Article

MaLang: A Decentralized Deep Learning Approach for Detecting Abusive Textual Content

Pranav Kompally ¹, Sibi Chakkaravarthy Sethuraman ^{1,*}, Steven Walczak ², Samuel Johnson ³
and Meenalosini Vimal Cruz ⁴

¹ School of Computer Science and Engineering & Artificial Intelligence and Robotics Research Center, VIT-AP University, Amaravati 522237, Andhra Pradesh, India; pranav.18bce7169@vitap.ac.in

² School of Information & Florida Center for Cybersecurity, University of South Florida, Tampa, FL 33620, USA; swalczak@usf.edu

³ School of Business, VIT-AP University, Amaravati 522237, Andhra Pradesh, India; samuel.johnson@vitap.ac.in

⁴ Department of Information Technology, Georgia Southern University, Statesboro, GA 30458, USA; mvimalcruz@georgiasouthern.edu

* Correspondence: chakkaravarthy.sibi@vitap.ac.in

Abstract: Cyberbullying is a growing and significant problem in today's workplace. Existing automated cyberbullying detection solutions rely on machine learning and deep learning techniques. It is proven that the deep learning-based approaches produce better accuracy for text-based classification than other existing approaches. A novel decentralized deep learning approach called MaLang is developed to detect abusive textual content. MaLang is deployed at two levels in a network: (1) the System Level and (2) the Cloud Level, to tackle the usage of toxic or abusive content on any messaging application within a company's networks. The system-level module consists of a simple deep learning model called CASE that reads the user's messaging data and classifies them into abusive and non-abusive categories, without sending any raw or readable data to the cloud. Identified abusive messages are sent to the cloud module with a unique identifier to keep user profiles hidden. The cloud module, called KIPP, utilizes deep learning to determine the probability of a message containing different categories of toxic content, such as: 'Toxic', 'Insult', 'Threat', or 'Hate Speech'. MaLang achieves a 98.2% classification accuracy that outperforms other current cyberbullying detection systems.

Keywords: cyberbullying; deep learning; abusive text; decentralization; cyber-harassment

Citation: Kompally, P.; Sethuraman, S.C.; Walczak, S.; Johnson, S.; Cruz, M.V. MaLang: A Decentralized Deep Learning Approach for Detecting Abusive Textual Content. *Appl. Sci.* **2021**, *11*, 8701. <https://doi.org/10.3390/app11188701>

Academic Editors: Giancarlo Mauri and Mauro Castelli

Received: 2 July 2021

Accepted: 15 September 2021

Published: 18 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

While cyberbullying is a widely recognized problem for adolescents [1], racism, misogyny, homophobia, and other causes of cyberbullying do not just disappear with the granting of a diploma or degree. These attitudes and the large percentage of type-A personalities present in STEM-related businesses [2] lay a fertile ground for cyberbullying of adults. The ever-increasing prevalence and importance of electronic communication mechanisms in organizations today including social networks, along with the ever-expanding global cyber-marketplace, are the foundations necessary to facilitate cyberbullying in the workplace [3]. Cyberbullying in the workplace is typically conducted using social networks and social media, email, and text (SMS) messages, which allow for cyberbullies to often remain anonymous [4]. Additionally, the recent migration of employees to online workplace environments due to the COVID-19 pandemic and consequent reliance on social media networks and the Internet for conducting everyday work may further exacerbate the problem [5,6].

Although the subjects of child and adolescent cyberbullying are well-documented in academic research [7–12], workplace cyberbullying research is still in its infancy and there

is a need for further research on the topic of workplace cyberbullying [3]. There is a need for automatic intelligent detection of cyberbullying in organizational contexts [13].

Cyberbullying in the workplace is an ongoing and significant problem. Studies conducted in Finland found from 12.6% to 17.4% of employees felt they had experienced cyberbullying on a monthly basis [4]. From 14% to 21% of higher education faculty have been documented to suffer from cyberbullying [14]. Cyberbullying can last for an extended period of time [15]. Various studies have shown how cyberbullying increases employee stress and job dissatisfaction [16,17]. These effects of workplace cyberbullying in turn may cause a decrease in worker productivity and if not resolved after being reported to human resources (HR), may result in litigation.

In this paper, a decentralized deep learning approach called Malicious Language Detector (MaLang) is proposed. While cyberbullying may include text, images, photographs, video, and audio content [18], the MaLang system only analyses textual content in user devices. The decentralized methodology facilitates keeping an employee's data private and secure. MaLang is simulated in a workplace environment to demonstrate the applicability of intelligent automated detection of cyberbullying messages for improving workplace safety and well-being. Previous machine learning and deep learning techniques are re-combined in a novel way to improve the overall accuracy of hostile content in an organization's electronic messaging systems. The two-part MaLang method helps to minimize deployed size, improve employee privacy, and increase hostile message detection.

2. Related Research Work

Around the world, more than one billion people use messaging platforms, such as WhatsApp or Facebook Messenger, and more than 30 billion messages are exchanged daily [19]. However, not all of these messages are beneficial to individuals or organizations as a lot of toxic textual content is shared in public, private, and workplace spaces. Abusive textual content includes words or phrases that either insult or show hate speech toward a person or community, or that pose a threat to a person, or project obscene opinions. Various applications have been deployed to control and minimize the sharing of such content [20].

Earlier information retrieval (IR) and natural language processing (NLP) methods are applied for resolving the text-related classification problems such as sentiment analysis [21]. IR and NLP have been previously applied for the detection of cyberbullying text-oriented messages [11]. Various other cyberbullying detection techniques have been previously developed, with most being dependent on textual and user features [7]. A simple program was developed by making use of a dictionary of keywords to perform cyberbullying detection over online chats [22], and performance was improved by collecting a much larger sample of labeled data [23].

Sensitive topics in the offensive and cyberbullying-oriented comments of YouTube such as racism or sexual aggression are considered in building the detection model [24]. The roles of bullies and other participants are studied and segregated into the groups of accuser, bully, and victim [21]. Feature engineering is a commonly used method for detecting cyberbullying from text-related representations. A twitter-based model extracts features from tweets and classifies them into the different groups mentioned in [21]. In addition, to generate additional features, the domain knowledge of data is also used for improving the performance of the classifiers. The performance of Support Vector Machines (SVM) is improved for the detection of cyberbullying as a result of adding features such as the occurrence and frequency of bad language, number of pronouns, length of comments, use of capital letters, and emoticons [20]. Similarly, pronoun counts are also considered a feature with the representation of n-grams and skip-grams [25]. The usage and application of soft computing for detecting cyberbullying are reviewed by [26].

