mathematics

MDPI

Article

Cyberbullying Detection on Twitter Using Deep Learning-Based Attention Mechanisms and Continuous Bag of Words Feature Extraction

Suliman Mohamed Fati [1], Amgad Muneer [2,3,*], Ayed Alwadain [4] and Abdullateef O. Balogun [3]

[1] Information Systems Department, Prince Sultan University, Riyadh 11586, Saudi Arabia; sgaber@psu.edu.sa
[2] Department of Imaging Physics, The University of Texas MD Anderson Cancer Center, Houston, TX 77030, USA
[3] Department of Computer and Information Sciences, Universiti Teknologi Petronas, Seri Iskandar 32160, Malaysia; abdullateef.ob@utp.edu.my
[4] Computer Science Department, Community College, King Saud University, Riyadh 11451, Saudi Arabia; aalwadain@ksu.edu.sa
* Correspondence: muneeramgad@gmail.com or amabdulraheem@mdanderson.org

Abstract: Since social media platforms are widely used and popular, they have given us more opportunities than we can even imagine. Despite all of the known benefits, some users may abuse these opportunities to humiliate, insult, bully, and harass other people. This issue explains why there is a need to reduce such negative activities and create a safe cyberspace for innocent people by detecting cyberbullying activity. This study provides a comparative analysis of deep learning methods used to test and evaluate their effectiveness regarding a well-known global Twitter dataset. To recognize abusive tweets and overcome existing challenges, attention-based deep learning methods are introduced. The word2vec with CBOW concatenated formed the weights included in the embedding layer and was used to extract the features. The feature vector was input into a convolution and pooling mechanism, reducing the feature dimensionality while learning the position-invariant of the offensive words. A SoftMax function predicts feature classification. Using benchmark experimental datasets and well-known evaluation measures, the convolutional neural network model with attention-based long- and short-term memory was found to outperform other DL methods. The proposed cyberbullying detection methods were evaluated using benchmark experimental datasets and well-known evaluation measures. Finally, the results demonstrated the superiority of the attention-based 1D convolutional long short-term memory (Conv1DLSTM) classifier over the other implemented methods.

Keywords: cyberbully; RNN; CNN; LSTM; BiLSTM; word2vec; text classification

MSC: 68T50; 68T07

Citation: Fati, S.M.; Muneer, A.; Alwadain, A.; Balogun, A.O. Cyberbullying Detection on Twitter Using Deep Learning-Based Attention Mechanisms and Continuous Bag of Words Feature Extraction. *Mathematics* **2023**, *11*, 3567. https://doi.org/10.3390/math11163567

Academic Editors: Ravil Muhamedyev and Evgeny Nikulchev

Received: 13 July 2023
Revised: 13 August 2023
Accepted: 15 August 2023
Published: 17 August 2023
Corrected: 31 October 2023

1. Introduction

The emergence and adoption of social media as an important communication and interaction platform in recent times cannot be overemphasized. As stated by Chaffey [1], there are over 3 billion active users that communicate and interact with each other daily via social media. Thus, social media is pivotal to the definition and spread of information and content in the modern world. Moreover, social media platforms enable relationship- and interest-based virtual communities that allow connection and networking among people [2–4]. The increased popularity of these social media platforms (Facebook, Twitter, Instagram, Tinder, etc.) allows the sharing of various forms of multimedia messages among its users [5]. Mining and deriving analytics from these social media platforms have been standardized as feasible and viable methods or options to identify real-time insights from most users. For instance, these social media platforms' operations were pivotal during the

COVID-19 pandemic, as real-time data on events and discoveries were easily disseminated among people in various places [6,7].

Although it is apparent that social media platforms offer numerous benefits to their users, these platforms may also be exploited for malevolent purposes [2,8,9]. Cyberbullying is a prominent and profound example of these malevolent trends [10]. Specifically, the pervasiveness of these social media platforms has undeniably sparked, advanced, and worsened bullying among its users [11].

Bullying seems to have been present throughout the history of human civilization, and it entails abusing someone by embarrassing or bothering them in any way that causes emotional, psychological, or physical harm. When this assault occurs via the Internet, it is known as cyberbullying or cybervictimization [11]. Cyberbullying can be described as bullying a person or a group of people, who are usually referred to as "victim(s)", with the aid of an Internet, mobile, or electronic device by sending improper textual or non-textual multimedia content [3]. In other words, cyberbullying is the continuous display of unpleasant acts carried out on the "victim(s)" to exact fear, annoyance, pain, or harm via electronic media and social media platforms [9]. Cyberbullying has been a major problem in the last decade, mostly impacting children and adolescents. For instance, a United States (US)-based study reported that over 43% of teenagers in the US are being cyberbullied [12]. According to ER statistics, around 18% of Europe's youngsters have been affected by someone bullying or harassing them via the Internet and mobile phones [3]. As stated in the 2014 EU Kids Online Report, more than 20% of children (aged between 11 and 16 years) experience cyberbullying [3,13]. In Sweden, which is a developed country, the prevalence of cyberbullying has hit a tipping point, gradually growing and worsening [2]. These reports demonstrate how critical it is to develop an acceptable, quick, and tested solution to this Internet-based problem. It is, therefore, imperative to evaluate and address cyberbullying from a variety of perspectives, including automatic identification and prevention of such incidents [2,3,9,12].

Due to technological development, the social privileges (anonymity) that social media provide, and access to a broader audience, cyberbullying has grown exponentially [11]. This issue necessitates the development of intelligent tools and strategies that recognize, detect, and analyze cyberbullying using existing social multimedia data to mitigate its harmful impact. Automated cyberbullying detection is a classification issue, with the goal being categorizing each abusive or insulting comment/post/message/image as bullying or non-bullying [9,13,14]. Although some social media platforms, such as YouTube and Twitter, have embedded safety centers to monitor and control cyberbullying, the problem still lingers, and there is a need for more definite solutions [2,13].

In addition, much research has been carried out to investigate new paradigms through which to deal with the uncertainty or fuzziness created by social media data, resulting in computationally effective automated cyberbullying detection systems [15–17]. Recently, the automatic identification of cyberbullying using deep learning (DL) and machine learning (ML) has received a lot of interest [3,9,14,15,18,19]. For instance, Muneer and Fati [18] investigated the performance of different ML algorithms in cyberbullying detection. Also, Alam et al. [20] analyzed the performance of four ML methods and three ensemble methods in cyberbullying detection. Experimental results derived from these studies showed the effectiveness of ML methods of cyberbullying detection. Similarly, Dadvar and Eckert [13] and Iwendi et al. [3] deployed DL techniques to aid cyberbullying detection. Their respective results indicated the applicability of DL methods to cyberbullying detection. Moreover, the findings of their studies indicated the superiority of DL techniques to ML techniques in cyberbullying detection. Also, DL techniques have the advantage of being more easily used on different datasets (transfer learning) than conventional ML-based approaches [13]. Nonetheless, the integration of an appropriate feature extraction method using a DL technique can further improve its detection performance [21]. Moreover, the proficiency of DL techniques needs to be investigated to ascertain and validate the efficacies of the DL technique in cyberbullying detection. Hence, this study aims to propose and conduct an

extensive empirical analysis of various DL techniques using multiple feature extraction methods. This study will guide researchers and practitioners on how to select appropriate DL and feature extraction methods in cyberbullying detection. Specifically, this paper has a multifold contribution as follows:

1. The performances of six DL-based attention mechanism techniques used in cyberbullying detection are investigated and evaluated;
2. The performances of two feature extraction methods used in the selection of prominent features of cyberbullying detection are evaluated;
3. The evaluation of the effectiveness and performance of DL methods using different feature extraction methods is performed through empirical analysis.

This paper is succinctly organized as follows: Section 2 discusses related studies and provides related works in the context of using machine learning in cyberbullying detection. Section 3 presents the theoretical background of the proposed cyberbullying model, while Section 4 designates the proposed method, including details such as experimental results, datasets, parameter settings, and visualized results with their discussions is provided, and concluding remarks are presented in Section 5.

2. Related Works

Teenagers' online activities have grown in recent years, especially on social media platforms, with more individuals being subjected to cyberbullying. Comments containing harsh words have a negative impact on adolescent psychology, demoralize teenagers, and might even escalate into despair. Chavan and Shylaja [22] provided two feature extraction (FE) methods that were used to identify perceived unfavorable and insulting remarks frequently aimed at peers. Their combination of hand-crafted features and conventional FE methods tends to improve the suggested system's detection accuracy. Even though current methods influenced by DL and ML techniques have enhanced cyber-bullying detection performance, the absence of acceptable standard labelled datasets continues to restrict the development of this approach. Consequently, Chen et al. [23] presented a method in which feedback in the form of comments was combined with crowdsourcing, including actual realistic scenarios of deliberately unpleasant or pleasant remarks.

Kumar and Sachdeva [9] investigated the concept of utilizing soft computing techniques to identify cyberbullying, particularly via social media sites. They compared their findings to prior research and concluded that social media companies should use a meta-analytic approach to cyberbullying detection. Frommholz et al. [24] developed a technique to identify, text categorize, and customize text-based cyberstalking. It was an ethical framework and represented a means of detecting text-based cyberstalking. The suggested technique emphasized the need to use additional methods, like forensic analysis, to identify bullies.

