Article

A Multichannel Deep Learning Framework for Cyberbullying Detection on Social Media

Munif Alotaibi [1] *, Bandar Alotaibi [2] * and Abdul Razaque [3] *

[1] Department of Computer Science, Shaqra University, Shaqra,11961 Saudi Arabia; munif@su.edu.sa
[2] Sensor Networks and Cellular Systems Research Center, University of Tabuk, Tabuk 71491, Saudi Arabia; b-alotaibi@ut.edu.sa
[3] Department of Computer Engineering & Cybersecurity, IITU, Almaty 050000, Kazakhstan; a.razaque@iitu.edu.kz
* Correspondence: munif@su.edu.sa, b-alotaibi@ut.edu.sa, a.razaque@iitu.edu.kz

Abstract: Online social networks (OSNs) play an integral role in facilitating social interaction; however, these social networks increase antisocial behavior, such as cyberbullying, hate speech, and trolling. Aggression or hate speech that takes place through short message service (SMS) or the Internet (e.g., in social media platforms) is known as cyberbullying. Therefore, automatic detection utilizing natural language processing (NLP) is a necessary first step that helps prevent cyberbullying. This research proposes an automatic cyberbullying method to detect aggressive behavior using a consolidated deep learning model. This technique utilizes multichannel deep learning based on three models, namely, the bidirectional gated recurrent unit (BiGRU), transformer block, and convolutional neural network (CNN), to classify Twitter comments into two categories: aggressive and not aggressive. Three well-known hate speech datasets were combined to evaluate the performance of the proposed method. The proposed method achieved promising results. The accuracy of the proposed method was approximately 88%.

Keywords: Online social networks (OSNs); Deep Learning; cyberbullying; Twitter

1. Introduction

Cyberbullying is a critical cybersecurity threat continuously targeting more Internet users–social media users in particular [1–3]. Debatably, hostile behavior by one person or group of people, known as bullying, that can be limited to particular scenes or specific times of the day (e.g., school hours) can instead take place anywhere and anytime by electronic means [4–7]. At the beginning of the 20th century, cyberbullying was not treated seriously when Internet usage (i.e., social media in particular) was still in its infancy. The idealistic suggestion to treat cyberbullying at that time was "disconnect" or "just turn off the screen". However, as online hate speech consequences reach predominant levels, these suggestions become inoperative. To avoid cybercrime, it is not enough to only follow the typical recommended cybersecurity standards and rules. [8]. In 2017, 41% of United States citizens personally encountered online harassment, whereas 66% observed online hate speech directed at others. In addition, it has been reported that approximately 50% of young people who use social media platforms are cyberbullied in different forms [9]. Famous social media sites such as Twitter are not invulnerable to this threat [10].

Cyberbullying detection has become an important NLP topic [11]. Like other NLP tasks, the aim of cyberbullying detection is to preprocess the text (e.g., a tweet) and extract meaningful information in a way that makes it possible for the machine learning algorithm to understand and classify each text. The traditional strategies for text classification utilize a technique to simplify the representation of text (e.g., BoW) followed by a machine learning classifier (e.g., SVM or LR) [12] as shown in Figure 1. Although traditional NLP techniques have become highly successful in detecting social media cyberbullying, there are still some challenges that need to be addressed: the short text allowed by social media platforms, imbalance between aggressive and nonaggressive comments, natural language ambiguity, and excessive use of slang [13].

Citation: Alotaibi, M.; Alotaibi, B.; Razaque, A. A Multichannel Deep Learning Framework for Cyberbullying Detection on Social Media. *Preprints* **2021**, *1*, 0. https://doi.org/

Academic Editor: Firstname Lastname

Received:
Accepted:
Published:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 October 2021 doi:10.20944/preprints202110.0070.v1

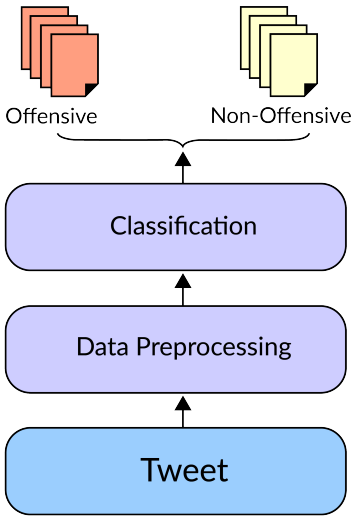


Figure 1. The general framework of deep learning architecture.

In the last decade, neural network-based models have achieved superior results on several NLP tasks compared to traditional machine learning techniques. These NLP techniques rely on dense vector representations provided by neural networks and deep learning in particular and the considerable success of word embeddings [14]. Unlike traditional machine learning algorithm-based approaches that depend heavily on handcrafted features that are considered incomplete and time-consuming, deep learning-based techniques utilize multilevel automatic feature representation to discriminate the input. Recently, neural network models such as MLP, RNN, and CNN-based models have achieved promising results in various NLP tasks. The authors of [15] proposed a distinctive method to classify text as a characterwise approach using a combination of a CNN model and RNN architecture. Another unique method was proposed by Tai et al. to classify text sentence by sentence to learn text semantics utilizing LSTM and then use CNN to extract local features from expressions [16]. Motivated by the considerable success presented by several deep learning architectures in the NLP research area where the task is to classify a lengthy text, we investigate the potential of classifying short text using the idea of the multichannel deep learning model based on three advanced deep learning architectures, namely, BiLSTM, transformer block, and CNN.

Table 1. Summery of the related methods.

