the context of unknown words by breaking them into smaller forms and matching the similarities with those within its training corpus.

Paragram [41]**:** Paragram is another word representation technique designed to capture better contextual similarities. It uses the ParaPhrase DataBase (PPDB) and performs fine-tuning, counter-fitting, and attack-repel to inject synonym and antonym features as a vectorization constraint. The technique is comparatively robust due to better contextual understanding.

3.2 Machine Learning Approaches

We employ four popular machine learning approaches for cyberbullying detection: XGBoost, Naïve Bayes, SVM, and Logistic Regression. After the pre-processing and feature extraction phases, the vectorized text is input to these classifiers to evaluate their performances.

XGBoost [42]**:** Extreme Gradient Boosting (XGBoost) and Gradient Boosting Decision Tree (GBDT) advancement. The algorithm employs multiple decision trees with low accuracy (weak learners) and combines them to provide higher accuracy. The trees are built in stages, and the residuals or errors of prior models from previous stages inform the next stage of trees using gradient descent. That is standard gradient boosting. Building on top of gradient boosting principle, XGBoost incorporates several other optimizations such as parallelized implementation, tree pruning, hardware optimization, regularization, sparsity awareness, cross validation, and weighted quantile sketch, to make its performance more efficient and effective.
The algorithm advances in the direction of the tree, which minimizes the objective function. For a document $D = \{(x_i, y_i)||D| = n, x_i \in R^m, y_i \in R\}$ with n samples and m eigenvalues, where x_i denotes a sample and y_i denotes its category, the predictions are calculated using Eq. 5.

$$\hat{y}_i = \theta(x_i) = \sum_{k=1}^{K} f_k(x_i) \tag{5}$$

where, $f_k(x_i)$ is the error value between true and predicted classes.

Naïve Bayes [43]**:** Naïve Bayes (NB) algorithm is used for probabilistic classification. It is widely used for various practical applications due to its efficiency in reducing computational costs. It is a scalable algorithm applicable to large-sized datasets, also resulting in high classification accuracies. Its principle assumes that a feature in a category is independent of its presence in another category. The probability of a document D pertaining to a class C is given by Eq. 6.

$$P(C|D) = \frac{P(D_j|C)P(C)}{P(D_j)} = \frac{P(D_j|C)P(C)}{P(D_j|C)P(C)+P(D_j|\bar{C})P(\bar{C})} \tag{6}$$

To predict that a data point x' with features $(a'_i \dots a'_d)$ belongs to a particular category, the prediction $\theta(x')$ is given by Eq. 7.

$$\theta(x') = argmax_{y_i \in C} P(y_i) \prod_{j=1}^{d} P(a'_j|y_i) \tag{7}$$

SVM [44]**:** Support Vector Machine (SVM) is a supervised algorithm that uses the separation margin between data points of classes as a classification criterion. The original m-dimensional feature space is reduced to a user-defined dimensional space. Support vectors are then determined and to optimize the margin distance among data points of different categories. The algorithm automatically determines these support vectors found nearest to the separating margins (hyperplanes). Eq 8 defines a linear SVM optimization equation.

$$\alpha^* = maximize_\alpha \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (x_i x_j) \tag{8}$$

$$subject\ to\ the\ constraints\ \sum_{i=1}^{l} y_i \alpha_i = 0 \tag{9}$$

$$0 \le \alpha_i \le C, i = 1 \dots l$$

where α_i denotes the term weight, and C represents the model's error and relative importance. To predict that a data point x' belongs to a particular category, the prediction $\theta(x')$ is given by Eq. 10.

$$\theta(x') = sign\left(\sum_{i=1}^{l} \alpha_i y_i (x_i, x') + b\right) = sign((w^*, x') + b) \tag{10}$$

where, $w^* = \sum_{i=1}^{l} \alpha_i y_i x_i$.

Logistic Regression [45]**:** Logistic Regression (LR) is another statistical algorithm that works on predicting probabilities rather than classes. The logistic function is used to form a hyperplane to classify data points in the given classes. Textual features are input to the algorithm employed to generate forecasts about a data point belonging to a particular class. The function is given by Eq. 11 where the positive class is determined by $h_\theta(x) \geq 0.5, y = 1$, and the negative class is determined by $h_\theta(x) \leq 0.5, y = 0$.

$$h_\theta(x) = \frac{1}{1+e^{-x_i\theta}} \tag{11}$$

3.3 Deep Learning Approaches

The elimination of manual feature extraction has made deep neural networks extremely popular in the research community. Neurons within the network are responsible for automatically extracting essential features that help to differentiate content belonging to different classes. The need of deep neural networks arises due to large dataset sizes which most of the machine learning algorithms fail to accommodate. Additionally, neural networks offer robustness and higher classification results. We compare the following architectures of popular neural networks for cyberbullying classification: CNN, LSTM, Bi-LSTM, GRU, Bi-GRU, CNN-BiLSTM, and Attention-BiLSTM. To execute our approach, we design a novel framework accommodating each of these models, depicted by figures 2-5. The methodology and architectures of these models are discussed below.

CNN [46]**:** After demonstrating extreme efficiency in image classification tasks, convolutional neural networks are widely adopted for text classification. We develop a vanilla Text-CNN and employ it for cyberbullying detection. A representation of the proposed architecture incorporation of a CNN is shown in Figure 2. The model intakes pre-processed text input and performs respective embedding. A convolution operation $*$ on functions f and g is performed by reversing and shifting one of these functions as described by Eq. 12.