Bruwaene et al. [25] showed several phases and numerous approach systems that integrated crowdsourcing for topic and keyword tagging and, subsequently, used ML methods to discover additional topics requiring evaluation. They concluded that if the dataset was trained using their method, the models might perform very well. They emphasize the need for positive and regular parental, instructor, or peer monitoring to improve cyberbullying prevention measures. Khan [26] investigated cross-lingual emotion recognition using four different languages: English, Urdu, Italian, and German. They obtained an accuracy rate of 91.25%. In another study, Zhao and Mao [27] presented a technique to evaluate the underlying structure of cyberbullying features and acquire a robust and discriminative representation of text. Their proposed technique outperformed existing basic text portrayal methods. Rosa et al. [28] utilized 22 studies and tests to verify existing methods of automated cyberbullying identification. They eventually concluded by presenting findings suggesting that cyberbullying is often misrepresented and the inherent imbalance nature of cyberbullying dataset is an often an issue. Sugandhi et al. [29] explored the same technique. In this regard, the system of response grading identified the cyberbullying's heinousness and reacted accordingly. Rakib and Soon [30] started with a Reddit word-embedding appli-

cation, before moving on to a cyberbullying detection model utilizing a public dataset of 6594 comments derived from Kaggle. The system was built and trained using the random forest (RF) algorithm. The prediction model provides 90% score in terms of area under the curve (AUC), while the precision was 0.89. The disadvantage of this study is that the sample was unbalanced, with cyberbullying messages accounting for just 25% of the total [30]. Moreover, Agrawal and Awekar [31] and Dadvar and Eckert [13] performed similar studies deploying DL techniques to build the prediction models. They utilized over-sampling techniques, which may be a drawback of their proposed methods, since additional details are added to the datasets.

Haidar et al. [32] developed a method to prevent cyberbullying in several languages. The suggested method was evaluated using an authentic Arabic dataset obtained from Arab countries. Haider et al. deployed two ML classifiers, namely support vector machine (SVM) and naive Bayes (NB), with acceptable results. Nonetheless, this study may be improved using DL techniques and expanding the dataset size. Also, Al-Ajlan and Ykhlef [33] used 20,000 random tweets to create a cyberbullying technique. To eliminate noisy and undesirable data, data pre-processing was used, whereby the data were partitioned and labeled. To label tweets for training data, tweet categorization was given. A dataset was later classified using deep convolutional neural networks (CNN). There were no promising experimental findings. The research must be broadened by considering a big data set and many languages. Similarly, Banerjee et al. [34] utilized deep convolutional neural networks (DCNN) to analyze the 69,874-tweet dataset derived from Twitter. Glove's open-source word-embedding model was used to map tweets to vectors. The testing findings revealed that the authors obtained a 93.7% accuracy rate using deep convolutional neural networks. Detecting cyberbullying in conversations that include both Hindi and English may, however, broaden the scope of the study. The primary focus of Wulczyn et al.'s study [35] was the Wiki-Detox dataset. They developed a classifier that provided results, in terms of AUC and Spearman's correlation, that are as excellent as those of three human workers combined. Bozyiğit et al. [2] aimed to evaluate cyberbullying detection in the Turkish language using eight diverse ANN algorithms. The study recorded a F1-measure score of 91%, which outperformed the existing ML classifiers. Pawar and Raje's [32] work is another example of a comparable situation. Their article provided a method of identifying and preventing cyberbullying, with a focus on Arabic-language material. Finally, their method was used to optimize DL, resulting in excellent parameter tuning. Jeyasheeli and Selva [36] utilized another example to demonstrate the inadequacy of prior techniques' categorization. The main research issue that is addressed in this study is the integration of an appropriate FE method that uses a DL technique to enable cyberbullying detection. Furthermore, the effectiveness of DL approaches must be studied to establish and verify the efficacy of DL approaches regarding cyberbullying detection. Table 1 shows the comparison between the related works.

Table 1. Comparison between the related works.

Authors	Methods	Datasets	Features	Language
[2]	Chi-Square and (SVM, LR, RF, kNN, and NBM Adaboost	Turkish datasets	Textual, social media	Turkish
[22]	Chi-Square) and (SVM, LR) text	Kaggle	Textual	English
[23]	SVM	Online services	Textual, content-based, and context-based features	English
[24]	Anti-cyberstalking text-based system (ACTS)	N/A	Textual	English
[25]	Multi-technique annotation and (SVM, CNN, and XGBoost)	VISR dataset	Textual	English

Table 1. *Cont.*

Authors	Methods	Datasets	Features	Language
[27]	Semantic-enhanced marginalized denoising auto-encoder	Twitter and MySpace	Textual and semantics	English
[29]	SVM, NB, and kNN	Twitter	Textual	English
[30]	word2vec skip-gram models with RF	Reddit	Textual	English
[30]	CNN, LSTM, BLSTM, and BLSTM	Formspring, Twitter, and Wikipedia	Textual	English
[13]	CNN, LSTM, BLSTM, and BLSTM	Formspring, Twitter, Youtube, and Wikipedia	Textual	English
[32]	NLP with NB, SVM, kNN, and DT	Twitter and Facebook	Textual	Arabic
[33]	CNN	Twitter	Textual	English
[34]	CNN	Twitter	Textual	English
[35]	Crowdsourcing using LR and MLP	Wikipedia	Textual	English
[23]	Diverse ANN (YSA) models	Twitter	Textual	Turkish
[37]	MNB, LR, and SGD	Twitter	Textual	Hindi and Marathi

3. Research Methodology

The research methodology that was followed is illustrated in Figure 1, whereby the following steps were applied to achieve this study's goal. Firstly, the dataset was loaded into a local machine to perform the necessary pre-processing on the dataset, including essential natural language processing (NLP) steps, such as text cleaning, stemming, tokenizing, and lemmatizing. Then, the problematic comment pattern was analyzed using linguistic techniques. Next, multiple deep learning algorithms were applied after data partitioning to allow them to be tested and evaluated using proper performance evaluation metrics.

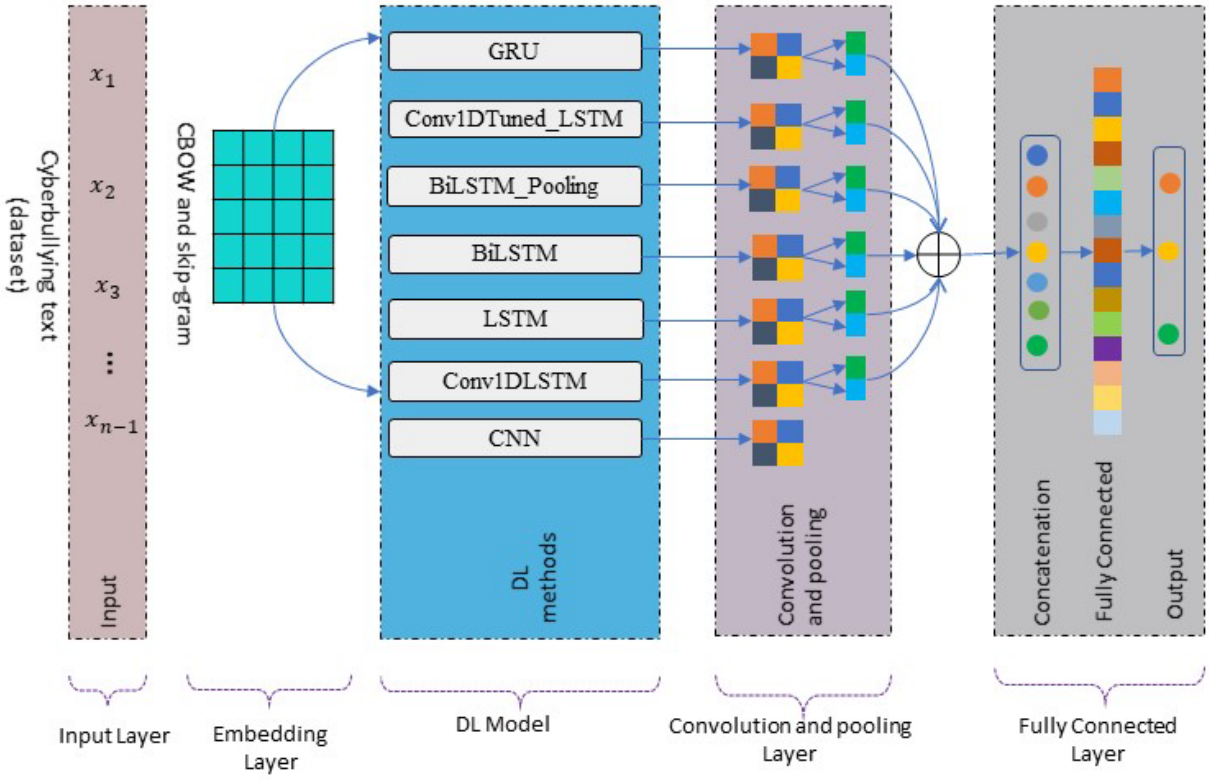


Figure 1. The proposed architecture of DL models used in cyberbullying detection.

3.1. Dataset Discription

This study utilized a global benchmark dataset of 37,373 tweets to detect cyberbullying by identifying offensive and non-offensive tweets [18]. The dataset had 37,373 columns and four rows (ID, label, tweet, and tag). It was numerically labeled, with tweets assigned a label of 1 for offensive tweets and 0 for neutral tweets that did not belong to the offensive category. The first characteristic evaluated was the timestamp of the comments, although this information was often missing or incomplete, making it challenging to obtain precise timestamps. The dataset primarily consisted of English-language tweets, and pre-processing methods and data cleaning techniques were applied, as explained in the subsequent subsections. To identify offensive tweets within the Twitter dataset, a set of unique bullying keywords was manually selected (e.g., "stupid", "idiot", "hoe", "nigga", "moron", "loser", "fool", "dumb", "retard", "slut", "bitch", and "ugly") for insults and name-calling tweets. Figure 2 illustrates the distribution of tweet lengths, with a significant number of tweets containing fewer than 10 words. Among these tweets, there were 1972 tweets specifically consisting of 9 words, which was the highest count within the dataset. Figure 3 provides a count of tweets with higher numbers of words, showcasing that there were 1865 tweets comprising exactly 11 words. The longest tweet in the dataset contained 52 words, as depicted in Figure 3. Additionally, Figure 4 presents an illustration of the dataset's unique bullying words alongside the most frequently used words. These unique bullying words were manually selected based on their presence within the dataset.

