9-1-2022

Using deep learning to detect social media ‘trolls’

Áine MacDermott
Liverpool John Moores University

Michal Motylinski
Liverpool John Moores University

Farkhund Iqbal
Zayed University, farkhund.iqbal@zu.ac.ae

Kellyann Stamp
Liverpool John Moores University

Mohammed Hussain
Zayed University, mohammed.hussain@zu.ac.ae

See next page for additional authors

Forensic Science International: Digital Investigation 43 (2022) 301446

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2022 APAC - Proceedings of the Second Annual DFRWS APAC

Using deep learning to detect social media 'trolls'

Áine MacDermott [a, *], Michal Motylinski [a], Farkhund Iqbal [b], Kellyann Stamp [a], Mohammed Hussain [b], Andrew Marrington [b]

[a] School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool, UK
[b] Zayed University, Dubai, United Arab Emirates

ARTICLE INFO

Article history:

Keywords:
Data mining
Digital forensics
Machine learning
Social media
Toxic data

ABSTRACT

Detecting criminal activity online is not a new concept but how it can occur is changing. Technology and the influx of social media applications and platforms has a vital part to play in this changing landscape. As such, we observe an increasing problem with cyber abuse and 'trolling'/toxicity amongst social media platforms sharing stories, posts, memes sharing content. In this paper we present our work into the application of deep learning techniques for the detection of 'trolls' and toxic content shared on social media platforms. We propose a machine learning solution for the detection of toxic images based on embedded text content. The project utilizes GloVe word embeddings for data augmentation for improved prediction capabilities. Our methodology details the implementation of Long Short-term memory Gated recurrent unit models and their Bidirectional variants, comparing our approach to related works, and highlighting evident improvements. Our experiments revealed that the best performing model, Bidirectional LSTM, achieved 0.92 testing accuracy and 0.88 inference accuracy with 0.92 and 0.88 F1-score accordingly.

1. Introduction

Practically every type of crime now involves some aspect of digital evidence. Digital forensics provides various techniques and tools that can help articulate this evidence in legal proceedings (Liu et al., 2017) (MacDermott et al., 2020). For example, social media data encompasses many social networks and applications, constituting a large part of forensic evidence in cases (Powell et al., 2020). According to a recent study by Smart Insights (Chaffey, 2022), "*Social media users are now spending an average of 2 h and 24 min per day multi networking across an average of 8 social networks and messaging apps*."

There is an increased need to gather and understand evidence, information, and intelligence from a wide variety of digital sources. Phones and other mobile devices provide valuable information such as communication data, geolocation information, a person's associations, and timeline evidence. Devices containing a range of new sensor types, such as RFID, GPS location tracking, health monitoring etc. are leading to improved information on individual user behaviour on their respective devices (Conti et al., 2012) (Quick and Choo, 2017). Although forensic analysis of specific devices is not outlined in detail in this article, notable examples include Android (Lwin et al., 2020) and iOS forensics (Huang et al., 2019), drone forensics (Al-Room et al., 2021) wearable devices (MacDermott et al., 2019) and smart home devices (Chung et al., 2017), (Li et al., 2019) - all of which could have potential social media data residing within them.

The use of artificial intelligence (AI) and machine learning in digital forensics has a key role in extending and advancing the capabilities of currently used tools, yet there is a need to prepare for approaches that can handle even larger variations of data sizes and attributes. Forensic acquisition of data from digital devices – particularly social media content – has received increasing interest in academic literature, due to the vast nature of platforms and content available (Quick and Choo, 2017) (Aghababaei and Makrehchi, 2017; Quick and Choo, 2018; Arshad et al., 2019; Iqbal et al., 2019). The large quantities of data contained within social media profiles can often be vital to an investigation, whether it is law enforcement, criminal defense, or internal investigation within a business. Digital media investigations can identify the obscure digital evidence that can be crucial to a case, involving techniques for social media analysis (including posts, comments, messages, pictures, videos) utilizing open-source research investigations, conducting Wi-Fi surveys, IP address identification and analysis, etc

* Corresponding author.
E-mail address: a.m.macdermott@ljmu.ac.uk (MacDermott).

https://doi.org/10.1016/j.fsidi.2022.301446

(Boast and Harriss, 2016). Increasingly, big data analytics and historical analyses of social networking/digital fingerprints will be needed to build a clearer picture of an individual's life.

