Iranian Journal of Fuzzy Systems

Volume 21, Number 4, (2024), pp. 141-162

Original paper

University of Sistan and Baluchestan

DKNN and HPFS: An efficacious deep learning approach with fuzzy sets for social network hostility

T. Charan Singh [1] and S. Srithar [2]

[1,2] *Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Deemed to be University, Vaddeswaram, Andhra Pradesh, 522302, India.*

tcharansingh.klef@gmail.com, sss.srithar@gmail.com

Abstract

Hate speech and hateful language have become more accessible to spread as a result of the increase in social media and digital contacts. Cyberbullying is the term used to describe these kinds of online insults, attacks, and harassment. Finding such offensive content has become difficult due to the overwhelming amount of user-generated content. But Deep Learning (DL) techniques outperform typical machine learning algorithms in various ways. This research proposes a new DL-based techniques to overcome the issues of cyberbullying content recognition. To detect and classify the bullying content from pre-processed data using selected essential features, the Deep Kronecker Neural Network (DKNN) technique was employed. Comparing different classification strategies with the proposed approach, the extensive tests conducted on the two datasets demonstrate the significance of this work. We provide a novel technique for cyberbullying detection: the DKNN technique outperforms existing state-of-the-art methods with up to 99.56% accuracy results.

Keywords: Cyberbullying, deep learning, detection, algorithms, Twitter, cybercrime, social media, sentiment analysis, cyberbullying natural language processing.

1 Introduction

Social media is a platform that facilitates information sharing between users. Online social networks, or OSNs, are primarily used to facilitate virtual communication between internet users. Nevertheless, these technological advancements have also given rise to some other societal problems, such as cyberbullying." Bullying has always existed in society; even in the pre-digital age, online interfaces are seen as more secure, and cyberbullying has become more common [12, 19, 20]. Cyberbullying is generally understood to be the deliberate use of electronic media by a group or an individual to target a victim online in a violent or abusive manner. This activity entails verbally abusing, harassing, or demeaning a target repeatedly online. Malevolent individuals on social media employ derogatory remarks, inflammatory vocabulary, hate speech, harmful remarks, and harsh language to target victims specifically [10, 13]. These kinds of posts hurt social media users, harming their emotional well-being and undermining the credibility of social media networks. Manually identifying such illegal stuff on the internet is neither practical nor productive. Cyberbullying detection requires an automated detection system [3, 11].

Social media platforms like Twitter, Facebook, Instagram, Telegram, Snapchat, and others have made it possible for millions of individuals worldwide to communicate effectively and instantly, regardless of their location or political affiliation [6, 7, 18]. Social media companies now consider cyberbullying while developing their policies since they understand the harm that can result from its use. These days, cyber bullies are anonymously threatening and hacking people, and they are not afraid to carry out other criminal acts. Cyberbullies have become more prevalent in recent years. Among the many negative consequences of cyberbullying are an increased risk of suicide, mental illness, depression, drug and alcohol misuse, decreased productivity, subpar performance at work or school, low self-esteem, self-harm, and many more [6, 14, 21].

Corresponding Author: S. Srithar
Received: April 2024; Revised: August 2024; Accepted: September 2024.
https://doi.org/10.22111/ijfs.2024.48439.8532

With over 500 million users, Twitter is a well-known social media platform where users may write, read, and send 140-character messages, known as tweets. Every day, the site produces more than 500 million tweets. Unique features on Twitter include following, usernames, profile images, locations, biographies, retweets, hashtags, and more [8, 17]. Approximately 80% of Twitter users are thinking of tweeting using a smartphone. Bullies who use social media as a playground are already starting to use it. By applying a deep learning technique, this work seeks to identify and diagnose cyberbullying across global social media networks by leveraging behavioural characteristics in tweet content.

1.1 Research contribution

This work contribution can be mentioned as follows:
• To enhance the quality of cyberbullying data, and improve the effectiveness of deep learning models, data cleaning, tokenization, stemming, stop word removal, count vectorization, and normalization are performed in the pre-processing phase.
• The essential features are selected with the help of the Farmland Fertility Algorithm (FFA). Then, DKNN is employed to detect cyberbullying.
• Categorizing the severity level as medium, high, or low utilizing a Linguistic hesitant Pythagorean Fuzzy sets (HPFSs).

The structure of this research study is as follows. An overview of the literature is conducted in Section 2 to identify the advantages and disadvantages of various methods for detecting cyberbullying. The proposed deep learning approach is presented in Section 4. The dataset utilized to assess the proposed method is presented in Section 5. In Section 6, the findings are presented, discussed, and contrasted with closely related methods. The research provided is concluded in Section 7. Future work is discussed in Section 8.

1.2 Background of deep learning and cyberbullying

Deep learning is a recent development in the topic of machine learning. Artificial neural networks are DL approaches modelled after the brain's anatomy and physiology. Furthermore, rather than following simple, conventionally written instructions, DL approaches are trained to retrieve and comprehend the relevant representations from the information. Creating basic, non-linear modules that individually convert the raw input information at one level into a higher-level representation is the first step toward creating a meaningful representation.

As a result, it has been demonstrated that deep learning algorithms effectively classify various data types. Three categories can be used to group these algorithms: Acquiring knowledge through supervised, semi-supervised, or unstructured methods. The supervised learning paradigm, in which the training data is labelled, was modified for this research. Furthermore, massive volumes of data and vital computing gear, like a potent graphics processing unit (GPU), are needed for Deep Learning. Conversely, bullying that occurs through digital devices and online social media platforms is known as cyberbullying on social media. It can appear in various ways, such as cyberstalking, social exclusion, distributing false information, spreading libellous material, harassing or discriminating behaviour, negative and hostile communications, and personal attacks. Furthermore, addressing cyberbullying on social media platforms is quite tricky.

2 Literature survey

This section reviews approaches to cyberbullying detection that concentrate on online social networks (OSNs). To ascertain if deep learning algorithms successfully identify insults in social commentary, Iwendi et al. [9] conducted an empirical analysis. Pre-processing procedures include text cleaning, stemming, lemmatization, tokenization, and stop word removal. Four deep-learning models were used to analyze the experimental data: an RNN, a BLSTM, a GRU, and an LSTM.

The DEA-RNN hybrid deep learning model is presented by Murshed et al. [15] as a way to identify CB on the social media network Twitter. To optimize the Dolphin Echolocation Algorithm (DEA) and the Elman Recurrent Neural Network (RNN) parameters, the suggested DEA-RNN model combines RNNs of the Elman type with DEA. Using a dataset of 10,000 tweets, we conducted a detailed evaluation of DEA-RNN and compared its results to those of cutting-edge algorithms like RF, RNN, SVM, Bi-LSTM, and SVM.

An effective hybrid machine-learning model for identifying cyberbullying in Bengali on social media is presented by Akhter et al. [2]. The suggested technique for addressing bullying in the Bengali language includes efficient preprocessing of the text to convert it into a format that can be used for analysis, feature extraction from the text data using TfidfVectorizer (TFID) to retrieve valuable information, and resampling utilizing the Instance Hardness Threshold (IHT) process to balance the dataset and prevent issues with overfitting or under-fitting. Suhas et al. [21] presented

an Extreme Learning Machine (ELM) method for detecting cyberbullying. Emojis included in tweets are categorized using Convolutional Neural Networks (CNN). After pre-processing, attributes are extracted from the text, hashtag, and emoji databases. Cyberbullying can be identified using the ELM classifier.

A CB detection technique called FAEO-ECNN was suggested by Murshed et al. [16] to identify and categorize cyberbullying on social media sites. The recommended method combines Extended Convolutional Neural Network (ECNN) and Fuzzy Adaptive Equilibrium Optimization (FAEO) clustering-based topic modelling to improve the precision of the CB identification process. To start, the dataset is cleaned up using pre-processing. The characteristics are then retrieved using several models. Fuzzy Adaptive Equilibrium Optimization (FAEO) automatically analyses the pre-processed input data, an unsupervised technique that automatically generates word clusters from the text data to identify the latent subjects. Lastly, cyberbullying categorization uses the Rain Optimization (RO) algorithm and ECNN to identify CB from posts or texts.

To enhance cyberbullying detection, Almomani et al. [4] developed a hybrid strategy that utilizes machine learning classifiers and deep learning models as feature extractors. Understanding the intricate circumstances in which cyberbullying occurs is improved by extracting features utilizing trained deep learning algorithms, such as InceptionV3, ResNet50, and VGG16, and then feeding those features into classifiers like Logistic Regression and Support Vector Machines. Alqahtani & Ilyas [5] are concentrating on improving a system to identify six different forms of cyberbullying tweets. Using a cyberbullying dataset, we applied multi-classification methods and found that our method produced good accuracy results, especially when using the TF-IDF (bigram) feature extraction. This suggests that it performed exceptionally well compared to other trials on the same dataset.

2.1 Problem statement

People, especially teenagers, are still frequently the targets of cyberbullying, which is a dangerous and widespread issue that involves various forms of emotional anguish, intimidation, and harassment. The rising incidence of cyberbullying presents formidable obstacles to preserving a secure and encouraging virtual space. Even with technological improvements and raised awareness, it is still difficult to identify instances of cyberbullying in a timely and precise manner. The intricate patterns of online aggressiveness, context-dependent emotions, and the constantly changing strategies used by bullies are just a few examples of the complex kinds of cyberbullying that are difficult for current detection tools to identify accurately. Furthermore, old ways could be challenging because online communication is dynamic, and new platforms and communication styles frequently appear.

There are concerns about people's mental health and the continuation of hostile online behaviour when there are insufficient and faulty cyberbullying detection systems. Thus, to precisely identify and mitigate instances of cyberbullying across various digital platforms, there is an urgent need to build sophisticated and context-aware cyberbullying detection systems that use cutting-edge technology and deep learning-based algorithms. The approach we propose successfully identifies cyberbullying.

