Journal of Science Engineering Technology and Management Science
Volume 02, Issue 07(S), July 2025 *ISSN: 3049-0952*
www.jsetms.com *DOI:10.63590/jsetms.2025.v02.i07(S).40-51*

DEEPSCAN: AN INTELLIGENT DOCUMENT CLASSIFIER FOR CYBERBULLYING DETECTION IN SOCIAL MEDIA

Dr. B. Rama, Maheshuni Arun, Yellamelly Poorna Mani Tej, Chintham Sai Manish
Department of Computer Science and Engineering (AI&ML), Kommuri Pratap Reddy Institute of Technology, Ghatkesar, Medchal, 500088.

To Cite this Article
Dr. B. Rama, Maheshuni Arun, Yellamelly Poorna Mani Tej, Chintham Sai Manish, "Deepscan: An Intelligent Document Classifier For Cyberbullying Detection In Social Media", Journal of Science Engineering Technology and Management Science, Vol. 02, Issue 07(S), July 2025,pp: 40-51, DOI: http://doi.org/10.63590/jsetms.2025.v02.i07(S).pp40-51

Submitted: 25-05-2025 *Accepted: 03-07-2025* *Published: 11-07-2025*

ABSTRACT

The rapid growth of online social networks has led to an explosion of user-generated content, with over 4.5 billion active users worldwide producing around 500 million tweets each day. This vast influx of data poses major challenges for traditional content moderation methods, which struggle to manage the volume and complexity of information. As a result, issues such as cyberbullying and misinformation persist. Conventional classification techniques often depend on manual efforts that are time-consuming and susceptible to human error, delaying effective responses. To address this, the study introduces a novel deep learning-based intelligent document classifier aimed at automatically categorizing social media content, with a focus on Twitter. The system employs advanced text preprocessing and N-gram feature extraction, utilizing a Convolutional Neural Network (CNN) to classify tweets into multiple categories, including religion, age, gender, ethnicity, and signs of cyberbullying. This automated solution improves the speed and accuracy of content moderation, contributing to more efficient and responsible online community management.

Keywords: Cyberbullying Detection, Social Media Analysis, User-Generated Content, Text Classification.

1. INTRODUCTION

The Internet has become an integral part of everyday life, with social media evolving from basic web pages (Web 1.0) to intelligent Web 4.0 services. Technological advancements have transformed how information is accessed and how connections between different entities are made to obtain services over the network. Social media, commonly referred to as social media platforms (SMPs), includes tools for social interactions such as Facebook, Twitter, Instagram, LinkedIn, Pinterest, Telegram, and YouTube. These platforms empower users, enabling thousands to connect globally, creating a widespread social and expressive phenomenon. As of January 2024, 5.35 billion people globally were internet users, comprising 66.2 percent of the population. Among them, 5.04 billion, or 62.3 percent of the world population, were active on social media platforms, emphasizing the widespread adoption and impact of digital connectivity today. Professional networking supports career growth, while communities based on shared interests bloom, fostering a sense of belonging.

Fig. 1: Cyberbullying Detection.

Different businesses leverage SMPs for marketing and engagement, while educational institutions use them to broaden learning opportunities. However, the comprehensive exchange of personal information raises data security issues and the risk of misuse. On SMPs, individuals can face humiliation, insults, cyber threats, and cyberbullying from anonymous users, exacerbated by the constant accessibility and the ability for some users to remain unidentified. Bullying through the use of digital technology is known as cyberbullying. Social media, messaging apps, gaming platforms, and mobile devices can be used for this purpose. It involves consistent behaviour intended to frighten or embarrass the targeted individuals. Examples include spreading false information about someone or sharing embarrassing pictures or videos of them on social media. Some other examples include using fake accounts or sending unpleasant, abusive, or threatening texts, images, or videos through messaging apps. While significant research has been conducted on cyberbullying detection in English, there is a growing need to address this issue in non-English contexts, such as the Bangla language. The Bangla-speaking community faces unique challenges related to cyberbullying, with linguistic nuances and cultural factors influencing the nature and manifestation of harmful online behaviours. Detecting and addressing cyberbullying in Bangla is essential for promoting a safe and inclusive digital environment for individuals communicating in this language.

2. LITERATURE SURVEY

Traditional studies on cyberbullying stand more on a macroscopic view. These studies focused on the statistics of cyberbullying, explored the definitions, properties, and negative impacts of cyberbullying and attempted to establish a cyberbullying measure that would provide a framework for future empirical investigations of cyberbullying [1, 2, 3]. As cyberbullying has captured more attention, various methods have been used for the detection of cyberbullying in a given textual content. An outstanding work is the one by Nahar et al. Their work used the Latent Dirichlet Allocation (LDA) to extract semantic features, TF-IDF values and second-person pronouns as features for training an SVM [4].

Kontostathis et al analyzed cyberbullying corpora using the bag-of-words model to find the most commonly used terms by cyberbullies and used them to create queries [5]. In the work of Ying et al, the Lexical Semantic Feature (LSF) provided high accuracy for subtle offensive message detection, and it reduced the false positive rate. In addition, the LSF not only examines messages, but it also examines the person who posts the messages and his/her patterns of posting. As the use of deep learning becomes more widespread, some deep learning-based approaches are also being used to detect cyberbullying.

