www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

DEEP LEARNING-BASED INTELLIGENT DOCUMENT CLASSIFIER FOR ONLINE SOCIAL NETWORKS

S.Vishnuvardhan Reddy[1], Bharath kumar[2], Dr. Ch. Srikanth[3]
[1,2,3] UG Scholar, Dept. of CSE(AI&ML) ,
St. Martin's Engineering College, Secunderabad, Telangana, India, 500100
[3]Associate Professor, Dept. of CSE(AI&ML), St. Martin's Engineering College, Secunderabad, Telangana, India, 500100
vishnuvardhanreddysareddy@gmail.com

Abstract:
The rapid proliferation of online social networks has resulted in an exponential increase in user-generated content, with over 4.5 billion active social media users globally, generating approximately 500 million tweets daily. However, this deluge of information presents significant challenges, as conventional content moderation methods struggle to keep pace with the scale and complexity of data, leading to issues like cyberbullying and misinformation. Existing classification techniques often rely on manual processes that are not only time-consuming but also prone to human error, which hampers timely intervention and response. This study proposes a novel deep learning-based intelligent document classifier designed to automatically classify user-generated content on social media platforms, specifically focusing on Twitter. The proposed system incorporates advanced preprocessing techniques and N-gram feature extraction to effectively analyze textual data. By employing a Convolutional Neural Network (CNN) architecture, the model classifies content into multiple target categories, includingreligion, age, gender, ethnicity, and indicators of cyberbullying. This automated approach enhances the accuracy and efficiency of content moderation, paving the way for more effective online community management.
Keywords: Social Neural Network, CNN, SVM,LRCN to detect,, TF-IDF, training dataset and testing dataset, SeamlessOperations.

1. INTRODUCTION

The rapid growth of online social networks has transformed the way people communicate and interact. With over 4.5 billion active users globally, platforms such as Facebook, Twitter, and Instagram have become primary sources of information exchange, discussions, and content sharing. Twitter alone witnesses approximately 500 million tweets per day, contributing to the ever-expanding digital landscape. However, the sheer volume of user-generated content presents significant challenges,
Particularly in monitoring and classifying content efficiently.

India, as one of the largest consumers of digital media, has witnessed an exponential rise in social media engagement. With over 624 million internet users, social networking sites have become the dominant platform for communication. India ranks among the top countries in terms of Twitter usage, with millions of daily active users sharing opinions, news, and multimedia content. Despite the benefits of online connectivity, there has been a notable increase in cyberbullying, misinformation, and hate speech. Studies have revealed that India is among the top three countries experiencing online harassment, with over 37% of internet users facing some form of cyber abuse. Fake news dissemination has also been a persistent issue, particularly during politically sensitive periods, contributing to unrest and misinformation.

Traditional moderation techniques, such as manual content review and rule-based filtering, fail to keep up with the velocity and volume of online content. These methods are not only inefficient but also susceptible to bias and human error. The emergence of deep learning-based document classification offers a scalable and accurate solution to these challenges. By leveraging artificial intelligence, machine learning models can analyze vast datasets, detect patterns, and classify content into predefined categories, enhancing content moderation and user safety.

Cyberbullying is an increasingly important and serious social problem, which can negatively affect individuals. It is defined as the phenomena of using the internet, cell phones and other electronic devices to willfully hurt or harass others.

R1	Sassy. More like trashy
R2	I HATE KAT SO MUCH
R3	Kat, a massive c*nt
R4	Shut up Nikki… That is all :)

Table 1: Some instances in dataset.

Another key challenge in cyberbullying research is the availability of suitable data, which is necessary for developing models that can classify cyberbullying. There are some datasets have been publicly available for this specific task such as the training set provided in CAW 2.0 Workshop and the Twitter Bullying Traces dataset [6]. Since cyberbullying detection has been fully illustrated as a natural language processing task, various classifiers have been masterly improved to accomplish this task, including the Naive Bayes [7], the C4.5 decision tree [8], random forests [9], SVMs with different kernels, and neural networks classifiers [6]. A variety of feature selection methods have also been carefully designed to improve the classification accuracy.