Authors	Platform of OSNs	Number of Used Instances	Methods	Limitation
[17]	Youtube	50,000 comments	Naive bayes, Rule-based Jrip, J48, and SVM	Low accuracy and relatively small dataset
[18]	Twitter	460 tweets	NB classifier	Low accuracy (70%) and relatively small dataset
[19]	Twitter	-	Naive Bayes and Random Forest	Not reported
[20]	Twitter	10,007 tweets	random forests, NB, SVM, and KNN	Relatively small dataset
[21,22]	Twitter	5453 tweets	Naive Bayes, Random Forest and J48	Relatively small dataset and narrowed to specific community

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 October 2021 doi:10.20944/preprints202110.0070.v1

Table 1. *Cont.*

Authors	Platform of OSNs	Number of Used Instances	Methods	Limitation
[2,9]	Twitter	9484 texts	probabilistic, tree-based, ensemble, and ANN	Relatively small dataset
[24,25]	Twitter, Wikipedia, and Formspring	9484 texts	SVM, CNN, LSTM	Relatively small dataset
[26]	Twitter	6655 tweets	CNN with softmax	relatively small dataset
[27]	Twitter	8710 tweets	pretrained models along with BERT approaches	Relatively small dataset

This research paper is organized as follows. In the next Section 2, a brief survey of the literature is carried out to pinpoint the strengths and weaknesses of cyberbullying detection approaches. Section 3 presents the combined deep learning method. Section 4 present the dataset used to evaluate the proposed method. Section 5 introduces and discusses the results and compares them with closely related approaches. Section 6 concludes our research paper. Section 7 discusses future work.

1.1. Research Problem

Automatic detection of cyberbullying utilizing natural language processing (NLP) advanced mechine learning method on Online social networks such as Twitter platform is a necessary and an important. Moreover, the existing dataset that used to train the mechine learning model are limited. Thus, it is very critical to have large dataset that can cover many cyberbullying cases.

1.2. Research Contribution

Our research contribution can be summarized as follow:

- The multichannel technique integrates the features of three deep learning models: transformer block, BiGRU and CNN. This integration helps to contribute to the final prediction.
- Three well-known hate speech datasets [33–35], were combined to evaluate the performance of the proposed method. The proposed method achieved good accuracy results.

2. Related Work

In this section, cyberbullying detection approaches focusing on OSNs are reviewed. Dinakar et al. [17] divided cyberbullying occurrence into various themes, including race, sexuality, culture, and intelligence. Consequently, they utilized some controversial videos from YouTube as a use case to classify the comments posted on them using four different classifiers (Naive bayes, Rule-based Jrip, Tree-based J48, and SVM). The dataset has around 50000 comments and divided into 50% training, 30% validation and 20% testing. However, the best accuracy as obtained by Rule-based Jrip has not exceeded 80%. Hee et al. [23] proposed a technique to detect fine-grained types of cyberbullying, such as insults and threats. The authors utilized cyberbullying content that has linguistic characteristics similar to those found in OSNs; this content (English and Dutch) was extracted from the Ask.fm website. The authors categorized the potential subjects of a cyberbullying conversation into three classes: harasser, victim, and bystander. The bystander class was split into two categories: the bystander who defends the victim, i.e., bystander-defender, and the bystander who encourages the harasser, i.e., bystander-assistant. Then, SVMs were used to differentiate the comments. However, in this paper we will focus on the detection of cyberbullying on Twitter. Detection of bullying words in the tweet contents is more challenging.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 October 2021 doi:10.20944/preprints202110.0070.v1

Sanchez et al. [18] were one of the first to propose a method to detect cyberbullying on the Twitter platform. The authors utilized the NB classifier to detect tweets that contained abusive behavior toward a specific gender. However, their method achieved only an accuracy of 70% and the size of the used dataset is relatively small. Moreover, the abusive cases should be generalized and not limited to specific topic so that it cover wide cases of cyberbullying. Saravanarj et al. [19] suggested general framework to detect rumor, bullying tweets using both NB and random forest classifiers and word2vec as a feature representation method. They also suggested that the framework can extract demographics about the abusers, such as name, gender, and age. However, the suggested methods can not give accurate results when compared to more advanced machine learning algorithm such as deep learning. Al-garadi et al. [20] presented a method that uses various unique features belonging to the Twitter platform, such as activity, network, user, and tweet, as a feature set to detect cyberbullying in Twitter. These features, along with their associated samples, were fed into a machine learning algorithm for classification purposes. The authors investigated four machine learning algorithms, i.e., random forests, NB, SVM, and KNN, and found that the random forest is the best performing algorithm in terms of the f-measure and area under the receiver operating characteristic curve. The authors used data set contains 10,007 tweets where the number of bullying tweets is only around 599 tweets.

Balakrishnan et al. [21,22] utilized the Big Five (e.g., extraversion, specifically, agreeableness, and neuroticism) and Dark Triad (e.g., psychopathy) models to determine the personality of Twitter users and sequentially detect cyberbullying. The objective of the proposed method was to investigate the relationship between cyberbullying and personality traits. The authors categorized the tweets into four categories representing the behavior of the user, namely, bully, spammer, aggressor, and normal. The authors then used the random forest ensemble method to classify each tweet into one of the previously mentioned classes. The proposed method using these personality traits achieved good results. However, the dataset contained 5453 tweets, collected using the hashtag (Gamergate), which still relatively small amount. Also, the tweets is narrowed to specific community (using the hashtag (Gamergate)) while it should be more generalized.

Chatzako et al. [2,9] analyzed a large number of Twitter comments to recognize abusive behavior characteristics. These tweets were extracted from users who participated in different topics, such as the NBA, the GamerGate controversy, and comments on gender pay inequality programs at BBC stations. The authors investigated several features extracted from Twitter, such as tweets, network-based features, and user attributes. Then, they tried different state-of-the-art classification methods to distinguish user accounts and accomplished an accuracy of 91%.

Gamback et al. [26] presented a deep learning detection system to identify Twitter cyberbullying comments. This system classified the comments into one of four possible categories: sexism, racism, both (i.e., sexism and racism), and non-offensive comments. For text representations, the authors utilized character four-grams. The authors also used word2vec for semantic analysis. Then, the authors reduced the feature set using one of the capabilities provided by a CNN layer (i.e., maxpooling layer). Consequently, they classified each tweet using a softmax function. The proposed method achieved an F-score of 78.3% when evaluated using 10-fold cross validation. The authers used datasets consist of 6655 tweets.

Sadiq et al. [12] investigated a neural network model and two deep learning architectures to detect cyberbullying presented on Twitter comments. The investigated neural network approach is known as MLP, and the two deep learning architectures are CNN-LSTM and CNN-BiLSTM. Both of the deep learning approaches achieved promising results of approximately 92% accuracy.