A combination of textual and social network features is utilized for cyberbullying detection in [27]. Social network analysis (SNA) features are represented as a graph with the combination of nodes and edges used for discriminating the activities of cyberbullying. The usage of textual features along with the term frequency-inverse document fre-

quency method improved the detection of cyberbullying. SNA has been further extended for the labeled features such as racism, sexual, and intelligence [28,29]. The classification of more indirect and hidden labels related to bullying can improve the accuracy of cyberbullying detection.

Zhao et al. [30] presented an insult-related word embedding method for extracting the features of bullies. They also used the additional features that were extracted from a dictionary of general words used in social networks [31]. However, their method works well only on a balanced dataset and suffers in real time with imbalanced datasets [32].

The issue of real time deployment and huge data processing is addressed by various researchers who used neural network-based models [33]. A simple convolutional neural network (CNN) model is utilized over pre-trained word vectors to perform classification based on the number of tokens presents in the target text [34,35]. The CNN model was used on 50 million tweets for detecting cyberbullying [35,36]. A pronunciation-based CNN is proposed to tackle the challenges of cyberbullying such as misspellings in the messages and posts in social media [37]. Due to their ability to identify spoken language, CNNs are frequently used to build models for detecting bullying tweets [8,38–42]. Abusive language detection was performed by training Bidirectional Encoder Representations from Transformers (BERT) on a Reddit Abusive Language Dataset. Hate speech detection in different languages was achieved using various hybrid deep learning models such as Bidirectional Encoder Representations from Transformers (BERT) [41], Graph Neural Network Model [43], BERT-CNN [44,45], Gated Recurrent Units (GRU) [46], CNN-GRU and mBERT, CNN-LSTM (Long Short-Term Memory) [10], and Heterogeneous Neural Interaction Networks [47].

Machine learning ensemble models for cyberbullying detection have also been proposed [12,48], though more infrequently. The deep learning architectures used for experimentation include the specialization in regularization, convolution, and bi-directional networks in which the authors defined various metrics for evaluation. It is also observed that the LSTM models are provided with more parameters for training than GRU, which performs with the highest median accuracy with minimal metrics. The major issues with the hybrid models are that the experimental results and evaluations are bound to a single dataset [10,28]. Automatic improper conversation (such as disgusting language and offensive chats) extraction is complicated and time consuming [32,49–51].

Moreover, the processing of huge knowledge bases on resource-constrained platforms such as on-boards and mobile devices is very difficult. Increasing the number features increases the difficulties in the process of feature extraction and selection. No current research uses a word's meaning or semantics for cyberbullying detection [52–54]. Transformers, mentioned in a few papers stated above, often require high computing resources while training, especially at regular intervals. These models combined with different types of recurrent neural networks produce predictions with decent accuracy and precision. A novel approach is introduced to detect and restrict abusive textual content being exchanged in a network with better accuracy and precision. MaLang solves the challenge of detecting abusive text at both the edge and cloud levels, without consuming high computation resources and producing optimal predictions, while dealing with small influx of data at regular intervals. Table 1 provides a summary of the various state-of-the-art approaches and the accuracy of these models for detecting bullying language.

Major Contributions of MaLang

- MaLang uses two-stage abuse detection and restriction analysis as opposed to more traditional single-stage detection processing.
- At the edge level, MaLang uses Bi-LSTM-based CASE to detect toxic text, and at the Cloud level, KIPP utilizes a combination of Bi-LSTM- and Bi-GRU-based models for Entity-based recognition. This combination of machine learning approaches has not been used previously.

- MaLang proposes an architecture that keeps data private at the user level and the cloud level, without disclosing or transmitting the profile of the user involved in the system. Thus, data privacy and confidentiality are a central focus of the MaLang system design.
- Unlike simple RNN approaches, MaLang uses an actively monitored pipeline to detect and alert users and administrators of possible toxic textual content being exchanged.

Table 1. State-of-the-art research on cyber bullying detection.

Approach	Reference	Methods and Models	Accuracy
Text Analytics	[22]	Keyword matching	58.6%
Natural Language Processing	[7]	Sentiment Analysis	95%
	[53]		N/A
Machine Learning (ML)	[23]	C4.5 decision tree	78.5%
	[20]	SVM	77%
	[21]		77%
	[30]		78%
	[48]	Ensembles of ML classifiers	90.57%
	[12]		91%
Deep Learning	[35]	CNN	81%
	[33]		84%
	[34]		92.8%
	[50]		96%
	[53]	Deep Neural Network	73.2%
	[40]		86.6%
	[8]		93.97%
	[39]	LSTM	92.7%
	[38]		93.97%
	[55]	Bi-directional LSTM	52.6%
	[42]	Tree and LSTM	90.3%
	[45]	Gradient Boosting BERT and LASER	62%
	[56]	CNN-GRU and mBERT	95%
	[10]	CNN-LSTM and CNN-BiLSTM	92%
	[46]	LSTM and GRU	98%

3. Materials and Methods

To overcome issues encountered in the previous related work described above, this paper proposes a novel approach called MaLang. MaLang is a decentralized learning approach to detect abusive textual content and initiate managerial response to detected cyberbullying events. MaLang leverages the use of Bi-LSTM with a Max Pooling Layer for abusive content classification in system-level modules and the use of Bi-LSTM and Bi-GRU with a combination of Average and Max Pooling layers. The cloud-level modules are interconnected for different departments in the workplace when deployed in medium to very large organizations. When a tagged entity is received in one module, a copy is sent to other cloud modules in the network to check if a similar entity is present. In the case of a match, it is concluded that two or more people were involved in the sharing, and an alert is sent to the Human Resource representative and the cyber protocol enforcement department.

As shown in Figure 1, the automated business-level cyberbully detection framework is divided into two modules: System Level and Cloud Level. The System-level module consists of the deep learning model called ‘CASE’ that classifies abusive content from non-abusive content on the user’s device. The collected abusive content is transmitted to the cloud to detect the intensity of the abusive content. The Cloud-level module consists of a cloud-based deep learning model called ‘KIPP’ that classifies the level of Toxic Content of each sentence into one of four classes: ‘Toxic’, ‘Threat’, ‘Insult’, or ‘Hate Speech’. A sentence can contain multiple classes of Toxic content as presented in Table 2, which shows examples of partial real-world organizational messages and their corresponding classifications (the second message has been modified for viewing purposes).

Figure 1. Architecture of MaLang.

Table 2. Toxic content and different classes.