3 Proposed methodology

Social media platforms have offered us more options than we could have imagined because they are frequently used and well-liked. Even with all of the established advantages, some users might make use of these chances to harass, abuse, bully, and humiliate other people. This problem explains why it's necessary to stop these harmful acts and, by identifying instances of cyberbullying, provide a secure online environment for innocent users. Data cleansing, tokenization, stemming, stop word removal, and normalization are the first steps in the pre-processing step. The Farmland Fertility Algorithm (FFA) was used to identify the features after the pre-processing stage in which the data was fed.

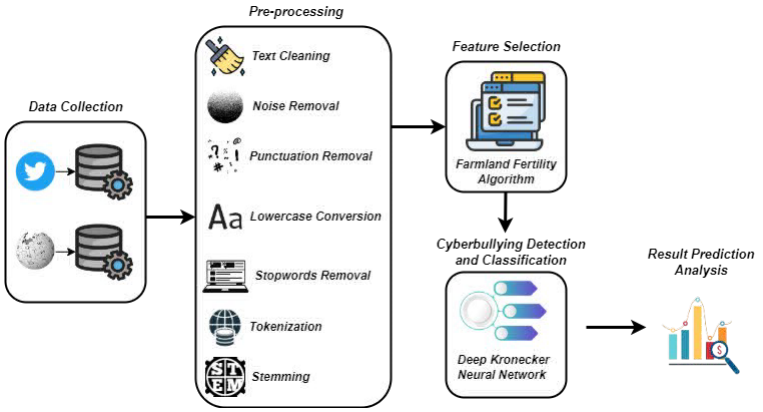


Figure 1: Proposed Architecture.

Next, a Deep Kronecker Neural Network (DKNN) based on deep learning is used to detect cyberbullying. The degree of the comments was assessed using linguistically hesitant Pythagorean fuzzy sets (HPFSs), and fuzzy rule sets were ultimately created using the DKNN prediction model's output. Several performance measures will be utilized to simulate the proposed approach's performance utilizing Twitter and cyberbullying data sets. Figure 1 shows the structure of the proposed methodology.

3.1 Pre-processing

The raw format of the Twitter and Wikipedia data gathered from the dataset required some refinement before the identification module was performed. Some pre-processing methods have been applied to obtain the data in an organized manner. The procedure involved removing punctuation, eliminating noise (tags, links, whitespace, and digits), and consistently rearranging the entire page in lowercase. After stop words—insignificant words—are eliminated from the documents, tokenization is performed. These are the most often used terms in sentences that, if added as text classification features, would have caused significant commotion. As a result, we have stemmed the tokens to their primary form to reduce the variety of terms on the page. Finally, the final pre-processing step involved detokenization, which produced clean and appropriate data for the subsequent phase. Figure 2 shows an overall picture of the pre-processing.

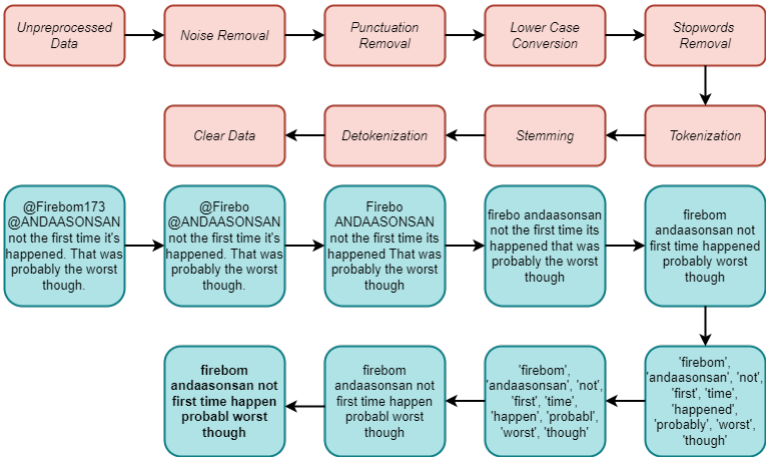


Figure 2: Pre-processing Steps.

3.2 Feature selection

The features included in the cyberbullying detection algorithm are the particular topic of this section. Because there are so many words, terms, and phrases in the text, it can be not easy to learn from large amounts of information. As a result, the entire operation incurs significant computing costs. Furthermore, repetitious and irrelevant features hinder

any classification model's performance and accuracy. Therefore, selecting features is a perfect way to minimize the large amount of data and avoid working with high-dimensional data. This study employed the Farmland Fertility Algorithm (FFA) to select the essential attributes from the dataset. This allows the methodology to produce vector features of fixed length, a numerical text representation.

Natural farmland's fertility served as the model for FFA, a novel metaheuristic algorithm. Each farm portion has a different soil quality than the others since farmers typically split the farm into distinct sections based on the soil quality. Each section's soil is altered in quality by adding specific materials. Thus, farmers attempt to maximize every field area by adding the nutrients the soil requires. Farmers change every portion of the land using this algorithm, and by documenting the soil quality in every piece, they may then determine how to enhance the soil in every region. The best and most necessary materials are assigned to every part to improve the soil's quality once the soil's quality in that portion has been determined. As a result, to increase the quality, every farm area needs a specific percentage of a particular material, with the worst land requiring the most significant amount of modification. Additionally, the highest-quality soil from the entire farming area is stored in warehouses in total memory (also known as global memory).

Global memory concepts are mixed with the farm's lowest-quality section's solutions, and all of the remaining sections' solutions are integrated. After introducing FFA, it's time to go over this algorithm's main ideas, formulas, and processes. The number of land sections, the number of solutions for every part, and their respective roles in the population's creation are determined in the generation of the initial population step. Since the optimization numbers determine the standard amount of farm sections, the search space is divided into K sections, and each part is given a specific number. N denotes an integer amount, a variable, and a solution amount for every land portion in this Equation.

$$N = k \times n. \tag{1}$$

The process of segmenting the land and determining the quality of each section involves determining the beginning population and then utilizing Equations (2) and (3) (which represent parts of the search space) to calculate the quality of each section based on the average of the section's answers.

$$Section_s = x(aj),\ a = n \times (s-1) : n \times s, \quad s = \{1, 2, \cdots, k\},\ j = \{1, 2, \cdots, 4\}. \tag{2}$$

Equation (2) divides the solutions of every phase to determine each component's average independently.

$$Fit_Section_x = Mean(all\ Fit(x_{ij}) in\ Section_s), s = \{1, 2, \cdots, k\},\ i = \{1, 2, \cdots, n\}. \tag{3}$$

In Equation (3), $Fit_Section_x$, where every phase has a unique value equal to the average fit of all the solutions inside that phase. Consequently, a calculation is made, and the average of the solutions inside every land segment is recorded in $Fit_Section_s$.

Updating Local and Global Memory: At this stage, each segment's global and local memories are updated. The best solutions are kept in the local memory for that part, while the global memory contains the best solutions for all phases combined.

$$M_Global = round(t \times N);\ 0.1 < t < 1. \tag{4}$$

$$M_local = round(t \times n);\ 0.1 < t < 1. \tag{5}$$

The best values are obtained in the local memory by Equation (4) and in the global memory by Equation (5). Following the selection of solutions that fit these memories, updates are made to local and global memories. It's time to proceed to the next part, "changing soil quality in each section," after identifying the poorest and best sections.

Changing Soil Quality in Every Phase: The most significant notable change is correlated with the lowest quality segment, as defined by each part's quality in Eq. (3). The amount of solutions in each segment is indicated by equation (2). The farm's worst part is where all solutions are coupled with global memory solutions, per Eqs. (6) and (7).

$$h = \alpha \times rand(-1.1). \tag{6}$$

$$X_{new} = h \times (X_{ij} - X_{MGlobal}) + X_{ij}. \tag{7}$$

In Equation (7), α, a whole amount between 0 and 1, is started at the start of the algorithm. Equation (6) yields h, and X_{ij} provides a solution to apply improvements to the farm's worst region. All farm sections should be joined with the answers for the total search space, as determined by Equations (8) and (9), just like the worst section should be changed.

$$h = \beta \times rand(0.1), \tag{8}$$

$$X_{new} = h \times (X_{ij} - X_{uj}) + X_{ij}. \tag{9}$$

Eqs. (8) and (9) use β to start the process, and X_ij allows selecting a solution to modify every part except the worst one. Equation (8) yields a decimal integer, h, and X_{new} provides a fresh solution for the applied modifications.

Updating Solutions Based on Global Memory and Local Memory: In the last stage, farmers combine the soil in each farm section with the best case in local memory, $Best_Local$, using the algorithm presented in Sect. 1, as long as no section's solutions are combined with local memory. To enhance the quality of the solutions, some are combined with the best solution thus far, $Best_{Global}$. This can be seen in Eq. (10).

$$H = \begin{cases} X_{new} = X_{ij} + \omega_1 * (X_{ij} - Best_{Global}(b)) & Q > rand \\ X_{new} = X_{ij} + rand(0.1) * (X_{ij} - Best_{Local}(b)) & \text{else} \end{cases} \tag{10}$$

It should be emphasized that there are two approaches to arrive at this new solution when interpreting Eq. (10). The initial determination of Q, a variable ranging from 0 to 1, indicates the degree to which the solutions are integrated with $Best_{Global}$. ω_1, an integer amount, is first calculated and displays the algorithm's variables; each iteration steadily lowers its value.

Fitness function

Regarding machine learning, feature selection (FS) can be viewed as a multi-objective optimization issue where two competing goals are considered: the least amount of features and classification accuracy. Applying FFA, or the binary optimization strategy, is suggested to solve the multi-objective issue. In addition, Eq. (11) defines a multi-objective fitness function for assessing the answers.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|N|}. \tag{11}$$

$\alpha\gamma_R(D)$ is the KNN categorization error rate in Eq. (11). Due to its ease of use, the KNN classifier is frequently utilized. In addition, $\mid R \mid$ denotes a subset of chosen attributes, $\mid N \mid$ represents the total amount of characteristics in the primary dataset, and two variables about the significance of categorization accuracy are shown by α and β.