Pradhan et al. [24] examined the effectiveness of self-attention models (these models achieved state-of-the-art results in various machine translation tasks) in cyberbullying detection. The authors explored the usefulness of a self-attention model known as transformer architecture using three data sources: Formspring, Wikipedia, and Twitter cyberbullying

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 October 2021 doi:10.20944/preprints202110.0070.v1

datasets. This architecture replaced the recurrent layers used for encoding and decoding by a multiheaded self-attention layer. The proposed method yielded satisfactory results. Agrawal et al. [25] presented a framework and proved experimentally that this approach could overcome some existing approach limitations, such as restricting the detection to a specific SMP, shortening the detection to a single kind of hate speech (i.e., cyberbullying), and depending on handcrafted features that conventional machine learning algorithms have to offer. The authors investigated four deep learning architectures to overcome these limitations, namely, CNN, LSTM, and BiLSTM with an attention layer. The authors also classified hate speech in social media platforms into four categories: bullying, racism, sexism, and attack. They also utilized transfer learning to divert knowledge learned by deep learning on a specific dataset to another similar dataset. The investigated architectures went through extensive experiments on three different datasets: Twitter, Wikipedia, and Formspring. The authors of [24,25] used twitter data set which contain around 16k tweets.

Plaza et al. [27] proposed an approach to detect cyberbullying in social media that is related to Spanish content. The authors investigated some deep learning techniques to identify Spanish hate speech. In particular, the authors utilized transfer learning to address the limited number of sample problems by pretraining the deep learning models to improve the performance. The authors also compared the performance of pretrained deep learning models such as CNN, LSTM, BiLSTM, BERT, and Enhanced-BERT models with conventional machine learning methods such SVM and LR. The experiments showed that applying pretrained models along with BERT approaches improved the accuracy performance compared to the other deep learning and conventional models. The authors used small dataset of only around 8710 tweets. Summery of the related methods is showing in Table 2.

3. Proposed Multichannel Deep Learning Framework

In this work, we used and integrated the idea of the multichannel technique, where three mod- els that contribute to the final prediction given as.:

- The transformer block,
- bidirectional GRU (BiGRU),
- Typical CNN architecture.

Different from other existing methods, our proposed combined the power of three advanced deep learning model namely: The transformer block, bidirectional GRU (BiGRU), CNN architecture. Thus, it has the ability to extract the meaningful features and can give accurate results. In the following sections, we describe each of them in more detail.

3.1. Transformer Block

The transformer method is one of the advanced deep learning models proposed for NLP applications and particularly for machine translation. It has achieved the state of art on the WMT 2014 English-to-French translation task as well as many other tasks. Vaswani et al. [28]. It highly depends on the idea of stacked attention and the fully connected layers to compute representations of its input and output for both the encoder and decoder. The encoder has encoding layers that process the input, and the decoder has decoding layers that process the output of the encoder.

The architecture of the transformer is shown on the left side of Figure 2. Unlike RNNs and LSTM, the transformer enables fast computation using parallel computing, which speeds up the training process. The attention mechanism can deal with the limitations of the encoder-decoder model on long sequences and thus speed up the learning process and improve model performance on any prediction problem.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 October 2021 doi:10.20944/preprints202110.0070.v1

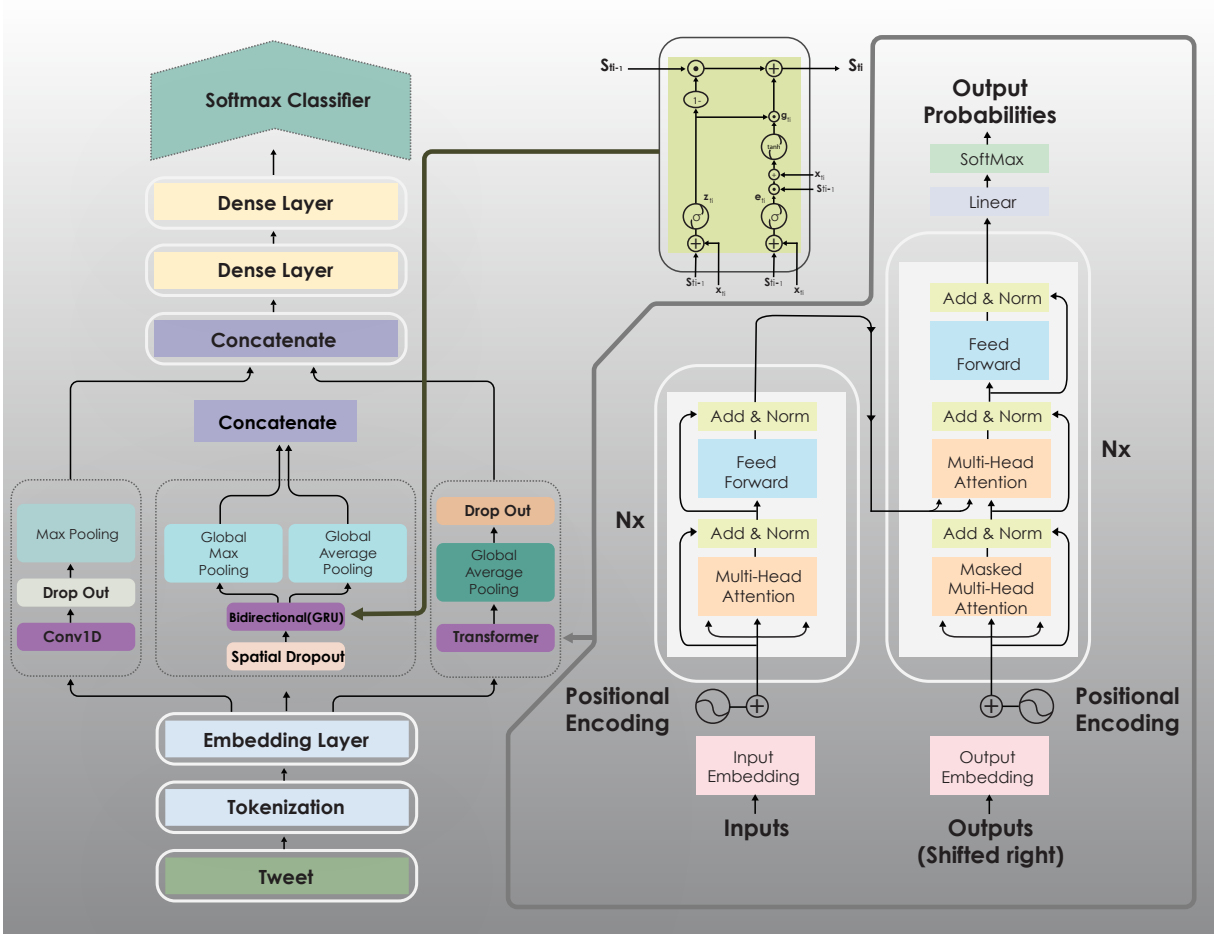


Figure 2. The proposed multichannel deep learning architecture.