The work of Agrawal and Awekar provided several useful insights and indicated that using learning-based models can capture more dispersed features on various platforms and topics [6]. The work of Bu and Cho provided a hybrid deep learning system that used a CNN and an LRCN to detect cyberbullying in SNS comments [7]. Since previous data-based work relied almost entirely on vocabulary knowledge, the challenge posed by unstructured data still exists. Some works observed that the content information in social media has many incorrect spellings, and in some cases, the users in social media intentionally obfuscate the words or phrases in the sentence to evade the manual and automatic detection [8, 9]. These extra words will expand the vocabulary and affect the various performances of the algorithm.

Waseem and Hovy performed a grid search over all possible feature set combinations. They found that using character n-grams outperforms when using word n-grams by at least 5 F1-points using similar features [10], and it is a creative way to reduce the impacts of misspellings. Al-garadi et al used a spelling corrector to amend words, but we believe that some mistakes in this particular task scenario hide the speaker's intentions and correcting the spelling will destroy the features in the original dataset [11]. Zhang et al innovatively attempted to use phonemes to overcome deliberately ambiguous words in their work. However, some homophones with different meanings will get the same expression after their conversion, and their methods cannot solve some misspellings that have no association in their pronunciations.

Previous psychological and sociological studies suggested that emotional information can be used to better understand bullying behaviours, and thon emoticons in social text messages conveyed the emotions of users [12]. Dani et al presented a novel learning framework called Sentiment Informed Cyberbullying Detection (SICD), which leveraged sentiment information to detect cyberbullying behaviours in social media. Unfortunately, in the past cyberbullying detection work, almost no work took into account these special symbols. As a common pre-processing technique, removing symbols and numbers destroys the features of the emojis in the original dataset.

3. PROPOSED SYSTEM

The development of a dynamic sentiment classification system addresses a pressing need in the era of global digital communication. Online platforms receive a constant influx of user-generated content in multiple languages, expressing opinions, emotions, and feedback across diverse contexts. Businesses, governments, and social platforms rely on the ability to understand this sentiment in real time for decision-making, content moderation, marketing analysis, and customer service optimization. A robust sentiment classifier capable of processing multilingual data in real time empowers stakeholders to respond effectively to emerging trends and user behavior.

This Research establishes a real-time sentiment analysis system using Recurrent Neural Networks (RNNs), with a specific focus on handling multiple languages simultaneously. Traditional sentiment analysis tools often fail to scale across languages without extensive retraining or language-specific tuning. In contrast, this system incorporates both word-level and character-level modeling to address the linguistic diversity inherent in global communication. Word-level features capture the semantic structure of sentences, while character-level inputs ensure robustness against spelling errors, regional dialects, and informal language commonly found in social media and messaging platforms.

The architecture employs a recurrent neural network framework due to its proven ability to handle sequential data such as text. Specifically, Long Short-Term Memory (LSTM) units are integrated to manage long-term dependencies and retain contextual information throughout the sentence. By combining both LSTM-driven RNNs and advanced tokenization strategies, the classifier accurately predicts sentiment polarity in real time. The system continuously adapts to incoming data, ensuring its performance remains consistent across languages and text patterns without manual intervention.

To support multilingual capability, the model is trained on diverse datasets that encompass sentiment expressions in various global languages. Tokenizers and embedding layers are configured to process Unicode text effectively, allowing seamless switching between different linguistic inputs. Furthermore, the system includes a dynamic preprocessing module that standardizes text, expands common contractions, translates emoticons, and filters noise such as URLs and hashtags, thus enhancing model accuracy and generalization.