3.3 Cyberbullying detection and classification

This section uses the critical characteristics chosen to represent cyberbullying. An innovative deep-learning system has been used to identify cyberbullying on social media. In this section, we employed a DKNN for cyberbullying identification and categorization. The KNN and standard feed-forward neural networks (FNNs) are comparable in that they both have an all-around flexible initiation function that looks like this:

$$\Psi_{\alpha,\omega} = \sum_{m=1}^{M} \alpha_m \Psi_m(\omega_m y), M \in N \geq 1, \quad \alpha = (\alpha_m), \ \omega = \omega_m. \tag{12}$$

Where the fixed activation functions α_m is and ω, trainable parameters are located. The actual processing of the Kronecker item is not required for KNN execution. However, compared to a KNN, the Kronecker product allows for developing a much larger organization while maintaining nearly the same number of boundaries. In the lth hidden layer, Nl neurons are available.

$$L_l(S^{l-1}) = W^l S^{l-1} + b^l. \tag{13}$$

Give each component of the modified vector a non-linear activation function $\varphi 1()$ before transmitting it as a commitment to the following layer. The same function denotes an initial function after a subsequent layer.

$$u^{FF}(s) = (L_D^{\circ} \Psi_1^{\circ} L_{D-1}^{\circ} \cdots^{\circ} \Psi_1^{\circ} L_1)(s). \tag{14}$$

Where the composition operator that is utilized is °. The trainable parameter of the network indicates:

$$\Theta_{FF} = \{W^l, b^l\}_{t=1}^{D}. \tag{15}$$

The different norms of a vector $w = [w1, \cdots, wn]T \in RN$ are as follows:

$$\| w \|_1 = \sum_{i=1}^{n} | w_i |, \ \| w \|^2 = \sum_{i=1}^{n} | w_i |^2, \quad \| w \|_\infty = \max_{1<i<q} | w_i | . \tag{16}$$

The definitions of the Frobenius norm and the spectral norm are as follows:

$$\| N \|_F^2 = \sum_{i=1}^{p} \sum_{i=1}^{p} N_{ij}^2, \quad \| N \|_\infty = \max_{\|x\|-1} | N_x | \tag{17}$$

N_{ij} is the (i, j)-component of. Assume that all of the entries in the matrix $1u \times t$ for size $u \times t$ have a value of $1u$. Let K be a fixed + ve integer. Define the weight matrix and block bias vector for the lth block.

$$l_{KXl} \otimes b' = \begin{bmatrix} b^l \\ \vdots \\ b^l \end{bmatrix} \in R^{N_l K} \ 1_{KXK} \otimes W_1 = \begin{bmatrix} W^l & \cdots & W^l \\ \vdots & \ddots & \vdots \\ W^l & \cdots & W^l \end{bmatrix} \in R^{N_l KXN_{l-1}K} \tag{18}$$

Where the sign $\otimes$ represents the Kronecker product. Let $s = [s_1, \cdots, s_k]T \in RmK$ and

$$\overrightarrow{\Psi}(s) = \begin{bmatrix} \overrightarrow{\Psi}_1(s_1) \\ \vdots \\ \overrightarrow{\Psi}_k(s_k) \end{bmatrix} \tag{19}$$

The φjs are function calls that are applied to each element separately. Next, build a neural network utilizing the following Equation and the weight matrices of the block and block bias vectors: Given an input of $s0 = s$, we have $s1 = (1KXK \otimes Wl)(1KX1 \otimes s0) + 1KX1 \otimes b1$.

$$S^D = (1_{LXK} \otimes W_D)\overrightarrow{\Psi}(S^{D-1}) + b^D \text{ and } s' = (1_{KXK} \otimes W')\overrightarrow{\Psi}(1_{KX1} \otimes b_l). \tag{20}$$

SD is a D-layer KNN with uFF, the number of network variables at the lth layer, and KNl amount of neurons. The definitions of the scaled block weight matrices and block bias vectors are as follows:

$$\hat{W}^1 = (\omega^l \otimes \alpha^l) \otimes W^l, \ \hat{b}^1 = (\omega^l) \otimes b^l, \ 1 \le l < D. \tag{21}$$

$$\hat{W}^D = (1_{LXK} \otimes W^D) \otimes W^l \text{ and } \hat{b}^D = b^D. \tag{22}$$

$$s^l = (\hat{L}_l s^{l-1}) = \hat{W}^l s^{l-1} + \hat{b}^l. \tag{23}$$

The obtained representation is as follows:

$$u_\theta^k(s) = (\hat{L}_D^\circ \Psi^\circ \hat{L}_{D-1}^\circ \Psi^\circ \hat{L}_1)(1_{KX1} \otimes s). \tag{24}$$

Below is a definition of the DNN network variable set:

$$\theta_K = \{W^l, b^l\}_{l=1}^{D} \ U\{\omega^l, \alpha^l\}_{l=1}^{D-1}. \tag{25}$$

The block weight matrices $\{W^l\}$ and block bias vectors $\{b^l\}$ are unnecessary for DKNN completion. KNN is used in this work.

$$u^k(s) = (L_D^\circ \hat{\Psi}^\circ L_{D-1}^\circ \cdots^\circ \hat{\Psi}^{1\circ} L_1)(s). \tag{26}$$

Thus, the initiation function of the lth layer is no longer deterministic and instead depends on the parameter variables $\{l, \alpha l\}$.

$$\hat{\Psi}^1(L_1(s); \omega^l, \alpha^l) = \sum_{k=1}^{K} \alpha_k^l \Psi_k(\omega_k^l L_1(s)), l = 1, \cdots, D-1. \tag{27}$$

The yellow circles in the proposed DKNN represent the neurons in corresponding hidden levels. When a neuron's output deviates from the standard neural network architecture, it passes to several initiation functions. This section examines the DKNN by comparing the supervised learning square loss function. Assume that $Tm = \{(xi, yi)\}mi = 1$ contains the m-training data points. As stated below, define the square loss:

$$L(\theta) = \frac{1}{2} \sum_{j=1}^{M} (u_\theta^{Type}(x_i) - y_i)^2. \tag{28}$$

The selected network, denoted by $uType\theta(xi)$, could be either DKNN uK $\hat{i}(x)$ or FF network uFF $\hat{i}(x)$. Examine the following two-layer networks:

$$u_{\theta FF}^{FF}(x) = \sum_{l=1}^{N} c_i \Psi_1(w_i^T x + b_i) u_{\Theta k}^k(x) = \sum_{l=1}^{N} c_i \sum_{k=1}^{K} \alpha_k \Psi_k(\omega_k(w_i^T x + b_i)). \tag{29}$$

Specifically, $FF = \{ci, wi, bi\}Ni = 1$ and $K = \{ci, wi, bi\}Ni = 1U\{\alpha k, wk\}Kk = 1$ are shown as the corresponding network variables. The learning aim is to determine the network parameters required for loss function reduction.

3.4 Bully strength (severity) determination

The DKNN prediction model's outcome was eventually used to create fuzzy rule sets once fuzzy logic was used to assess the severity of the remarks. A series of if-then clauses is how fuzzy rules operate. Table 1 displays the fuzzy rule sets applied to the DKNN's output. This classifier separated the input data into bullying types, and only bullying-related comments were subjected to fuzzy rule sets. It is feasible to forecast whether there is a chance of bullying happening again and whether there is a risk in real life that might materialize soon by determining the extent of the bullying. Three levels of bullying severity were identified: high, low, and medium.

The ambiguous answer "High" can be understood as the most extreme kind of bullying, where there's a potential that it will persist and that it may finally materialize in real life. It is replicable, but there is a little chance of it going wrong in the future. The fuzzy output "Low" indicates that the bully is manageable and only a little aggressive. The amount of nasty terms in every bully's comment is tallied. The DKNN method was utilized in the provided dataset to generate these comments. The dataset and the comments found for different kinds of bullies are compared to ascertain the number of offensive phrases.

Table 1: The three-level function is for the number of bad words.

No	Rule
Rules 1	If $n \geq 1$ and $n < 3$, then Bully Words Count Level = Low.
Rules 2	If $n \geq 2$ and $n < 5$, then Bully Words Count Level = Medium.
Rules 3	If $n \geq 4$ and $n < 7$, then Bully Words Count Level = High.

3.4.1 Linguistic hesitant Pythagorean fuzzy entropy

The most widely used uncertainty measure is the entropy measure. Many scholars have proposed various entropy measures under multiple conditions to address the evaluation challenge effectively. In 2018, Xue et al. proposed interval-valued PF entropy and PF entropy. In 2019, Song et al. proposed uncertainty metrics and divergence-based cross entropy for IFSs. Entropy measurements in LPFSs were defined in 2019 by Lin et al. Han (2019a) and Han (2019b) proposed new Pythagorean fuzzy entropies and linguistic Pythagorean fuzzy entropies. Pythagorean fuzzy soft entropy and linguistic Pythagorean fuzzy soft entropy surfaced on this foundation. According to the complementary set and fuzzy set views, Chen et al. (2019) presented a few innovative entropy metrics that do not require value addition. Nevertheless, only a few studies have examined linguistically hesitant Pythagorean and hesitant Pythagorean fuzzy entropies up to this point.