Sentence	Toxic	Threat	Insult	Hate Speech
You are gay or antisemmitian?	Yes	No	Yes	Yes
C*****ER BEFORE YOU PISS AROUND ON MY WORK	Yes	Yes	Yes	Yes
Bye! Don't look, come or think of coming back! Tosser.	No	No	Yes	No

3.1. System-Level Module of MaLang

Figure 2 depicts the system module architecture of MaLang. The application is deployed as an extension to most of the messaging and communication applications. The application collects communication data such as conversations from messaging applications such as WhatsApp or Messenger as a text file. Every sentence is in raw textual format. To convert it into usable form, every sentence is tokenized into entities. The tokenized entities are cropped to uniform length to be converted into input vectors. These entities are represented as arrays of vectors using one-hot encoding. One-hot encoding simply converts categorical variables into unique features or numerical values that can be easily fed to a deep learning model. The obtained vectors are passed through the deep learning model CASE. CASE is built with a sequence of Bi-directional LSTM and fully connected neural

network layers that classify the entities into Abusive and Non-abusive. The categorized Abusive entities are then pushed to the cloud to detect levels of toxicity. Subsequently, all the classified entities are used as a training data set for continuous learning to improve the model. Every entity received by the application is used to improve the model.

Figure 2. System-level Module of MaLang.

3.2. Cloud-Level Module of MaLang

Figure 3 depicts the Cloud module architecture of MaLang. The detected abusive entities containing the array of tokens are tagged with the user's identification number and transmitted to the cloud module. The received tokens are converted into input vectors of 32 indices by mapping the word to the id included in the pre-trained GloVe vectors. This creates an embedding matrix that is passed through the cloud-based deep learning model 'KIPP'. KIPP consists of a sequence of Bi-directional LSTM, Bidirectional GRU, pooling layers, and fully connected layers. The KIPP system classifies the level and type of abusive content present in the sentence as shown in Figure 4.

3.3. Recurrent Neural Networks

Many text classification operations have been performed using recurrent neural networks (RNN) [57,58]. They enable the modeling of sequence-based tasks; however, they are known to suffer from the problem of vanishing gradients that diminishes the learning of long sequences [59]. To avoid the problem of a vanishing gradient, an ensemble of Bi-directional LSTM and Bi-directional GRU are used.

Bidirectional LSTM or Bi-LSTM allows the neural networks to preserve both forward and backward information received from input about the sequence at every step with respect to time, making it time-dependent. Inputs are used in two ways, past to future and future to past, allowing the Bi-LSTM to better understand the context of the text. Figure 5 shows the design of a Bi-directional LSTM layer along with the operations happening in each cell. As seen in the figure, LSTM has three gates, input, output, and forget gates. Unlike RNN, the cell state derivative of cell state vectors prevent gradient vanishing since the long-term dependencies are encoded in them.

Figure 3. Cloud-level Module of MaLang.

Figure 4. Probability of the presence of different types of toxic content.

Figure 5. (a) Architecture of the Bi-LSTM neural network (b) Functions inside a single LSTM cell.

The operations that happen in LSTM are:

Previous state:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \dots \quad (1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \dots \quad (2)$$

where i_t input/update the gate's activation vector.

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \dots \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \dots \quad (4)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \dots \quad (5)$$

$$h_t = o_t * \tanh(C_t) \dots \quad (6)$$

where W is the weight matrix, b is the bias vector parameter, o_t is the activation vector of output vector, $\tilde{C}_t$ is the cell input activation vector, C_t is the cell state vector, and h_t is the hidden state vector [60].

Similar to BiLSTM, bidirectional GRUs predict the current state of a sequence using the information preserved from both the previous and successive time steps. Unlike LSTMs, GRUs are equipped with only two gates: an update gate and a reset gate. Information is transferred using the hidden state, and no memory unit is used. GRUs are more efficient and train faster on less amounts of data giving results on par with those of LSTMs. Figure 6 shows the architecture of the Bidirectional GRU or Bi-GRU and the operations happening in every cell.

Figure 6. (a) Architecture of the Bi-GRU neural network. (b) Functions inside a single GRU cell.

The operations that happen inside a GRU cell are:

Reset Gate:

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r) \dots \quad (7)$$

Update Gate:

$$z_t = \sigma(X_t W_{xz} + H_{t-1} w_{hz} + b_z) \dots \quad (8)$$

where W_{xr} , w_{xz} and W_{hr} , w_{hz} are weight parameters, whereas b_r and b_z are biases.

Hidden State:

$$\tilde{H}_t = \tanh(x_t W_{xh} + (R_t \odot H_{t-1} W_{hh}) + b_h) \dots \quad (9)$$

Update State:

$$H_t = Z_t \odot H_{t-1} + (1 - z_t) \odot \tilde{H}_t \dots \quad (10)$$

The Reset gates capture short-term sequences, whereas the update gates capture long-term sequences.

3.4. Architecture of the System-Level Module

3.4.1. Data Collection Unit

The application is deployed as an extension to messaging applications, which allows it to read and collect user data. For every 3 h, the user's communication data are collected and stored as a simple text file with each sentence separated by a newline character. Non-alphabetic characters and HTML content are removed from each sentence of the collected data, thus obtaining clean sentences. The cleaned sentences are tokenized into an array of tokens or words and one-hot encoded into binary vectors. These vectors act as the input for the deep learning model CASE. CASE processes the input vectors and determines whether each sentence consists of abusive content. After classifying all the entities, they are initialized as a training dataset for the deep learning model to improve its classification metrics and update its weight matrix. Once detected, the respective abusive sentences are labelled with a corresponding category. If CASE predicts the presence of an abusive class of text with a probability of more than 0.8, it is fed into the dataset for training. The 0.8 cutoff for the collection of future training data is to ensure that clean data that have been correctly identified are used for training purposes. After the CASE model is trained, these entities are completely deleted from the MaLang application, making space for the next iteration of collected textual data. The entire process of cleaning, tokenizing, and training data with CASE takes an average time of 15 min.

3.4.2. Deep Learning Model for the Classification of Abusive Content

To identify abusive textual content, the Hate-speech dataset of [61] was considered. The dataset contains more than 28,000 sentences that were categorized into Normal, Offensive, and Identity-hate. The dataset was modified into two categories, i.e., abusive and non-abusive content. All the sentences were converted into lowercase and cleaned by removing non-alphabetic characters and the HTML content present in them; 25,000 sentences were used as the training set, and 3000 sentences were used for the testing set.

All the sentences were tokenized using the NLTK package [62] and were cropped to uniform length. These were converted into an array of words for each sentence called entities. Simultaneously, every word from each entity was one-hot encoded, thus obtaining a matrix representing categorical variables as binary vectors. The vectors act as an input to the embedding layer in the CASE deep learning model.