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{12}$$

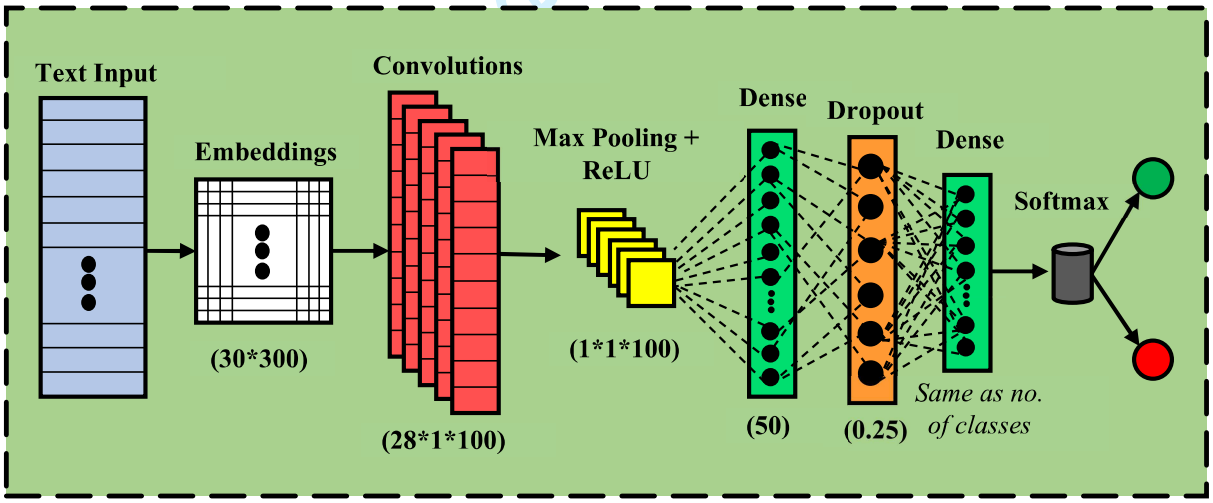

Figure 2: Architectural representation of proposed CNN framework

The embedded text passes through a one-dimensional convolutional layer with a kernel that moves over the convolutions with a filter size of 3. The information is then passed through a max-pooling layer that outputs the max value from each resulting data matrix. ReLU activation function is used, post which the input is fed to a series of fully connected layers. A dense layer of dimension 50 sends the information through a dropout layer to a final dense layer. The dimension of this final layer is variable depending upon the number of classes in a particular dataset. Predictions are then generated using a softmax function that outputs the classes for each item in the document set.

LSTM [47]**:** Long Short-Term Memory networks (LSTMs) are a special type of Recurrent Neural Networks (RNNs) that are advantageous than RNNs in terms of information retention. LSTMs overcome the problem of gradient descent encountered in traditional RNNs. LSTMs are highly preferred for tasks like text classification and predictive modeling due to their extensive memory capacity. Such a network selectively decides which information is necessary to be transferred to further neurons and which data can be forgotten or omitted. These networks employ backpropagation and a gated mechanism. A basic LSTM network consists of an input (i_t), output (o_t) and a forget gate (f_t), represented by the following equations.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (13)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (14)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (15)$$

Here, x_t denotes an input text, h is used to represent the state of the input where h_t is called current state and h_{t-1} denotes the previous state. W and b are the weights and bias for each gate, respectively. Here, σ denotes the activation function used, which is ReLU in the case of the proposed model.

Bi-LSTM [48]**:** Bidirectional LSTM (Bi-LSTM) is a robust mechanism to enhance backpropagation in LSTM networks. While the information in an LSTM travels unidirectionally, a Bi-LSTM allows data to move in both forward and backward directions. A Bi-LSTM processes input both reverse and serially. Architecturally, it is simply combining two LSTMs but in opposite directions. This allows the network to remember information from past to future using the forward layer and future to past using the backword LSTM layer.

GRU [49]**:** Gated Recurrent Units (GRU) are also a type of RNN with a gated mechanism designed to deal with the vanishing and exploding gradient problem. These provide more testing accuracies than traditional RNNs because of the ability to remember long-term dependencies. GRUs is a more straightforward and dynamic version of LSTM networks specially designed for updating or resetting information in their memory cells. The network constitutes an update gate that combines input, and a forget gate present in LSTMs. Additionally, there is a reset gate for refreshing the memory contents. These are lightweight and have fewer parameters than LSTMs. For an input vector x_t at time t with vector, parameter, and matrices as b, W and U respectively, the update gate and reset gate are given by Eq. 16 and 17.

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \quad (16)$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \quad (17)$$

where h_t is defined by Eq. 18 with $\odot$ as the Hadamard product.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widehat{h_t} \quad (18)$$

$$\widehat{h_t} = \emptyset_h(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \quad (29)$$

σ_g and $\emptyset_h$ are the activation function and hyperbolic tangent, respectively.

Bi-GRU [48]**:** A bidirectional GRU is a dual-layered structure similar to a Bi-LSTM with forward and backward neural networks. The idea of this structure is to transfer entire contextual information from the input to the output layer. Similar to a bidirectional LSTM, in a Bi-GRU, the input information travels through a neural network in the forward direction and a neural network in the backward direction. The outputs from both these forward and backward layers are fused to provide the final output. An architectural representation for classification using LSTM, Bi-LSTM, GRU, and Bi-GRU is displayed in Figure 3. To use a specific model at a time, the outputs from the embedding matrix are fed to the supposed deep learning model and then sent to the series of fully connected layers. The parameters of fully connected layers stay the same as in CNN for all models proposed in this work.

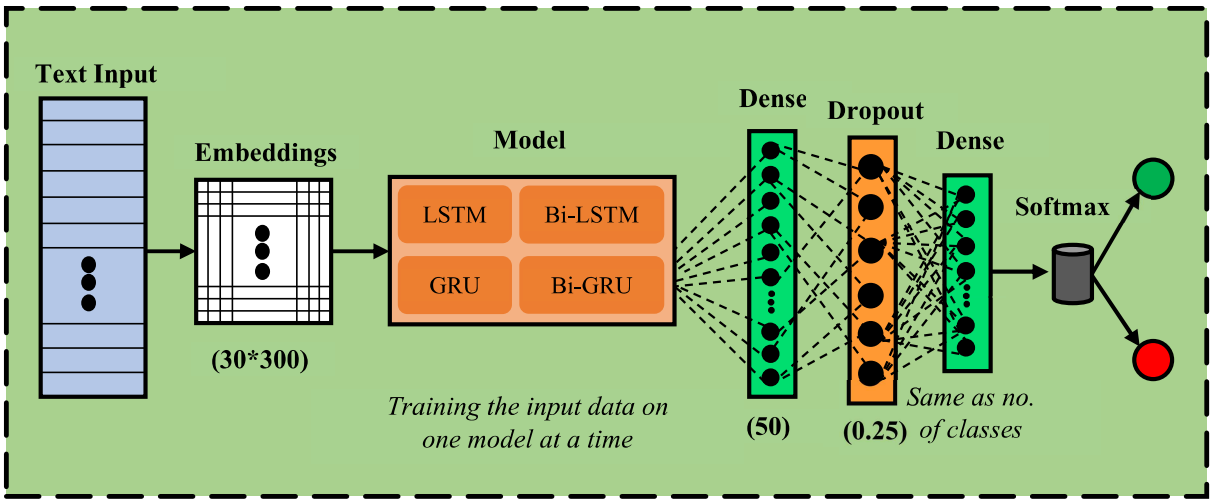

Figure 3: Architectural representation of framework for using LSTM, Bi-LSTM, GRU, and Bi-GRU