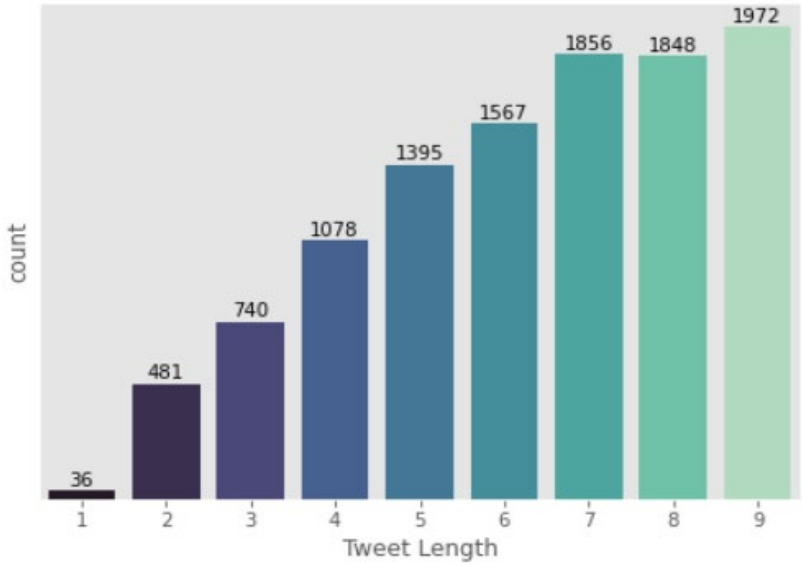


Figure 2. Count of tweets with less than 10 words.

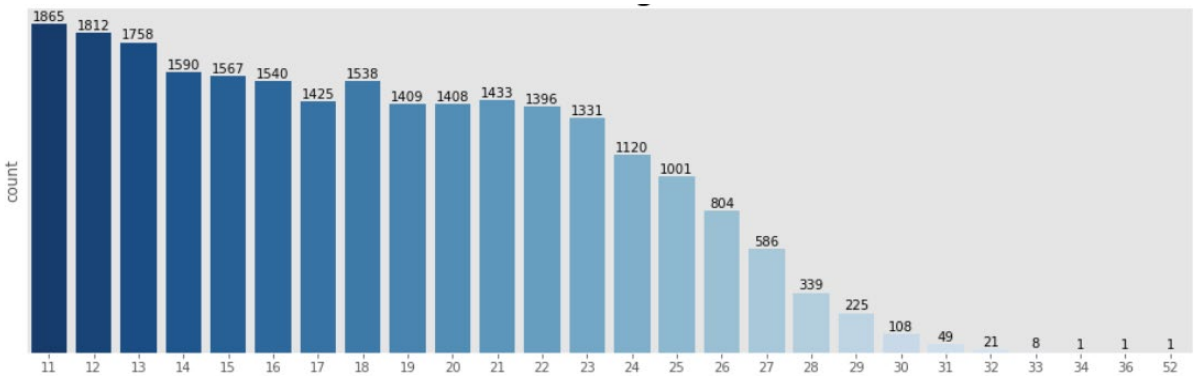


Figure 3. Counts of tweets containing a higher number of words.

Figure 4. Illustration of the unique bullying words and most used words in the datasets.

3.2. Embedding Layer

As mentioned earlier, this study used NLP in the pre-processing stage. In NLP, the word was represented as a one-hot vector, whereby all of the cells should be filled with 0, except for the cell that contained the word. Such representation was somehow impractical due to the sparsity and high dimensionality. Thus, in this study, we used the continuous vector space, whereby similar words were aggregated in a cluster. Such a vector space was more efficient.

Recently, using neural networks to obtain word representations has attracted researchers' attention as the learner vectors explicitly encode uneven patterns in the texts with several linguistic. Word-embedding representations can be learned using the word2vec with skip-gram [38] and continuous bag-of-words (CBOW) [39] models. Both models have the same objective, except that the skip-gram model relies on maximizing the prediction probability of the adjacent attributes based on the main word. Figure 5 shows how the CBOW uses word vector representations to anticipate the middle words in context.

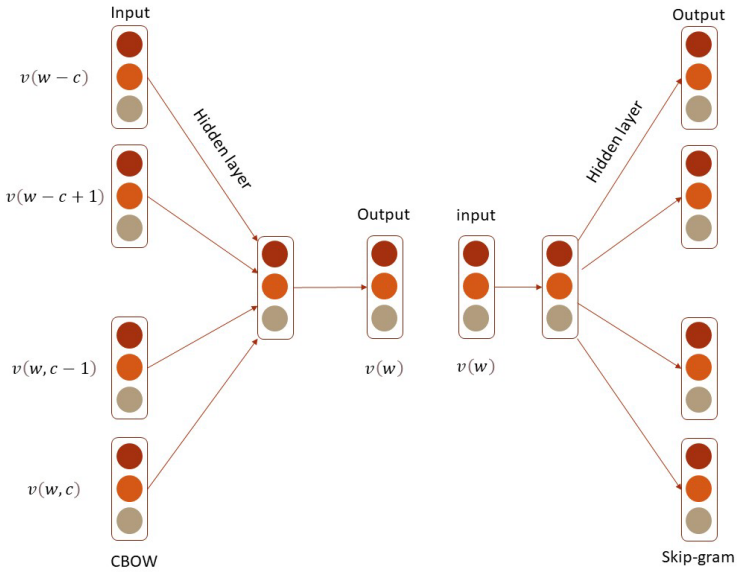


Figure 5. CBOW and skip-gram models' architectures learn dense vector representations of words.

Due to the similarities between the CBOW and skip-gram models, we delved into their derivation. As both models entail a substantial computational cost, training methods such as hierarchical SoftMax or negative sampling were employed to mitigate this issue. Hierarchical SoftMax entailed the representation of all words in the vocabulary as tree units at the output layer. This process was accomplished using a frequency-based Huffman tree, which was typically a binary tree [40]. In the CBOW model that used hierarchical SoftMax, the output was replaced by a Huffman tree, facilitating more efficient computation. In the CBOW model, the hidden layer performed the task of averaging the input word vectors. As a result, the output of the hidden layer could be represented as the average of the following vectors:

$$h = \frac{1}{C} \sum_{u \in (\text{context})(w)} v(u) \tag{1}$$

In the given expression, $v(u)$ represents the vector of the word u, context(w) represents the set of contextual information associated with the word w, and C represents the cardinality of the context set. Consequently, within each context, the conditional probability of the word w could be defined as follows:

$$p(w \mid \text{context}(w)) = \prod_{j=1}^{k(w)-1} \parallel h^T v'_{n_{w,j}} \parallel \tag{2}$$

where $n_{(w,j)}$ represent the jth inner point from root to word w in the Huffman tree, $v'_{\eta_{w,j}}$ and represents the vector of an inner point n, where $k(w) - 1$ represents the length of the Huffman tree for word w and $\mid \mid$ is a function defined as follows:

$$\parallel x \parallel = \sigma(x)^{d^w_{j+1}} \left[1 - \sigma(x)^{(1-d^w_{j+1})}\right] \tag{3}$$

where d^w_{j+1} is the jth bit of the Huffman code for word w. In this study, we implemented it by maximizing the conditional probability of the equation during the model's training in Figure 5 for the context (or target) of the word w. The log of the conditional probability provides the loss function as follows:

$$l = \log p(w \mid \text{context(w)}) \tag{4}$$

The derivative = obtained l as a loss function of the vector of the inner point $\eta_{(w,j)}$ as follows:

$$\frac{\partial l}{\partial v'_{n_{w,j}}} = \frac{\partial l}{\partial h^T v'_{n_{w,j}}} \frac{\partial h^T v'_{n_{w,j}}}{\partial v'_{n_{w,j}}} = h^T \parallel 1 - h^T v'_{n_{w,j}} \parallel \tag{5}$$

where $j = 1, 2, \cdots, l(w) - 1$. We define the derivative of l of vector of information contextual of words u as follows:

$$\frac{\partial l}{\partial v(u)} = \sum_{j=1}^{l(w)-1} \parallel 1 - h^T v'_{n_{w,j}} \parallel v'_{n_{w,j}} \tag{6}$$

They are mirror images of one another. The CBOW model's learning goal is to train a word vector that predicts the cantered word inside of a certain context; the skip-gram is used to learn a word vector that predicts surrounding words based on the cantered word.

3.3. Deep Neural Networks (DNN) Models

The deep neural network models in this work comprise LSTM (long short-term memory), Conv1DLSTM (convolutional 1D long short-term memory), CNN (convolutional neural network), and BiLSTM_Pooling (bidirectional long short-term memory with pooling), which were employed to detect cyberbullying via social media LSTM-, Conv1DLSTM-, CNN-, and BiLSTM_Pooling-based attention mechanism models using the same tweeter

dataset. The following sub-sections briefly describe the building model procedure of each DNN architecture used in cyberbullying detection.

3.3.1. Bidirectional Long Short-Term Memory and Gated Recurrent Unit Layer

Recurrent neural networks (RNN) are believed to be the most effective designs used in sequence modeling. As the internal structure of the RNN grants for time-based feeding, their method of operation is built to sequentially handle the inputs. Figure 6 depicts a graph illustration of an RNN neuron with a cycle surrounding it at the point where the graph expanded, and the temporal sequence can be seen. Therefore, the input from prior steps fed into the current step.