Figure 7 shows the architecture of the CASE deep learning model. The system-level deep learning model consists of one embedding layer with an input dimension of 21,000 and an output dimension of 300. The input length sequence is 53. The embedding layer is followed by a Bidirectional LSTM layer with 128 cells and a one-dimensional global Max Pooling layer. These are connected to two sets of a fully connected layer and Dropout layer. Finally, a SoftMax function is used as an output layer to predict the probabilities of either of the classes. The model is used to determine whether the sentences collected contain any abusive content. The usage of Bi-LSTM over a traditional RNN such as LSTM provides a solution to two traditional problems associated with abusive text detection:

1. When dealing with abusive sentences, many phrases might not contain a foul word; however, the sentence might be abusive or offensive when its meaning is considered. Bi-LSTM was chosen as it accesses past information from past words and the future information in the reverse direction, therefore attempting to understand the context and overall meaning of the sentence.
2. Because CASE is deployed locally on the user's mobile device, it should only use minimum storage space and processing power. The current implementation requires 8 megabytes (MB) of storage space. Although transformers fulfill the purpose

of understanding the context, they are known to occupy a huge amount of space and computation resources, which is practically infeasible for this purpose. The transformer type of deep learning used in prior research can use up to 200 MB of storage space.

Figure 7. Architecture of CASE using Bi-LSTM and the Max-Pooling Layer.

The application generates a user identification tag. The tag consists of a randomly generated 5-digit text that consists of both alpha and numeric characters. The tag is unique to only one user. After the entities are classified into abusive and non-abusive, the abusive entities are linked with the user identification tag and sent to the cloud-level module. Similarly, all such entities from multiple devices or users are sent to the cloud-level module along with the tagged user's unique identification number.

3.5. Architecture of the Cloud-Level Module

3.5.1. Data Collection Unit

The cloud-level module receives the tagged entities. These entities are received every 3 h from multiple users. All the entities that belong to a single tag are extracted into an array of tokens of uniform length for each sentence. These arrays of tokens are converted into input vectors of 32 indices, making them suitable as input to the embedding layer in the KIPP deep learning model.

3.5.2. Deep Learning Model for Identification of Toxicity of Abusive Content

The Jigsaw/Google Toxic Comment Dataset [61] is used to classify the toxicity in the text message content uploaded to KIPP into four classes: 'Toxic', 'Threat', 'Insult', and 'Hate Speech'. The dataset contains labelled examples of Wikipedia comments that were manually rated by human raters based on the comment's toxicity in six classes, which are 'Toxic', 'Severely Toxic', 'Threat', 'Obscene', 'Insult', and 'Hate-speech'. Although the dataset consisted of six classes, classes such as 'Toxic' and 'severely toxic' were merged since severely toxic is a subset of Toxic text, and classifying both as a single set seemed conceptually convincing. Similarly, the class 'obscene' was merged with 'Insult'.

The dataset consists of more than 157,000 sentences. To separate the dataset, a 60–20–20 split was considered for training–validation–testing, respectively. All the samples or sentences were cleaned by removing non-alphabetic characters and html content. The cleaned samples are tokenized using the Keras tokenizer, obtaining an array of tokens of unique words for each sentence. Subsequently, every token is cropped to uniform length in each array, obtaining an array of uniform tokens for each sample called entities.

The obtained entities are converted into input vectors of 32 indices. This is done by mapping tokens from entities to their corresponding vector representations present in pre-trained GloVe vectors [63]. Word embedding for each token is performed using the 300-dimension pre-trained global vector representation of word–word occurrences,

producing 840 billion tokens from the Massive Common Crawl Corpus with a vocabulary size of 2.2 million.

Some abusive words may not be present in MaLang's (or KIPP's) recognized vocabulary. When any missing word is encountered, it is given a value of zero, or if any token belongs to the substring of the main word in the vocabulary, the token is mapped to the root or main string vector. Mapping the token from entities to pre-trained GloVe vectors results in input vectors that are passed through the embedding layer of the KIPP deep learning model.

Figure 8 shows the architecture of the deep learning model KIPP. The cloud-level model consists of an embedding layer followed by a bidirectional LSTM layer and a bidirectional GRU layer with 128 cells in each layer. These layers are connected to a combination of a one-dimensional Average Pooling Layer and a Max Pooling Layer. The output layer consists of a sigmoid function to represent the output in the form of probabilities to detect the toxicity of the content in four classes. The use of Bidirectional GRUs and Bidirectional LSTM is solely for the purpose of handling single labelled entities sent to KIPP from mobile devices. KIPP receives a very large amount of unstructured data, which GRUs are known to handle and produce better results without monopolizing computational resources.

Figure 8. Architecture of KIPP using Bi-LSTM and Bi-GRU.

KIPP predicts the level of Toxicity in each entity and returns the probability of presence of the kind of toxic content present as shown in Figure 4. After all the entities are analyzed, a report is generated stating the average toxicity score of the user's content received at the cloud level of MaLang every 3 h.

3.6. Methodology to Calculate the Toxic Score of a User's Content

After predicting the probability of the presence of different types of toxic content in each entity, the entity toxic score (ET) is obtained by calculating the mean of the score of the different types of classes as shown below:

$$UT_k = \frac{\sum_{i=1}^n ET_i}{n} \text{ where } ET_i = \frac{P(T) + P(Th) + P(I) + P(H)}{4} \quad (11)$$

where n is the number of text entities for user k , $P(T)$ is the probability of the presence of Toxic content, $P(Th)$ is the probability of the presence of Threat content, $P(I)$ is the probability of the presence of Insulting content, and $P(H)$ is the probability of hate speech content, in each entity. Similarly, an entity toxic score is calculated for all entities that

belong to one user identification number. The highest entity scores are considered, and their average is calculated to obtain a User Toxic Score (UT).

Similarly, a User Toxic Score is calculated for all users in the network. A threshold of permitted User Toxic Score less than or equal to 0.4 is considered relatively benign and will not result in any user or administrator notification. When a user's Toxic Score exceeds 0.4, the user is prompted with a warning. In addition, a flag is raised in the area's cyber security patrol. Any abusive textual sentence, on average, scores an Entity Score of 0.6–0.8 when at least two classes of abusive text are present. The 0.4 value used to indicate toxic content is meant to allow for a certain level of profane language that may be used humorously or to relieve stress and is considered acceptable by the organization. The organization may elect to change this number, lowering it to make text monitoring more restrictive or raising the value in team cultures where a certain level of profanity is accepted if not expected.