With the growing use of social media, we observe an increasing problem with cyber abuse and online toxicity. The term "online toxicity" encompasses rude, aggressive, and degrading attitudes and behavior that can be exhibited on online platforms (also known as being a 'troll' or 'trolling'). It can range from excessive use of profanity to outright hate speech and can be observed in the context of online interactions between one or more individuals. For example, recent studies show a significant rise in toxicity concerning the coronavirus pandemic and vaccine. The anti-mask and pro-mask conversations quickly divert from constructive discussions to disrespectful exchanges. Similar trends are also observable in 'charged' topics such as politics, sport, education etc (Salminen et al., 2020; Pascual-Ferrá et al., 2021).

Moreover, it is increasingly common that toxic messages are embedded in images and then shared online via an array of messaging applications and platforms, e.g., online platform stories, posts, memes. This surge in sharing negative and hateful images is a current problem for many social media platforms including Facebook, Reddit, Twitter, and Discord. It can be difficult to detect negative sentiment in the comments, especially when the message includes an image (Mielly, 2017; Bhattacharya, 2019). We cannot assume that 'negative language' (i.e., swearing) infers toxicity, but similarly, we cannot assume the absence of swearing or inappropriate language indicates that there is no toxicity. For example, images and text uploaded to social media can often be taken out of context especially when they concern recent news, e.g., political issues and events. Images are often ambiguous in indicating what a person had in mind. Sharing such vague messages leads to misunderstandings and arguments. Detection of hateful content must include extraction of the embedded content and subsequent analysis of said content. This, however, cannot be reduced to a simple keyword search.

However, machine learning has been used for sentiment analysis with great success. Deep learning models can detect patterns and provide accurate sentiment detection which has been confirmed by multiple studies in the field. The key is to apply the model to the extracted text content to analyze if the context of the message is insulting to a person or a group of people (Zhang and Zheng, 2017; Poornima and Priya, 2020; Maipradit et al., 2019).

Therefore, the focus and contributions of this paper are out as follows: we first present a review of similar works covering the method of data processing and machine learning approaches in Section 2. Section 3 provides a description of our methodology and experiments for detecting 'trolls' and toxic content. In Section 4 we cover the results of the developed machine learning models. In Section 5 we present our conclusions to the above analyses and discuss possible future applications. The novel contribution of this study is the ability of the developed framework to accurately extract and classify text from images that can be attached to messages sent online. The framework can be used to train models on different data and labels. This work aims to validate a deep learning approach in the field of digital forensics, specifically focusing on embedded device forensics. Our research indicates a need to continuously enhance our forensics preparedness to account for the addition of new content types which may occur in future.

2. Related work

In recent years there has been much effort to improve analysis of social media data and the information contained within the posts. Text embedded images are being sent at an increasing rate on various social media platforms. From a forensic perspective, this poses a new challenge as text must first be extracted from the image and then undergo a process of analysis of its content. Currently, there are three methods being employed for the analysis of image content. The first method involves extraction of the embedded text and its' analysis. The second approach utilizes neural networks for the analysis of the content of the image in search of patterns. The third option is a hybrid approach which involves the use of both techniques to increase the prediction confidence [25. 26]. Our research is focused on the utilization of embedded text for image classification thus further discussion will cover a review of similar works on this subject. The crucial part of the embedded text image classification is the extraction of the content. Tesseract is one of the most used optical character recognition (OCR) engines that can be used to extract text from images. In (Cao et al., 2019), the authors employed the Tesseract engine to detect text on a book spine for a book inventory system. The experiments showed that the developed solution had 90% detection accuracy. Ravindran (Ravindran et al., 2019) integrated Tesseract with the underlying deep neural network (DNN) architecture for detection of text on the traffic signals. The methodology involved the implementation of the Faster R−CNN Inception V2 model for balanced good accuracy and a reasonable processing time of the image. The Tesseract module was used to reduce the error rate of the implemented deep learning model. The results indicate that the developed model was able to successfully identify text on the traffic signs.