The main building blocks of the transformer are scaled dot-product attention units and multihead attention (MHA). In addition, the model consists of encoder and decoder stacks, embeddings, a fully connected feed-forward network, and softmax. The scaled dot-product in the transformer can be calculated as:

$$a(q,k,v) = SM(\frac{qk^T}{\sqrt{d_k}}v) \tag{1}$$

where a denotes each attention unit, q is a query, k is a key, v is a value, SM is softmax, and d_k is the dimensionality of the key vector.

The attention weights are divided by the square root of the dimension of the key vectors The softmax function in the equation normalizes the weights to a sum of 1. MHA is calculated as in (2)

$$MHA(q,k,v) = Concat(hd_1,...,hd_h)W^O \tag{2}$$

where hd_i is calculated as follow:

$$hd_i = a(qw_i{}^Q, kw_i{}^K, vw_i{}^V) \tag{3}$$

and $w_i{}^Q, w_i{}^K, w_i{}^V$ correspond to the weight matrices to be learned.

Although the transformer uses only the attention mechanism without RNN, it is still powerful enough to outperform NLP deep models such as LSTMs and GRUs on many tasks.

For each token in the transformer block (T-block), we set the embedding size to 40, and we set the number of neurons of the hidden layer inside the transformer to 32.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 October 2021 doi:10.20944/preprints202110.0070.v1

3.2. Bidirectional Recurrent Neural Networks

In our model, the bidirectional network starts with a spatial dropout layer, which performs dropout on entire feature maps rather than individual elements. Then, the output of this layer is fed to the BiRNN layer [29] based on the GRU, which connects two hidden layers (forward and backward) of opposite directions to the same output. After that, the output of the BiRNN layer is fed the global average pooling layer and global maximum pooling layer simultaneously, and the outputs of the two layers are combined to form new input to the next stage, as shown in Figure 2.

Each input feature map is divided into many windows or partitioned into feature map grids (windows). The average pooling function calculates the average of a window of size n as follows:

$$\frac{x_1 + x_2 + ... + x_n}{n} \tag{4}$$

while the max-pooling picks the number that has the maximum value from the input window $\{x_1, ..x_n\}$. The intention of both average pooling and max pooling is to reduce the dimensionality of the data without losing essential information.

The function of the single GRU cell, which is explained in [30,31], is as follows:

$$z_{ti} = \sigma(W_{xz}x_{ti} + W_{sz}s_{ti-1} + b_z) \tag{5}$$

$$e_{ti} = \sigma(W_{xe}x_{ti} + W_{se}s_{ti-1} + b_e) \tag{6}$$

$$g_{ti} = \tanh(W_{xg}x_{ti} + W_{hg}(e_{ti} \odot h_{ti-1}) + b_g) \tag{7}$$

$$s_t i = (1 - z_{ti}) \odot s_{ti-1} + z_{ti} \odot g_{ti}. \tag{8}$$

where x_{ti} is the input to the GRU cell at time ti. W_{xg}, W_{xz} and W_{xe} are the weight matrices that receive input X_{ti}. W_{sg}, W_{se} and W_{sz} are the weight matrices that receive input from the previous cell state vector.

tanh is a hyperbolic tangent activation function, and σ is a sigmoid activation function. b_e, b_g, and b_z are the bias units. s_{ti} is the output at time ti. $\odot$ refers to the Hadamard product.

In BiRNNs, each GRN cell calculates the hidden state in the forward direction $\overrightarrow{s_{ti-1}}$ and the backward direction $\overleftarrow{s_{ti+1}}$. Thus, the BiGRU can take advantage of features in both directions. The following equation explains the idea of BiRNNs.

$$s_{ti} = \overrightarrow{s_{ti-1}} \oplus \overleftarrow{s_{ti+1}} \tag{9}$$

where $\oplus$ denotes the elementwise sum for both vectors from both directions left and right. Figure 2 shows the architecture of the GRU.

3.3. Basic CNN architecture

We also have a simple CNN, which has one CNN layer. The layer has 32 filters, each with a size of 4, and the activation ReLU is applied. Each filter in CNN takes input x and convolves CV with the filer map W

$$F = CV(W, X) \tag{10}$$

Then, the bias unit b adds the feature map FM and applies the ReLU activation function, as shown in (11)

$$Relu(F + b) \tag{11}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 October 2021 doi:10.20944/preprints202110.0070.v1

Filters in the CNN layer are randomly initialized at the beginning of the training process using the Glorot normal initializer [32] After that, the dropout technique is applied at 50%, and the output is fed to the max pooling layer, which has a pool (window) size of 2.

3.4. Multichannel

In this paper, we introduce a multichannel deep learning model that uses three networks, namely, CNN, BiRNN, and a transformer block, to process the input jointly. Our model is shown in Figure 2. Each network produces output, which is combined and fed to fully connected layers, which are two dense layers. The first dense layer has 60 neurons, and the second has 30 neurons. Then, the output is fed to the softmax classifier. Softmax classifies the input data into one of two classes: cyberbullying or not.

The outputs of the three networks are combined using the concatenate layer. If we have three output vectors U, I, O, where the vector U is as follow $U = \{U_1, U_2, ..U_i\}$,, the vector I is as follow $I = \{I_1, I_2, ..I_i\}$, and the vector O is as follow $O = \{O_1, O_2, ..Oi\}$,. Then, the concatenate layer would combine them into one vector as follow:

$$V = Conc(U, I, O) \tag{12}$$

The result V would be as follow: $V = \{U_1, U_2, U_i, I_1, I_2, I_i, O_1, O_2, Oi\}$

Dense layer calculates its output DL as follows:

$$DL = F(\sum_i w_i + \cdot x_i + b) \tag{13}$$

Where w is the wights multiplied by the input value and the b is added after that. Then the activation function is applied at the end.