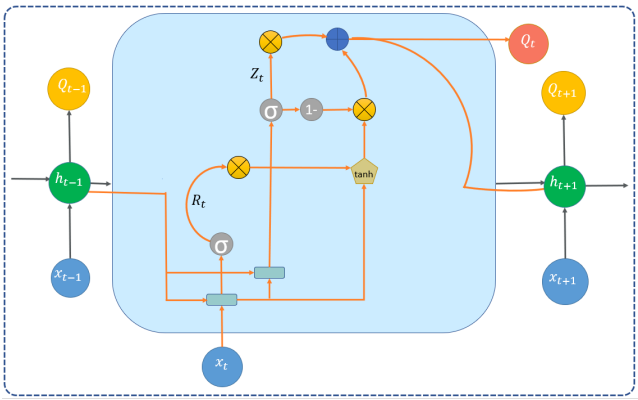


Figure 6. Architecture of standard gated recurrent unit model used in cyberbullying detection.

The RNN neuron can utilize prior information at each time step and has two enhancements, namely the long-short term memory (LSTM) and the gated recurrent unit (GRU). The capacity of RNN to parse sequences of indefinite length is an important attribute. This state is critical to processing language sentences because in a natural language phrase, all words are required to understand the meaning. A sentence could be long, and as the method processes the words one by one, authors know that the first words are essential for the whole meaning, and they must be learned when the final words in the sequence are processed.

The weights learnt via separate neurons prevent typical DNNs from determining exact representations of the attributes related to cyberbullying tweets due to the complicated language structure. To tackle the aforementioned problem, the RNN used a repetition loop over timesteps to circumvent the restriction. A sequence vector $\{x_1,\ldots, x_n\}$ was handled by employing a recurrence of the form $r_t = f_\alpha(r_{t-1}, x_t)$, where f was the activation function, α was a set of parameters employed at each time step t, and x_t was the input at timestep t [41,42].

To construct the potential RNN model for this work, three kinds of re-current neurons, such as the simple RNN unit, the GRU unit (shown in Figure 6), and the LSTM unit, were employed. The parameters defining the connections between the input and hidden layers, as well as the horizontal relationship between activations and the hidden layer to the output layer, were allocated for each timestep in a basic recurrent neuron. The forward pass of a primary recurrent neuron was represented as follows:

$$a^t = g\left(W_a\left[a^{<t-1>}, X^t\right] + b_a\right) \tag{7}$$

$$y^t = f\left(W_y.a^t + b_y\right) \tag{8}$$

In the given context, the activation function is represented by the variable g, and "t" represents the current timestep. The input at timestep t is represented by X^t, b_a represents the bias term, and W_a represents the cumulative weights with respect to timestep t for the

activation output, which is denoted as a^t. If necessary, this activation a^t can be used to estimate the value y_t at time t.

In addition, DNNs with simple RNN neurons indicated beneficial results in numerous applications. Thus, these neurons remained prone to vanishing gradients and struggled to learn long-term dependencies [41]. To solve the gradient disappearance issue and enable the learning of long-term dependencies, the research community has suggested several altered recurrent neuron architectures to resolve the simple RNN neuron limitation, such as the GRU method suggested by [43] and the LSTM method introduced by [44]. The work in [45] suggested a GRU that could show improved implementation for long-term relationship learning using input data. Additionally, $H^t = a^t$ factor memory was employed in the GRU unit at each stage t that offered a revised list of the entire samples handled by the GRU unit. Therefore, the GRU unit considered overwriting the H^t at each timestep t. However, the regulation of factor memory overwriting was employed using the update gate Γ_u when the GRU unit superimposed the H^t value at each step "t" with the candidate value $\overline{H}^t$. GRU neuron functionality was represented using the following equations:

$$\overline{H}^t = tanh\left(W_c\left[\Gamma_r * H^t,\ X^t\right] + b_c\right)$$

$$\Gamma_r = \sigma\left(W_r\left[H^{t-1},\ X^t\right] + b_r\right)$$

$$\Gamma_u = \sigma\left(W_u\left[H^{t-1},\ X^t\right] + b_u\right)$$

$$H^t = \Gamma_u * \overline{H}^t + (1 - \Gamma_u) * H^{t-1}$$

$$a^t = H^t$$

where W_r, W_c, and W_u correspond to the respective weights, and b_r, b_c, and b_u correspond to the subsequent bias terms for input X_t at timestep t. σ is the logistic regression function, and the activation value at timestep t is symbolized by a^t. Apart from for the usage of GRU neurons, the employed RNN model built using GRU was similar to those of the simple RNNs models. Figure 7 shows the LSTM-based model architecture used in cyberbullying detection.

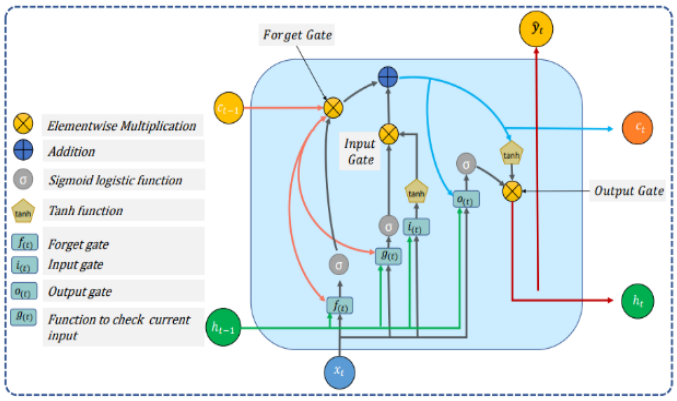


Figure 7. Architecture of standard long short-term memory model used in cyberbullying detection.

As mentioned earlier, the authors in [44] suggested the LSTM neuron with several enhancements to the design of the simple RNN unit that delivers a strong generalization of GRU. The following examples are some of the noticeable variances in LSTM and GRU cells:

1. There was no importance gate Γ_r that was utilized in standard LSTM units for $\overline{H}^t$ computation.
2. LSTM units employed two distinctive gates as substitutes for an update gate Γ_u These two gates were output gate Γ_o and update gate Γ_u . The output gate determined the next hidden state value of the H^t memory cell to process the LSTM unit activation

outputs of additional concealed network components. The forget gate dealt with the extent of overwriting using H^{t-1} to achieve H^t, such as how much memory cell information could be ignored for memory cells to work properly.

3. As the memory cell content H^t may not have been comparable to the activation at time t, the LSTM-based network differs from the GRU-based network.

Furthermore, the RNN approach-based LSTM was built using the same architecture as the GRU and basic RNN models. The sole distinction was that the LSTM units were located in recurrent layers [46].

3.3.2. Convolutional Neural Network

CNNs are specifically designed to handle learning tasks that involve high-dimensional input data with complex spatial structures. They have been successfully applied to various types of data, including images [47,48], videos [49], protein sequences [50,51], etc. CNNs aim to minimize the number of trainable parameters while effectively learning hierarchical filters that can accurately classify large volumes of incoming data. This goal is achieved by enabling sparse interactions between the input data and the trainable parameters through a technique called parameter sharing. This technique allows the network to develop a transformation process through which the learned filters are applied to different parts of the input, facilitating the extraction of meaningful features. Through this process, CNNs learn equivariant representations, also known as feature maps, of the complex and spatially structured input data. These feature maps capture important patterns and structures present in the data, enabling the network to effectively extract and utilize relevant information required to complete classification tasks [52]. The hierarchical nature of the learned filters allowed the network to progressively capture more abstract and high-level features, leading to improved classification accuracy.

CNNs comprise various convolution layers. These layers are utilized in NLP applications to better understand the local distinctive feature. The study conducted convolution operations using the feature vector from the attention layer by adding a linear filter. For a provided post on social media in sentence X with distinct x words, firstly, the embedding vector of size e was generated, and a filter F of size $e \times h$ was repeatedly used as a sub-matrix to represent the input data. The results of this generated a feature map $M = [m_1, m_2, \cdots, m_{x-h}]$ as follows:

$$m_i = F \times X_{i:i+h-1} \tag{9}$$

where $i = 0, 1, \cdots, x - h$, and $X_{i:j}$ is a sub-matrix of X from row I to j, as popular method is to input feature maps into a pooling or sub-sample layer to increase their dimension. The max-pooling is a regular pooling layer that choses the highly significant feature b from the map as follows:

$$m_i = F \times X_{i:i+h-1} \tag{10}$$

The output of the pooling layer is combined or concatenated to create a pooled feature vector, which serves as the input for the fully connected layer (FCL) (Figure 8).

3.4. Attention Mechanism

Attention models were used to assign various weights to words differently contributing to the bullying tweets to be detected via the proposed DL models. A customary practice of assigning various weights to diverse offensive contents in an unlabeled tweet was to use a weighted combination (Figure 9) of all hidden states X_{AW} as follows:

$$\begin{aligned} \alpha_t &= \frac{e^{(v^T \times \tilde{h})}}{e^{(v \times \tilde{h})}} \\ X_{A_w} &= \sum_t (\alpha_t h_t) \end{aligned} \tag{11}$$

where $\widetilde{h}_t$ and h are defined, as revealed in Equation (11), the element is a trainable parameter [53].