The aim of MaLang is not only to detect such content but also to restrict and curb its use in the network among the users. Therefore, if the system detects a user crossing the threshold of 0.4 more than six times, a notification along with the report is sent to the nearby cyber forensic analyst to take any necessary action. The threshold of 6 is arbitrary and is meant to allow some level of inappropriate language, similar to the threshold value of 0.4 used to detect toxic text. Again, this value can be lowered to allow the organization to maintain much tighter control over potential abusive or toxic messaging and resulting cyberbullying, with a value of 0 being the most restrictive. The threshold counter resets every month. If MaLang is deployed in an organization, exchange of abusive content between two or more employees present in the network can easily be detected.

3.7. Methodology to Identify Users Involved in the Exchange of Identical Abusive Content

In an organization, every department or every team is connected to a dedicated cloud module with KIPP. As shown in Figure 9, various cloud modules are interconnected and are given access to share and compare the incoming abusive entities coming from users' devices. When the entities tagged along with the user's identification number are transmitted to the cloud module, the extracted arrays of tokens are compared with different arrays of tokens of other user's entities in the same cloud module or connected cloud modules. When an exact match is found wherein an array of tokens exactly matches another entity's array of tokens, it is concluded that there was an exchange of abusive content between the devices in the network. The toxicity is analyzed using KIPP, and a report is generated stating the level of toxicity in each sentence. In such cases, the organization's Human Resource department is notified along with the involved user's identification number to take necessary action. The developed methodology can potentially reduce workplace harassment.

3.8. Methodology to Update the Weight Matrix of CASE in Various Devices over the Network

After CASE has trained itself using user's data for more than five iterations in the devices, the model is evaluated and results including accuracy and area under the receiver operating characteristics curve (AUROC) score are produced. These results along with CASE's weight matrix are transmitted to the cloud module wherein a comparison is performed between the accuracies and AUROC scores of CASE received from different devices. The weight matrix that projects the highest accuracy and AUROC score is selected and sent to all the devices in the network to update their respective CASE's weight matrix. Another set of two iterations of training is performed to compare CASE's performance with the updated and previous weight matrix. In the case where no improvement is observed, the existing matrix is used as usual and the updated matrix is permanently deleted. This approach helps CASE learn the presence of new abusive content in one device and detect the same in other devices as well. Additionally, this strategy helps prevent the CASE learning models from becoming trapped in a local minima.

Figure 9. Graphical Representation of a Network of various cloud-level modules to detect the exchange of similar abusive content across a network or organization.

3.9. Metrics for Evaluating Deep Learning Models

The evaluation of the deep learning models CASE and KIPP is performed using metrics such as Precision, Recall, Accuracy, Loss, and the AUROC score and curve. To implement these metrics, the following basic concepts are defined:

True Positive (TP): Textual Content that belongs to the toxicity class and is predicted as belonging to the same class

True Negative (TN): Textual content that does not belong to the toxicity class and is predicted as not belonging to the same class.

False Positive (FP): Textual Content that does not belong to the toxicity class and is predicted as not belonging to the same class

False Negative (FN): Textual Content that belongs to the toxicity class and is predicted as not belonging to the same class.

The above defined concepts are used to compute the following metrics:

1. **Precision (or positive predictive value):** The ability of the model to identify the possible textual content from the input.

$$P = \left(\frac{TP}{TP + FP} \right) \quad (12)$$

2. **Recall (or sensitivity or true positive rate (TPR)):** The ability of the model to identify all the relevant abusive textual content from the predicted possible textual content.

$$R = \left(\frac{TP}{TP + FN} \right) \quad (13)$$

3. Accuracy: The ratio of correct predictions made by the model to the total number of predictions by the model.

$$\alpha = \left(\frac{TP + TN}{TP + TN + FN + FP} \right) \quad (14)$$

4. Selectivity (or specificity or true negative rate): The ability of the model to identify all non-abusive textual content from the predicted non-abusive content.

$$S = \left(\frac{TN}{TN + FP} \right)$$

5. AUROC: measures the ability of the model to distinguish between classes whereas the ROC curve is a probability curve where TPR (R) is plotted against the false positive rate (1 – S).

AUROC determines the accuracy of the model being able to predict the true values as true and false values as false. As the AUROC increases, the MaLang model becomes better at recognizing different types of toxic content.

4. Results

4.1. Validation of the System-Level Deep Learning Module CASE

The rate of learning of CASE was 0.001. The optimizer and loss function used were Adam and a categorical cross-entropy function, respectively. The model achieved an average accuracy of 92.2%. The average precision and recall of CASE were 92% and 91%, respectively. The average AUROC score of CASE was 0.977. The pattern of maintaining accuracy and measuring loss in classifying the textual content based on the defined class is shown in Figure 10.

(a) Model Accuracy of CASE

Figure 10. Cont.

(b) Model Loss of CASE

Figure 10. Performance analysis of the deep learning model CASE.

4.2. Validation of the Cloud-Level Deep Learning Module KIPP

Figure 11 shows the ROC curves that depict the performance of the KIPP deep learning model in predicting abusive text categorization values correctly. The rate of learning of KIPP was 0.001. The optimizer and loss function used were Adaptive Movement Estimation (Adam) and Binary Cross-entropy function, respectively. The KIPP model achieved an average accuracy of 98.21%. The average precision and recall of KIPP were 93% and 92%, respectively. The average AUROC score of KIPP was approximately 0.986. Figure 12 shows the KIPP toxic message type classification prediction accuracy and the corresponding loss function. Figure 13 shows the ROC curves that depict the performance of the KIPP model in accurately predicting the values of the different toxic message classes.

Figure 11. ROC Curves for Abusive and Non-Abusive predictions made by CASE.

(a) Model accuracy of KIPP

(b) Model Loss of KIPP

Figure 12. Performance analysis of the deep learning model KIPP.

Figure 13. ROC curves of the predictions made for different types of Toxic content.

4.3. Validation of the Decentralized Approach of MaLang

The system-level module classifies more than 98% of the data into abusive and non-abusive text messages for smartphones or personal computers. Post classification, only the abusive content is pushed to the cloud in the form of entities that contain an array of tokens that describe the sentence. Therefore, all of the user's data other than tokenized text messages are secure and are not transmitted to the cloud. Moreover, the tokenized version of data sent to the cloud, when accessed by a human, will not be able to make any sense, structurally or contextually, due to the lack of articles or stop words. Reframing the articles transmitted to the cloud back into a sentence is difficult and unlikely for a human. This would result in giving no exploitable meaning or value to the user's data in the cloud module.

Since 98% of a user's data are classified on the user's device, the cloud-level module consumes less storage space and computing time to identify the toxicity in the text. It is estimated that the consumption of computing resources and time can be reduced by 35–40% over prior cyberbullying detection systems by deploying MaLang.