Definition 3.1. *Assume that have two HPFEs, $\Psi_1 = (h_1, g_1)$ and $\Psi_2 = (h_2, g_2)$. The entropy for HPFEs, $E : HPFE(X) \to [0,1]$, represents the crisp function that satisfies the following conditions:*
(D1) *If Ψ_1 is $\{\{1\},\{0\}\}$ or Ψ_1 is $\{\{0\},\{,E(\Psi_1)=0\}\}$;*
(D2) *If $\theta_1 = \nu_1$, where $\theta_1 \in h_1$, $\nu_1 \in g_1, E(\Psi_1) = 1$;*
(D3) *$E(\Psi_1) = E(\Psi_1^c)$;*
(D4) *$E(\Psi_1) \le E(\Psi_2)$, if Ψ_1 is less fuzzy than Ψ_2, that is, $\theta_1 \le \theta_2$ and $\nu_1 \ge \nu_2$ for $\theta_2 \le \theta_2$; or $\theta_1 \ge \theta_2$ and $\nu_1 \le \nu_2$ for $\nu_2 \ge \nu_2$, where $\theta_1 \in h_1$, $\nu_1 \in g_1$, $\theta_1 \in h_2$, $\nu_2 \in g_2$.*

And H is known as Pythagorean fuzzy entropy with hesitation. The entropy measure of HPFEs is established, drawing inspiration from Chen's entropy measure for dual hesitant fuzzy elements.

$$H_{P1}(\Psi) = 1 - \frac{1}{2}\left(\frac{1}{l(h)}\sum_{\nu\in h}\min_{\nu\in g} \mid \theta - \nu \mid + \frac{1}{l(g)}\sum_{\nu\in g}\min_{\nu\in h} \mid \nu - \theta \mid\right). \tag{30}$$

The following formula is used to determine the entropy of A given a universe of discourse $X = \{x_1, x_2, \cdots, x_n\}$ and LHPFSs $A = \{x_i, S_{hA}(x_i), S_{gA}(x_i) \mid x_i\}$, $i = 1, 2, \cdots, n$.

$$H_{P1}(A) = 1 - \frac{1}{2n} \times \sum_{i=1}^{n}\left(\frac{1}{l(h_i)}\sum_{\nu_i\in h_i}\min_{\nu_i\in g_i} \mid \theta_i - \nu_i \mid + \frac{1}{l(g_i)}\sum_{\nu_i\in g_i}\min_{\nu_i\in h_i} \mid \nu_i - \theta_i \mid\right). \tag{31}$$

Definition 3.2. *Given two linguistic HPFEs (LHPFEs), let $\Psi_1 = (S_{h_1}, S_{g_1})$ and $\Psi_2 = (S_{h_2}, S_{g_2})$ be such that the entropy for LHPFEs $E : HPFE(X) \to [0,1]$ as the crisp function satisfies the following requirements:*
(D1) *If Ψ_1 is $\{\{$or Ψ_1 is $\{\{,E(\Psi_1)=0\}\}\}\}$;*
(D2) *If $I(S_{\theta_1}) = I(S_{\nu_1})$, where $S_{\theta_1} \in S_{h_1}$, $\nu_1 \in S_{g_1}$, $E(\Psi_1) = 1$;*
(D3) *$E(\Psi_1) = E(\Psi_1^c)$;*
(D4) *$E(\Psi_1) \le E(\Psi_2)$, if Ψ_1 is less fuzzy than Ψ_2, that is, $S_{\theta_1} \le S_{\theta_2}$ and $S_{\nu_1} \ge S_{\nu_2}$ for $S_{\theta_1} \le S_{\theta_2}$; or $S_{\nu_1} \ge S_{\nu_2}$ and $S_{\nu_1} \le S_{\nu_2}$ for $S_{\nu_1} \ge S_{\nu_2}$, where $S_{\theta_1} \in S_{h_1}$, $S_{\nu_1} \in S_{g_1}$, $S_{\theta_2} \in S_{h_1}$, $S_{\nu_1} \in S_{g_1}$ and E can be called linguistic hesitant Pythagorean fuzzy entropy.*

Regarding LHPFEs and LHPFSs, we can define the entropy measures $E_1(\Psi)$ and $E_1(A)$. Then, to demonstrate it, we looked at $E_1(\Psi)$.

$$H_1(\Psi) = 1 - \frac{1}{2}\left(\frac{1}{l(h)}\sum_{\theta\in h}\min_{\nu\in g}\frac{\mid I(S_\theta) - I(S_\nu) \mid}{\tau} + \frac{1}{l(g)}\sum_{\nu\in g}\min_{\theta\in h}\frac{\mid I(S_\nu) - I(S_\theta) \mid}{\tau}\right). \tag{32}$$

$$H_1(A) = 1 - \frac{1}{2n}\sum_{i=1}^{n}\left(\frac{1}{l(h_i)}\sum_{\theta_i\in h_i}\min_{\nu_i\in g_i}\frac{\mid I(S_{\theta_i}) - I(S_{\nu_i}) \mid}{\tau} + \frac{1}{l(g_i)}\sum_{\nu_i\in g_i}\min_{\theta_i\in h_i}\frac{\mid I(S_{\nu_i}) - I(S_{\theta_i}) \mid}{\tau}\right). \tag{33}$$

Lemma 3.3. *Definition 3.2's four conditions are satisfied by $H_1(\Psi)$.*

Proof. (D1) Equation (16) states that if $\Psi_1 = \{\{S_\tau\},\{S_0\}\}$ or $\{\{S_0\},\{S_\tau\}\}$, then $E_1(\Psi) = 0$. If $E_1(\Psi) = 0$, it can be obtained as per Equation (16).

$$\frac{1}{2}\left(\frac{1}{I(h)}\sum_{\theta\in h}\min_{\nu\in g}\frac{\mid I(S_\theta) - I(S_\nu) \mid}{\tau} + \frac{1}{I(g)}\sum_{\theta\in g}\min_{\theta\in h}\frac{\mid I(S_\nu) - I(S_\theta) \mid}{\tau}\right) = 1. \tag{34}$$

Because $0 \le \dfrac{\mid I(S_\theta) - I(S_\nu) \mid}{\tau} \le 1$, $0 \le \dfrac{\mid I(S_\nu) - I(S_\theta) \mid}{\tau} \le 1$, together with Equation (18), we obtain.

$$\min_{\nu\in g}\frac{\mid I(S_\theta) - I(S_\nu) \mid}{\tau} = 1, \min_{\theta\in h}\frac{\mid I(S_\nu) - I(S_\theta) \mid}{\tau} = 1. \tag{35}$$

Either $S_\theta = S_0$, $S_\nu = S_\tau$ or $S_\theta = S_0$, $S_\nu = S_0$ satisfy Equation (19) according to the definition of LHPFSs. Accordingly, $\Psi_1 = \{\{S_\tau\},\{S_0\}\}$ or $\{\{S_0\},\{S_\tau\}\}$.
(D2) When $E_1(\Psi) = 1$, we have $\dfrac{1}{I(h)}\sum_{\theta\in h}\min_{\nu\in g}\dfrac{\mid I(S_\theta) - I(S_\nu) \mid}{\tau}$. Since $\dfrac{1}{I(g)}\sum_{\theta\in g}\min_{\theta\in h}\dfrac{\mid I(S_\nu) - I(S_\theta) \mid}{\tau} = 0$, $S_\theta = S_\nu$.

(D3) It is obvious.
(D4) If $I(S_{\theta_2}) \leq I(S_{\nu_2})$, $I(S_{\theta 1}) \leq I(S_{\theta_2})$ and $I(S_{\nu_1}) \geq I(S_{\nu_2})$, we have $I(S_{\theta_1}) - I(S_{\nu_1}) \leq I(S_{\theta_2}) - I(S_{\nu_2}) \leq 0$; therefore, we can obtain $\mid I(S_{\theta_1}) - I(S_{\nu_1}) \mid \geq \mid I(S_{\theta_2}) - I(S_{\nu_2}) \mid$.

If $I(S_{\theta_2}) \geq I(S_{\nu_2})$, $I(S_{\theta 1}) \geq I(S_{\theta_2})$ and $I(S_{\nu_1}) \leq I(S_{\nu_2})$, then we have $I(S_{\theta_1}) - I(S_{\nu_1}) \geq I(S_{\theta_2}) - I(S_{\nu_2}) \geq 0$. Moreover, we can obtain $\mid I(S_{\theta_1}) - I(S_{\nu_1}) \mid \geq \mid I(S_{\theta_2}) - I(S_{\nu_2}) \mid$.

$$\left(\frac{1}{l(h_1)} \sum_{\theta_1 \in h_1} \min_{\nu_1 \in g_1} \frac{\mid I(S_{\theta_1}) - I(S_{\nu_1}) \mid}{\tau} + \frac{1}{l(g_1)} \sum_{\nu_1 \in g_1} \min_{\theta_1 \in h_1} \frac{\mid I(S_{\nu_1}) - I(S_{\theta_1}) \mid}{\tau} \right) \geq$$

$$\left(\frac{1}{l(h_2)} \sum_{\theta_2 \in h_2} \min_{\nu_2 \in g_2} \frac{\mid I(S_{\theta_2}) - I(S_{\nu_2}) \mid}{\tau} + \frac{1}{l(g_2)} \sum_{\nu_2 \in g_2} \min_{\theta_1 \in h_2} \frac{\mid I(S_{\nu_2}) - I(S_{\theta_2}) \mid}{\tau} \right).$$

Hence, $H_1(\Psi_1) \leq H_2(\Psi_2)$. □

4 Result and discussion

This section contains the dataset that was used to assess the proposed approach. The data preparation to get the raw text ready to be fed into the proposed system is next covered. Information about the tools used to train and validate the model's efficacy is provided at the end of this section. Eighty per cent of the manually annotated datasets were utilized for model training, while twenty per cent was used for testing after randomly selecting it. The evaluation criteria included F1-score, accuracy, recall, and precision.

4.1 Hyper-parameter configuration

In this section, the system's development results are presented, focusing on the configuration of hyper-parameters. The details of the specific settings employed for various hyper-parameters during the system's development process. These settings include a batch size of 15, training epochs totalling 100, and a dynamic dropout rate ranging from 0.5 to 0.1, ensuring regularization and preventing overfitting. The activation function utilized in the output layer is Sigmoid, while Rectified Linear Unit (ReLU) is applied as the activation function throughout the network layers. Binary Cross entropy is utilized for loss calculation, catering to the binary nature of the classification task. This comprehensive setup of hyper-parameters forms the backbone of the system's architecture, contributing to its effectiveness and performance in subsequent evaluations.