CNN-BiLSTM: We propose a combination network constituting a convolutional and a BiLSTM layer. The model is illustrated in Figure 4. The input text, after undergoing embedding, is initially fed to a BiLSTM layer of size 100. Features from this layer further undergo convolution operation using a one-dimensional convolutional layer with a ReLU activation. Outputs are processed under max-pooling operation and passed on to the series of fully connected layers, as depicted in Figure 2. This combination allows us to use the retentive power of LSTMs and feature extraction capability of CNNs, thus forming a more robust classifier.

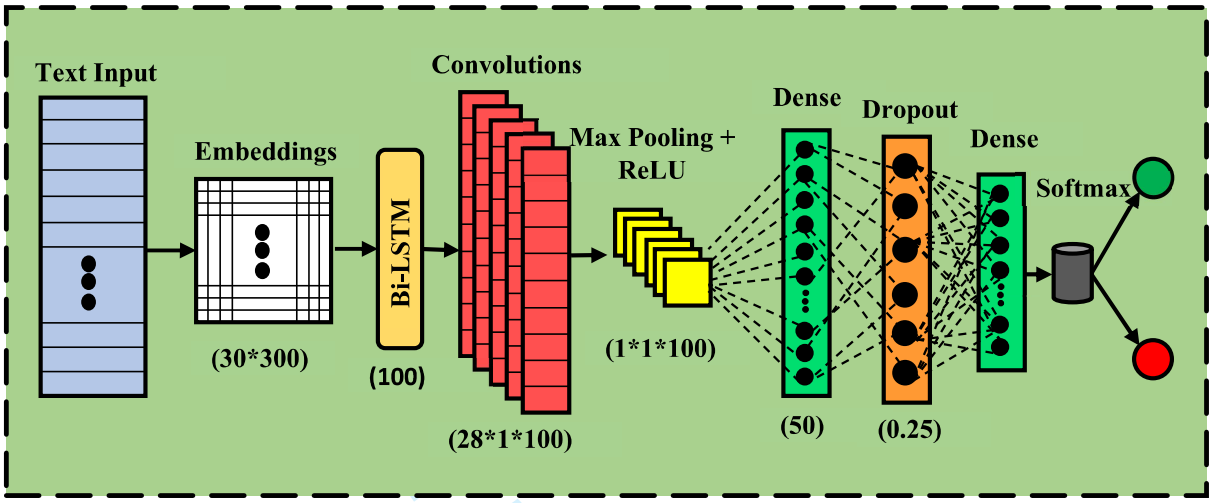

Figure 4: Architectural representation of CNN-BiLSTM framework

Attention-BiLSTM: Another model that we propose combines a Bi-LSTM network with a hierarchical attention model. As suggested by the name, the attention model [50] pays special attention to words having higher importance in the document. In the proposed architecture represented in Figure 5, information processed through the Bi-LSTM network is passed through an attention layer with multiple neurons and then to the fully connected layers. The mechanism encodes only selective valuable information by understanding the context and enhancing the final output. This allows the model to run successfully on sufficiently large input texts. The model assigns weights because only a few of the input items shall be used to compute the outputs. The context vector c_i for an output y_i is calculated by Eq. 20 where α_{ij} is the set of attention weights and T_x denotes the desired window size.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \quad (20)$$

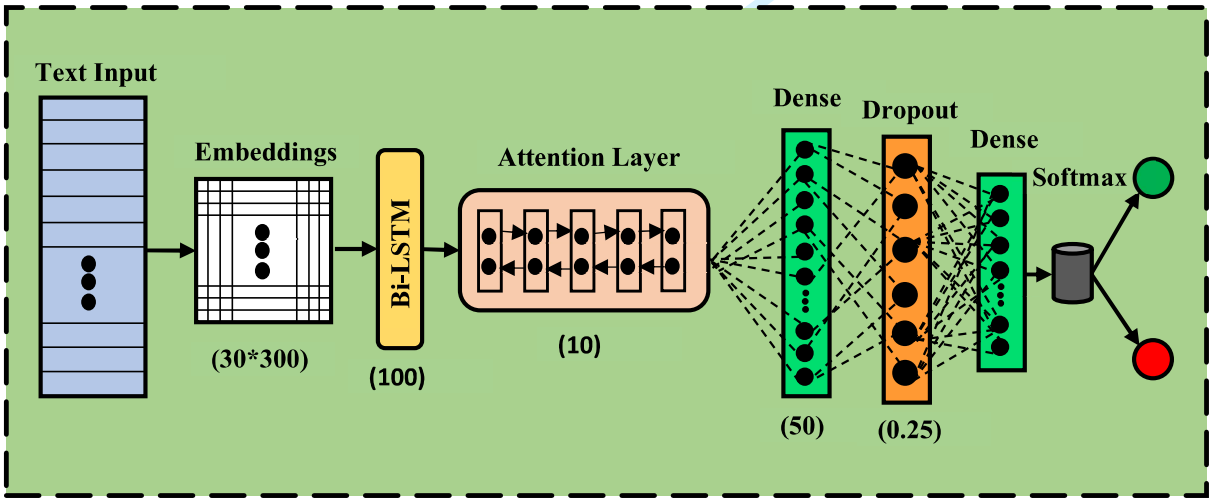

Figure 5: Architectural representation of Attention-BiLSTM framework

4. Experimental Result Analysis

In this section, we describe the datasets used and report the experiment results. The results are evaluated using four metrics, accuracy, precision, recall, and F1 scores. We graphically compare the performances of all

algorithms used on two real-world datasets. The baseline comparison with existing literature is provided in subsection 4.3