5. Discussion

5.1. Comparison of MaLang with Existing State-of-the-Art Models

Table 3 depicts the performance comparison of MaLang compared against the best existing state-of-the-art models taken from Table 1, where best performing is based on the accuracy of overall predictions. Logistic regression is added as a reference value since this is a common method for traditional research. Other values of precision, recall, F-score, and AUROC values are provided when available in the previously reported research. Prior research utilized different data sources for analyzing cyberbullying detection systems and as such, only the reported performance measures are used for comparison.

Table 3. Performance comparison of MaLang with respect to existing state-of-the-art detection models.

Approach	Accuracy	Precision	Recall	F1-Measure	AUROC
MaLang	0.982	0.987	0.960	0.973	0.986
Text mining [22]	0.586				
NLP [7]	0.950				
Logistic Regression [64]	0.905	0.910	0.900		
Decision Tree [23]	0.785				
SVM [30]	0.780			0.780	
LSTM [38]	0.9397				
CNN [50]	0.960				
Deep Learning Ensemble [46]	0.980				

Tolba et al. [65] recently performed an independent comparison of various machine learning and deep learning approaches using various encoding schemas for the detection of online harassment (no age limitations specified). Tolba et al.'s findings report an optimal geometric mean, F1 score, and AUROC of: 0.8107, 0.8107, 0.7612; all for systems that utilized GloVe with either a Bi-LSTM or LSTM based deep learning classifier.

As may be seen in Table 3, the MaLang approach outperforms current existing natural language, parametric statistical, text analytics, and machine learning-based abusive language detection models. While ensemble deep learning methods come close to MaLang's performance, the decentralized approach of MaLang has the additional benefit of preserving data privacy for employees. The decentralized methodology of MaLang also makes this approach scalable to very large organization data networks because the CASE component runs on separate individual devices.

5.2. Limitation

A limitation of the MaLang approach is that corporations must install and maintain CASE on all corporate computing devices that may be used for text communications

and also requires that all employees install CASE on any of their personal devices that are brought to work. Fortunately, regardless of employee compliance on their personal devices, cyberbullying may still be detected via receipt of messages on corporate-owned communication endpoint devices that have CASE installed.

6. Conclusions

Electronic communications are becoming ever more ubiquitous in today's organizations as well as in general life. The ever-increasing use of electronic communication technologies has enabled and is increasing the occurrence of workplace cyberbullying and the exchange of other hostile or abusive communications [3]. Various attempts were previously made to utilize machine learning to try and identify abusive language in textual communications but are not being widely deployed due to system overhead or other business factors. In this paper, a new method for identifying and reacting to abusive text messages, called MaLang, is developed and tested.

The MaLang approach ensures data privacy by not sharing sensitive or raw data being transmitted to the cloud. With the increasing global emphasis on data privacy, MaLang's anonymization features could be critical adoption factors for organizations.

The Deep Learning models CASE and KIPP perform with an average accuracy of 92.2% and 98.21%, respectively. They achieved an average AUROC score of 0.977 and 0.986, respectively, in classifying abusive content. With the methodology presented in the current paper, it can be concluded that MaLang outperforms existing methods with respect to overall accuracy for detecting abusive textual content in a user's electronic communications using their device and over the business's network.

The MaLang framework can be deployed in an organization to detect the exchange of abusive content among employees, even across different departments. The methodology to improve the model by updating the weight matrix forms the basis of federated learning and helps multiple devices benefit from the mechanism of training one model on multiple datasets. The decentralized approach can be used to detect abusive textual content in the downloaded media on user's device.

A potential future upgrade to the framework would scale down the functionalities of Cloud-level classification to an edge platform preserving data securely and performing classification on mobile devices without pushing any data to the cloud. The approach can be scaled up to perform real-time text prediction without suggesting any abusive content to the user to type in. This could potentially decrease the usage of abusive or toxic content.

Author Contributions: Conceptualization, S.C.S. and P.K.; Methodology, S.C.S. and P.K.; Software, S.C.S., P.K. and M.V.C.; Validation, S.C.S. and S.W.; Formal Analysis, S.J.; Data Curation, S.J.; Writing—Original Draft Preparation, S.C.S.; Writing—Review & Editing, S.W.; Visualization, S.C.S. and S.W.; Supervision, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: <https://www.kaggle.com/mrmorj/hate-speech-and-offensive-language-dataset>, accessed on 30 August 2021.

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

References

1. Hamm, M.P.; Newton, A.S.; Chisholm, A. Prevalence and Effect of Cyberbullying on Children and Young People: A Scoping Review of Social Media Studies. *JAMA Pediatr.* **2015**, *169*, 770–777. [CrossRef]
2. Caplan, R.D.; Jones, K.W. Effects of Work Load, Role Ambiguity, and Type A Personality on Anxiety, Depression, and Heart Rate. *J. Appl. Psychol.* **1975**, *60*, 713–719. [CrossRef]