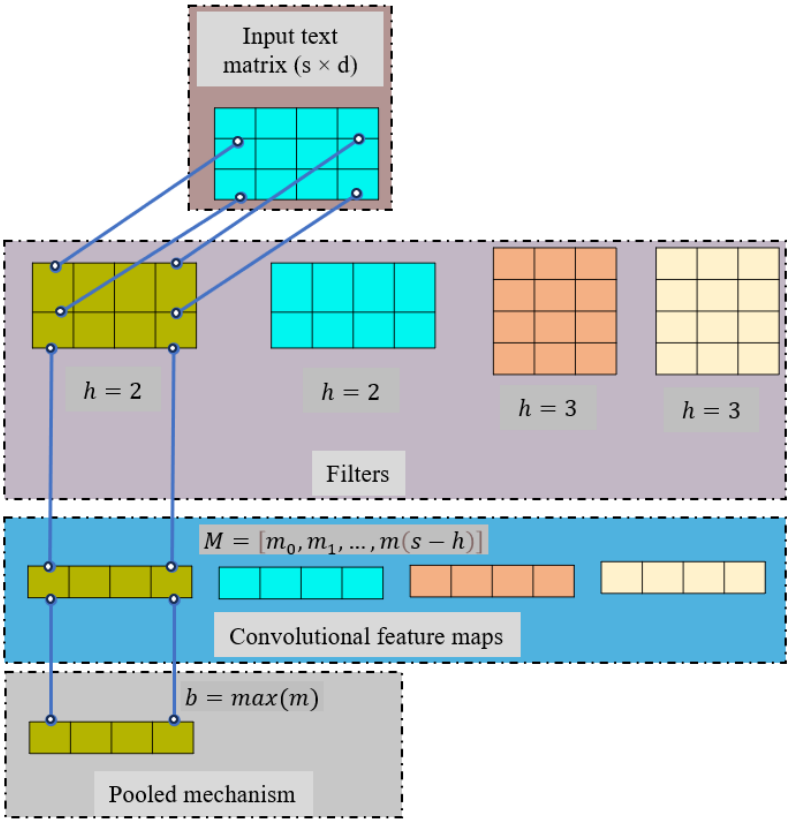


Figure 8. Convolutional layer used to extract local features. This layer utilizes filters to perform convolution operations on the input data. The filter slides over the input data, computing a dot product between the filter weights and the corresponding input values at each spatial position. The pooled mechanism used to perform down-sampling aids in capturing robust features, as it retains the most prominent features while discarding some spatial information.

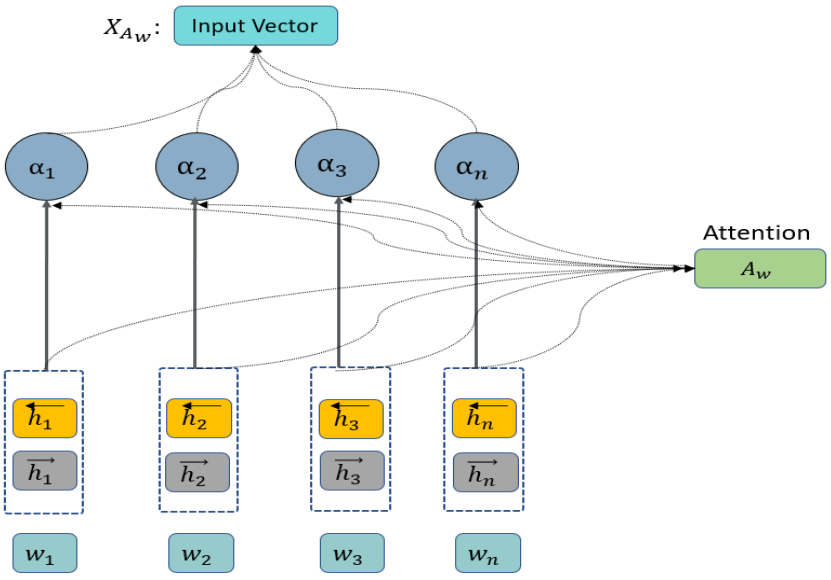


Figure 9. The attention mechanism structure employed in the proposed cyberbullying detection model. The attention mechanism allows the model to dynamically allocate attention to relevant words or segments, enabling a more fine-grained analysis and capturing crucial features related to cyberbullying behavior.

3.5. Fully Connected Layer (FC)

In the FC layer, the feature vector representation, which obtained by concatenating the pooling layers' weight vectors, was mapped to the input vector using a matrix of weights. This mapping process enabled the learning of the bullying patterns required to construct the cyberbullying model. The FC layer comprised multiple dense layers, non-linear activation functions, SoftMax, and a prediction function used to accurately classify instances as either bullying or non-bullying as follows:

$$\mathbb{H}_t = SoftMax(w_t h_{t-1} + b_t) \tag{12}$$

where w_t and b_t are parameters learned in training, H_t is obtained from the pooled concatenated feature vector and h_{t-1} is the feature map received from the CNN layers. The output layer performs the correct classification using the *SoftMax* function, as shown in Figure 1. The cross-entropy loss was minimized to learn the model parameters, which was the training objective when using the Adam optimization algorithm [54]. It was provided by

$$CrossEntropy(p, q) = -\sum_{p}(x) log\ (q(x)) \tag{13}$$

In this scenario, p represents the true distribution of a one-hot vector that represents characters in the messages posted onto social media. On the other hand, q represents the output of the SoftMax function. This calculation involves computing the negative logarithm probability of the true bullying tweet.

4. Results and Discussion

This section presents the experimental findings, along with commentary on their importance. The Keras library in the TensorFlow machine learning framework was used to implement the proposed cyberbullying model and the other baseline models. The objective was to minimize the complexity of the model by removing unnecessary elements, such as the number of hidden nodes, and, in the dense layer, by finding optimal hyperparameters with the hyperband method in Keras tuner [55]. An input matrix of 35,873 words was constructed to divide the raw input data into tokens, which helped the cyberbullying model to understand the context and interpret the vital information in the text by analyzing the word sequence using tokenization in the Keras library. A pre-processing step was applied before tokenization by removing irregular text formats, text content loss, and incomplete and duplicate documents.

Words in the text that added no meaning to the sentence were removed; they would not affect text processing for the defined purpose and were removed from the vocabulary to reduce noise and the dimension of the feature set. The word2vec with CBOW concatenated formed the weights in the embedding layer. The 75-dimension of word2vec was trained using word vectors of 147 words and phrases of 35873 words derived from a tweeter cyberbullying dataset. In the proposed DL methods, each neuron spanned between 32 and 256 memory units, having a step size of 32, but the Conv1DLSTM, CNN, and LSTM provided an optimum value using the Adam optimization in the Keras tuner. The library was used to establish the optimum value while restricting the number of iterations to a low value. The maximum number of trials was set between five and 10, corresponding to two or three per execution trial, with a dropout rate of 0.4. In the convolutional layer, 480 filters with kernel sizes of four and six provided the optimum values, as shown in Figure 1.

The size of the fully connected layer was 416, which initialized word embeddings using Glorot uniform initialization [56] for the model to converge over a SoftMax classifier. The entire model was trained to cover 30 epochs using the Adam stochastic optimizer. A mini-batch size of 64 yielded better performance for tweet datasets when the class label had over 10 or 20 words per tweet; however, the learning rate of 0.0001 and the dropout of 0.55 varied from 0.3 to 0.7 and were constant for all training datasets, irrespective of the class label. The SoftMax function was employed in the output layer without the hashing trick.

Finally, the training process was accelerated using the dataset with a class label of less than 50 by setting the learning rate, embedding size, mini-batch size, and number of epochs to 0.001, 50, 32, and 20, respectively. A 10-fold cross-validation and early stopping through monitoring validation loss in max mode with the patience of five trials were applied to the training process to prevent overfitting problems.

4.1. Evaluation Metrics

In this study, the effectiveness of a proposed model was examined by employing various assessment metrics to evaluate its ability to distinguish between cyberbullying and non-cyberbullying content. Several deep learning-based attention mechanisms, such as LSTM, Conv1DLSTM, CNN, BiLSTM Pooling, and GRU, were developed in this study. Evaluation criteria play a crucial role in understanding the performances of competing models in the scientific community. The following evaluation criteria are commonly used to assess the performance of cyberbullying classifiers for social media networks: Accuracy in Equation (14) is the proportion of actual identified instances to all cases, and it is frequently used to assess cyberbullying prediction models. Precision in Equation (15) determines the percentage of relevant tweets among tweets that are both true positives and false positives for a given group. Recall in Equation (16) measures the proportion of relevant tweets retrieved from all relevant tweets. The F-measure (17) provides a means of combining recall and precision into a single metric that takes into account both aspects.

$$\text{Accuracy} = \frac{(\text{tp} + \text{tn})}{(\text{tp} + \text{fp} + \text{tn} + \text{fn})} \tag{14}$$

$$\text{Precision} = \frac{\text{tp}}{(\text{tp} + \text{fp})} \tag{15}$$

$$\text{Recall} = \frac{\text{tp}}{(\text{tp} + \text{fn})}, \tag{16}$$

$$\text{F} - \text{measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{recision} + \text{recall}} \tag{17}$$

where fp stands for false positive, *fn* for false negative, *tp* for true positive, and tn for true negative. Therefore, (i) True positive is when a sample contains offensive text and the model correctly predicts its presence as offensive, it is considered a true positive. The model's prediction aligns with the actual presence of offensive words. (ii) False positive (FP): If a sample does not contain offensive text, but the model incorrectly predicts it as offensive, it is considered a false positive. The model mistakenly identifies non-offensive content as offensive. (iii) False negative (FN): When a sample contains offensive words, but the model fails to detect them and predicts the absence of offensive content, it is a false negative. The model misses the presence of offensive words. (iv) True negative (TN): If a sample does not contain offensive words, and the model accurately predicts the absence of offensive content, it is a true negative. The model correctly identifies non-offensive content as such. Figure 10 shows confusion matrices of Conv1DLSTM-based attention predictors models.

4.2. Performance Result of DL Models

The proposed work utilizes six0 DL detector models without theword2vec-based CBOW feature extractor and without the word2vec-based CBOW feature extractor. Firstly, we experimented with the six proposed DL detector models without implementing the word2vec-based CBOW feature extractor, and the DL-based models' performance was recorded and tabulated, as demonstrated in Table 2. The Conv1DLSTM model outperformed other predictors, having an accuracy of 0.8649, a precision of 0.8146, a recall of 0.8919, and an F1-score of 0.8515. Therefore, the GRU model obtained the lowest performance in detecting cyberbullying, having an accuracy of 0.7093, a precision of 0.7089, a

recall of 0.7561, and an F1-score of 0.7317. Figure 11 summarizes the accuracy, precision, recall, and F1-score of DL-based attention predictors without the feature extractor.