4.2 Experimental setup

Python was utilized to implement the proposed approach. Some Python libraries were used to pre-process, cleanse the data, and implement the model. The pandas' package was used to pre-process the information. The scikit-learn library was utilized to analyze the deep model once created using the Keras library. The experiment is conducted in the Anaconda3 environment with an Intel i5 2.60 GHz processor running Windows 10 and 16 GB of RAM, utilizing Python and KERAS as background applications.

4.3 Data collection

Dataset 1: Twitter dataset
The Python programming language was used to accomplish the proposed method. The data was pre-processed using Tensorflow, OpenCV tools, and Python packages like Numpy and Pandas. In addition, the deep multichannel model was made using the Keras package. 39,748 of the 47,733 tweets in the Kaggle dataset can be classified as offensive, and 7,985 as non-offensive. Table 2 presents the dataset's class distribution. There were testing and training sets inside the dataset.
Dataset 2: Wikipedia dataset
Wulczyn et al. gathered this corpus in 2017 using Wikipedia articles. The dataset is made up of comments on English-language debates that were taken from Wikipedia's "talk pages." To obtain all interactions, even deleted comments, the change history of Wikipedia pages is accessed to gather the comments. After the corpus is gathered, HTML content is eliminated, and plain text is retained. Only comments left by humans were kept in the dataset after all automated messages were removed. With 13,590 cases of cyberbullying text and 102,274 cases of non-cyberbullying remarks, the dataset version we use has 115,864 user comments.

Table 2: Classes in the dataset.

Dataset	Label	Counts
Tweet (47,733)	Non Bullying	7985
	Bullying	39748
Wikipedia (115,864)	Non Bullying	102,274
	Bullying	13,590

A train set (80%) and a test set (20%) have been separated from the cleaned label dataset shown in Table 2. After the dividing was finished, features were added to the train set.

4.4 Evaluation metrics

The pertinent measures and metrics were calculated. Some of these measures are false Positive, True Positive, True Negative, False Negative, F1-score, Accuracy, Precision, Recall, and Boxplot features.

Accuracy: This is the most well-known performance metric. This can be ascertained by calculating the ratio of accurately anticipated observations to total observations. This refers to the proportion of accurately categorized tweets to the total number of tweets.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}. \tag{36}$$

A model's applicability only sometimes equates to its high accuracy. Accuracy is a remarkable statistic if the values of false positives and false negatives in a dataset are symmetric.

Precision: This measures how many correctly anticipated positive observations out of all the predicted positive observations. Here, precision addresses issues like how many tweets are genuine examples of cyberbullying behaviour in relation to all tweets labelled as such.

$$Precision = \frac{TN}{TN + FN}. \tag{37}$$

Recall: This is the proportion of correctly identified and anticipated positive observations to all observations made during the class.

$$Recall = \frac{TP}{TP + FN}. \tag{38}$$

F1-Score: The recall and precision are averaged and weighted together. False positives and false negatives must both be acknowledged. Given the unequal class distribution in this work, F1 is typically more valuable than accuracy. Finding a balance between recall and precision can be greatly aided by it.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision \times Recall}. \tag{39}$$

4.5 Implementation of the cyberbullying performance

This experiment evaluates the performance of two datasets, Twitter and Wikipedia. Figure 3 shows the first page of cyber-content detection; on this page, select the dataset, enter the content or comment from these two platforms and click the detection button.

The given content or comment is checked by proposed techniques and given an output as to whether the provided content is bullying; if it's bullying content, then it'll show the severity of that content, as shown in Figure 4.

4.6 Performance on dataset 1

The most common terms in the Twitter dataset are displayed in Figure 5. The names of politicians, counties, derogatory terms, and some common words are the most often occurring words in the offending tweets. When combined with additional words in the same sentence, some words become offensive phrases even though they are not.

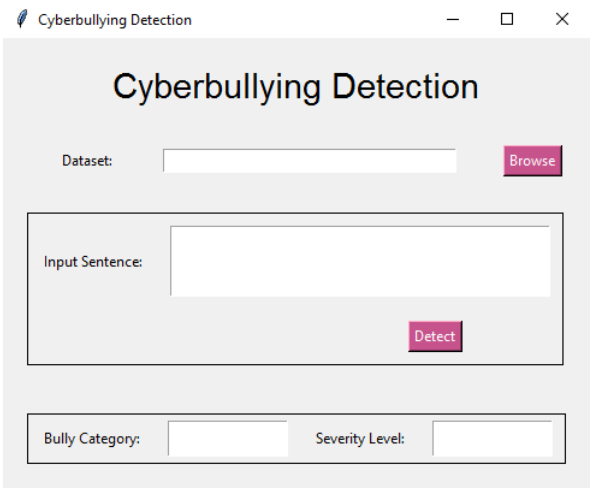


Figure 3: Front Page.

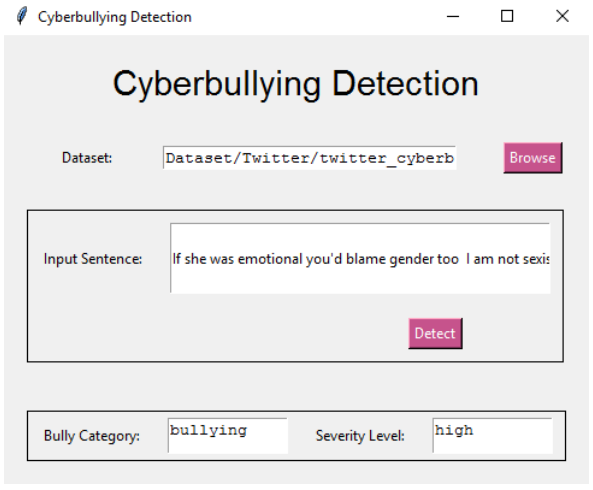


Figure 4: Cyber-content Detection Page.



(a) (b)

Figure 5: A word cloud containing the most frequently occurring words from both classes in the dataset. (a) The most commonly used terms in the offensive category. (b) These are the most widely used terms in the non-offensive category.

A commonly used word cloud for offensive and non-offensive is shown in Figure 3. The proposed approach's classification report for Dataset 1 across all classes is shown in Table 3. The proposed approach performs remarkably well in both the bullying and non-bullying categories.

Table 3: Classification report of proposed method for all classes on Dataset 1.

Category	Accuracy	Precision	Recall	F1-score
Bullying	99.52	99.42	99.18	99.30
Non-Bullying	99.61	99.30	99.35	99.32

The model obtains 99.52% accuracy for the Bullying class and 99.42%, 99.18%, and 99.30% for precision, recall, and F1-score, respectively. Comparably, the model achieves a high accuracy of 99.61% in the Non-Bullying category, with precision, recall, and F1-score values of 99.30%, 99.35%, and 99.32%, respectively. These findings demonstrate the method's robustness and efficacy in correctly identifying bullying and non-bullying behaviour in the dataset.

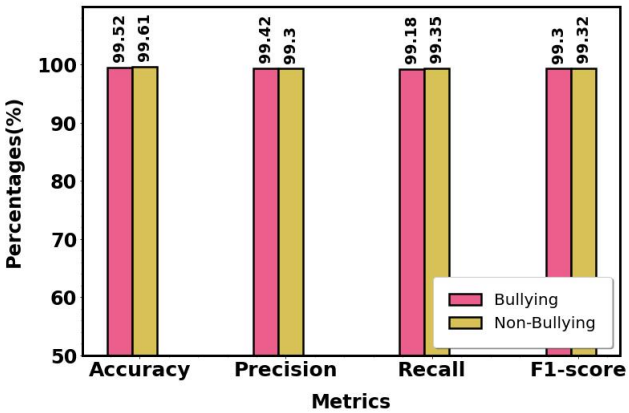


Figure 6: Classification results of the proposed method for all classes on Dataset 1.

The categorization outcome is presented in Figure 6. Table 4 compares the proposed approach's precision, recall, and F1-score metrics with different techniques on Dataset 1. Recall evaluates the percentage of true positives that are

accurately detected; precision gauges how well a model predicts positive outcomes and the F1-score. Logistic Regression outperforms the other approaches in metrics of precision (80.8%), recall (81.7%), and F1-score (81.2%). With precision, recall, and F1-score values of 81.3%, 80.6%, and 80.8%, respectively, the Convolutional Neural Network (CNN) performs better. With 84.5% precision, 84.2% recall, and an F1-score of 84.3%, the C-LSTM (Convolutional-Long Short-Term Memory) model shows even more improvement.

Table 4: Precision, Recall, and F1-score of the proposed approach vs other approaches on Dataset 1.

Methods	**Precision (%)**	**Recall (%)**	**F1-score (%)**
Logistic Regression	80.8	81.7	81.2
CNN	81.3	80.6	80.8
C-LSTM	84.5	84.2	84.3
BiLSTM	74	76	75
SVM	81.4	80.8	81.1
Self-Attention	89.7	88.5	89.1
Proposed Method	99.56	99.26	99.31

By comparison, the BiLSTM (Bidirectional Long Short-Term Memory) approach obtains 74% precision, 76% recall, and 75% F1 score. The recall, precision, and F1-score values of the Support Vector Machine (SVM), which produce values of 81.4%, 80.8%, and 81.1%, respectively, are comparable to those of CNN. The Self-Attention mechanism performs effectively with precision, recall, and F1-score metrics of 89.7%, 88.5%, and 89.1%, respectively. However, the proposed approach outperforms all previous methods with remarkable F1-score, precision, and recall values of 99.26%, 99.31%, and 99.56%, respectively.