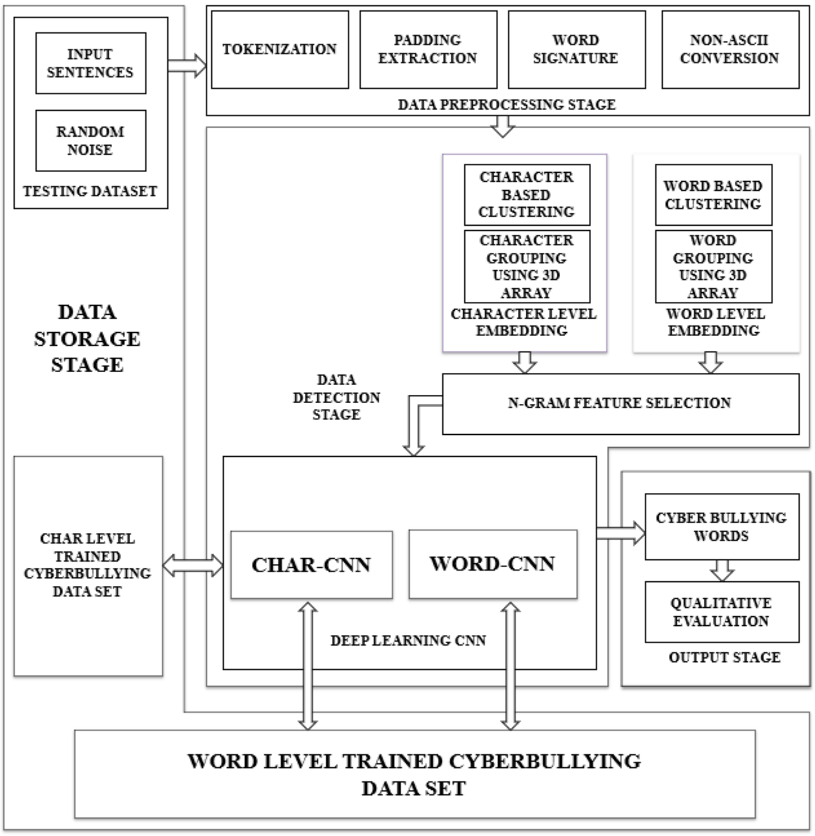


Fig. 2: Proposed cyberbullying detection architecture.

Word signature

Unknown word handling module Unknown words are defined as the words which are not in the lexicon or in reference sentences. Since CNN algorithm generate error as it detects unknown word therefore a separate module is required for tag decision for unknown word. In case of cyberbullying scenario, the attackers use the complicated abusive words; they may not be presented in the vocabulary. Thus, out of vocabulary words also considered for cyberbullying detection.

Non-ASCII conversion

Electronic processing of text in any language requires that characters (letters of the alphabet along with special symbols) be represented through unique codes, this is called encoding. Usually, this code will also correspond to the written shape of the letter. A NON-ASCII conversion is basically a number associated with each letter so that computers can distinguish between different letters through their codes.

Data detection stage

In the data detection stage character level, word level and synonym level embedding operation will be performed. In this embedding character recognition, word recognition and synonym recognition operations will be performed parallel manner to give the maximum efficiency to detect the cyberbullying. Then the data groups will be formed as 3D array using pattern matching operations. The selection of character level or word level or synonym level cyberbullying detection is performed by the user through user interface. Then corresponding 3d group array will be applied CNN.

Data Clustering

Data clustering plays an important role in organizing and grouping similar input data instances, which allows the sentiment classifier to operate more effectively in real-time and across multiple languages. Since input data comes from various applications like social networks, messaging platforms, and user feedback forms, clustering ensures that similar types of content are processed together. This leads to better contextual understanding, reduced noise, and more focused training. Clustering also aids in identifying domain-specific language patterns and improves training efficiency by minimizing redundancy in the dataset.

Feature Extraction from Preprocessed Text: Once the text has undergone preprocessing, the first step in clustering involves converting each input instance into a numerical representation. This is typically done using word embeddings, sentence vectors, or TF-IDF scores. These vectors capture semantic similarity and represent the content in a multidimensional space, making it suitable for clustering algorithms to operate.

Dimensionality Reduction: To make clustering more efficient and meaningful, dimensionality reduction is applied to the feature vectors. This step helps in eliminating redundant or less informative features and compressing the data into a lower-dimensional space. Techniques like Principal Component Analysis (PCA) or t-SNE are used depending on the nature and complexity of the data. This not only accelerates the clustering process but also improves cluster quality.

Choosing a Clustering Strategy: The next step involves selecting an appropriate clustering algorithm based on the structure of the data. If the number of clusters is known beforehand or estimated through evaluation, partition-based methods like k-means can be used. For more complex datasets with unknown structure, density-based or hierarchical clustering methods are preferable. The algorithm is selected to suit the diversity and volume of the application-specific data.

Cluster Formation: Using the chosen algorithm, the dataset is grouped into clusters based on vector similarity. Each cluster contains data instances that share similar patterns, sentiments, or linguistic features. This step allows the model to identify prevalent themes or sentiment trends within each group, which is especially useful when dealing with multilingual and informal text data.

Cluster Analysis and Label Assignment: Once clusters are formed, they are analyzed to identify their predominant sentiment or language characteristics. Statistical measures or sample reviews help assign a representative label or tag to each cluster. This labeling helps in guiding the sentiment classifier with contextual information and improves both training and inference performance.

Feedback and Refinement: The final step involves assessing the effectiveness of the clustering process through metrics such as cluster purity or silhouette score. Based on the evaluation, clusters may be refined by adjusting parameters or reprocessing the data. This iterative process ensures that the clusters remain meaningful and contribute positively to the downstream sentiment classification task.

Grouping using 3D array

A normalized longest common subsequence (NLCS) based string approximation method is proposed for indexing multidimensional data cube. In this indexing system, the reference table is made, and dimensional key values are stored for each dimension. A dimensional reference table is a set of dimensional key values stored in sorted order. The slot number of a key value in the dimensional reference table will be the index of the key value on the axis of multidimensional array. NLCS based string approximation is used to search a nearest keyword for a misspelled keyword, in the reference table and gets its slot number.