3. Farley, S.; Coyne, I.; D'Cruz, P. Cyberbullying at work: Understanding the influence of technology. In *Handbooks of Workplace Bullying, Emotional Abuse and Harassment—Concepts, Approaches and Methods*; D'Cruz, P., Noronha, E., Notelaers, G., Rayner, C., Eds.; Springer Nature: Berlin, Germany, 2021; pp. 233–263.
4. Oksanen, A.; Oksa, R.; Savela, N.; Kaakinen, M.; Ellonen, N. Cyberbullying victimization at work: Social media identity bubble approach. *Comput. Hum. Behav.* **2020**, *109*, 106363. [\[CrossRef\]](#)
5. Kowalski, R.M.; Toth, A.; Morgan, M. Bullying and cyberbullying in adulthood and the workplace. *J. Soc. Psychol.* **2017**, *158*, 64–81. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Oksa, R.; Saari, T.; Kaakinen, M.; Oksanen, A. The Motivations for and Well-Being Implications of Social Media Use at Work among Millennials and Members of Former Generations. *Int. J. Environ. Res. Public Health* **2021**, *18*, 803. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Al-Ajlan, M.A.; Ykhlef, M. Deep learning algorithm for cyberbullying detection. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 199–205. [\[CrossRef\]](#)
8. Banerjee, V.; Telavane, J.; Gaikwad, P.; Vartak, P. Detection of Cyberbullying Using Deep Neural Network. In Proceedings of the 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 15–16 March 2019; pp. 604–607.
9. Beauchere, J. Microsoft Study Shows Bullying Remains an Issue with 4 in 10 Teens Involved; Adults, Too. 2020. Available online: <https://blogs.microsoft.com/on-the-issues/2020/09/14/microsoft-online-bullying-study-covid-19/> (accessed on 18 January 2021).
10. Sadiq, S.; Mehmood, A.; Ullah, S.; Ahmad, M.; Choi, G.S.; On, B.-W. Aggression detection through deep neural model on Twitter. *Futur. Gener. Comput. Syst.* **2020**, *114*, 120–129. [\[CrossRef\]](#)
11. Singh, N.; Sinhasane, A.; Patil, S.; Balasubramanian, S. Cyberbullying Detection in Social Networks: A Survey. In Proceedings of the 2nd International Conference on Communication & Information Processing, Tokyo, Japan, 27 November 2020; pp. 8–13. [\[CrossRef\]](#)
12. Talpur, B.A.; O'Sullivan, D. Cyberbullying severity detection: A machine learning approach. *PLoS ONE* **2020**, *15*, e0240924. [\[CrossRef\]](#)
13. Van Hee, C.; Jacobs, G.; Emmery, C.; Desmet, B.; Lefever, E.; Verhoeven, B.; De Pauw, G.; Daelemans, W.; Hoste, V. Automatic detection of cyberbullying in social media text. *PLoS ONE* **2018**, *13*, e0203794. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Coyne, I.; Farley, S. Cyberbullying within working contexts. In *Cyberbullying at University in International Contexts*, 1st ed.; Cassidy, W., Faucher, C., Jackson, M., Eds.; Routledge: Abingdon, UK, 2018; pp. 80–96.
15. Noakes, T.; Noakes, T. Distinguishing online academic bullying: Identifying new forms of harassment in a dissenting Emeritus Professor's case. *Heliyon* **2021**, *7*, e06326. [\[CrossRef\]](#)
16. Coyne, I.J.; Farley, S.; Axtell, C.; Sprigg, C.A.; Best, L.; Kwok, O. Understanding the relationship between experiencing workplace cyberbullying, employee mental strain and job satisfaction: A disempowerment approach. *Int. J. Hum. Resour. Manag.* **2016**, *28*, 945–972. [\[CrossRef\]](#)
17. Loh, J.; Snyman, R. The tangled web: Consequences of workplace cyberbullying in adult male and female employees. *Gend. Manag. Int. J.* **2020**, *35*, 567–584. [\[CrossRef\]](#)
18. Cohen-Almagor, R. Social responsibility on the Internet: Addressing the challenge of cyberbullying. *Aggress. Violent Behav.* **2018**, *39*, 42–52. [\[CrossRef\]](#)
19. Tankovska, H. Mobile Messenger Apps—Statistics & Facts. 2021. Available online: <https://www.statista.com/topics/1523/mobile-messenger-apps/> (accessed on 18 February 2021).
20. Dadvar, M.; Trieschnigg, D.; Ordelman, R.; de Jong, F. Improving Cyberbullying Detection with User Context. In *European Conference on Information Retrieval*; Springer: Berlin, Germany, 2013; pp. 693–696.
21. Xu, J.M.; Zhu, X.; Bellmore, A. Fast learning for sentiment analysis on bullying. In Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining, Beijing, China, 12 August 2012; pp. 1–6.
22. Bayzick, J.; Kontostathis, A.; Edwards, L. Detecting the presence of cyberbullying using computer software. In Proceedings of the 3rd International Web Science Conference WebSci '11, Koblenz, Germany, 15–17 June 2011; pp. 93–96.
23. Reynolds, K.; Kontostathis, A.; Edwards, L. Using Machine Learning to Detect Cyberbullying. In Proceedings of the 10th International Conference on Machine Learning and Applications and Workshops, Honolulu, HI, USA, 18–21 December 2011; Volume 2, pp. 241–244.
24. Dinakar, K.; Reichart, R.; Lieberman, H. Modeling the detection of textual cyberbullying. In Proceedings of the International AAAI Conference on Web and Social Media, Barcelona, Spain, 17–21 July 2011; Volume 5, pp. 11–17.
25. Chavan, V.S.; Shylaja, S.S. Machine Learning Approach for Detection of Cyber-Aggressive Comments by Peers on Social Media Network. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, India, 10–13 August 2015; pp. 2354–2358.
26. Kumar, A.; Sachdeva, N. Cyberbullying Detection on Social Multimedia Using Soft Computing Techniques: A Meta-Analysis. *Multimed. Tools Appl.* **2019**, *78*, 23973–24010. [\[CrossRef\]](#)
27. Huang, Q.; Singh, V.K.; Atrey, P.K. Cyber Bullying Detection Using Social and Textual Analysis. In Proceedings of the 3rd International Workshop on Socially-Aware Multimedia, Orlando, FL, USA, 7 November 2014; pp. 3–6.
28. Agrawal, S.; Awekar, A. Deep Learning for Detecting Cyberbullying Across Multiple Social Media Platforms. *European Conference on Information Retrieval*; Springer: Cham, Switzerland, 2018; pp. 141–153.