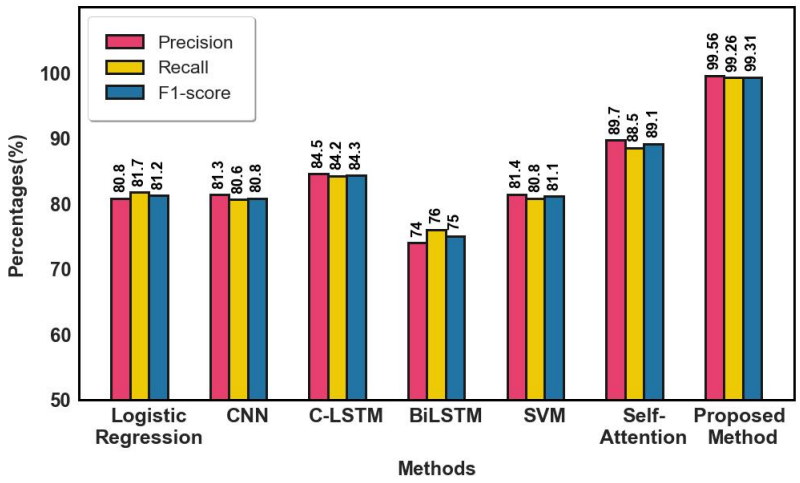


Figure 7: Comparison of proposed and other techniques on Dataset 1.

Figure 7 compares the outcomes of the proposed and existing methods. This notable increase in performance highlights how effectively the proposed hybrid strategy accurately classified cases within Dataset 1. The proposed method's robustness and superiority over current techniques are indicated by its excellent precision, recall, and F1-score metrics, which make it an appealing option for emotion classification tasks. As indicated in Table 5, we employ three sets of experiments to verify the accuracy of the proposed method. The dataset is split between 25% testing and 75% training in the first trial. The second experiment's dataset is divided in half (i.e., 50% testing and 50% training). The third trial uses 30% of the dataset for training and 70% for testing. The dataset is then set to 90% testing and 10% training in the last trial, for 80% and 20% training.

Table 5: Accuracy with distinct testing-training splits.

Training (%)	Testing (%)	Accuracy (%)
80	20	99.56
50	50	99.37
30	70	99.10
20	80	98.24
10	90	98.12

The accuracy increases when we divide the dataset into 25% testing and 75% training, 85.36%, 82.78%, 30% training and 70% testing, 20% training and 80% testing, 79.75%, and 10% training and 90% testing. The dataset is split into 25% testing and 75% training, 75.36% training, and 82.76% testing. In the final research series, the approach's performance drops as the quantity of training data also drops.

4.7 Performance of dataset 2:

The most common terms in the Twitter dataset are displayed in Figure 8. The names of politicians, counties, derogatory terms, and some common words are the most often occurring words in the offending tweets. When combined with additional words in the same sentence, some words become offensive phrases even though they are not. Figure 8 shows the word clouds for offensive and non-offensive classes for Dataset 2.



(a) (b)

Figure 8: Word clouds of Dataset 2: (a) Offensive and (b) Non-Offensive.

Table 6 shows the results of a proposed method's classification of Dataset 2, divided into two categories: bullying and non-bullying. The proposed method performs admirably in every category, with high accuracy scores of 99.65% for classes that are not bullying and 99.78% for bullying classes.

Table 6: Classification report of proposed method for all classes on Dataset 2.

Category	Accuracy	Precision	Recall	F1-score
Bullying	99.78	99.28	99.62	99.45
Non-Bullying	99.65	99.45	99.54	99.49

Bullying precision is 99.28%, and non-bullying precision is 99.45%. These are very high precision numbers for both groups. Furthermore, the model exhibits remarkable recall rates: 99.62% for bullying and 99.54% for non-bullying.

Additionally, bullying achieves an F1-score of 99.45%, while Non-Bullying achieves 99.49%. The F1 scores, which balance recall and precision, are high. These outcomes demonstrate how well the proposed method classified Dataset 2 instances of bullying and non-bullying categories.

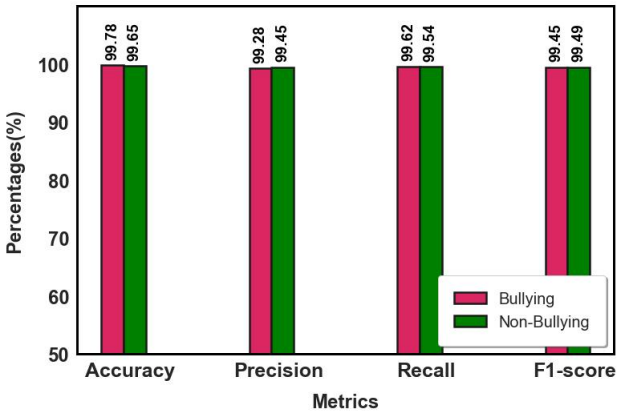


Figure 9: Classification results of the proposed method for all classes on Dataset 2.

The categorization outcome is presented in Figure 9. The proposed approach and baseline methods on Dataset 2 are compared and evaluated in Table 7. Using GloVe embedding, the baseline techniques include Bayes, CNN (Convolutional Neural Network), Bidirectional Gated Recurrent Unit (Bi-GRU), BERT (Bidirectional Encoder Representations from Transformers), and Logistic Regression.

Table 7: Comparison of proposed method and baseline other methods on Dataset 2.

Method	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
LR	80.90	79.36	80.97	79.74
Bayes	83.11	81.78	83.14	81.58
CNN	92.91	92.09	83.78	88.63
BERT	95.34	92.61	93.57	95.70
Bi-GRU with GloVe	96.98	99.22	96.74	98.56
Proposed Method	99.71	99.36	99.58	99.47

With an accuracy of 99.71%, precision of 99.36%, recall of 99.58%, and an F1-score of 99.47%, the proposed approach performs substantially better than the baseline methods on all parameters. This shows how better the proposed approach is at correctly categorizing Dataset 2 instances. Notably, the Bi-GRU model with GloVe embeddings performs well, but the proposed method outperforms even this very efficient baseline, demonstrating how effectively it solves the classification issue.

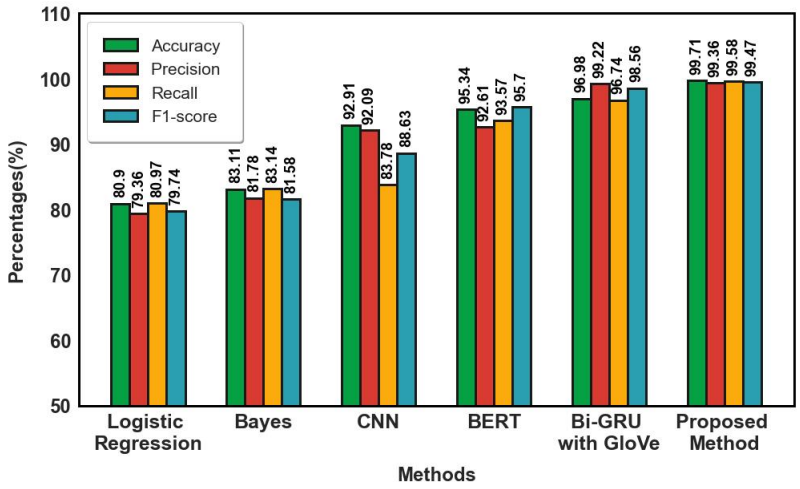


Figure 10: Comparison of proposed and other techniques on Dataset 2

Figure 10 compares the outcomes of the proposed and existing methods. This research presented a holistic approach to determining the dataset's cyberbullying level. This was accomplished using a deep learning classifier to determine whether a comment was cyberbullying. Then, the severity was classified into several groups using fuzzy criteria. The present study aimed to provide a systematic approach for assessing the severity of cyberbullying using a multi-class classification system. The outcome of the DKNN method recognizes and classifies bullying comments. Based on the remarks made by different bullies, the fuzzy system assesses the bullying intensity. A random selection of posts from the dataset was used to evaluate the fuzzy rule; the posts were already classified when the evaluation process began, and the model tested them before comparing the results to those included in the dataset.

4.8 Comparison of the proposed method with machine learning-based techniques

Regarding accuracy in emotion classification tasks on two distinct datasets, designated as Datasets 1 and 2, Table 8 compares some machine learning techniques and the proposed approach. The Proposed Method, Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), and Logistic Regression (LR) are among the techniques that are contrasted. SVM obtains an accuracy of 83.06% for Dataset 1, DT attains 78.63%, NB attains 73.86%, and LR attains 82.06%. Still, with an outstanding accuracy of 99.56%, the proposed approach performs substantially better than current conventional machine learning algorithms. This exceptional result shows how effective the proposed method is in correctly categorizing emotions in Dataset 1.

Table 8: Comparison with difference machine learning methods and the proposed method.

Method	**Accuracy (%)**	
	Dataset 1	Dataset 2
SVM	83.06	89.17
DT	78.63	78.78
NB	73.86	88.88
LR	82.06	89.03
Proposed Method	99.56	99.71

Comparably, on Dataset 2, SVM scores 89.17%, DT scores 78.78%, NB scores 88.88%, and LR scores 89.03% in accuracy. The proposed method outperforms all others again, attaining the most excellent accuracy of 99.71%.

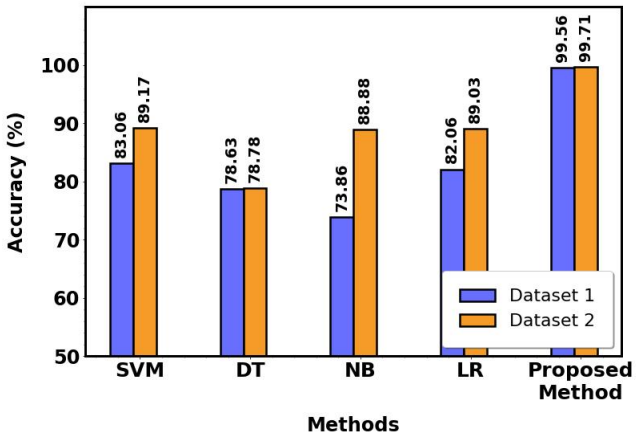


Figure 11: Evaluation of the proposed method with some machine learning techniques.