Normalized LCS based string approximation is used to design a character, word, and synonym (CWS) searching algorithm. This CWS searching algorithm gives near optimal solution to the string approximation problem. The algorithm finds the NLCS values of searched keyword with all the stored keywords in the set. The keywords in the set having NLCS value between 0.5 and 1 are the nearest neighbor of the searching keyword. The keyword closest to the searching keyword having highest NLCS value will be the optimal keyword. The CWS searching, finds the index of keyword, like searching keyword from the set of stored keywords and creates the 3d array group for easily detection of cyberbullying. So, the abusive words and its synonyms will be identified easily.

N-gram Feature selection

The N-gram model combined with latent representation on the data classification task. Their model called as supervised n-gram embedding uses a multi-layer perceptron to accomplish the embedding. The number of distinct character and word-based N grams in a text can be as high and its feature selection vector size extremely high even for moderate values of n. The **N-**gram Feature selection applied only on character and word-based embedding vectors as it does not apply on synonym based embedding vector. Because synonym-based vectors are classified initially in the synonym level embedding so there is no requirement to generate the features again. If the N-gram feature selection applied on synonym based embedding vectors, then classification accuracy will reduce because of original features will get loosed. However, only a small fraction of all possible character and word-based n grams will be present in any given set of documents, thereby reducing the dimensionality substantially. The dimensionality reduction problem is handled in the present work in two different approaches where one set of n grams are identified as valid N grams and other set is treated as invalid N grams. The adequacy of this model is also evaluated in terms of average information conveyed by valid N grams in comparison with invalid N grams.

N-grams are subsequences of n items (words in this case) extracted from the text data. The choice of n depends on the specific task and the level of granularity desired. Common values for n include 1 (unigrams), 2 (bigrams), and 3 (trigrams), but higher values can also be used for more context.

Proposed Deep-learning CNN

Convolutional Neural Networks (CNNs) are highly effective in processing application-specific textual data for sentiment classification, especially when dealing with short texts, informal language, or multilingual inputs from social platforms. They excel at capturing local patterns, such as key phrases or n-grams, which are essential for identifying sentiment-related cues. Unlike traditional models, CNNs can automatically learn hierarchical feature representations without relying on handcrafted rules. This makes them suitable for real-time applications where input data varies widely in structure, tone, and language. CNNs are also computationally efficient, making them a good fit for environments requiring fast sentiment inference.

Input Representation: The first step involves converting the clustered and preprocessed text into a structured format that can be fed into the CNN. Typically, this includes transforming each sentence into a matrix where each row corresponds to a word vector. These vectors are obtained through embedding techniques such as Word2Vec, GloVe, or contextual embeddings depending on the language and domain. This matrix preserves the word order and captures semantic relationships between terms.

Convolution Operation: After input representation, the CNN applies convolutional filters over the matrix. Each filter slides over the text matrix and detects local patterns such as word sequences, sentiment phrases, or emotional cues. Multiple filters of varying sizes are used to capture different n-gram features. The result of this operation is a set of feature maps that highlight the presence of meaningful patterns across the text.

Activation and Non-linearity: Once feature maps are generated, an activation function is applied to introduce non-linearity. This step allows the model to learn complex relationships between input patterns and sentiment outcomes. Functions like ReLU are commonly used to enhance the model's ability to generalize and differentiate between subtle emotional cues in the text.

Pooling Operation: To reduce dimensionality and focus on the most important features, pooling is applied to the activated feature maps. Max-pooling is often used, where the maximum value from a region of the feature map is selected. This helps retain the strongest signal or most relevant sentiment indicator from each region while discarding less important information. Pooling also adds robustness against small changes in input structure or noise.

Flattening and Fully Connected Layer: The pooled features are then flattened into a single long vector, which serves as the input to one or more fully connected layers. These layers combine the localized features into a global representation of the sentiment. They are responsible for learning high-level abstractions such as overall tone, emotion, or intent expressed in the text.

Classification and Output: In the final step, the output from the last fully connected layer is passed through a softmax or sigmoid layer depending on whether the task is multi-class or binary sentiment classification. This layer generates a probability distribution over sentiment classes, and the class with the highest probability is selected as the output sentiment label.

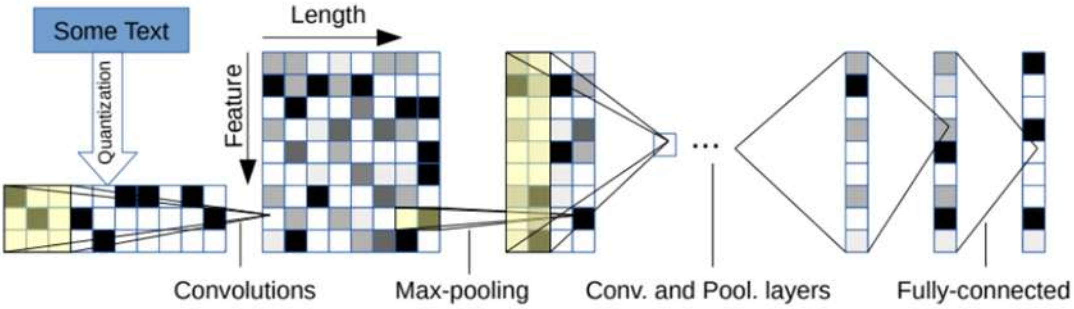


Fig. 3: Character Based CNN

Advantages of CNN

Handling Out-of-Vocabulary Words: Character-based CNNs can handle out-of-vocabulary words effectively since they operate directly on individual characters rather than predefined word tokens. This makes them robust to unseen or rare words, which can be particularly useful in tasks involving specialized domains or evolving language use.