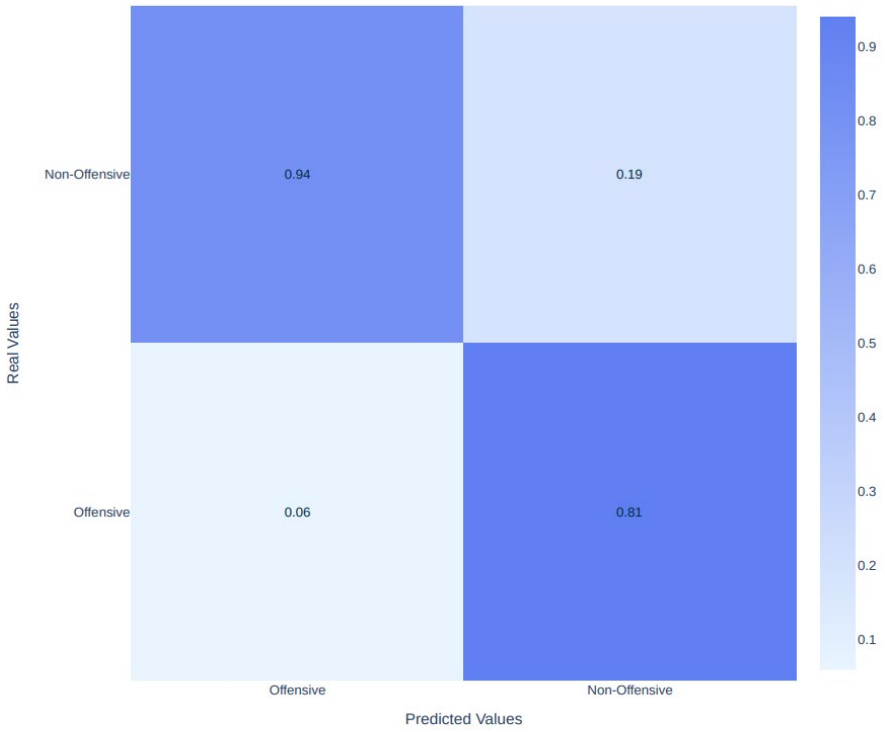


Figure 10. Confusion matrix of Conv1DLSTM-based attention predictor model.

Table 2. Comparative analysis of investigated DL models without feature extractor (word2vec-based CBOW).

No.	Algorithm	Accuracy (%)	Precision	Recall	F1 Score
1	LSTM	0.8011	0.8142	0.7281	0.7687
2	Conv1DLSTM	0.8649	0.8146	0.8919	0.8515
3	CNN	0.8496	0.8836	0.7908	0.8346
4	BiLSTM	0.7795	0.8373	0.8130	0.8250
5	BiLSTM_Pooling	0.7982	0.8167	0.7862	0.8012
6	GRU	0.7093	0.7089	0.7561	0.7317

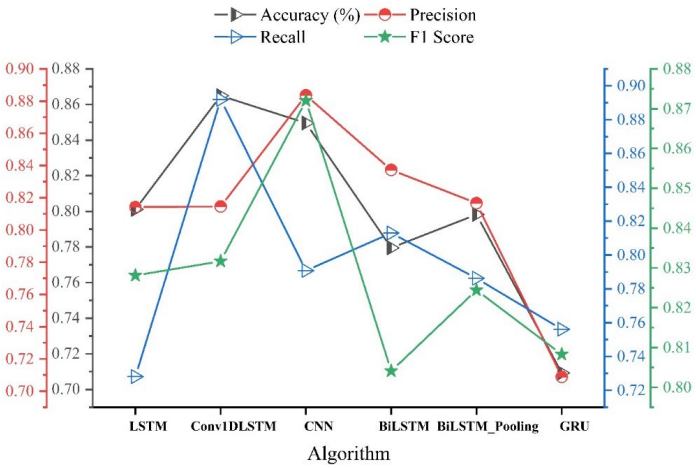


Figure 11. Accuracy, precision, recall, and F1-score of each DL-based attention predictor without feature extractor.

Additionally, the proposed work utilizes six DL detector models with word2vec-based CBOW concatenated formed weights in the embedding layer for feature extraction. These methods were set empirically to attain higher accuracy. For example, Conv1DLSTM had the best accuracy in our dataset, where the proposed model obtained a classification accuracy of 94.49% and an F1 score of 0.9518. Meanwhile, the CNN-based attention mechanism obtained the same accuracy in this classification problem (93.96%); on the other hand, the CNN-based attention mechanism achieved slightly lower performance in terms of F1-score (0.9025) than the LSTM-based attention mechanism, which had a value of (0.9218). Accuracies of 86.92%, 88.92%, and 89.90% were obtained for BiLSTM, BiLSTM_Pooling, and GRU-based attention, respectively. This result means that the Conv1DLSTM attention-based model performs better than other classifiers, as shown in Table 3. Moreover, we observed that by employing the word2vec-based CBOW feature extractor, the DL-based models were significantly improved in terms of distinguishing between the cyberbullying tweets and the non-cyberbullying tweets. Figure 12 summarizes the accuracy, precision, recall, and F1-score of each DL-based attention predictor with feature extractor.

Table 3. Comparative analysis of investigated DL-based attention models with feature extractor (word2vec-based CBOW).

No.	Algorithm	Accuracy (%)	Precision	Recall	F1 Score
1	LSTM	0.9396	0.9185	0.9251	0.9218
2	**Conv1DLSTM**	**0.9449**	**0.9836**	**0.9219**	0.9518
3	CNN	0.9396	0.9146	0.8908	0.9025
4	BiLSTM	0.8692	0.9372	0.9130	0.9249
5	BiLSTM_Pooling	0.8892	0.9166	0.8862	0.9011
6	GRU	0.8990	0.9083	0.8956	0.9019

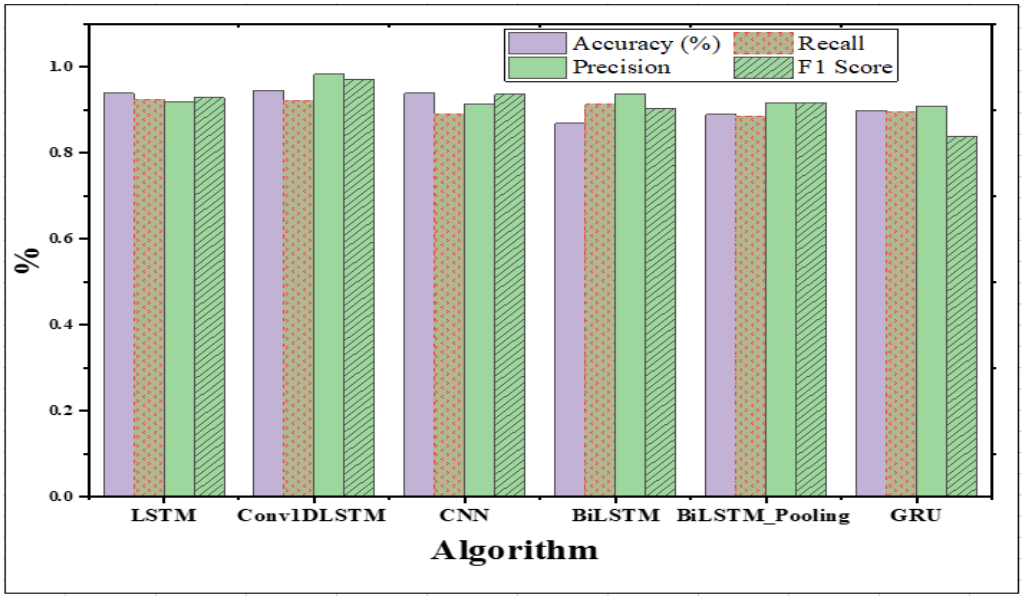
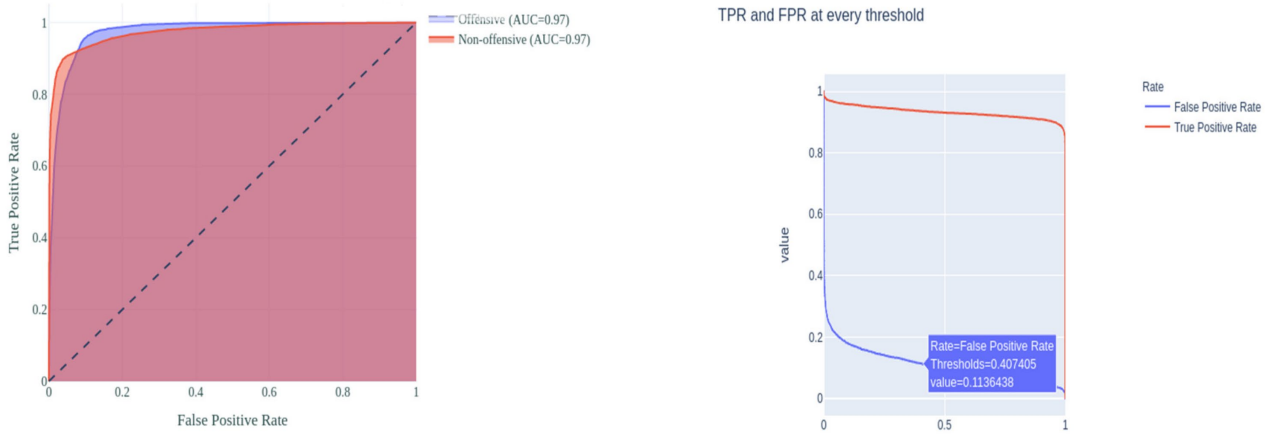


Figure 12. Accuracy, precision, recall and F1-score of each DL-based attention predictor0 with feature extractor.