In this model, we choose to use a binary cross-entropy loss function since we are dealing with a binary classification problem. The equation in (14) explains the process of the entropy loss function.

$$BC = -\sum_c y_c \cdot log(s_\theta(x)_c) \tag{14}$$

where c is the index for the classes. We have two classes in this problem. y is the correct value for class c, s is the predicted probability for class c, and x is the current input data. Moreover, we use the Adam optimizer to optimize the network. The total number of trainable parameters in our proposed model is 17,868,016.

4. Testing

In this section, we present the dataset used to evaluate the proposed method. Subsequently, we discuss the data preprocessing used to prepare the raw text to be fed into our model. This section concludes with information on the equipment utilized to train our model and validate its effectiveness.

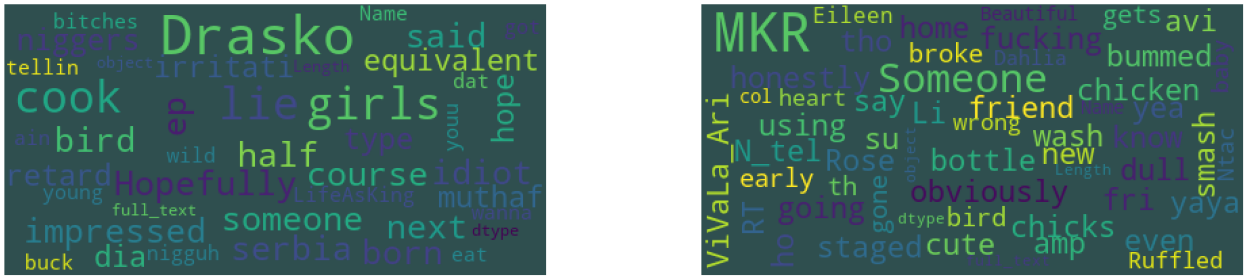
4.1. Dataset

Our data are taken from three sources, [33–35], which have been combined to form a dataset of 55,788 tweets. The dataset is divided into two categories: offensive and non-offensive. Table 2 shows more details about our datasets in which the percentage of offensive samples is 65.8% (i.e., 23,548) and the percentage of the non-offensive samples is 34.2% (i.e., 12,239).

Table 2. Category of tweets in the dataset.

Category	Number of Instances	Percentile
Offensive	31353	56.2%
Non-offensive	24435	43.8%

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 October 2021 doi:10.20944/preprints202110.0070.v1



(a) The most frequent words that belong to the offensive class.

(b) The most frequent words that belong to the non-offensive class.

Figure 3. Word cloud of most frequent words in the dataset that belong to the two classes.

Offensive: tweets that appear to be cyberbullying behavior. Tweets with negative content that have been posted to insult or harm other people.

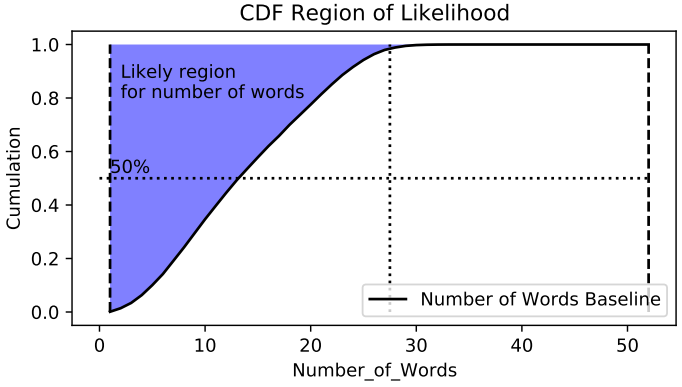
Non-offensive: tweets that appear to be ordinary (i.e., do not manifest cyberbullying behavior). Tweets with normal content posted with no intent to insult or harm other people.

4.2. Data Preprocessing

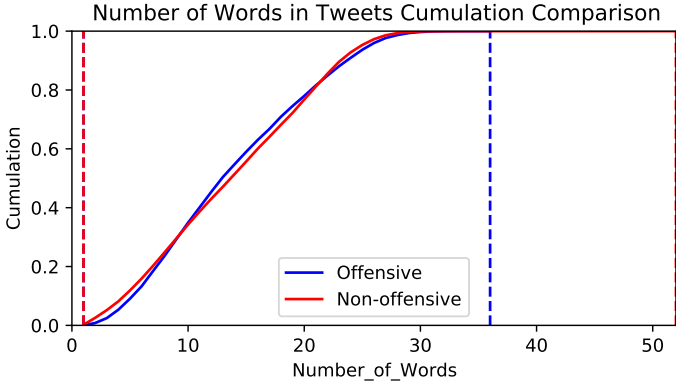
Tokenization or (lexical analysis) is the procedure of transforming a sequence of characters into a sequence of tokens. We converted all the collections of text (tweets) to a sequence of tokens. This procedure vectorizes a text collection by turning every text into a sequence of integers. We set the maximum number of words to keep during the tokenization to 80,000 words.

4.3. Equipment and Tools

The proposed method was implemented using the Python programming language. Several Python libraries were utilized for data cleaning, preprocessing, and model implementation. The data were preprocessed using the pandas library [36]. The deep multichannel model was implemented utilizing the Keras library [37] and evaluated using the scikit-learn library [38]. We utilized the tensor processing unit (TPU) provided by Google Colab, which is an open source iPython notebook that runs on the cloud to facilitate collaboration, instead of a GPU to validate our proposed method. We trained our method using a batch size of 100.



(a) The cumulative distribution function for both classes.



(b) The accumulation comparison for offensive and non-offensive classes in terms of the number of words in tweets.

Figure 4. The cumulative distribution function for both classes where the majority of tweets have less than 20 words and the accumulation comparison for offensive and non-offensive classes in terms of the number of words in tweets.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 October 2021 doi:10.20944/preprints202110.0070.v1

5. Evaluation and Results

To investigate the most frequent words in the dataset, we used a useful visualization technique known as a word cloud(documentation of this technique can be found at https://amueller.github.io/word_cloud to illustrate the most frequent words present in encoded text such as tweets.