29. Dinakar, K.; Jones, B.; Havasi, C.; Lieberman, H.; Picard, R. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Trans. Interact. Intell. Syst.* **2012**, *2*, 1–30. [\[CrossRef\]](#)
30. Zhao, R.; Zhou, A.; Mao, K. Automatic detection of cyberbullying on social networks based on bullying features. In Proceedings of the 17th International Conference on Distributed Computing and Networking, Singapore, 4–7 January 2016; pp. 1–6.
31. Chatzakou, D.; Kourtellis, N.; Blackburn, J.; De Cristofaro, E.; Stringhini, G.; Vakali, A. Mean Birds: Detecting Aggression and Bullying on Twitter. In Proceedings of the 2017 ACM on Web Science Conference, Troy, NY, USA, 25–28 June 2017; pp. 13–22.
32. Al-Garadi, M.A.; Hussain, M.R.; Khan, N.; Murtaza, G.; Nweke, H.F.; Ali, I.; Mujtaba, G.; Chiroma, H.; Khattak, H.A.; Gani, A. Predicting Cyberbullying on Social Media in the Big Data Era Using Machine Learning Algorithms: Review of Literature and Open Challenges. *IEEE Access* **2019**, *7*, 70701–70718. [\[CrossRef\]](#)
33. Rosa, H.; Matos, D.; Ribeiro, R.; Coheur, L.; Carvalho, J.P. A “Deeper” Look at Detecting Cyberbullying in Social Networks. In Proceedings of the 2018 International Joint Conference on Neural Networks, Rio, Brazil, 8–13 July 2018; pp. 1–8.
34. Hani, J.; Nashaat, M.; Ahmed, M.; Emad, Z.; Amer, E.; Mohammed, A. Social media cyberbullying detection using machine learning. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 703–707. [\[CrossRef\]](#)
35. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
36. Severyn, A.; Moschitti, A. Twitter sentiment analysis with deep convolutional neural networks. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 959–962.
37. Zhang, X.; Tong, J.; Vishwamitra, N.; Whittaker, E.; Mazer, J.P.; Kowalski, R.; Hu, H.; Luo, F.; Macbeth, J.; Dillon, E. Cyberbullying detection with a pronunciation based convolutional neural network. In Proceedings of the 15th IEEE International Conference on Machine Learning and Applications, Anaheim, CA, USA, 18–20 December 2016; pp. 740–745.
38. Bleiweiss, A. LSTM Neural Networks for Transfer Learning in Online Moderation of Abuse Context. In Proceedings of the 11th International Conference on Agents and Artificial Intelligence, Prague, Czech Republic, 19–21 February 2019; pp. 112–122.
39. Khieu, K.; Narwal, N. Detecting and Classifying Toxic Comments. 2017. Available online: <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184> (accessed on 30 July 2020).
40. Nguyen, H.; Nguyen, M.L. A deep neural architecture for sentence-level sentiment classification in twitter social networking. In *International Conference of the Pacific Association for Computational Linguistics*; Springer: Singapore, 2017; pp. 15–27.
41. Park, J.H.; Fung, P. One-step and Two-step Classification for Abusive Language Detection on Twitter. In Proceedings of the First Workshop on Abusive Language Online, Vancouver, BC, Canada, 17–30 August 2017; pp. 41–45.
42. Yu, L.C.; Wang, J.; Lai, K.R.; Zhang, X. Refining word embeddings for sentiment analysis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 17 September 2017; pp. 534–539.
43. Cécillon, N.; Labatut, V.; Dufour, R.; Linarès, G. Abusive Language Detection in Online Conversations by Combining Content- and Graph-Based Features. *Front. Big Data* **2019**, *2*, 8. [\[CrossRef\]](#) [\[PubMed\]](#)
44. Safaya, A.; Abdullatif, M.; Yuret, D. KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media. In Proceedings of the Fourteenth Workshop on Semantic Evaluation, Barcelona, Spain, 12–13 December 2020; pp. 2054–2059.
45. Saha, P.; Mathew, B.; Goyal, P.; Mukherjee, A. Hatemonitors: Language agnostic abuse detection in social media. In Proceedings of the Working Notes of FIRE 2019—Forum for Information Retrieval Evaluation, Kolkata, India, 12–15 December 2019; Volume 251, pp. 246–253.
46. Markoski, F.; Zdravevski, E.; Ljubešić, N.; Gievska, S. Evaluation of Recurrent Neural Network architectures for abusive language detection in cyberbullying contexts. In Proceedings of the 17th International Conference on Informatics and Information Technologies, Virtual, 8–10 May 2020.
47. Chen, H.Y.; Li, C.T. HENIN: Learning Heterogeneous Neural Interaction Networks for Explainable Cyberbullying Detection on Social Media. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Virtual, 16–20 November 2020; pp. 2543–2552.
48. Muneer, A.; Fati, S.M. A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter. *Future Internet* **2020**, *12*, 187. [\[CrossRef\]](#)
49. Dadvar, M.; Eckert, K. Cyberbullying detection in social networks using deep learning based models; a reproducibility study. In Proceedings of the 2nd International Conference on Computational Intelligence and Intelligent Systems, Phuket, Thailand, 17–19 November 2018; pp. 17–21.
50. Lu, N.; Wu, G.; Zhang, Z.; Zheng, Y.; Ren, Y.; Choo, K.R. Cyberbullying detection in social media text based on character-level convolutional neural network with shortcuts. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e5627. [\[CrossRef\]](#)
51. Yao, M.; Chelms, C.; Zois, D.S. Cyberbullying ends here: Towards robust detection of cyberbullying in social media. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3427–3433.
52. Rosa, H.; Pereira, N.; Ribeiro, R.; Ferreira, P.; Carvalho, J.P.; Oliveira, S.; Coheur, L.; Paulino, P.; Simão, A.V.; Trancoso, I. Automatic cyberbullying detection: A systematic review. *Comput. Hum. Behav.* **2018**, *93*, 333–345. [\[CrossRef\]](#)
53. Singh, V.; Varshney, A.; Akhtar, S.S.; Vijay, D.; Shrivastava, M. Aggression detection on social media text using deep neural networks. In Proceedings of the 2nd Workshop on Abusive Language Online, Brussels, Belgium, 31 October–1 November 2018; pp. 43–50.

54. Vishwamitra, N.; Zhang, X.; Tong, J.; Hu, H.; Luo, F.; Kowalski, R.; Mazer, J. MCDefender: Toward effective cyberbullying defense in mobile online social networks. In Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics, New York, NY, USA, 24 March 2017; pp. 37–42.
55. Zhou, P.; Qi, Z.; Zheng, S.; Xu, J.; Bao, H.; Xu, B. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. In Proceedings of the 26th International Conference on Computational Linguistics COLING 2016, Osaka, Japan, 13–16 December 2016; pp. 3485–3495.
56. Aluru, S.S.; Mathew, B.; Saha, P.; Mukherjee, A. A Deep Dive into Multilingual Hate Speech Classification. In *European Conference on Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track Proceedings, Part V*; Springer: Cham, Switzerland, 2021; pp. 423–439.
57. Liu, P.; Qiu, X.; Huang, X. Recurrent neural network for text classification with multi-task learning. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16), New York, NY, USA, 9–15 July 2016; pp. 2873–2879.
58. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; Volume 8, pp. 1310–1318.
59. Arras, L.; Arjona-Medina, J.; Widrich, M.; Montavon, G.; Gillhofer, M.; Müller, K.R.; Hochreiter, S.; Samek, W. Explaining and interpreting LSTMs. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*; Springer: Cham, Switzerland, 2019; pp. 211–238.
60. Alshalan, R.; Al-Khalifa, H. A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere. *Appl. Sci.* **2020**, *10*, 8614. [\[CrossRef\]](#)
61. Kaggle.com. Toxic Comment Classification Challenge. 2018. Available online: <https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data> (accessed on 30 July 2020).
62. Bird, S. NLTK: The Natural Language Toolkit. In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, Sydney, Australia, 17–18 July 2006; pp. 69–72.
63. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
64. Davidson, T.; Warmley, D.; Macy, M.; Weber, I. Automated hate speech detection and the problem of offensive language. *Proc. Int. AAAI Conf. Web Soc. Med.* **2017**, *11*, 512–515.
65. Tolba, M.; Ouadfel, S.; Meshoul, S. Hybrid ensemble approaches to online harassment detection in highly imbalanced data. *Expert Syst. Appl.* **2021**, *175*, 114751. [\[CrossRef\]](#)