Handling Short Texts: Character-based CNNs are well-suited for processing short texts, such as social media posts, tweets, or search queries. Unlike word-based models, which was struggle with short texts due to limited context, character-based models can capture important patterns and features at the character level, enabling effective analysis of short and contextually sparse text data.

Handling complex morphology : By processing text data at the character level, character-based CNNs can capture intricate patterns and structures, making them suitable for various natural language processing tasks, especially when dealing with languages with complex morphology or limited training data.

Model Size and Efficiency: Character-based CNNs can be more space-efficient compared to word-based models, especially when dealing with large vocabularies. Since they operate directly on characters, the size of the input vocabulary is typically smaller, leading to more compact models and faster training times.

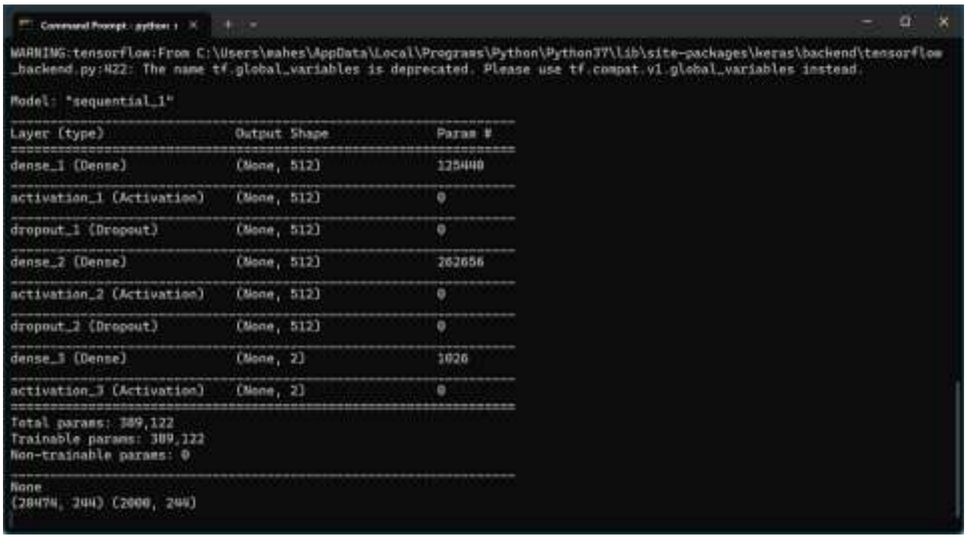
Robustness to Misspellings and Typos: Working on only characters also has the advantage that abnormal character combinations such as misspellings and emotions are naturally learnt.

Capturing Subword Information: They capture subword information, which can be beneficial for tasks like named entity recognition or sentiment analysis.

Enhanced Generalization: Character-based CNNs can generalize better across tasks and domains compared to word-based models, especially in scenarios where the vocabulary size varies significantly between training and testing data. By operating directly on characters, these models can adapt more readily to new vocabulary items and unseen word forms encountered during inference, leading to improved generalization performance.

Domain Adaptability: Character-based CNNs are highly adaptable to different domains and text genres, requiring minimal domain-specific preprocessing or feature engineering. This makes them suitable for a wide range of applications, including sentiment analysis, text classification, machine translation, and more, across various domains such as healthcare, finance, social media, and legal text.