Interpretation of the extracted text can be performed using machine learning and deep learning techniques. The machine learning models are used for various tasks concerning natural language processing (NLP). One of the most common uses is sentiment detection. In (Zhang and Zheng, 2017), the authors conducted sentiment analysis applying Support Vector Machine (SVM) and Extreme Machine Learning (Huang et al., 2004). The results show that ELM performance was better however both models were able to successfully classify over 88% of the tested texts. In (Maipradit et al., 2019) N-Gram inverse document frequency was explored for feature extraction. The method was tested on various publicly available datasets using the SVM model for sentiment detection. The results indicate that the applied method is superior to other techniques it was compared to including NLTK, Stanford CoreNLPSO or SentStrength.

Sulke (Sulke et al., 2019) proposed the use of standard machine learning algorithms for the classification of online toxic comments. Their experiments were conducted on Google's dataset consisting of 6 different types of toxicity (Jigsaw, 2022). The research involved the application of Logistic Regression, K-Nearest Neighbour, SVM and Decision Tree adopted for multi-label classification problems. The authors used binary relevance and classifier chains methods to transform the multi-label problem into a binary classification task. The results comparison indicates that the most robust results are achieved using binary relevance in combination with the SVM classifier. The best-trained model achieved 98.97% accuracy and a 97% f1-score. While overall accuracy was very high it is important to note a significant difference in metric scores between binary classes. The class "0" metrics of 0.99 and class "1" of 0.76 may indicate a bias towards the former class. The majority of the comments in the dataset are not toxic which could have impacted the performance of the classifiers. Rahul et al. (2020), conducted similar research using the same dataset. The pre-processing stage involved stemming, lemmatization and removal of stop words and punctuation marks. They analyzed the comments according to their length and used it as a threshold to remove lengthy comments for better results. Six algorithms were used in the experiments namely Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, K-nearest neighbour, and SVM classifiers. The results of the study

indicate that Logistic Regression achieved the lowest hamming loss score (2.43) and the highest accuracy (89.46%), while Random Forest had the lowest log-loss of 0.58.

Anand (Anand and Eswari, 2019) proposed the use of deep learning methods for the classification of toxic comments. The authors implemented a word embeddings model GloVe in their pre-processing stage to improve the accuracy of the solution. The GloVe is a model for distributed word representation that obtains vector representation of each word from a text corpus. The model uses relations between different words to produce global and local statistics of a corpus which results in linear substructures that improve the model's prediction capabilities. The authors compared the performance of models with and without word embeddings. The experiments were conducted using a standard neural network, a network with a convolutional layer and a Long-Short Term Memory layer. The results indicate that the model utilizing convolutional layer and word embeddings performed had the best results, however, all models had a very similar performance (~98% accuracy).

In (Ibrahim et al., 2019), the authors analyzed the data imbalance problem of the toxicity dataset and proposed a novel solution. The study shows the entire dataset consists of less than 7% of toxic comments. The skewed distribution might create a bias towards a majority label; thus, augmentation is applied to the data. The methods applied include removal of duplicates from minority classes, creation of new comments using random 20% content of the original text and creation of new comments but with some words replaced for their synonyms. The method was tested using a convolutional neural network, Bidirectional LSTM and GRU models. The results indicate that each method provided an improvement over the not augmented data. The best F measure (0.88) was recorded utilizing the CNN ensemble model and data processed using all three aforementioned methods.

Husnain (Husnain et al., 2021) proposed a different approach to the pre-processing of toxic comments to address the classification problems. After the data cleaning process that involved removal of stop words, stemming and tokenization authors extracted features according to the word length. The analysis of the features indicates that bigrams or words composed of two tokens are giving better results. The created training set was tested on a binary classification problem of detection of toxicity in the text and a multi-label classification problem. The algorithms used included Logistic Regression, Naïve Bayes, and Decision Tree classifiers. The results presented show over 95% accuracy of all models in a binary task and ~90% accuracy in multi-label problems. The best performing model was Logistic Regression in both case scenarios.

Our analysis of related works indicates that appropriate pre-processing of the content of the comments is crucial for algorithms to achieve good accuracy. The incorporation of word embeddings and data balancing techniques into the pipeline may provide a significant performance increase. A good choice of machine learning models is also important as the deep learning approach provides better and more robust results. Furthermore, it will be necessary to choose appropriate and correct evaluation metrics. Accuracy while easy to interpret is unreliable especially for imbalanced classification. A high accuracy score tells us that model correctly classified toxic comments as toxic, however it does not take into consideration true negative values predicted by the model. To ensure the robustness of the solution other metrics will be considered.