2.LITERATURESURVEY

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 [15-18]. 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 [19]. Reynolds et al used the labelled data, in conjunction with the machine learning techniques provided by the Weka tool kit, to train a C4.5 decision tree learner and instance-based learner to recognize bullying content [8]. Xu et al showed that the SVM with a linear kernel using unigrams and bigrams as features can achieve a recall of 79% and a precision of 76% [6]. Dadvar et al took into account the various features in hurtful messages, including TF-IDF unigrams, the presence of swear words, frequent POS bigrams, and topic-specific unigrams and bigrams, and the approach was tested using JRip, J48, the SVM, and the naive Bayes [10].

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 [20]. 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 [12]. 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 [21]. 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 [22]. 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 [23, 24]. 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 [25], 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 [26]. 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 [24].

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 [27]. 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 [23]. 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.

We believe that spelling mistakes can be learned. Most of the spelling mistakes have an edit distance of less than 2, and there is a certain regular pattern, which is related to people's pronunciation habits and the key distribution on a keyboard [28, 29]. In addition, on social networks, in order to convey a special meaning, some spelling mistakes are customary and common. Almost all factors suggest that these errors that we regarded as noise in previous works can be memorized by learning the combinations of characters.

3.PROPOSED METHODOLOGY

The proposed architecture for cyberbullying detection as shown in Figure 3 is broadly divided into four stages namely data storage stage, data preprocessing stage, data detection stage and output stage. In the data storage stage, data will be trained based on word, character and synonyms. Finally creates the three individual trained datasets such as word level trained dataset, character level trained dataset and synonym level trained dataset. These trained data sources consisting of malicious data generated by numerous attackers and contains the spelling and grammatical errors, these datasets available from the different sources of social networking platforms.

1.1. Data preprocessing stage

In the data preprocessing stage, input test data (T) will be applied and will be spitted into words. Then white space will be removed using padding extraction operation. In the extracted words, there might be the special characters, unknown symbols, and encrypted format data. This may cause to creation of abusive content in text generates bullying. Thus, these missing unknown text data will be replaced by the known relevant text. The text data is in ASCII format generally, but neural networks neither be trained nor be tested with the text content. Thus, the input text data will be converted into special type of non-ASCII value and will be represented in digital numeric's for every character like **"a** will be transformed to **0"**, similarly b:1, c:2, d:3 and goes on for all characters.

1.2.Tokenization

Over here the input text data is split into a set of words by removing all punctuation marks, tabs and other non-text characters and replacing them with white spaces. The part-of-speech (POS) tagging is also applied in some cases where words are tagged according to the grammatical context of the word in the sentence, hence dividing up the words into nouns, verbs, etc. This is important for the exact analysis of relations between words. Another approach was to ignore the order in which the words occurred and instead focus on their statistical distributions (the bag-of-words approach). In this case it is necessary to index the text into data vectors. The POS becomes important if the research is related to NLP. In one algorithm as part of extension work POS has been implemented.

1.3. Padding extraction

Padding refers to the white space between words, thus in padding extraction the space between two conjugative words will be extracted. In most of the times, the attackers wantedly use the whiter space to utilize the abusive text in the data. Thus, by using the padding extraction, the words contain white space will be precisely

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

analyzed for cyberbullying detection.

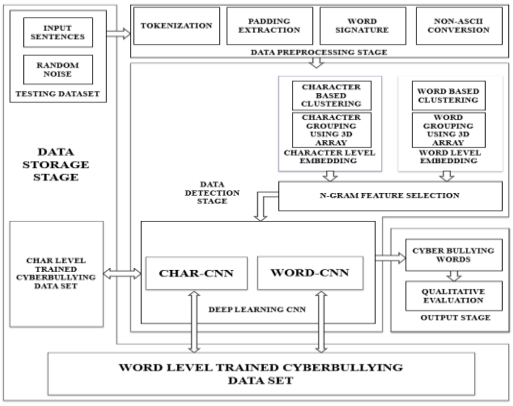


Figure 3: Proposed cyberbullying detection architecture.