Figure 3 shows the most frequent words present in the cyberbullying detection dataset. The most frequent words present in the offensive tweets include politicians' names, county names, disrespectable words, and some normal words. Some of these words are not offensive; however, when they combine with other words in the same sentence, they present an offensive phrase. Most of the frequent words appearing in the non-offensive tweets include normal words except for some outliers that, when combined with other words, have a non-offensive meaning (e.g., f***ing nice).

To explore the data and determine the number of words for offensive and non-offensive tweets, we used a technique called CDF (a.k.a., CDG) inspired by the work that is available at the following link: https://www.kaggle.com/jell9265/streamlined-eda-cumulative-distribution-graphs. Figure 4a shows the number of words for the two classes, and it is obvious that the majority of tweets have fewer than 20 words. A detailed plot for each class is shown in Figure 4b.

For both classes, the minimum number of words is one word. The maximum number of words for the offensive class is 37, while the maximum number of words for the non-offensive class is 52 words. Both classes have similar distributions, where the mean number of words is 14. Four evaluation metrics were used to validate our proposed method: precision, F-measure, recall and accuracy. Prior to identifying these four metrics, the four building blocks for these metrics are defined as TP, TN, FP, and FN. Cyberbullying tweets (i.e., offensive instances) that are correctly classified as offensive are TPs, while ordinary tweets (i.e., non-offensive instances) that are correctly classified as non-offensive TNs. Cyberbullying tweets that are misclassified (i.e., classified as non-offensive) FNs, while non-offensive tweets that are mistakenly classified as offensive instances are FPs.

The accuracy is calculated using the following equation:

$$accuracy = \frac{tp + tn}{t} \tag{15}$$

where t is the total population $t = tp + tn + fp + fn$.

Accuracy measures the correct predictions of both the number of offensive tweets that are correctly classified as offensive (i.e., TP) and the number of non-offensive tweets that are correctly classified as non-offensive (i.e., TN) among the whole testing set.

The precision is presented to measure the number of correctly classified offensive tweets among all the instances in the testing set that are classified as offensive either correctly or mistakenly.

$$precision = \frac{tp}{tp + fp} \tag{16}$$

The recall is introduced to measure the number of correctly classified offensive tweets among all the offensive instances in the testing set.

$$recall = \frac{tp}{tp + fn} \tag{17}$$

The F-measure or F-score is the harmonic mean of both recall and precision and is calculated as in equation 18:

$$f - score = 2 * \frac{precision * recall}{precision + recall} \tag{18}$$

Three sets of experiments are utilized to validate our proposed method in terms of accuracy, as shown in Table 3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 October 2021 doi:10.20944/preprints202110.0070.v1

Table 3. Accuracy using 55,788 tweets with different training-testing splits.

Training	Testing	Accuracy
75%	25%	87.99%
50%	50%	85.36%
30%	70%	82.78%
20%	80%	79.75%
10%	90%	78.23%

In the first experiment, the dataset is divided into 75% training and 25% testing. In the second experiment, the dataset is split into halves (i.e., 50% training and 50% testing). In the third experiment, the dataset is divided into 30% training and 70% testing. Then, for 20% training and 80% testing, in the last experiment, the dataset is set to 10% training and 90% testing. The accuracy is 87.99% when we split the dataset into 75% training and 25% testing, 85.36% when we split the dataset into 50% training and 50% testing, 82.78% when we split the dataset into 30% training and 70% testing, 79.75% when we split the dataset into 20% training and 80% testing, and 78.23% when we split the dataset into 10% training and 90% testing. As the number of training samples decreases in the last set of experiments, the performance of our method decreases as well.

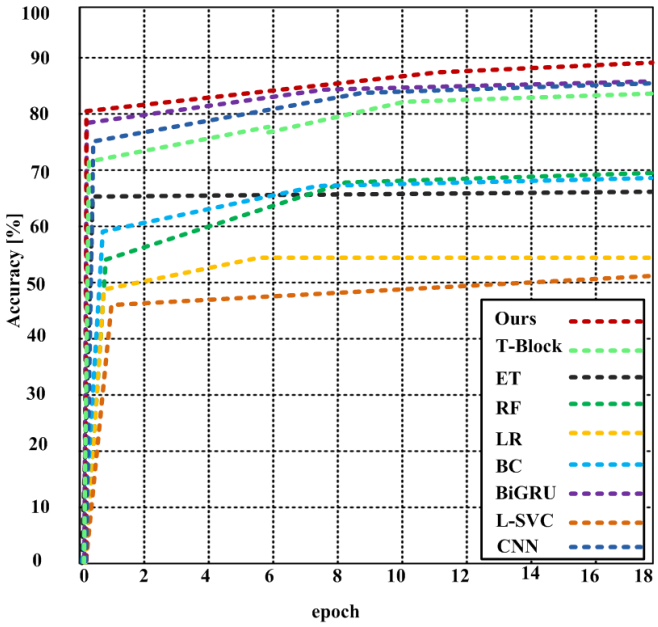


Figure 5. Comparison with different methods in terms of accuracy.

To evaluate the performance of the proposed method in terms of accuracy, we compare it with eight well-known machine learning algorithms (i.e., CNN, BiLSTM, transformer, linear SVC, bagging, LR, RF, and ET) that achieved high accuracy in various natural language processing tasks. We used tokenization as a preprocessing step with all of these algorithms. The performance of our proposed method is better than the rest of the algorithms when we divided the data into 75% training and 25% testing, as shown in Table 4 and depicted in Figure 5.

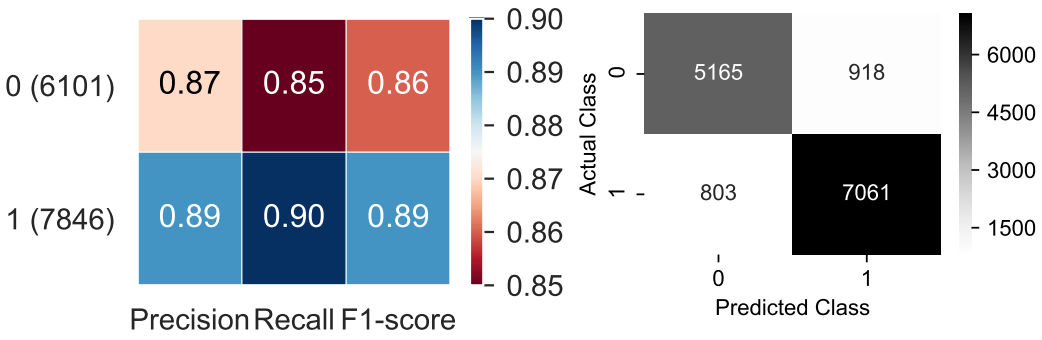
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 October 2021 doi:10.20944/preprints202110.0070.v1

Table 4. Comparison with different methods.