3. Methodology for troll/toxicity detection

In this section our proposed approach is discussed including two parts of the proposed software solution. In production, the final solution would also require a separate module for scraping data from a specific social media profile. Depending on the targeted platform the results can be very different as different platforms allow for various data to be acquired via API.

Our approach includes the development of two modules: an image text extraction module and a text classification module. Fig. 1 presents a simplified process of prediction followed during this research. For extraction of the text from images we are using a Python-tesseract module that provides OCR support (Hoffstaetter, 2022). Tesseract is one of the most commonly used OCR engines that allows highly accurate extraction of characters from images.

While Tesseract can be used for character detection of other languages this research is focused on messages written in pure English. The text classification module implements a deep learning model developed for the detection of toxicity in the text embedded in the image. The models chosen for the process include recurrent neural networks: Bidirectional LSTM and Bidirectional GRU. Both architectures are considered state-of-the-art solutions for NLP problems.

3.1. Dataset

For training the toxic/hate speech/troll detection model, we use a publicly available dataset for toxicity classification. The dataset was created by the Conversation AI team for the purpose of an NLP challenge on Kaggle (Jigsaw, 2022). The dataset consists of 6 labels: toxic, severe toxic, obscene, threat, insult, and identity hate. The original data is already split into train and test sets, however, for the purpose of this research both sets are merged. Overall, there are 223549 comments and clean texts constitute the majority of the dataset - 201081. The toxic label is the most common and many comments are labelled as toxic only. A small group of 45 comments have all 6 toxic labels assigned. Table 1 presents the distribution of toxic labels in the dataset across all labels.

3.2. Label selection and uneven distribution

The original problem was a multi-label classification of comments, however, for the purpose of this research, only binary classification will be necessary. We have selected a toxic label as it contains the largest number of samples and aligns with the aim of this research of toxicity detection. The uneven distribution is a very common issue in machine learning. In fact, it is very difficult to find perfectly even datasets especially with thousands or millions of records. Depending on the scale of irregularity this can be a serious problem in some cases leading to very poor results of prediction. While some classifiers like decision trees, logistic regression and SVM can work with imbalanced data, they will most likely fail when there is a high disproportion of classes.

To tackle the problem of imbalanced attributes two methods can be employed: over-sampling and under-sampling. Application of the former technique requires instances of the under-represented data to be copied. Under-sampling on the other hand can be applied by deleting instances of the major class. It is generally advised to use oversampling on small datasets and under-sampling when there is a lot of data so removal of values will not have a negative impact on the model. For the purpose of this research, only under-sampling will be employed because clean comments constitute a majority of the dataset. This method will allow reducing bias towards not toxic comments.

3.3. Text processing

The first text cleaning method applied was the replacement of short versions of various words such as ‘s, ‘re, ‘ll, ‘d etc. Then all

Fig. 1. Toxicity prediction process.

Table 1
Data distribution across original toxicity levels.

Label	Number of comments
Toxic	21384
Severe toxic	1962
Obscene	12140
Threat	689
Insult	11304
Identity hate	2117

comments were converted to the lower case because we want to avoid capitalized words being treated differently by a model (as this could lead to a decline in accuracy). We then applied Tokenizer from the TensorFlow library. The tokenization process refers to the process of separating a piece of text into smaller parts. Characters, words, or sub-words can be assigned as tokens. We have decided to use word tokenization using default delimiter characters provided by the TensorFlow Tokenizer function. Various settings were tested; however, the best results were achieved with a maximum number of features set to 10000. Based on other research and our own experiments, we have decided to remove comments longer than 150 characters. The reason for this action was that long comments were increasing the processing and training time of the algorithms. Moreover, it was observed that the inclusion of the aforementioned texts did not provide any improvement over the results achieved when training without inclusion of long comments.

As a data augmentation technique, we implemented word embeddings. GloVe word embeddings are a standard vector used for many NLP problems. The results of Anand (2019) showed that semantic relationships between words provide improvement to the trained models.

3.4. Data split and model selection

The data was divided into three parts: training, validation, and test split. The test split consists of only 100 text samples that are then embedded on the images using a random set of fonts. The split is stratified to ensure equal representation of toxic and non-toxic comments. The remaining data is split using an 80:10:10 ratio for training, validation and test sets as our experimental results showed that the best results are achieved using this setting. As a result of processing the data used in the experiments constitutes 20942 training samples, 2617 validation comments and 2617 texts for testing. Every split contains an equal amount of toxic and non-toxic comment samples.