4.1 Datasets

Wikipedia Attack Dataset [21]**:** This corpus was crowdsourced by Wulczyn et al. in 2017 using Wikipedia articles. The dataset consists of discussion comments in the English language extracted from the 'talk page' of Wikipedia websites. The comments are extracted by accessing the revision history of Wikipedia pages to get all interactions, including removed comments. The collected corpus is cleaned by removing HTML content and keeping plain text only. The dataset is strapped out of all bot messages, and only human-made comments are retained. The dataset version that we use consists of 1,15,864 user comments with 13,590 cyberbullying text and 1,02,274 non-cyber bullying comments. Figure 6 illustrates the word clouds for attack and non-attack classes for this dataset.

(a) (b)

Figure 6: Word clouds of Wikipedia Attack dataset (a) Attack (b) Non-Attack

Wikipedia Web Toxicity Dataset [21]**:** Another corpus of comments is proposed by Wulczyn et al. from Wikipedia collected from 'article talk namespace'. It is a binary labeled corpus with 159689 comments containing 15,365 toxic and 1,44,324 non-toxic comments. The scraping procedure of the dataset is the same as the Wikipedia Attack dataset using the revision history of article pages. The discussion comments are collected, and administrative and bot comments are removed to constitute the final corpus. Figure 7 illustrates the word clouds for toxic and non-toxic classes for this dataset.

(a) (b)

Figure 7: Word clouds of Wikipedia Web Toxicity dataset (a) Toxic (b) Non-Toxic

4.2 Result Analysis

We evaluate and compare the results of all classifiers used on two datasets. Table 1 illustrates the experimental results of machine learning algorithms using four feature extraction approaches. The results on XGBoost, Naïve Bayes, SVM, and Logistic Regression are graphically represented in Figures 8, and 9. Table 2 illustrates the results of deep learning classifiers. The performance comparison for deep learning approaches is shown in Figures 10, and 11. While accuracy is the simplest and most intuitive metric of model performance, it is not suitable for unbalanced datasets. The precision and recall as well as F1-measure (which is the harmonic mean of precision and recall) have also been reported, and their performance is also over 90% for majority of cases.

Table 1: Results of four machine learning models with four feature extraction techniques on two datasets

Feature Extraction	Machine Learning Algorithms	Wikipedia Attack Dataset				Wikipedia Web Toxicity			
		A	P	R	F1	A	P	R	F1
Count Vectorization	XG Boost	94.43	99.55	95.36	**97.14**	94.19	99.14	94.35	97.21
	Naïve Bayes	**95.44**	**99.73**	96.27	96.92	95.62	98.48	96.9	**98.14**
	SVM	93.8	96.78	**96.6**	96.24	95.55	98.65	**97.82**	97.88
	Logistic Regression	94.55	98.34	96.11	96.79	**96.49**	**99.39**	97.55	97.73
TFIDF Word unigram	XG Boost	**95.23**	**99.51**	95.6	97.86	94.33	**99.95**	94.46	96.82
	Naïve Bayes	91.34	93.12	90.67	95.42	92.57	97.56	92.89	96.36
	SVM	95.02	98.94	**96.41**	**98.12**	**96.29**	99.93	**97.59**	**98.77**
	Logistic Regression	95.11	99.09	95.27	97.18	95.91	99.33	96.43	98.55
TFIDF Word bigram & trigram	XG Boost	93.26	**98.71**	92.93	**96.79**	92.38	96.46	91.74	96.12
	Naïve Bayes	89.11	92.12	89.25	94.77	91.81	98.12	90.91	95.54
	SVM	**93.58**	98.52	**94.51**	96.31	**95.32**	**100**	**95.64**	**97.97**
	Logistic Regression	91.6	94.55	91.76	96.13	93.24	97.17	92.93	96.82
TFIDF Char bigram & trigram	XG Boost	93.87	**99.97**	94.22	96.86	95	**99.74**	94.62	97.8
	Naïve Bayes	91.24	99.9	90.91	96.05	92.07	99.8	92.91	96.64
	SVM	81.01	98.92	69.05	81.24	**96.21**	99.35	**97.34**	**98.32**
	Logistic Regression	**95.16**	99.52	**95.29**	**97.13**	96.13	99.46	96.45	98.15

Table 1: Results of seven deep learning models with three embedding techniques on two datasets

Feature Extraction	Deep Learning Algorithm	Wikipedia Attack Dataset				Wikipedia Web Toxicity Dataset			
		A	P	R	F1	A	P	R	F1
GloVe	CNN	95.6	98.43	96.69	97.16	96.53	99.33	97.56	**98.69**
	LSTM	95.08	98.53	96.85	97.61	96.44	99.14	97.24	98.34
	GRU	95.46	99.19	96.17	97.16	96.24	98.99	97.28	98.42
	Bi-LSTM	96.88	98.29	**97.01**	98.1	95.99	98.94	97.48	98.05
	Bi-GRU	**96.98**	**99.22**	96.74	**98.56**	96.01	99.45	96.8	98.63
	CNN-BiLSTM	95.36	98.88	96.84	98.03	96.74	**99.63**	96.78	98.07
	Att-BiLSTM	95.4	98.45	96.91	97.88	**96.84**	98.5	**97.84**	98.06
FastText	CNN	94.94	98.21	96.54	97.6	96.36	99.19	**97.87**	97.88
	LSTM	95.34	99.22	**96.82**	97.25	96.57	99.64	96.72	98.37
	GRU	95.8	98.72	96.72	**98.23**	95.93	98.42	97.17	98.55
	Bi-LSTM	**95.93**	98.81	96.58	98.15	96.51	99.18	97.34	98.07
	Bi-GRU	95.5	**99.37**	96.06	97.68	96.45	98.12	96.61	98.58
	CNN-BiLSTM	95.1	98.72	96.06	97.05	96.1	**99.71**	97.23	98.2
	Att-BiLSTM	95.01	99.12	96.51	97.81	**96.6**	99.18	97.06	**98.65**
Paragram	CNN	95.25	98.2	97.36	97.36	96.31	99.41	97.06	**98.61**
	LSTM	95.32	98.28	96.51	98.01	96.18	99.09	96.82	97.81
	GRU	95.78	98.03	97.39	97.48	96.58	98.92	96.83	98.22
	Bi-LSTM	95.12	**99.17**	96.74	97.95	95.87	99.41	96.63	98.46
	Bi-GRU	**95.88**	98.66	**97.6**	97.82	96.65	**99.92**	97.27	98.43
	CNN-BiLSTM	94.94	97.76	96.63	97.64	96.53	99.35	97.31	98.34
	Att-BiLSTM	95.12	98.21	96.73	**98.03**	**96.77**	98.91	**97.72**	98.49