The proposed work utilizes additional important evaluation methods,0 such as the receiver ROC and precision–recall curves. The ROC curve provides a valuable visual representation of the balance between the true positive and false positive rates. To assess the generalization capabilities of each deep learning (DL) model, separate test data collections were used during the training phase. This approach guarantees unbiased outcomes and allows the testing of the detectors' ability to generalize.

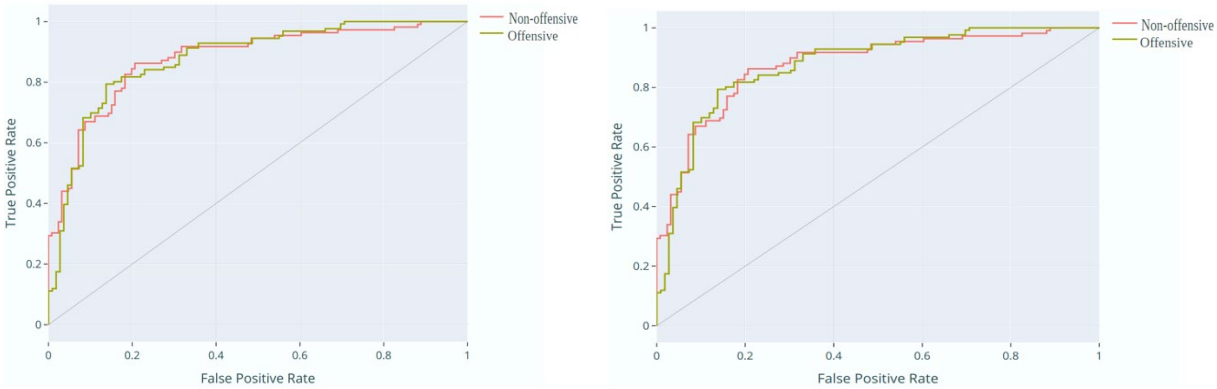
The Area under the curve (AUC) is a measure used to assess how effectively a classifier can distinguish between different classes, and it is often employed as a summary of the ROC curve. A higher AUC indicates a better performance of the model in distinguishing between positive and negative samples [57]. In essence, the ROC curve is a graphical representation of the trade-off between the recall of false positive rates and true positive rates [57]. It provides valuable insights into evaluating the costs and benefits of the classifier. The false positive rate is computed by dividing the number of false positives by the total number of negative samples. This rate is considered a cost, since any subsequent action taken based on a false positive outcome would be wasted, as it is a misprediction. On the other hand, the true positive rate, which represents the proportion of positive cases accurately predicted, can be seen as a benefit, as it indicates successful predictions of the analyzed issue. Figure 13a shows AUC curve for the classical Conv1DLSTM predictor, and Figure 13b shows the TPR and FPR for different threshold values for the classification Conv1DLSTM model used to enable a better understanding of the classification performances of cyberbullying prediction models.



(**a**) (**b**)

Figure 13. AUC curve of Conv1DLSTM-based attention predictor. (**a**) AUC curve for Conv1DLSTM-based model; (**b**) TPR and FPR for different threshold values.

Figure 14 shows AUC curves for the second and third models depicted in our study. Figure 14a shows the LSTM-based model, while Figure 14b shows the AUC curve-based CNN predictor performance.



(**a**) (**b**)

Figure 14. AUC curve of the two best models, namely Conv1DLSTM- and CNN-based attention mechanisms. (**a**) AUC curve for LSTM-based attention predictor; (**b**) AUC curve for CNN-based attention predictor.

The proposed attention-based Conv1DLSTM classifier presents several significant advantages in terms of cyberbullying detection. Firstly, the integration of attention mechanisms allows the model to focus on the most relevant parts of the input text, facilitating a deeper understanding of the context in which offensive language or cyberbullying occurs. This contextual understanding is crucial to distinguishing between harmless content and harmful behavior in social media posts. Additionally, the Conv1DLSTM architecture proves advantages related to handling the variable length nature of social media text, enabling effective processing of tweets with differing word counts. The model's feature extraction capability further enhances its cyberbullying detection accuracy by capturing spatial patterns and temporal dependencies within the text.

Moreover, the Conv1DLSTM's ability to learn position-invariant features using the input text ensures the identification of offensive words or phrases, regardless of their specific location within the tweet. This feature is particularly relevant in the context of social media, where abusive content can be interspersed with non-offensive language. The incorporation of LSTM units within the Conv1DLSTM allows the model to capture long-term dependencies in the text, enabling a holistic consideration of the entire tweet's context when making predictions of cyberbullying. The attention mechanism's capacity to highlight offensive words or phrases within the text also aids in understanding the model's decision-making process and enables the interpretability of the results. Furthermore, the superior performance of the attention-based Conv1DLSTM classifier to those of other deep learning methods, as demonstrated using benchmark experimental datasets and evaluation measures, underscores its effectiveness in accurately identifying instances of cyberbullying. Lastly, the attention-based Conv1DLSTM classifier showcases its potential to significantly advance cyberbullying detection methodologies and contribute to the creation of a safer online environment for social media users.

5. Conclusions and Future Work

Complex and multifaceted problems, such as cyberbullying, are challenging to track down using standard methods. This study investigated the use of deep learning and attention mechanisms to identify the best model to predict text-based cyberbullying tweets on social media. Conv1DLSTM achieved the best accuracy in our dataset, where the classification accuracy and F1 score were 94.49%. Meanwhile, LSTM- and CNN-based attention mechanisms obtained the same accuracy in this classification problem (93.96%), whereas the LSTM-based attention mechanism achieved a better performance in terms of F1-score measure (0.9218) than CNN, which scored a value of (0.9025). The attention-based Conv1DLSTM consistently produced more precise predictions. Therefore, it is possible to conclude that the proposed approach detects most offensive cyberbullying tweets. The suggested model includes the following limitations: (i) this work did not consider image-based cyberbullying detection, which means a post solely containing images was not part of this research, and (ii) the scope of this research was confined to text-based cyberbullying detection. As a result, the future scope of this research is up for debate, as it involves numerous subproblems. The proposed system achieved an accuracy rate of 94.49%, which could be enhanced by increasing the training sample size. Additionally, an ensemble or stacking model will be explored in future research to improve prediction accuracy.

Author Contributions: Conceptualization, A.M. and S.M.F.; methodology, A.M.; software, A.M.; validation, A.M. and S.M.F.; formal analysis, A.O.B.; investigation, A.M. and A.A.; resources, A.M.; data curation, A.M.; writing—original draft preparation, A.M. and A.O.B.; writing—review and editing, S.M.F. and A.A.; visualization, A.M.; supervision, S.M.F.; project administration, A.M. and S.M.F.; funding acquisition, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia under project no. (IFKSUOR3–057-1). The authors would also like to thank Prince Sultan University, Riyadh, Saudi Arabia, for their support.