The proposed method involves the implementation of Recurrent Neural Networks (RNN). The first type of RNN that is considered for this task is Long short-term memory (LSTM). LSTM is commonly used for processing sequential data achieving state-of-the-art performance. LSTM was developed to mitigate vanishing and exploding gradient problems of RNN networks by introducing an additional output cell that has four gates. The forget gate decides what data should be dropped from memory units. Input gate decides what data should be accepted into the neuron. Update gate updates the memory, while the output gate returns the new long-term memory.

The other RNN that will be used in this project is the Gated Recurrent Unit (GRU). GRU is very similar to LSTM, however, it has only 2 gates: reset and updated. The former gate determines a combination of inputs with previous data. The update gate decides how much of the previous memory should be retained. Due to a simpler architecture GRU provides faster training times. While accuracy is usually slightly worse when compared to LSTM on some datasets GRU might over perform LSTM. Both models achieve state-of-the-art performance on time series and NLP data thus, it is expected that both will yield similar accuracy scores.

While regular RNN networks learn from left to right-hand side, recent developments of neural networks allow training to be performed in two directions. The characteristic of this type of network is that there are two sequences that are considered. The input is being fed in two directions backwards and forward. This means that essentially two models are being trained. In many cases, it provides additional context to the network and allows the model to learn more from the combined pieces of information. Bi-directional RNN's excel in speech recognition and NLP thus, they might be a better solution for this task. Fig. 2 presents the architecture of a Bidirectional RNN that can be applied to both LSTM and GRU networks.

The implementation of a model training pipeline starts from an embedding layer that transforms input data into a vector representation. The created word embeddings are then passed to the

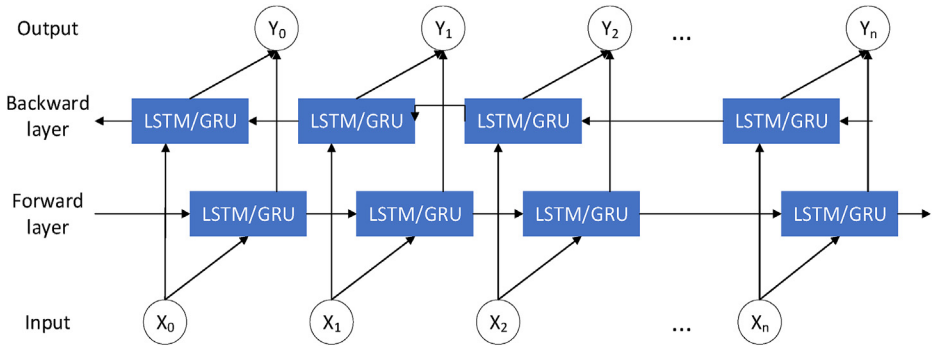

Fig. 2. Bidirectional RNN architecture.

RNN layer consisting of 128 units. The resulting tensor is passed through the Global Max Pooling layer which converts multi-dimensional input into a single-dimensional object. The resulting data is then passed to two dense layers with 64 and 16 nodes accordingly. Both fully connected layers utilize the Relu activation function that provides improvement over sigmoid and hyperbolic tangent functions. Dropout of 0.1 is applied after each dense layer. The output layer returns a single value that is an estimation of a processed text being toxic.

3.5. Evaluation metrics

The performance of each classifier is evaluated using four metrics: accuracy, precision, recall and F1-score. The accuracy is defined as a fraction of predictions that were correct. Accuracy provides general feedback about the model's performance; however other metrics should be used as accuracy score is negatively impacted by a severe class imbalance. The formula for accuracy is presented in (1). A True Positive (TP) is an outcome where the model correctly predicts the positive class. Similarly, a True Negative (TN) is an outcome where the model correctly predicts the negative class. A False Positive (FP) is an outcome where the model incorrectly predicts the positive class. A False Negative (FN) is an outcome where the model incorrectly predicts the negative class.

$$\text{Accuracy} = \text{TP} + \text{TN} \,/\, \text{TP} + \text{TN} + \text{FP} + \text{FN} \quad (1)$$

Precision is a metric that determines the number of correctly predicted positive values. This metric is especially useful for binary detection of toxic messages as the solution should not flag normal messages as toxic. To calculate precision (2), the score of all true positive values should be divided by the sum of true positive and false positive predictions.