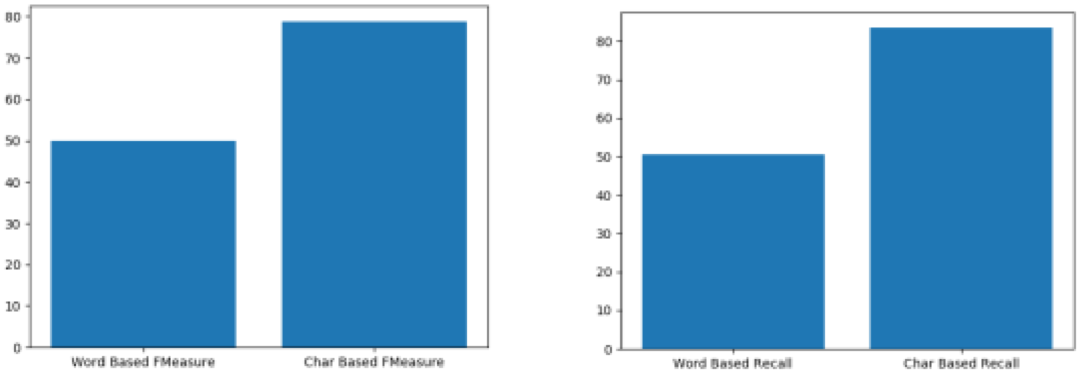
4. RESULTS AND DISCUSSION



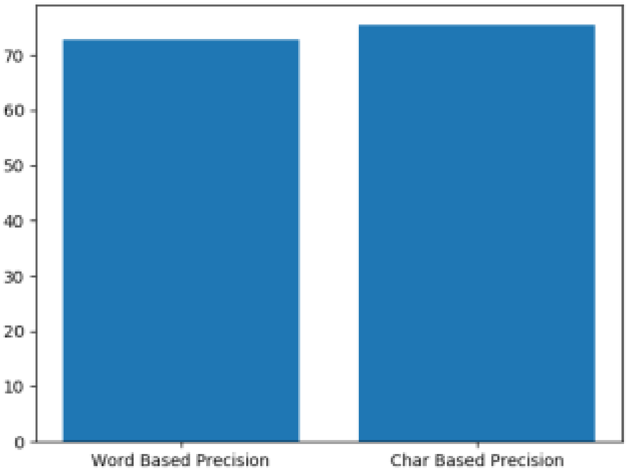
```
WARNING:tensorflow:From C:\Users\mahes\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\backend\tensorflow
_backend.py:422: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 512)               125440
_________________________________________________________________
activation_1 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 512)               262656
_________________________________________________________________
activation_2 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 1026
_________________________________________________________________
activation_3 (Activation)    (None, 2)                 0
=================================================================
Total params: 389,122
Trainable params: 389,122
Non-trainable params: 0
_________________________________________________________________
None
(28474, 244) (2000, 244)
```

Fig. 4: Architecture of Proposed CNN Model.

Figure 4 illustrates the detailed architecture of the Proposed Character-Based Convolutional Neural Network (Char-CNN) model used for text classification. It includes layers such as embedding, convolutional, pooling, and fully connected layers, showcasing how input text is processed through deep learning. The architecture highlights feature extraction, learning mechanisms, and how it improves classification performance.



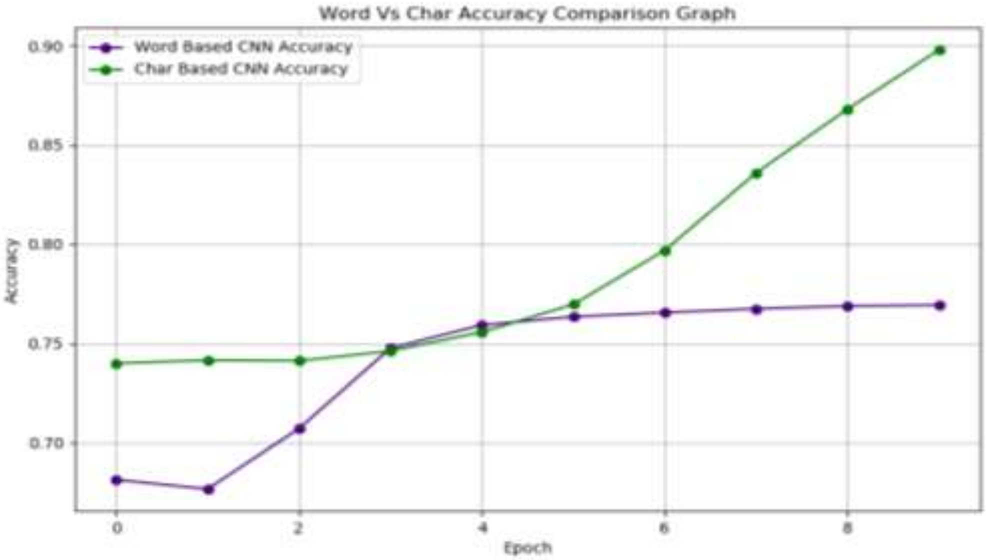
(a) (b)



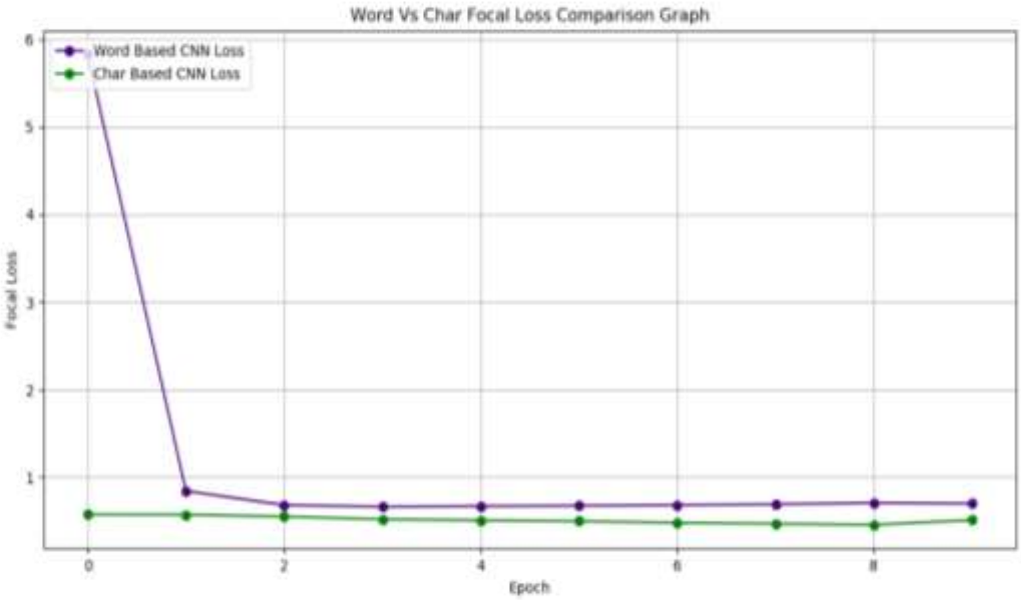
(c)