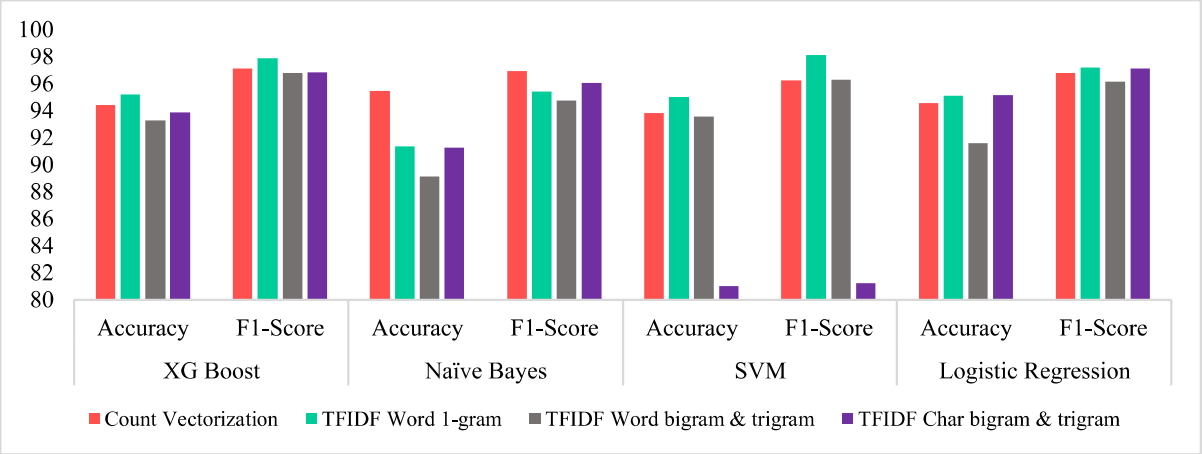

Figure :8 Performance comparison of Machine learning techniques on Wikipedia Attack dataset

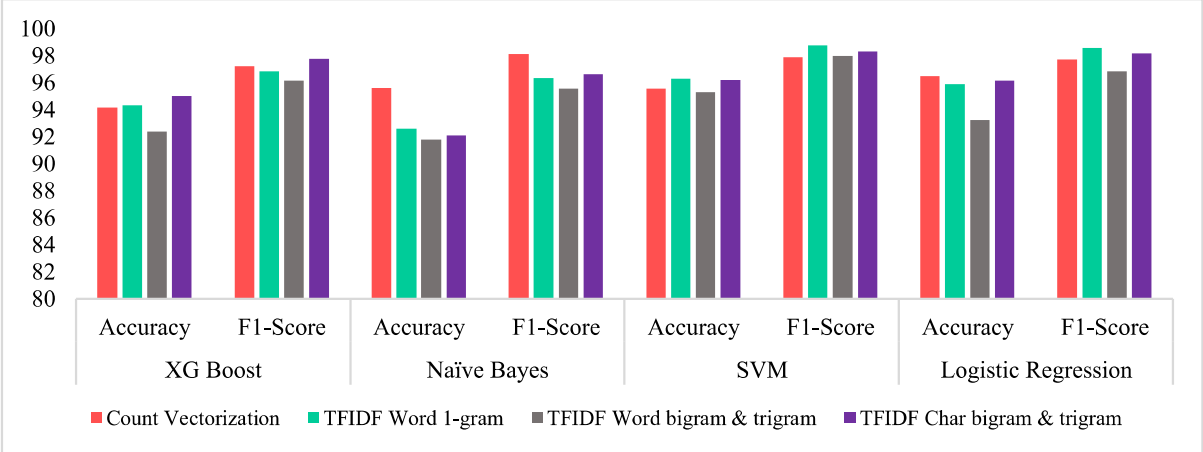

Figure 9: Performance comparison of Machine learning techniques on Wikipedia Web Toxicity dataset

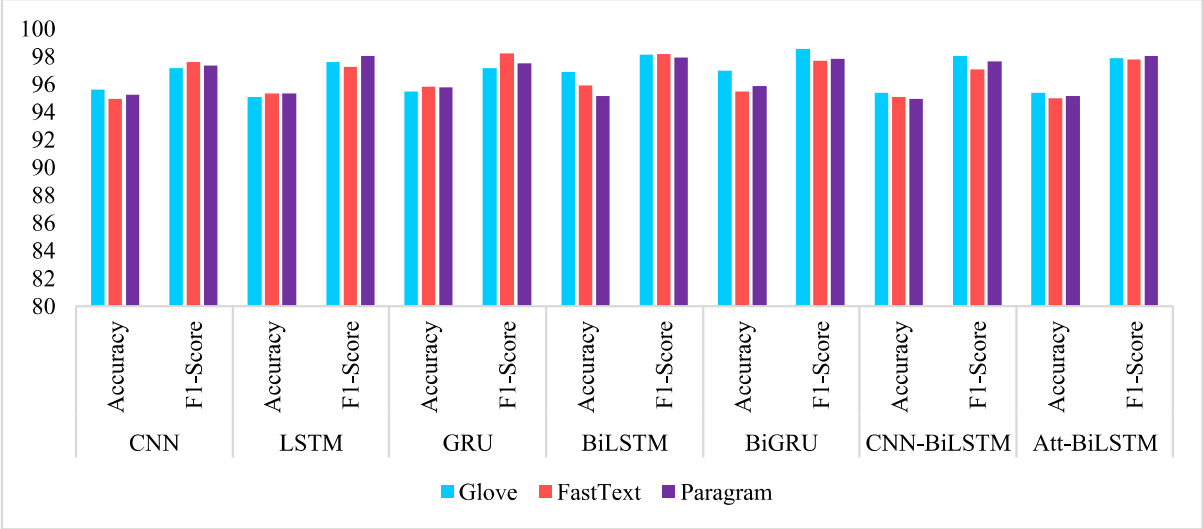

Figure 10: Performance comparison of Deep learning techniques on Wikipedia Attack dataset