$$\text{Precision} = \text{TP} \,/\, \text{TP} + \text{FP} \quad (2)$$

Recall measures how many positive values were predicted as such by the model. The recall is defined as follows:

$$\text{Recall} = \text{TP} \,/\, \text{TP} + \text{FN} \quad (3)$$

F1-score or F-measure is a metric that is a function of precision and recall. The calculated balance is a harmonic mean of the model's precision and recall that is defined as follows:

$$\text{F1} = 2*(\text{Precision}*\text{Recall}/\text{Precision} + \text{Recall}) \quad (4)$$

The model performance during training was assessed based on training accuracy, loss, as well as validation loss and accuracy. The validation loss metric was used for the early stopping function in order to avoid overfitting of the models.

4. Results

In this section the performance of the trained models is evaluated using metrics outlined in the methodology section. The experiments were performed using images with embedded text from the original dataset.

4.1. Training

The training of all models has been performed over 30 epochs with patience set at 5. Below are the training metrics for the best-trained models for all network types. Validation loss was used to ensure that training is stopped as soon as there is no improvement shown on a validation set. Fig. 3 presents the training metrics for a GRU model trained without a word embedding layer. While training lasted 23 epochs the last checkpoint was saved at epoch 18 as further training did not improve validation loss. The training loss was 0.5984 while the validation loss 0.6086. The GRU model obtained a 0.6825 training accuracy score and validation accuracy of 0.6693 at epoch 18 when the model was saved. In the case of LSTM, the training stopped at epoch 13. The model attained a training loss of 0.5965 and a validation loss of 0.6083. The training accuracy of the model was 0.6671 while validation accuracy was 0.6532 when the model was saved. Fig. 4 shows the model training history.

In Fig. 5 and Fig. 6 the training of the Bidirectional versions of GRU and LSTM models is illustrated. The training of both models takes a similar time to their regular versions. The last checkpoints were saved at epoch 16 and 19 accordingly. The Bi-GRU model achieved a training loss of 0.5852 and a validation loss of 0.6014. The training and validation accuracy of the model were 0.6754 and 0.6547 accordingly. In the case of Bi-LSTM, the training loss obtained was 0.5789 while validation loss was 0.6068. The best model that was trained achieved 0.6903 training and 0.6610 validation accuracy. The application of GloVe embeddings visibly improves the performance of all models. Figs. 7 and 8 show the training history of regular GRU and LSTM models with the addition of an embedding layer. The significant change is the number of epochs necessary to train both models decreased. The training loss and accuracy of the GRU model were 0.1950 and 0.9280 while the validation process achieved a loss of 0.2079 and an accuracy of 0.9192. The LSTM model with applied embedding layer obtained a training loss of 0.1607 and an accuracy of 0.9329. The validation metrics were 0.1903 and 0.9239 accordingly.

The last type of trained models involved Bidirectional versions of GRU and LSTM with the implementation of an embedding layer for improved predictions. Figs. 9 and 10 illustrate the training

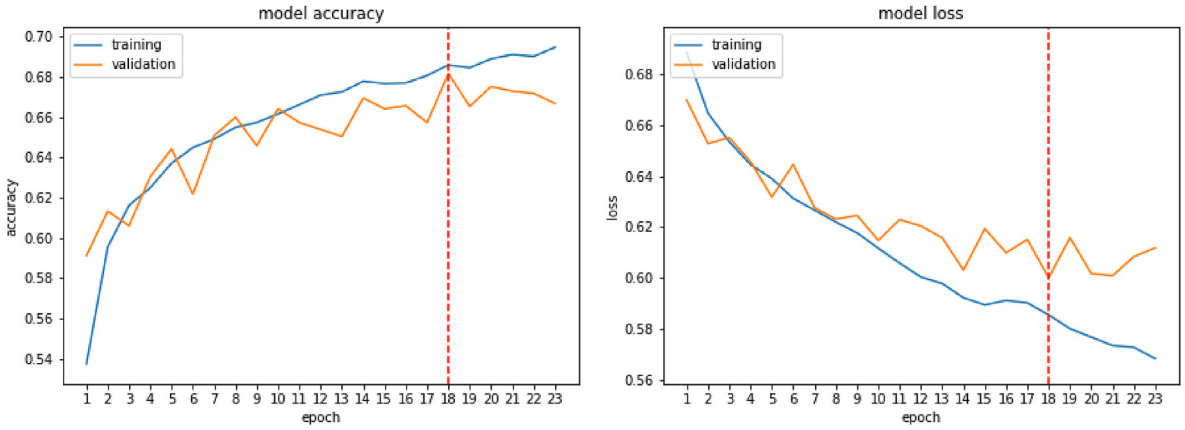

Fig. 3. GRU training accuracy (left) and loss (right).