Method	Accuracy
ET	65.62%
RF	69.29%
LR	54.27%
Bagging Classifier	68.30%
Linear SVC	50.33%
Transformer block	86.99%
CNN	87.28%
BiGRU	87.43%
Our method	**87.99%**

The accuracy of our method is 87.99%, which is better than the second-most accurate algorithm (i.e., BiGRU) by approximately half a percent, better than the third- and fourth-most accurate algorithms by approximately one percent, and better than the rest by a good margin.

We also evaluated our method using four other metrics (i.e., precision, recall, F1-score, and a confusion matrix). Our proposed method achieved good performance of 87% precision, 85% recall, and 86% F-score when recognizing non-offensive instances, as shown in Figure 6a. Our method yielded an outstanding performance of 89% precision, 90% recall, and 89% F1-score when recognizing offensive instances. Figure 6b shows the confusion matrix of our proposed method in which approximately 85% of non-offensive instances are classified correctly (i.e., 5165 out of 6083 samples), while 90% of offensive instances are classified correctly (i.e., 7061 out of 7864 samples).



(a) The three evaluation metrics used to evaluate our method. **(b)** The confusion matrix for the two classes.

Figure 6. The three measures (i.e., precision, recall, and F1-score) and the confusion matrix used to evaluate our proposed method.

6. Conclusions

Online social networks have become an important aspect of our daily routines due to the provided ease of social interaction. However, the increase in antisocial behavior such as hate speech, trolling, and cyberbullying in social networks such as Twitter and the consequences that social media users encounter makes cyberbullying detection an important topic to explore. This paper presents a new cyberbullying detection technique using a combination of three deep learning architectures (i.e., a multichannel architecture consisting of BiGRU, a transformer block, and CNN models). The proposed method is evaluated using three famous cyberbullying datasets (i.e., we combined the three datasets to have enough samples to train our model). The experimental results show the significance of this method in classifying short messages (e.g., tweets). The proposed method achieved good results compared to the state-of-the-art methods on the three datasets, achieving

 doi:10.20944/preprints202110.0070.v1

an accuracy of approximately 88% when the dataset was split into 75% training and 25% testing.

7. Future Work

In the future, we plan to apply our method to a larger dataset. We believe that by using a larger dataset, the performance of our method can be enhanced. Deep learning algorithms require large data sets in order to have a good performance. Furthermore, We will try to enlarge the proposed framework by adding many channels. Increasing the number of channels when using a large dataset could improve the performance of the framework. A large dataset can help to optimize the weights and other parameters of deep and large neural networks. Moreover, we also plan to test our proposed framework with tweets of different languages.

Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript

Funding: M.A., conceptualization, writing, idea proposal, methodology, software, submission, and results; B.A, data curation, resources, Validation, software, writing,, and preparation; A.R., conceptualization, review, editing, and Supervision. This research received no external funding.

Informed Consent Statement: Not applicable

Conflicts of Interest: The authors declare no conflict of interest

Abbreviations

The following abbreviations are used in this manuscript:

NLP	natural language processing
BoW	bag of words
SVM	support vector machine
LR	logistic regression
MLP	multilayer perceptron
CNN	convolutional neural network
RNN	recurrent neural network
BiRNN	bidirectional RNN
LSTM	long sort-term memory
BiLSTM	bidirectional LSTM
GRU	gated recurrent unit
BiGRU	bidirectional GRU
BERT	bidirectional encoder representations from transformers
OSN	online social network
ANN	Artificial neural network
NB	naive Bayes
KNN	k-nearest neighbor
NBA	National Basketball Association
BBC	British Broadcasting Corporation
SMP	social media platform
GPU	graphics processing unit
CDF	cumulative distribution function
CDG	cumulative distribution graphs
RF	random forests
ET	extra trees
MH	multihead attention
WMT	workshop on statistical machine translation
TP	true positive
TN	true negative
FP	false positive
FN	false negative