For the Wikipedia Attack Dataset, we observe that SVM achieves the highest F1-score of 98.12% using TF-IDF word unigram, followed by XG Boost giving 97.14% and Logistic Regression giving 97.13% F1-scores with Count Vectorization and TF-IDF character bigram & trigram respectively. XG Boost and Logistic Regression appear to perform better than Naïve Bayes on this dataset, despite Naïve Bayes achieving the highest accuracy of 95.44% using Count Vectorization. SVM demonstrates a lower performance on this dataset, especially with TFIDF character bigram and trigram. For the Wikipedia Web Toxicity dataset, highest F1-score of 98.77% is

displayed by SVM with TFIDF word unigram embeddings. SVM with 98.32% F1-score follows it using TFIDF character bigram & trigram. Naïve Bayes using Count Vectorization follows it with 98.14% score. Overall, Logistic Regression demonstrates high results with all types of feature extraction techniques. SVM follows it, followed by XG Boost and then Naïve Bayes.

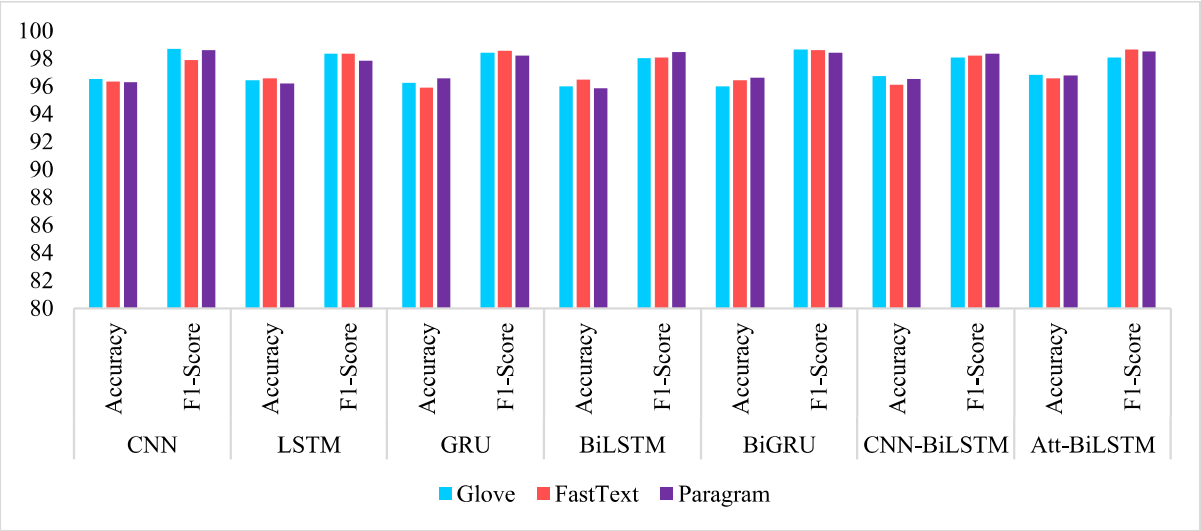

Figure 11: Performance comparison of Deep learning techniques on Wikipedia Web Toxicity dataset

Observing the performances of deep learning models, majority of the results are over 95% considering all evaluation measures. Also, these figures are higher than those reported for machine learning models. For the Wikipedia Attack dataset, Bi-GRU with GloVe embeddings is the best performing model with 98.56% F1-score and 96.98% accuracy. We observe that GloVe embeddings offer a higher and consistent rate in better classification than Paragram and FastText. Though the other two embedding methods have also performed greatly, GloVe is simply the winner. Amongst the deep learning models, Bi-LSTM and Bi-GRU models performed better than the rest. For the Wikipedia Web Toxicity Dataset, Bi-LSTM models have demonstrated great performance. The highest goes to 98.69% F1-score with GloVe embeddings followed by 98.65 with Attention-BiLSTM using FastText and then 98.61% with CNN using Paragram. On this dataset, the best performing models are CNN-BiLSTM, Attention-BiLSTM and BiGRU. Though the results, majorly over 95% are quite similar we assume that the proposed framework allows all the above models to perform classification with high accuracies.

Summarizing the results, all common metrics indicate good performance. While accuracy is well over 90% for all deep learning models and over 80% for all machine learning models across all datasets. The results obtained through deep neural networks incorporated in the architecture that we propose achieve higher results than traditional machine learning algorithms. With only a simple vanilla layer, each neural network architecture provides high performances. Additionally, the parametric settings of the network, neuron count, size of fully connected dense layers, and dropout probabilities which have been decided upon experimentation, yield optimum results. The proposed architecture is observed to work well with all the neural networks utilized. We summarize the key observations as follows:

1. Deep neural networks demonstrated higher performance than state-of-the-art machine learning algorithms, owing to their robustness and capability to handle large datasets.
2. Count Vectorization, though being an old statistical technique manages to consistently provide good results.
3. Logistic Regression displayed highest performances amongst all machine learning techniques used, followed by SVM, XG Boost and Naïve Bayes in the said order.
4. GloVe embeddings resulted in maximum number of high outputs than FastText and Paragram. Though, similar results are achieved by the other two methods in a pretty fashion.
5. F1-measures convey high performance through all deep learning models. Observing the accuracy scores alongwith, we conclude that RNN networks like GRU, Bi-GRU, and Bi-LSTM offered highest performance. Attention mechanism is also close to achieving results similar to these.