1.4. 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.

Data Clustering

Clustering is a powerful and broadly acceptable data mining technique which is used to partition voluminous data into different classes, known as clusters, to support the businessman or an end user by providing different views and various patterns of same data suitable to the requirements. The cyberbullying detection focuses on the different levels clustering's such as character level, word level and synonym-character level.

Phase I: The set of prototype vectors are much higher than the expected number of clusters. The prototypes are grouped to form the actual character-based clusters.

Phase II: In this phase word level clustering algorithm is executed on the prototypes vectors to find clusters' word centroid. Clustering of vast amount of words in text samples is a key process in providing a higher level of knowledge about the underlying inherent classification of the abusive content causes to create cyberbullying.

Phase III: The word-based cluster centres obtained in Phase II are used in phase III. Synonym-character identification algorithm utilizing the high standard vocabulary is applied in this phase to generate the actual synonym-character-based clusters. The result from word level clustering which is found in phase II is used as the initial seed of the Synonym-character identification algorithm. Phase III converges quickly when the centroids from Phase II are used.

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 oncharacter 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.

CNN architecture

This section describes about the implementation details of HCNN based approach for cyberbullying detection with respect to character, and word recognition. Figure 1 represents the seven layered deep learning network architecture with F number of filers, Kernel size as K; it consists of multiple hidden layers which will allow it to compute much more complex features of the input.

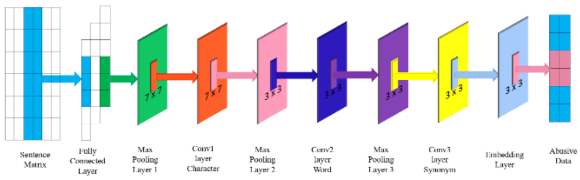


Figure 1: CNN model.

Because each hidden layer computes a non-linear transformation of the previous layer, a deep network can have significantly greater representational power (i.e., can learn significantly more complex functions) than a shallow one. By using a deep network, in the case of text data, one can also start to learn part-whole decompositions. For example, the first layer learns to group together characters in text to detect abusive data. The second and third layers might then group together words to detect cyberbullying content, or perhaps detect simple "parts of objects." An even deeper layer might then group together these word-based synonyms or detects even more complex features.

Initially the features of the sentence will be applied as matrix to the fully connected layer. The fully connected layer extracts the features of input text samples and parallel it will map the different types of features of text. The fully connected layer is used for binary classification of data. In computer vision tasks pooling layers are introduced for invariance of the nodes to rotation, translation, or scaling, modelling that textual data is the same, even occurring on another patch of the data, or in another size. Another handy property is that a variable sized feature map is thus reduced to a fixed size entity. Another handy attribute of Max pooling layer is that the network now can handle differently sized texts because despite the size of the input tensor or the then computed feature map the pooling operation yields only one value. Also pooling layers can make features invariant to translation or scaling.

Applications

1. **Social Media Content Moderation** - AI-based classifiers can be integrated into platforms like Twitter, Facebook, and Instagram to automatically filter harmful content.
2. **Cyberbullying Prevention** - Schools and organizations can use deep learning models to monitor online discussions and identify bullying behavior.
3. **Hate Speech Detection** - Government agencies can deploy AI-based classifiers to monitor and curb hate speech, ensuring social harmony.
4. **Misinformation Control** - Media houses and fact-checking organizations can utilize AI classifiers to verify news authenticity before publication.
5. **Corporate Communication Monitoring** - Companies can use document classifiers to monitor internal communications and ensure adherence to corporate policies.
6. **E-commerce Review Analysis** - AI classifiers help detect fake reviews and inappropriate content in customer feedback.
7. **Legal Document Classification** - Law firms and regulatory bodies can use AI-based classifiers to sort legal documents based on predefined categories.
8. **Healthcare Sentiment Analysis** - Hospitals can employ classifiers to analyze patient feedback and improve healthcare services.
9. **Political Content Analysis** - Governments can use AI models to monitor political discourse and prevent the spread of propaganda.
10. **Academic Research** - Researchers can leverage AI classifiers to analyze large textual datasets efficiently.