Data Availability Statement: The data is available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Smart Insights. Global Social Media Research Summary. West Yorkshire, UK, 2018. Available online: https://www.smartinsights.com (accessed on 20 October 2022).
2. Bozyiğit, A.; Utku, S.; Nasibov, E. Cyberbullying detection: Utilizing social media features. *Expert Syst. Appl.* **2021**, *179*, 115001. [CrossRef]
3. Iwendi, C.; Srivastava, G.; Khan, S.; Maddikunta, P.K.R. Cyberbullying detection solutions based on deep learning architectures. *Multimed. Syst.* **2020**, *29*, 1839–1852. [CrossRef]
4. Ali, H.; Farman, H.; Yar, H.; Khan, Z.; Habib, S.; Ammar, A. Deep learning-based election results prediction using Twitter activity. *Soft Comput.* **2022**, *26*, 7535–7543. [CrossRef]
5. Adewole, K.S.; Balogun, A.O.; Raheem, M.O.; Jimoh, M.K.; Jimoh, R.G.; Mabayoje, M.A.; Usman-Hamza, F.E.; Akintola, A.G.; Asaju-Gbolagade, A.W. Hybrid feature selection framework for sentiment analysis on large corpora. *Jordanian J. Comput. Inf. Technol.* **2021**, *7*, 130–151. [CrossRef]
6. Thakur, N. Sentiment Analysis and Text Analysis of the Public Discourse on Twitter about COVID-19 and MPox. *Big Data Cogn. Comput.* **2023**, *7*, 116. [CrossRef]
7. Badawi, D. Intelligent Recommendations Based on COVID-19 Related Twitter Sentiment Analysis and Fake Tweet Detection in Apache Spark Environment. *IETE J. Res.* **2023**, 1–24. [CrossRef]
8. Mahbub, S.; Pardede, E.; Kayes, A. Detection of harassment type of cyberbullying: A dictionary of approach words and its impact. *Secur. Commun. Netw.* **2021**, *2021*, 5594175. [CrossRef]
9. Kumar, A.; Sachdeva, N. Cyberbullying detection on social multimedia using soft computing techniques: A meta-analysis. *Multimed. Tools Appl.* **2019**, *78*, 23973–24010. [CrossRef]
10. Hilal, A.M.; Hashim, A.H.A.; Mohamed, H.G.; Alharbi, L.A.; Nour, M.K.; Mohamed, A.; Almasoud, A.S.; Motwakel, A. Spotted Hyena Optimizer with Deep Learning Driven Cybersecurity for Social Networks. *Comput. Syst. Sci. Eng.* **2023**, *45*, 2033–2047. [CrossRef]
11. Slonje, R.; Smith, P.K. Cyberbullying: Another main type of bullying? *Scand. J. Psychol.* **2008**, *49*, 147–154. [CrossRef]
12. Xu, J.-M.; Jun, K.-S.; Zhu, X.; Bellmore, A. Learning from bullying traces in social media. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Montréal, QC, Canada, 3–8 June 2012; pp. 656–666.
13. Dadvar, M.; Eckert, K. Cyberbullying detection in social networks using deep learning based models. In Proceedings of the Big Data Analytics and Knowledge Discovery: 22nd International Conference, DaWaK 2020, Bratislava, Slovakia, 14–17 September 2020; Proceedings 22. Springer: Berlin/Heidelberg, Germany, 2020; pp. 245–255.
14. Arif, M. A systematic review of machine learning algorithms in cyberbullying detection: Future directions and challenges. *J. Inf. Secur. Cybercrimes Res.* **2021**, *4*, 1–26. [CrossRef]
15. Ali, A.; Syed, A.M. Cyberbullying detection using machine learning. *Pak. J. Eng. Technol.* **2020**, *3*, 45–50.
16. Ahmed, M.T.; Rahman, M.; Nur, S.; Islam, A.; Das, D. Deployment of machine learning and deep learning algorithms in detecting cyberbullying in bangla and romanized bangla text: A comparative study. In Proceedings of the 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 19–20 February 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–10.
17. Ghasem, Z.; Frommholz, I.; Maple, C. Machine learning solutions for controlling cyberbullying and cyberstalking. *J. Inf. Secur. Res.* **2015**, *6*, 55–64.
18. Muneer, A.; Fati, S.M. A comparative analysis of machine learning techniques for cyberbullying detection on twitter. *Future Internet* **2020**, *12*, 187. [CrossRef]
19. Akande, O.N.; Nnaemeka, E.S.; Abikoye, O.C.; Akande, H.B.; Balogun, A.; Ayoola, J. TWEERIFY: A Web-Based Sentiment Analysis System Using Rule and Deep Learning Techniques. In *Proceedings of the International Conference on Computational Intelligence and Data Engineering: ICCIDE 2021*; Springer Nature: Singapore, 2022; pp. 75–87.
20. Alam, K.S.; Bhowmik, S.; Prosun, P.R.K. Cyberbullying detection: An ensemble based machine learning approach. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 710–715.
21. Ahuja, R.; Chug, A.; Kohli, S.; Gupta, S.; Ahuja, P. The impact of features extraction on the sentiment analysis. *Procedia Comput. Sci.* **2019**, *152*, 341–348. [CrossRef]
22. Chavan, V.S.; Shylaja, S. Machine learning approach for detection of cyber-aggressive comments by peers on social media network. In Proceedings of the 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, India, 10–13 August 2015; pp. 2354–2358.
23. Chen, H.; Mckeever, S.; Delany, S.J. Presenting a labelled dataset for real-time detection of abusive user posts. In Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, 23–26 August 2017; pp. 884–890.

24. Frommholz, I.; Al-Khateeb, H.M.; Potthast, M.; Ghasem, Z.; Shukla, M.; Short, E. On textual analysis and machine learning for cyberstalking detection. *Datenbank Spektrum* **2016**, *16*, 127–135. [CrossRef] [PubMed]
25. Van Bruwaene, D.; Huang, Q.; Inkpen, D. A multi-platform dataset for detecting cyberbullying in social media. *Lang. Resour. Eval.* **2020**, *54*, 851–874. [CrossRef]
26. Khan, A. Improved multi-lingual sentiment analysis and recognition using deep learning. *J. Inf. Sci.* **2023**, 01655515221137270. [CrossRef]
27. Zhao, R.; Mao, K. Cyberbullying detection based on semantic-enhanced marginalized denoising auto-encoder. *IEEE Trans. Affect. Comput.* **2016**, *8*, 328–339. [CrossRef]
28. Rosa, H.; Pereira, N.; Ribeiro, R.; Ferreira, P.C.; Carvalho, J.P.; Oliveira, S.; Coheur, L.; Paulino, P.; Simão, A.V.; Trancoso, I. Automatic cyberbullying detection: A systematic review. *Comput. Hum. Behav.* **2019**, *93*, 333–345. [CrossRef]
29. Sugandhi, R.; Pande, A.; Agrawal, A.; Bhagat, H. Automatic monitoring and prevention of cyberbullying. *Int. J. Comput. Appl.* **2016**, *8*, 17–19. [CrossRef]
30. Bin Abdur Rakib, T.; Soon, L.-K. Using the reddit corpus for cyberbully detection. In Proceedings of the Intelligent Information and Database Systems: 10th Asian Conference, ACIIDS 2018, Dong Hoi, Vietnam, 19–21 March 2018; Proceedings, Part I 10. Springer: Berlin/Heidelberg, Germany, 2018; pp. 180–189.
31. Agrawal, S.; Awekar, A. Deep learning for detecting cyberbullying across multiple social media platforms. In Proceedings of the European Conference on Information Retrieval, Grenoble, France, 26–29 March 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 141–153.
32. Haidar, B.; Chamoun, M.; Serhrouchni, A. Multilingual cyberbullying detection system: Detecting cyberbullying in Arabic content. In Proceedings of the 2017 1st Cyber Security in Networking Conference (CSNet), Rio de Janeiro, Brazil, 18–20 October 2017; pp. 1–8.
33. Al-Ajlan, M.A.; Ykhlef, M. Optimized twitter cyberbullying detection based on deep learning. In Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, Saudi Arabia, 25–26 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–5.
34. Banerjee, V.; Telavane, J.; Gaikwad, P.; Vartak, P. Detection of cyberbullying using deep neural network. In Proceedings of the 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 15–16 March 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 604–607.
35. Wulczyn, E.; Thain, N.; Dixon, L. Ex machina: Personal attacks seen at scale. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 1391–1399.
36. Jeyasheeli, P.G.; Selva, J.J. An IOT design for smart lighting in green buildings based on environmental factors. In Proceedings of the 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 January 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–5.
37. Bozyiğit, A.; Utku, S.; Nasiboğlu, E. Cyberbullying detection by using artificial neural network models. In Proceedings of the 2019 4th International Conference on Computer Science and Engineering (UBMK), Yogyakarta, Indonesia, 12–13 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 520–524.
38. Pawar, R.; Raje, R.R. Multilingual cyberbullying detection system. In Proceedings of the 2019 IEEE International Conference on Electro Information Technology (EIT), Brookings, SD, USA, 20–22 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 40–44.
39. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
40. Wang, Q.; Xu, J.; Chen, H.; He, B. Two improved continuous bag-of-word models. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2851–2856.
41. White, L. *On the Surprising Capacity of Linear Combinations of Embeddings for Natural Language Processing*; The University of Western Australia: Perth, Australia, 2019.
42. Muneer, A.; Taib, S.M.; Naseer, S.; Ali, R.F.; Aziz, I.A. Data-driven deep learning-based attention mechanism for remaining useful life prediction: Case study application to turbofan engine analysis. *Electronics* **2021**, *10*, 2453. [CrossRef]
43. Naseer, S.; Fati, S.M.; Muneer, A.; Ali, R.F. iAceS-Deep: Sequence-based identification of acetyl serine sites in proteins using PseAAC and deep neural representations. *IEEE Access* **2022**, *10*, 12953–12965. [CrossRef]
44. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
45. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* **2014**, arXiv:1409.1259.
46. Graves, A. *Long Short-Term Memory. Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 37–45.
47. Muneer, A.; Fati, S.M. Efficient and automated herbs classification approach based on shape and texture features using deep learning. *IEEE Access* **2020**, *8*, 196747–196764. [CrossRef]
48. Ghandour, C.; El-Shafai, W.; El-Rabaie, S. Medical image enhancement algorithms using deep learning-based convolutional neural network. *J. Opt.* **2023**, 1–11. [CrossRef]

49. Durairajah, V.; Gobee, S.; Muneer, A. Automatic vision based classification system using DNN and SVM classifiers. In Proceedings of the 2018 3rd International Conference on Control, Robotics and Cybernetics (CRC), Penang, Malaysia, 26–28 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 6–14.
50. Naseer, S.; Ali, R.F.; Muneer, A.; Fati, S.M. IAmideV-deep: Valine amidation site prediction in proteins using deep learning and Proceedings pseudo amino acid compositions. *Symmetry* **2021**, *13*, 560. [CrossRef]
51. Naseer, S.; Ali, R.F.; Fati, S.M.; Muneer, A. iNitroY-Deep: Computational identification of Nitrotyrosine sites to supplement Carcinogenesis studies using Deep Learning. *IEEE Access* **2021**, *9*, 73624–73640. [CrossRef]
52. Muneer, A.; Fati, S.M.; Akbar, N.A.; Agustriawan, D.; Wahyudi, S.T. iVaccine-Deep: Prediction of COVID-19 mRNA vaccine degradation using deep learning. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 7419–7432. [CrossRef]
53. Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* **2019**, *337*, 325–338. [CrossRef]
54. Creswell, A.; Arulkumaran, K.; Bharath, A.A. On denoising autoencoders trained to minimise binary cross-entropy. *arXiv* **2017**, arXiv:1708.08487.
55. Sinha, A.; Gunwal, S.; Kumar, S. A Globally Convergent Gradient-based Bilevel Hyperparameter Optimization Method. *arXiv* **2022**, arXiv:2208.12118.
56. Gao, T.; Chai, Y. Improving stock closing price prediction using recurrent neural network and technical indicators. *Neural Comput.* **2018**, *30*, 2833–2854. [CrossRef]
57. Narkhede, S. *Understanding AUC-ROC Curve: Towards Data Science, 2018 ed.*; Medium: San Francisco, CA, USA, 2018.