4.3 Baseline Comparison

To validate the efficacy of our work, we performed a baseline comparison with recent state-of-the-art techniques for cyberbully detection. The comparison of results on two datasets is provided in Table 3. The

techniques used for comparison have been re-implemented in accordance with the environment settings mentioned in the existing works. Bourgonje et al. [51] performed attack detection using Bayes, Bayes Expectation Maximization, C4.5 Decision Trees, Multivariate Logistic Regression, Maximum Entropy and Winnow2 on Wikipedia dataset [21]. The highest performance is obtained by their Bayes Expectation Maximization on Wikipedia Attack dataset with 67.40% accuracy and by Logistic Regression on Wikipedia Web Toxicity dataset with 69.98% accuracy. Agrawal and Awekar [52] used CNN, LSTM, Bi-LSTM and Att-BiLSTM on the Wikipedia dataset achieving highest accuracies of 92.91% and 93.52% with their CNN model. Bodapati et al. [53] achieved accuracy scores of 95.34% and 95.69% on Wikipedia Attack and Wikipedia Web Toxicity datasets, respectively using Bidirectional Encoder Representations from Transformers (BERT). As observed by Table 3, results obtained by our proposed methods have outperformed the existing approaches on these datasets. On the Wikipedia Attack dataset, our proposed model, Bi-GRU with GloVe embedding technique achieves 96.98% accuracy, and 98.56% F1-score, higher than the existing methods. On the Wikipedia Web Toxicity Dataset, the results achieved by Bi-GRU with GloVe embeddings outperform the existing baselines with 96.01% accuracy and 98.63% F1-score. The evaluation metrics descripted in Table 3 validate the efficiency of our proposed methods. In addition, most of our proposed models outperform the existing-state-of-the-art, as observable from Table 2. For the Wikipedia Attack Dataset, the precision, recall and f-measures achieved from all our deep learning methods are higher than the existing methods [35, 51, 52, 53]. For the Wikipedia Web Toxicity dataset, all of the results achieved using deep learning methods are exceptionally higher than the existing ones.

Table 2: Baseline comparison on Wikipedia Attack and Wikipedia Web Toxicity datasets

Dataset	Method	Accuracy	Precision	Recall	F1-Score
Wikipedia Attack Dataset	Bourgonje et al. [51] (Bayes Exp. Max)	67.40	56.05	66.94	59.00
	Agrawal et al. [35] (Att-BiLSTM)	--	89.00	86.00	88.00
	Agrawal and Awekar [52] (CNN)	92.91	92.09	83.78	88.63
	Bodapati et al. [53] (BERT)	95.34	92.61	93.57	95.70
	Our Approach (Bi-GRU with GloVe)	**96.98**	**99.22**	**96.74**	**98.56**
Wikipedia Web Toxicity Dataset	Bourgonje et al. [51] (Logistic Regression)	69.98	64.31	68.02	68.34
	Agrawal and Awekar [52] (CNN)	93.52	92.79	88.67	91.56
	Bodapati et al. [53] (BERT)	95.69	92.71	95.11	96.82
	Our Approach (Bi-GRU with GloVe)	**96.01**	**99.45**	**96.8**	**98.63**

5. Conclusion

With the expansion in the online space, cyberbullying has emerged as a ubiquitous problem having dire consequences on people and society. This research focuses on investigating several dimensions of cyberbullying detection. We explored eleven classification techniques, including machine learning and deep learning algorithms. In addition, we have also used seven types of feature extraction and embedding techniques. The results are established through experiments on two real-world datasets. We propose a novel deep learning framework, establishing optimum network settings, dense and dropout layer sizes. The framework accommodates various classifiers and achieves high results overall, outperforming several baselines. We provided a comparative study discussing the performances of all the methods utilized. The results are compared on a scale of four evaluation metrics to establish the concreteness of this study. The usefulness of this study lies in identifying robust mechanisms for online cyberbullying detection. We observe that deep learning highly outperforms traditional machine learning algorithms. We establish that bidirectional neural networks perform better in all scenarios. The attention mechanism is also observed to perform exceptionally well. We observe that older algorithms like SVM, Naïve Bayes, XGBoost, and Logistic Regression provide lower results compared to other methods. Overall, we suggest using bidirectional RNNs, and attention-based models for further advances in cyberbullying detection. This study paves a way towards developing better mechanisms to fight this online ailment.

References

[1] Moreno, M. A. (2014). Cyberbullying. *JAMA pediatrics*, *168*(5), 500-500.

[2] Bu, S. J., & Cho, S. B. (2018, June). A hybrid deep learning system of CNN and LRCN to detect cyberbullying from SNS comments. In *International Conference on Hybrid Artificial Intelligence Systems* (pp. 561-572). Springer, Cham.

[3] Mishra, P., Del Tredici, M., Yannakoudakis, H., & Shutova, E. (2018, August). Author profiling for abuse detection. In *Proceedings of the 27th international conference on computational linguistics* (pp. 1088-1098).

[4] Pavlopoulos, J., Malakasiotis, P., Bakagianni, J., & Androutsopoulos, I. (2017). Improved abusive comment moderation with user embeddings. *arXiv preprint arXiv:1708.03699*.

[5] Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017, May). Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 11, No. 1).

[6] Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., & Bhamidipati, N. (2015, May). Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web* (pp. 29-30)..

[7] Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016, April). Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web* (pp. 145-153).

[8] Muneer, A., & Fati, S. M. (2020). A comparative analysis of machine learning techniques for cyberbullying detection on twitter. *Future Internet*, *12*(11), 187.

[9] Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017, April). Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on World Wide Web companion* (pp. 759-760).

[10] Rawat, C., Sarkar, A., Singh, S., Alvarado, R., & Rasberry, L. (2019, April). Automatic Detection of Online Abuse and Analysis of Problematic Users in Wikipedia. In *2019 Systems and Information Engineering Design Symposium (SIEDS)* (pp. 1-6). IEEE.

[11] Waseem, Z., & Hovy, D. (2016, June). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop* (pp. 88-93).