References

1. Khan, M.U.S.; Abbas, A.; Rehman, A.; Nawaz, R. HateClassify: A service framework for hate speech identification on social media. *IEEE Internet Comput.* **2020**, *25*, 40–49.
2. Chatzakou, D.; Kourtellis, N.; Blackburn, J.; Cristofaro, E.D.; Stringhini, G.; Vakali, A. Mean birds: Detecting aggression and bullying on twitter. In Proceedings of the 2017 ACM on Web Science Conference, 25-28 June 2017; pp. 13–22.
3. Wullach, T.; Adler, A.; Minkov, E.M. Towards hate speech detection at large via deep generative modeling. *IEEE Internet Comput.* **2020**, *25*, *48–57.*
   Awais, M.; Hassan, S.U.; Ahmed, A. Leveraging big data for politics: Predicting general election of Pakistan using a novel rigged model. J. Ambient Intell. Human. Comput. ***2019***, *1–9, doi:10.1007/s12652-019-01378-z.*
4. *Razaque, Abdul; Fathi Amsaad; Dipal Halder; Mohamed Baza; Abobakr Aboshgifa; and Sajal Bhatia. Analysis of Sentimental Behaviour over Social Data Using Machine Learning Algorithms.* In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. ***2021***, *pp. 396-412. Springer, Cham.*
5. *Grigg, D.W. Cyber-aggression: Definition and concept of cyberbullying.* J. Psychol. Couns. Sch. ***2010***, 20, *143–156.*
6. *Rosa, H.; Pereira, N.; Ribeiro, R.; Ferreira, P.C.; Carvalho, J.P.; Oliveira, S.; Trancoso, I. Automatic cyberbullying detection: A systematic review.* Comput. Hum. Behav. ***2019***, 93, *333–345.*
7. *Cheng, L.; Silva, Y.N.; Hall, D.; Liu, H. Session-based cyberbullying detection: Problems and challenges.* IEEE Internet Comput., ***2020***, *doi:10.1109/MIC.2020.3032930.*
8. *Razaque A, Al Ajlan A, Melaoune N, Alotaibi M, Alotaibi B, Dias I, Oad A, Hariri S, Zhao C. Avoidance of Cybersecurity Threats with the Deployment of a Web-Based Blockchain-Enabled Cybersecurity Awareness System.* Applied Sciences. ***2021***, *11, 17, 7880.*
9. *Chatzakou, D.; Leontiadis, I.; Blackburn, J.; Cristofaro, E.D.; Stringhini, G.; Vakali, A.; Kourtellis, N. Detecting cyberbullying and cyberaggression in social media.* ACM Trans. Web (TWEB) ***2019***, 13, *1–51, 2019.*
10. *Rozsa, M.* Twitter Trolls Are Now Abusing the Company's Bottom Line*; 2016; Salon, goo:gl/SryS3k.* .
11. *Jacob, S.S.; Vijayakumar, R. Sentimental analysis over twitter data using clustering based machine learning algorithm.* J. Ambient Intell. Human. Comput. ***2021***, *1–12.*
12. *Sadiq, S.; Mehmood, A.; Ullah, S.; Ahmad, M.; Choi, G.S.; On, B.W. Aggression detection through deep neural model on Twitter.* Future Gener. Comput. Syst. ***2021***, 114, *120–129.*
13. *Gencoglu, O. Cyberbullying detection with fairness constraints.* IEEE Internet Comput. ***2021***, 25, *20–29.*
14. *Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing.* IEEE Comput. Intell. Mag. ***2018***, 13, *55–75.*
15. *Xiao, Y.; Cho, K. Efficient character-level document classification by combining convolution and recurrent layers.* arXiv ***2016***, *arXiv:1602.00367.*
16. *Tai, K.S.; Socher, R.; Manning, C.D. Improved semantic representations from tree-structured long short-term memory networks.* arXiv ***2015***, *arXiv:1503.00075.*
17. *Dinakar, K.; Reichart, R.; Lieberman, H. Modeling the detection of textual cyberbullying.* Proc. Social Mobile Web ***2011***.
18. *Sanchez, H.; Kumar, S. Twitter bullying detection.* Ser. NSDI ***2011***, 12, *15.*
19. *Saravanaraj, A.; Sheeba, J.I.; Devaneyan, S.P. Automatic detection of cyberbullying from twitter.* Int. J. Comput. Sci. Info. Technol. Secur. ***2016***, 6.
20. *Al-garadi, M.A.; Varathan, K.D.; Ravana, S.D. Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network.* Comput. Hum. Behav. ***2016***, 63, *433–443.*
21. *Balakrishnan, V.; Khan, S.; Fernandez, T.; Arabnia, H.R. Cyberbullying detection on twitter using Big Five and Dark Triad features.* Pers. Individ. Differ. ***2019***, 141, *252–257.*
22. *Balakrishnan, V.; Khan, S.; Arabnia, H.R. Improving cyberbullying detection using Twitter users psychological features and machine learning.* Comput. Secur. ***2020***, 90, *101710.*
23. *Hee, C.V.; Lefever, E.; Verhoeven, B.; Mennes, J.; Desmet, B.; Pauw, G.D.; Hoste, V. Automatic detection and prevention of cyberbullying.* Int. Conf. Human and Social Analytics (HUSO 2015) ***2015***, IARIA, *13–18.*
24. *Pradhan, A.; Yatam, V.M.; Bera, P. Self-attention for cyberbullying detection. In Proceedings of the 2020 International Conference Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), Dublin, Ireland, 15–19 June 2020; pp. 1–6.*
25. *Agrawal, S.; Awekar, A. Deep learning for detecting cyberbullying across multiple social media platforms. In* European Conference on Information Retrieval*; Springer: Cham, Switzerland, 2018; pp. 141–153.*
26. *Gamback, B.; Sikdar, U.K. Using convolutional neural networks to classify hate-speech. In Proceedings of the First Workshop on Abusive Language Online, August 2017; Association for Computational Linguistics: Vancouver, Canada, pp. 85–90.*
27. *Plaza-del-Arco, F.M.; Molina-Gonzalez, M.D.; Urena-Lopez, L.A.; Martin-Valdivia, M.T. Comparing pre-trained language models for Spanish hate speech detection.* Expert Syst. Appl. ***2021***, 166, *114120.*
28. *Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Polosukhin, I. Attention is all you need.* Adv. Neural Inf. Process. Syst. ***2017***, *5998–6008.*
29. *Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks.* IEEE Trans. Signal Process. ***1997***, 45, *2673–2681.*
30. *Cho, K.; Merriënboer, B.V.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches.* arXiv ***2014***, *arXiv:1409.1259.*

 doi:10.20944/preprints202110.0070.v1

31. *Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling.* arXiv **2014**, *arXiv:1412.3555.*
32. *Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference Artificial Intelligence and Statistics, 2010; pp. 249–256.*
33. *Davidson, T.; Warmsley, D.; Macy, M.; Weber, I. Automated hate speech detection and the problem of offensive language.* arXiv **2017**, *arXiv:1703.04009.*
34. *Waseem, Z.; Hovy, D. Hateful symbols or hateful people? Predictive features for hate speech detection on twitter. In Proceedings of the NAACL Student Research Workshop, June 2016; Association for Computational Linguistics, pp. 88–93.*
35. *DataTurks. Kaggle. Tweets Dataset for Detection of Cyber-Trolls. 2018. Available online: https://www.kaggle.com/datat urks/dataset-for-detection-of-cybertrolls (accessed on 15, January 2021).*
36. *McKinney, W. Pandas: A foundational Python library for data analysis and statistics.* Python High Perform. Sci. Comput. **2011**, 14.
37. *Chollet, F. Keras: The Python deep learning API. GitHub repository. 2015. Available online: https://github.com/fchollet/keras (accessed on 15, January 2021).*
38. *Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Vanderplas, J. Scikit-learn: Machine learning in Python.* J. Mach. Learn. Res. **2011**, 12, *2825–2830.*