This remarkable precision indicates the resilience and efficacy of the proposed approach in precisely categorizing emotions in Dataset 2. The comparison demonstrates that the proposed approach performs consistently better than typical machine learning techniques on both datasets, demonstrating its enhanced reliability in classifying bully content, as shown in Figure 11.

4.9 Overall comparison of the proposed method and previous techniques

A thorough comparison of proposed and existing methods for identifying unsafe substances and cyberbullying in diverse datasets is given in Table 9, along with a breakdown of each method's accuracy. Each approach's efficacy in identifying harmful online conduct is demonstrated by the list of authors and the datasets they used. Starting with Iwendi et al. [9], their method uses the Wiki-Detox dataset and Bidirectional Long Short-Term Memory (BiLSTM) to achieve 82.18% accuracy. On the Twitter platform dataset, the DEA-RNN technique achieved a significantly higher accuracy of 90.45%, as reported by Murshed et al. [15]. Using the Bengali cyberbully text dataset, Akhter et al. [2] experimented with Decision Trees (DT), Random Forests (RF), Logistic Regression (LR), and Multi-Layer Perceptrons (MLP) and achieved the highest accuracy of 98.57%.

Table 9: Overall comparison of proposed and existing methods on literature.

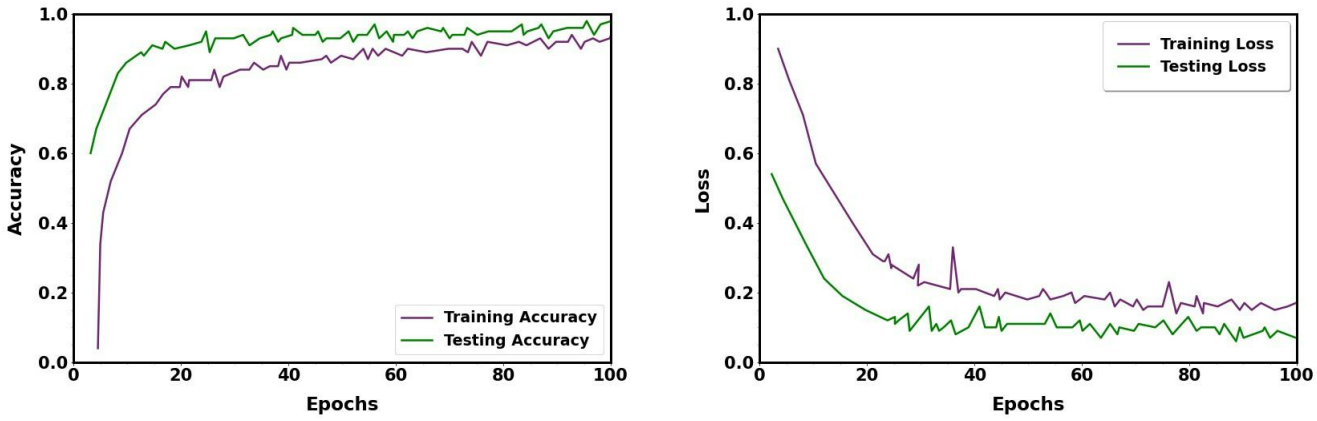
Authors	Methods	Datasets	Accuracy (%)
Iwendi et al. [9]	BiLSTM	Wiki-Detox dataset	82.18
Murshed et al. [15]	DEA-RNN	Twitter platform dataset	90.45
Akhter et al. [2]	DT, RF, LR, MLP	Bengali cyberbully text dataset	98.57
Suhas et al. [21]	ELM	Twitter platform dataset	99.41
Murshed et al. [16]	FAEO-ECNN	RW-CB-Twitter dataset	91.89
Murshed et al. [16]	FAEO-ECNN	CB-MNDLY dataset	92.91
Almomani et al. [4]	SVM	Cyberbullying image dataset	66.9
Alqahtani & Ilyas [5]	DT	Cyberbullying dataset	89
Proposed Method	DKNN	Twitter Dataset	99.56
		Wikipedia Attack dataset	99.71

The Extreme Learning Machine (ELM) method was suggested by Suhas et al. [21] and achieved a high accuracy of 99.41% on the Twitter platform dataset. The FAEO-ECNN approach was initially suggested by Murshed et al. [16], who obtained accuracy rates of 92.91% on the CB-MNDLY dataset and 91.89% on the RW-CB-Twitter dataset. Using Support Vector Machines (SVM) on the Cyberbullying picture dataset, Almomani et al. [4] obtained a slightly lower accuracy of 66.9%. Last, Alqahtani & Ilyas [5] used Decision Trees (DT) with an 89% accuracy rate on a Cyberbullying dataset. In comparison, the proposed technique, DKNN, showed impressive gains in accuracy, obtaining 99.56% on the Twitter dataset and an even greater accuracy of 99.71% on the Wikipedia Attack dataset. Overall, the proposed

DKNN approach performs better than existing techniques on several datasets, demonstrating its effectiveness in precisely detecting harmful content and cyberbullying on the internet. These findings highlight the significance of ongoing research and innovation in creating efficient approaches for reducing hazardous behaviours in online communities.

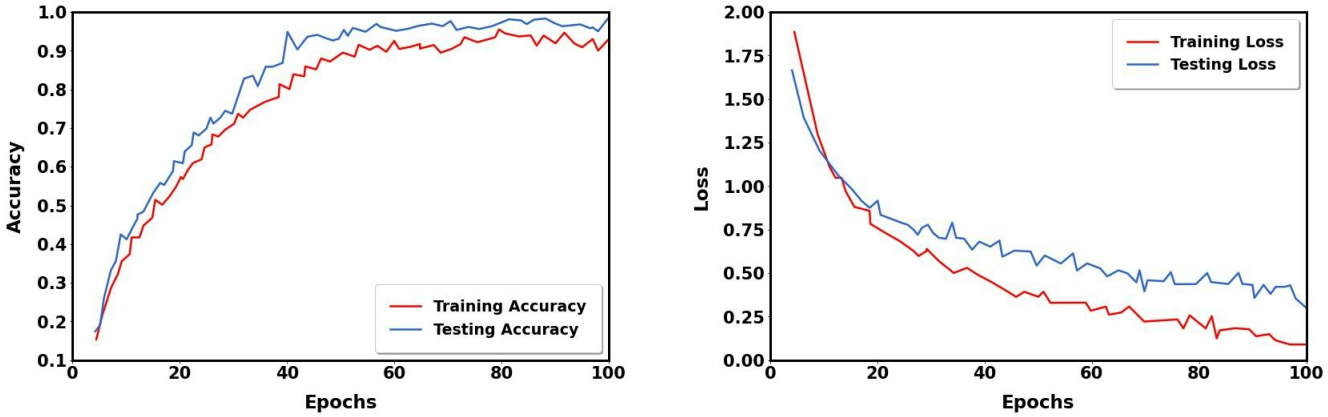
4.10 Evaluation of the performance of testing and training

Figures 12 and 13 demonstrates a graph of loss and detection accuracy as the amount of iteration phases increased. This graph shows that convergence is positively impacted by the strategy covered in this research. There were testing and training steps for the dataset.



(a) (b)

Figure 12: Training and testing assessment, (a) accuracy, (b) loss for the IITR-IAR Dataset 1.



(a) (b)

Figure 13: Training and testing assessment, (a) accuracy, (b) loss for the IITR-IAR Dataset 2.

For this research, 25% of the testing and 75% of the training data were created specially. At this moment, the learning rate is 0.1.

4.11 Computational time complexity

For data analysis and machine learning tasks, Table 10 compares the computational time complexity of several methodologies presented by different researchers. Milliseconds (ms) indicate the computation time, and each row represents a distinct author or research group and their methodology. BiLSTM, a bidirectional Long Short-Term Memory network with a computation time of 34 ms, was suggested for use by Iwendi et al. [9]. With the introduction of DEA-RNN, Murshed et al. [15] achieved a marginally reduced computation time of 29 ms. With a total computation time of 48

ms, Akhter et al. [2] investigated some approaches, including Decision Trees (DT), Random Forests (RF), Logistic Regression (LR), and Multi-layer Perceptrons (MLP).

Table 10: Comparison of computational time complexity of proposed and existing methods.

Authors	Methods	Computation Time (ms)
Iwendi et al. [9]	BiLSTM	34
Murshed et al. [15]	DEA-RNN	29
Akhter et al. [2]	DT, RF, LR, MLP	48
Suhas et al. [21]	ELM	32
Murshed et al. [16]	FAEO-ECNN	36
Almomani et al. [4]	SVM	42
Alqahtani & Ilyas [5]	DT	23
Proposed Method	DKNN	18

Extreme Learning Machines (ELM) were suggested by Suhas et al. [21] and took 32 ms to calculate. On the other hand, FAEO-ECNN, a method based on Fisher's Discriminant Analysis and Evolutionary Convolutional Neural Networks, was presented by Murshed et al. [16] and required 36 ms. Support Vector Machines (SVM) with a 42 ms computation time were used by Almomani et al. [4]. Decision Trees (DT), which have the shortest computation time of any algorithm now in use (23 ms), were introduced by Alqahtani & Ilyas. [5] With the shortest computation time of 18 ms, the proposed DKNN (Distance-based K-Nearest Neighbors) method in the table is a potential solution for practical data analysis and machine learning applications.