[12] Kim, Y. (2014). Convolutional neural networks for sentence classification. EMNLP.

[13] Lu, N., Wu, G., Zhang, Z., Zheng, Y., Ren, Y., & Choo, K. K. R. (2020). Cyberbullying detection in social media text based on character-level convolutional neural network with shortcuts. *Concurrency and Computation: Practice and Experience*, *32*(23), e5627.

[14] Zhang, X., Tong, J., Vishwamitra, N., Whittaker, E., Mazer, J. P., Kowalski, R., ... & Dillon, E. (2016, December). Cyberbullying detection with a pronunciation based convolutional neural network. In *2016 15th IEEE international conference on machine learning and applications (ICMLA)* (pp. 740-745). IEEE.

[15] Schmidt, A., & Wiegand, M. (2017, April). A survey on hate speech detection using natural language processing. In *Proceedings of the fifth international workshop on natural language processing for social media* (pp. 1-10).

[16] Qaiser, S., & Ali, R. (2018). Text mining: use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications*, *181*(1), 25-29.

[17] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*.

[18] Shah, F. P., & Patel, V. (2016, March). A review on feature selection and feature extraction for text classification. In *2016 international conference on wireless communications, signal processing and networking (WiSPNET)* (pp. 2264-2268). IEEE.

[19] Dzisevič, R., & Šešok, D. (2019, April). Text classification using different feature extraction approaches. In *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)* (pp. 1-4). IEEE.

[20] Reynolds, K., Kontostathis, A., & Edwards, L. (2011, December). Using machine learning to detect cyberbullying. In *2011 10th International Conference on Machine learning and applications and workshops* (Vol. 2, pp. 241-244). IEEE.

[21] Wulczyn, E., Thain, N., & Dixon, L. (2017, April). Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th international conference on world wide web* (pp. 1391-1399).

[22] Warner, W., & Hirschberg, J. (2012, June). Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media* (pp. 19-26).

[23] Kwok, I., & Wang, Y. (2013, June). Locate the hate: Detecting tweets against blacks. In *Twenty-seventh AAAI conference on artificial intelligence*.

[24] Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016, April). Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web* (pp. 145-153).

[25] Yin, D., Xue, Z., Hong, L., Davison, B. D., Kontostathis, A., & Edwards, L. (2009). Detection of harassment on web 2.0. *Proceedings of the Content Analysis in the WEB*, *2*, 1-7.

[26] Tokunaga, R. S. (2010). Following you home from school: A critical review and synthesis of research on cyberbullying victimization. *Computers in human behavior*, *26*(3), 277-287.

[27] Agrawal, S., & Awekar, A. (2018, March). Deep learning for detecting cyberbullying across multiple social media platforms. In *European conference on information retrieval* (pp. 141-153). Springer, Cham.

[28] Aroyehun, S. T., & Gelbukh, A. (2018, August). Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)* (pp. 90-97).

[29] Aglionby, G., Davis, C., Mishra, P., Caines, A., Giannakoudaki, H. Y., Rei, M., ... & Buttery, P. (2019). CAMsterdam at SemEval-2019 Task 6: Neural and graph-based feature extraction for the identification of offensive tweets.

[30] Chen, H., McKeever, S., & Delany, S. J. (2019, July). The use of deep learning distributed representations in the identification of abusive text. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 13, pp. 125-133).

[31] Chu, T., Jue, K., & Wang, M. (2016). Comment abuse classification with deep learning. *Von https://web. stanford. edu/class/cs224n/reports/2762092. pdf abgerufen*.

[32] Anand, M., & Eswari, R. (2019, March). Classification of abusive comments in social media using deep learning. In *2019 3rd international conference on computing methodologies and communication (ICCMC)* (pp. 974-977). IEEE.

[33] Pavlopoulos, J., Malakasiotis, P., & Androutsopoulos, I. (2017). Deep learning for user comment moderation. *arXiv preprint arXiv:1705.09993*.

[34] Banerjee, V., Telavane, J., Gaikwad, P., & Vartak, P. (2019, March). Detection of cyberbullying using deep neural network. In *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)* (pp. 604-607). IEEE.
[35] Agarwal, A., Chivukula, A. S., Bhuyan, M. H., Jan, T., Narayan, B., & Prasad, M. (2020, November). Identification and Classification of Cyberbullying Posts: A Recurrent Neural Network Approach using Under-sampling and Class Weighting. In *International Conference on Neural Information Processing* (pp. 113-120). Springer, Cham.
[36] "sklearn.feature_extraction.text.CountVectorizer — scikit-learn 0.19.0 documentation." [Online].
[37] Shi, C. Y., Xu, C. J., & Yang, X. J. (2009). Study of TFIDF algorithm. *Journal of Computer Applications*, *29*(6), 167-170.
[38] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).
[39] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135-146.
[40] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.
[41] Wieting, J., Bansal, M., Gimpel, K., & Livescu, K. (2015). From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*, *3*, 345-358.
[42] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., & Cho, H. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, *1*(4), 1-4.
[43] Murphy, K. P. (2006). Naive bayes classifiers. *University of British Columbia*, *18*(60), 1-8.
[44] Sarkar, A., Chatterjee, S., Das, W., & Datta, D. (2015). Text classification using support vector machine. *International Journal of Engineering Science Invention*, *4*(11), 33-37.
[45] Wright, R. E. (1995). Logistic regression.
[46] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.
[47] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.
[48] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, *45*(11), 2673-2681.
[49] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
[50] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).
[51] Bourgonje, P., Moreno-Schneider, J., Srivastava, A., & Rehm, G. (2017, September). Automatic classification of abusive language and personal attacks in various forms of online communication. In *International Conference of the German Society for Computational Linguistics and Language Technology* (pp. 180-191). Springer, Cham.
[52] Agrawal, S., & Awekar, A. (2018, March). Deep learning for detecting cyberbullying across multiple social media platforms. In *European conference on information retrieval* (pp. 141-153). Springer, Cham.
[53] Bodapati, S. B., Gella, S., Bhattacharjee, K., & Al-Onaizan, Y. (2019). Neural word decomposition models for abusive language detection. *arXiv preprint arXiv:1910.01043*.