Significance

1. **Scalability and Efficiency** - Traditional content moderation methods struggle to keep pace with the rapid generation of online content. An AI-based classifier provides scalable solutions capable of handling large volumes of data efficiently.
2. **Real-Time Content Filtering** - With millions of posts being uploaded every minute, real-time classification is crucial for prompt intervention. Deep learning models enable instant detection and categorization of content.
3. **Reduction in Human Bias** - Manual moderation is often influenced by personal biases and inconsistencies. Automated classifiers ensure an objective approach to content classification.
4. **Combating Cyberbullying** - The growing prevalence of cyberbullying necessitates advanced detection mechanisms. AI-driven models can identify offensive language and harmful intent, aiding in early intervention.
5. **Mitigating Fake News** - Misinformation can have dire consequences, especially in politically or socially sensitive contexts. An AI-based classifier can detect and flag misleading content, preventing its widespread dissemination.
6. **Improved Community Management** - Social media platforms require robust tools for managing user interactions. Intelligent classifiers enhance community management by automatically filtering harmful content.
7. **Enhancing User Safety** - Online platforms must ensure a safe digital environment for users. AI-driven moderation tools contribute to a safer social media experience.
8. **Regulatory Compliance** - Many governments mandate strict content moderation policies. AI-powered classifiers help platforms comply with legal frameworks efficiently.
9. **Industry Adoption and Innovation** - Companies investing in AI-based moderation tools stay ahead of competitors by offering a secure and user-friendly experience.

4.EXPERMENTAL

1. Uploading Dataset

The first step involves loading the dataset containing customer reviews and ratings. The dataset is typically in CSV format and is loaded using **pandas** to facilitate data processing. Each review corresponds to a rating, which will be used to determine sentiment polarity. After loading, basic exploratory data analysis (EDA) is performed to check for missing values, duplicates, and overall data distribution.

2. NLP Processing

To prepare textual data for sentiment analysis, several NLP preprocessing techniques are applied. The text is converted to lowercase to maintain uniformity, and punctuation, special characters, and unnecessary spaces are removed. Tokenization is performed to break down sentences into individual words, followed by the removal of stopwords to eliminate common but unimportant words. Stemming or lemmatization is then applied to reduce words to their base forms, ensuring better generalization during analysis.

3. Sentiment Analysis using VADER

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a rule-based sentiment analysis tool specifically designed for social media and customer reviews. It assigns a sentiment score to each review, classifying it as **Positive, Neutral, or Negative** based on predefined lexical features. This step helps in quickly analyzing textual data and serves as a baseline for sentiment classification before implementing machine learning models.

4. Feature Engineering

To convert textual data into a numerical format suitable for machine learning, **TF-IDF (Term Frequency-Inverse Document Frequency)** is applied. This technique assigns importance to words based on their occurrence in the document while reducing the impact of commonly occurring words. Additionally, other features like word count, sentiment polarity scores, and n-grams can be extracted to enhance model performance. The target variable is defined based on sentiment labels.

5. Splitting Data for Training & Testing

The dataset is split into **training and testing subsets** using the train_test_split function from **scikit-learn**. Typically, 80% of the data is used for training, while 20% is reserved for testing. This ensures that the model learns from a portion of the data while its performance is evaluated on unseen data. Stratified sampling can be used if class imbalance exists to maintain a balanced distribution of sentiment classes in both sets.