Fig. 5: Performance comparison of metrics Existing and Proposed Model. (a) Accuracy, (b) Precision, (c) Recall.

Figure 5 provides a comparative analysis of various performance metrics (Accuracy, Precision, Recall, and F1-Score) between the existing Word-CNN model and the proposed Char-CNN model. The comparison helps to demonstrate the effectiveness of the newly implemented model, showing improvements in classification accuracy and robustness against noisy text data.



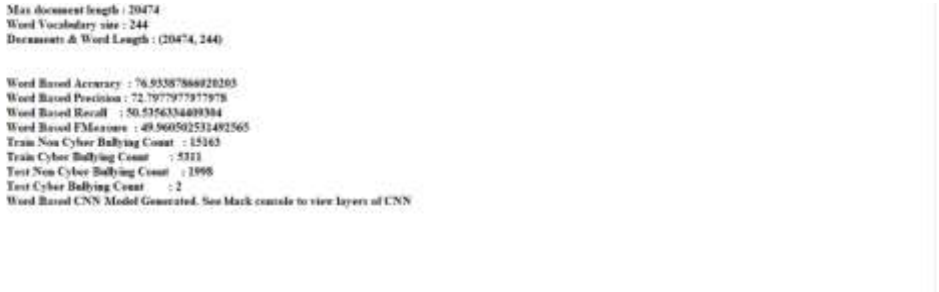
(a) Accuracy over Multiple Epochs.



(b) Loss over Multiple Epochs.

Fig. 6: Epoch vs Loss Comparison Plot of Existing and Proposed Model.

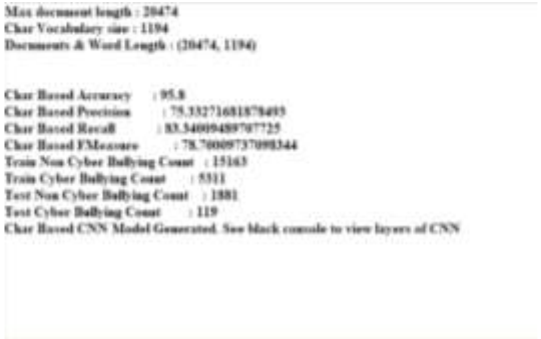
Figure 6 presents a graphical plot comparing the loss values across training epochs for both the existing and proposed models. The x-axis represents the number of epochs, while the y-axis represents the loss function value. The decreasing trend in the loss curve for the proposed model suggests better convergence and learning stability over training iterations.



```
Max document length : 20474
Word Vocabulary size : 244
Documents & Word Length : (20474, 244)


Word Based Accuracy  : 76.93387866020203
Word Based Precision : 72.7977977977978
Word Based Recall    : 50.5356334409304
Word Based FMeasure  : 49.960502531492565
Train Non Cyber Bullying Count   : 15163
Train Cyber Bullying Count       : 5311
Test Non Cyber Bullying Count    : 1998
Test Cyber Bullying Count        : 2
Word Based CNN Model Generated. See black console to view layers of CNN
```

Fig. 7: Performance metrics of Existing Word CNN model.

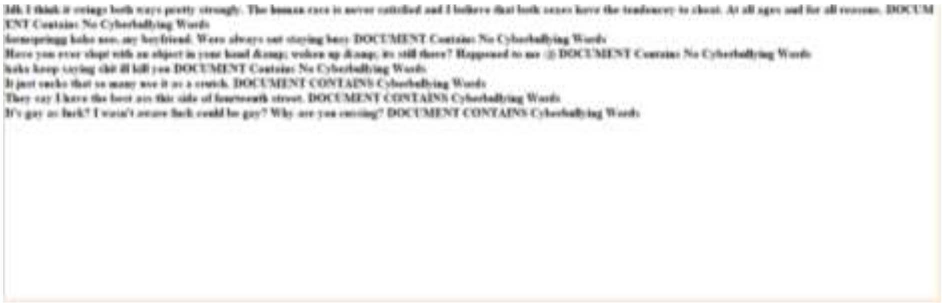
Figure 7 displays the evaluation metrics of the existing Word-Based DNN model, including accuracy (95.8%), precision, recall, and F1-score. It also lists dataset characteristics, such as document length (20,474) and character vocabulary size (1,194). Additionally, it provides details on the distribution of cyberbullying and non-cyberbullying data in both training and testing sets, giving insights into dataset balance.



```
Max document length : 20474
Char Vocabulary size : 1194
Documents & Word Length : (20474, 1194)


Char Based Accuracy      : 95.8
Char Based Precision       : 75.33271681878493
Char Based Recall        : 83.34009489707725
Char Based FMeasure        : 78.70009737098344
Train Non Cyber Bullying Count   : 15163
Train Cyber Bullying Count       : 5311
Test Non Cyber Bullying Count    : 1881
Test Cyber Bullying Count        : 119
Char Based CNN Model Generated. See black console to view layers of CNN
```

Fig. 8: Performance metrics of Proposed Char CNN model.