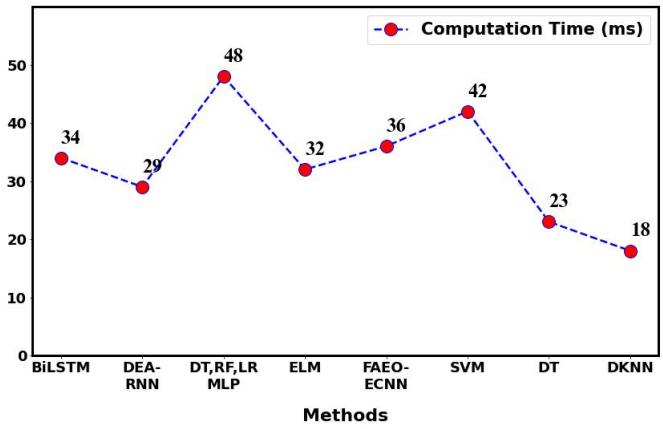


Figure 14: Comparison of proposed and existing method's running time.

Figure 14 shows the overall comparison of the proposed and existing method's computational time complexity. Analyzing the temporal complexity at each step of the process is essential to accurately calculating the computational complexity of the proposed approach. Each activity in the pre-processing step has its own computational demands; these activities include data cleansing, tokenization, stemming, stop word removal, count vectorization, and normalization. The quantity of tokens or attributes generated during this phase, as well as the size of the dataset, all have an impact on these needs. The amount of attributes being processed and the number of iterations required for the algorithm to converge to an optimal solution allow the complexity of the feature selection step, where the Farmland Fertility Algorithm (FFA) is used. The amount of examples in the dataset, the dimensionality of the feature space, and the computational steps necessary to determine the nearest neighbors and make classification assessments all influence the DKNN algorithm's complexity during the detection phase. Furthermore, fuzzy logic operations whose complexity depends on the quantity of fuzzy sets and rules involved in the categorization process are included in the severity categorization phase, which makes use of Linguistic Hesitant Pythagorean Fuzzy Sets (HPFSs). The computational needs associated with our proposed approach can be better understood by computing the theoretical time complexity for each of these steps, which can be stated in terms like $O(n)$ or $O(n^2)$. The findings are meaningful and accurate

because of this thorough analysis, which also makes the algorithm's scalability and efficiency clear. It also makes it possible to compare our results with those of other approaches in a more thorough manner.

Conclusion

Due to the ease of social connection it provides, online social networks have grown to be a significant part of everyday life. Cyberbullying detection is an important topic to investigate; nevertheless, because of the rise in antisocial conduct on social media platforms like Twitter and Wikipedia, including hate speech, trolling, and cyberbullying, as well as the penalties that these users face. In this study, a novel method of detecting cyberbullying is proposed. Two well-known cyberbullying datasets are used to assess the proposed approach. The experimental findings demonstrate the importance of this approach for categorizing brief messages. The proposed approach outperformed the most advanced techniques on the two datasets, with an accuracy of almost 99.56% and 99.71% for Twitter and Wikipedia datasets, respectively, when the dataset was divided into 20% testing and 80% training. Despite this, the study had certain shortcomings. First, the size and tweet categories of the dataset are constrained. We intend to use a more extensive dataset using this method. We believe a larger dataset will improve the approach's performance. For deep learning algorithms to function well, they need to work with big data sets. In addition, we will expand the proposed structure by including several channels. A larger dataset may require more channels, enhancing the framework's performance. Optimizing the weights and other variables of deep and extensive neural networks can also be aided by a large dataset. Additionally, we intend to test the framework we propose using multilingual tweets.

Declaration

Ethics approval: This material is the authors' own original work, which has not been previously published elsewhere. The paper reflects the authors' own research and analysis in a truthful and complete manner.

Competing interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We declare that this manuscript is original, has not been published before and is not currently being considered for publication elsewhere.

Funding: No funding was received to assist with the preparation of this manuscript.

Availability of data and material: Data will be available when requested.

Authors' contributions: The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

[1] T. Ahmed, S. Ivan, M. Kabir, H. Mahmud, K. Hasan, *Performance analysis of transformer-based architectures and their ensembles to detect trait-based cyberbullying*, Social Network Analysis and Mining, **12**(1) (2022), 99. https://doi.org/10.1007/s13278-022-00934-4

[2] A. Akhter, U. K. Acharjee, M. A. Talukder, M. M. Islam, M. A. Uddin, *A robust hybrid machine learning model for detecting Bengali cyberbullying in social media*, Natural Language Processing Journal, **4** (2023), 100027. https://doi.org/10.1016/j.nlp.2023.100027

[3] A. Al-Marghilani, *Artificial intelligence-enabled cyberbullying-free online social networks in smart cities*, International Journal of Computational Intelligence Systems, **15**(1) (2022), 9. https://doi.org/10.1007/s44196-022-00063-y

[4] A. Almomani, K. Nahar, M. Alauthman, M. A. Al-Betar, Q. Yaseen, B. B. Gupta, *Image cyberbullying detection and recognition using transfer deep machine learning*, International Journal of Cognitive Computing in Engineering, **5** (2024), 14-26. https://doi.org/10.1016/j.ijcce.2023.11.002

[5] A. F. Alqahtani, M. Ilyas, *An ensemble-based multi-classification machine learning classifiers approach to detect multiple classes of cyberbullying*, Machine Learning and Knowledge Extraction, **6**(1) (2024), 156-170. https://doi.org/10.3390/make6010009

[6] M. Alzaqebah, G. M. Jaradat, D. Nassan, R. Alnasser, M. K. Alsmadi, I. Almarashdeh, S. Alkhushayni, *Cyberbullying detection framework for short and imbalanced Arabic datasets*, Journal of King Saud University-Computer and Information Sciences, **35**(8) (2023), 101652. https://doi.org/10.1016/j.jksuci.2023.101652

[7] K. I. Arce-Ruelas, O. Alvarez-Xochihua, L. Pellegrin, L. Cardoza-Avendaño, J. Á. González-Fraga, *Automatic cyberbullying detection: A Mexican case in high school and higher education students*, IEEE Latin America Transactions, **20**(5) (2022), 770-779. https://doi.org/10.1109/TLA.2022.9693561

[8] C. E. Gomez, M. O. Sztainberg, R. E. Trana, *Curating cyberbullying datasets: A human-AI collaborative approach*, International Journal of Bullying Prevention, **4**(1) (2022), 35-46. https://doi.org/10.1007/s42380-021-00114-6

[9] C. Iwendi, G. Srivastava, S. Khan, P. K. R. Maddikunta, *Cyberbullying detection solutions based on deep learning architectures*, Multimedia Systems, **29**(3) (2023), 1839-1852. https://doi.org/10.1007/s00530-020-00701-5

[10] A. Kumar, N. Sachdeva, *A Bi-GRU with attention and CapsNet hybrid model for cyberbullying detection on social media*, World Wide Web, **25**(4) (2022), 1537-1550. https://doi.org/10.1007/s11280-021-00920-4

[11] A. Kumar, N. Sachdeva, *Multi-input integrative learning using deep neural networks and transfer learning for cyberbullying detection in real-time code-mix data*, Multimedia Systems, **28**(6) (2022), 2027-2041. https://doi.org/10.1007/s00530-020-00672-7

[12] A. Kumar, N. Sachdeva, *Multimodal cyberbullying detection using capsule network with dynamic routing and deep convolutional neural network*, Multimedia Systems, **28**(6) (2022), 2043-2052. https://doi.org/10.1007/s00530-020-00747-5

[13] K. Maity, A. Kumar, S. Saha, *A multitask multimodal framework for sentiment and emotion-aided cyberbullying detection*, IEEE Internet Computing, **26**(4) (2022), 68-78. https://doi.org/10.1109/MIC.2022.3158583

[14] A. Mangaonkar, R. Pawar, N. S. Chowdhury, R. R. Raje, *Enhancing collaborative detection of cyberbullying behaviour in Twitter data*, Cluster Computing, **25**(2) (2022), 1263-1277. https://doi.org/10.1007/s10586-021-03483-1

[15] B. A. H. Murshed, J. Abawajy, S. Mallappa, M. A. N. Saif, H. D. E. Al-Ariki, *DEA-RNN: A hybrid deep learning approach for cyberbullying detection in Twitter social media platform*, IEEE Access, **10** (2022), 25857-25871. https://doi.org/10.1109/ACCESS.2022.3153675

[16] B. A. H. Murshed, Suresha, J. Abawajy, M. A. N. Saif, H. M. Abdulwahab, F. A. Ghanem, *FAEO-ECNN: Cyberbullying detection in social media platforms using topic modelling and deep learning*, Multimedia Tools and Applications, **82**(30) (2023), 46611-46650. https://doi.org/10.1007/s11042-023-15372-3

[17] V. L. Paruchuri, P. Rajesh, *CyberNet: A hybrid deep CNN with N-gram feature selection for cyberbullying detection in online social networks*, Evolutionary Intelligence, **16**(6) (2023), 1935-1949. https://doi.org/10.1007/s12065-022-00774-3

[18] S. Paul, S. Saha, *CyberBERT: BERT for cyberbullying identification: BERT for cyberbullying identification*, Multimedia Systems, **28**(6) (2022), 1897-1904. https://doi.org/10.1007/s00530-020-00710-4

[19] S. Paul, S. Saha, M. Hasanuzzaman, *Identification of cyberbullying: A deep learning based multimodal approach*, Multimedia Tools and Applications, **1** (2022), 1-20. https://doi.org/10.1007/s11042-020-09631-w

[20] P. K. Roy, F. U. Mali, *Cyberbullying detection using deep transfer learning*, Complex and Intelligent Systems, **8**(6) (2022), 5449-5467. https://doi.org/10.1007/s40747-022-00772-z

[21] R. Suhas Bharadwaj, S. Kuzhalvaimozhi, N. Vedavathi, *A novel multimodal hybrid classifier-based cyberbullying detection for social media platforms*, In Proceedings of the Computational Methods in Systems and Software, **1** (2022), 689-699. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-21438-7_57

[22] F. Wu, B. Gao, X. Pan, Z. Su, Y. Ji, S. Liu, Z. Liu, *FACapsnet: A fusion capsule network with congruent attention for cyberbullying detection*, Neurocomputing, **542** (2023), 126253. https://doi.org/10.1016/j.neucom.2023.126253