6. Model Selection and Training

Several machine learning models are trained to classify customer sentiment. Common classifiers include **Logistic Regression**, **Random Forest**, **Naïve Bayes**, and **XGBoost**, each with different advantages. The models are trained using the transformed features, and hyperparameter tuning is applied to optimize performance. Cross-validation techniques like **k-fold cross-validation** ensure that the models generalize well to unseen data.

7. Model Evaluation & Performance Metrics

After training, the models are evaluated using various performance metrics. **Accuracy, precision, recall, and F1-score** provide insights into how well the models distinguish between different sentiment classes. A **confusion matrix** is used to analyze misclassifications, while **ROC-AUC curves** measure overall classification effectiveness. These metrics help in selecting the most reliable model for real-world predictions.

8. Predicting Sentiments of New Reviews

Once the best-performing model is selected, it is deployed to classify new, unseen customer reviews. The same preprocessing and feature extraction techniques are applied to new inputs before making predictions. The model outputs sentiment labels for each review, allowing businesses to gauge customer satisfaction and identify trends in feedback.

Results Description

The figure 1 represents the **Graphical User Interface (GUI)** of the project, where users can upload the dataset containing text data for analysis. The interface provides an easy-to-use platform to browse and select the dataset, ensuring smooth integration of data into the system.

It verifies the dataset format and performs initial checks before proceeding with processing.

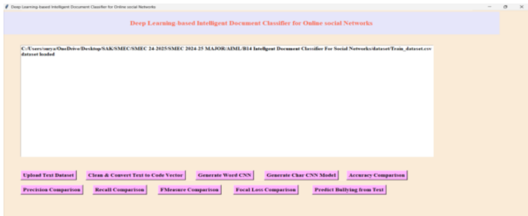


Fig. 1: Uploading Dataset in the Project GUI Interface.

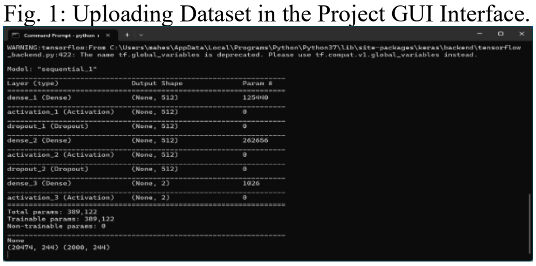


Fig. 2: Architecture of Proposed CNN Model.

This figure 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.

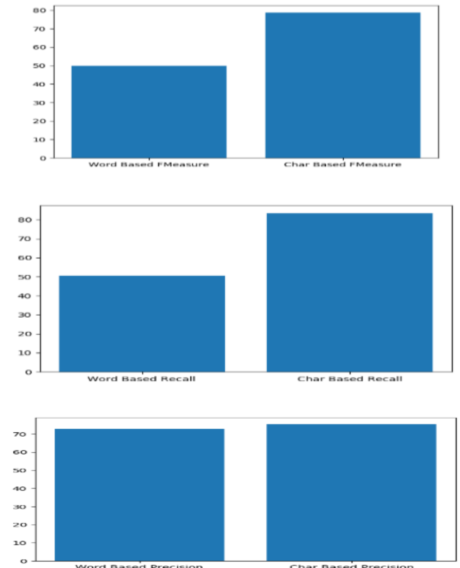
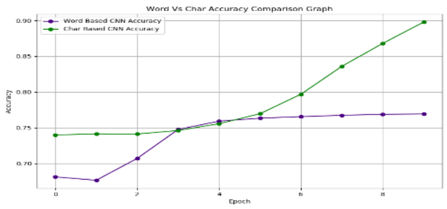


Fig. 3: Performance comparison of metrics Existing and Proposed Model.

This figure 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.