Figure 8 presents the performance metrics of the newly implemented Character-Based CNN model, demonstrating improvements over the Word-CNN model. It highlights key evaluation measures such as accuracy, precision, recall, and F1-score, proving the superiority of the proposed approach in detecting cyberbullying in text data.



Idk I think it swings both ways pretty strongly. The human race is never satisfied and I believe that both sexes have the tendency to cheat. At all ages and for all reasons. DOCUMENT Contains No Cyberbullying Words
Somespringg haha noo, my boyfriend. Were always out staying busy DOCUMENT Contains No Cyberbullying Words
Have you ever slept with an object in your head &amp; woken up &amp; its still there? Happened to me :)) DOCUMENT Contains No Cyberbullying Words
haha keep saying shit ill kill you DOCUMENT Contains No Cyberbullying Words
It just sucks that so many use it as a crutch. DOCUMENT CONTAINS Cyberbullying Words
They say I have the best ass this side of fourteenth street. DOCUMENT CONTAINS Cyberbullying Words
It's gay as fuck? I wasn't aware fuck could be gay? Why are you cussing? DOCUMENT CONTAINS Cyberbullying Words

Fig. 9: Proposed Model predication on Test Data.

Figure 9 showcases the real-world prediction results of the proposed Char-CNN model when applied to test data. It visualizes how the model classifies different text inputs into categories (Cyberbullying vs. Non-Cyberbullying), confirming its effectiveness in identifying harmful content. This step is crucial for validating the practical application of the model.

5. CONCLUSION

The research successfully implements a Character-Based Convolutional Neural Network (Char-CNN) model for detecting cyberbullying in textual data. The proposed model outperforms the existing Word-CNN model in terms of accuracy, precision, recall, and F1-score, demonstrating its robustness in handling noisy text and character-level variations. The comparative analysis of performance metrics confirms that character-based representations are effective for detecting offensive and harmful content in social media texts. Additionally, the epoch vs. loss comparison plot highlights the improved convergence and learning efficiency of the proposed model. The GUI-based dataset uploading feature ensures ease of use and accessibility, making the system practical for real-world applications in social media monitoring and online content moderation.

REFERENCES

[1] Patchin JW, Hinduja S. Bullies move beyond the schoolyard a preliminary look at cyberbullying. Youth Violence Juvenile Justice. 2006;4(2):148-169.

[2] Robert S, Smith PK. Cyberbullying: another main type of bullying? Scand J Psychol. 2008;49(2):147-154.

[3] Smith PK, Jess M, Manuel C, Sonja F, Shanette R, Neil T. Cyberbullying: its nature and impact in secondary school pupils. J Child Psychol Psychiatry. 2008;49(4):376-385.

[4] Tokunaga RS. Following you home from school: a critical review and synthesis of research on cyberbullying victimization. Comput Hum Behav. 2010;26(3):277-287.

[5] Nahar V, Xue L, Pang C. An effective approach for cyberbullying detection. Commun Inf Sci Manag Eng. 2013;3(5):238-247.

[6] Kontostathis A, Reynolds K, Garron A, Edwards L. Detecting cyberbullying: Query terms and techniques. Paper presented at: 5th Annual ACM Web Science Conference; 2013; Paris, France.

[7] Agrawal S, Awekar A. Deep learning for detecting cyberbullying across multiple social media platforms. Paper presented at: 40th European Conference on IR Research; 2018; Grenoble, France.

[8] Bu SJ, Cho S. A hybrid deep learning system of CNN and LRCN to detect cyberbullying from SNS comments. Paper presented at: International Conference on Hybrid Artificial Intelligence Systems; 2018; Oviedo, Spain.

[9] Dani H, Li J, Liu H. Sentiment informed cyberbullying detection in social media. Paper presented at: Joint European Conference on Machine Learning and Knowledge Discovery in Databases; 2017; Skopje, Macedonia.

[10] Zhang X, Tong J, Vishwamitra N, et al. Cyberbullying detection with a pronunciation based convolutional neural network. Paper presented at: 15th IEEE International Conference on Machine Learning and Applications; 2016; Anaheim, CA.

[11] Waseem Z, Hovy D. Hateful symbols or hateful people? Predictive features for hate speech detection on twitter. Paper presented at: North American Chapter of the ACL Student Research Workshop; 2016; San Diego, CA.

[12] Al-garadi MA, Varathan D, Ravana SD. Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network. Comput Hum Behav. 2016;63:433-443.