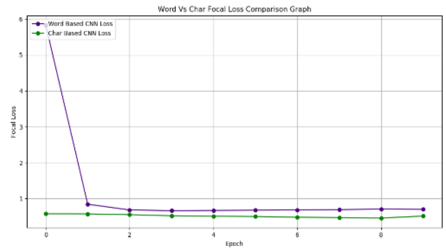


Fig. 4: Epoch vs Loss Comparison Plot of Existing and Proposed Model.

This figure 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.

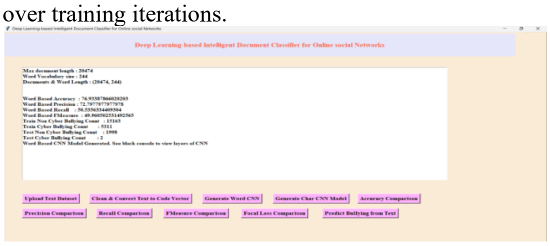


Fig. 5: Performance metrics of Existing Word CNN model.

This figure displays the evaluation metrics of the existing **Word-Based CNN 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.

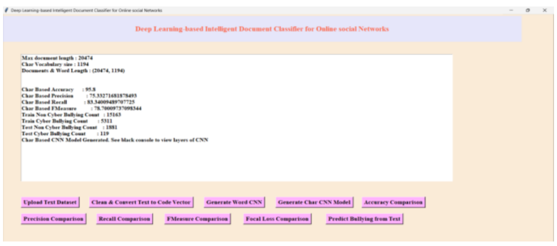


Fig. 6: Performance metrics of Proposed Char CNN model.

This figure 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.

Fig. 7: Proposed Model predication on Test Data.

This figure 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

This 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] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad and Y. Chang, "Abusive language detection in online user content", *Proc. 25th International Conference on World Wide Web (WWW2016)*, pp. 145-153, Apr. 2016.

[2] Y. Kim, "Convolutional neural networks for sentence classification", *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*, pp. 1746-1751, Oct. 2014.

[3] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks", *Proc. 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2015)*, pp. 103-112, May 2015.

[4] Y. Zhang and B. C. Wallace, "A sensitivity analysis of (and practitioners guide to) convolutional neural networks for sentence classification", *CoRR*, vol. abs/1510.03820, Oct. 2015.

[5] R. Zhang, H. Lee and D. Radev, "Dependency sensitive convolutional neural networks for modeling sentences and documents", *Proc. 2016 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016), pp. 1512-1521, Jun. 2016.

[6] D. Bertero and P. Fung, "A long short-term memory frame-work for predicting humor in dialogues", *Proc. 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pp. 130-135, Jun. 2016.

[7] C. Zhou, C. Sun, Z. Liu and F. C. Lau, "A c-lstm neural network for text classification", *CoRR*, vol. abs/1511.08630, Nov. 2015.

[8] Z. Z. Wint and M. Aritsugi, "Cyberbullying detection system for social network sites", *Proc. AUN/SEED-NET Regional Conference on Computer and Information Engineering (RCCIE 2016)*, pp. 283-288, Oct. 2016.

[9] A. Mousa and B. Schuller, "Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis", *Proc. 15th Conference of the European Chapter of the Association for Computational Linguistics*, vol. 1, pp. 1023-1032, Apr. 2017.

[10] H. Nguyen and M.-L. Nguyen, "A deep neural architecture for sentence-level sentiment classification in twitter social networking", *Proc. International Conference of the Pacific Association for Computational Linguistics (PACLING 2017)*, pp. 15-27, Aug. 2017.

[11] Available:https://fasttext.cc/docs/en/support.html.T. Chen, R. Xu, Y. He and X. Wang, "A c-lstm neural network for text classification" in Expert Systems with Applications, ELSEVIER, vol. 72, pp. 221-230, Apr. 2017.

[12] Z. Z. Wint, T. Ducros and M. Aritsugi, "Spell corrector to social media datasets in message filtering systems", *Proc. 2017 Twelfth International Conference on Digital Information Management (ICDIM)*, pp. 209-215, Sep. 2017.