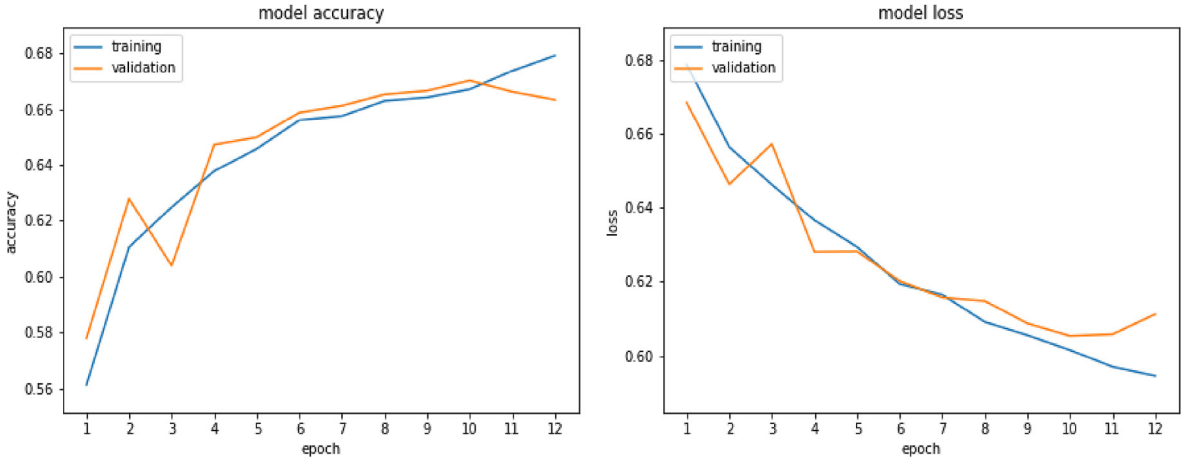

Fig. 4. LSTM training accuracy (left) and loss (right).

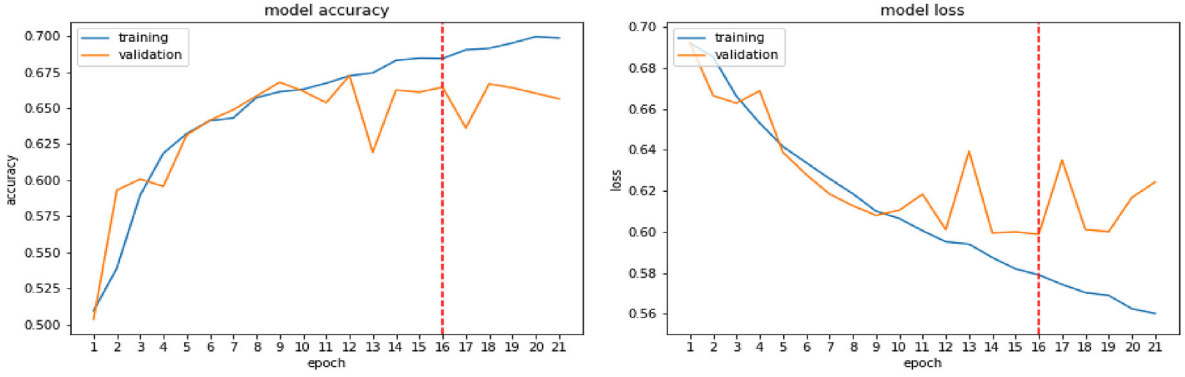

Fig. 5. Bidirectional GRU training accuracy (left) and loss (right).

process of both models that took 8 epochs while the last checkpoints were saved at epoch 3. The Bi-GRU training stopped when the model achieved a training loss of 0.1587 and an accuracy of 0.9372. The validation results were 0.1932 and 0.9226 accordingly. In the case of the best Bi-LSTM model, the training loss was 0.1686 while accuracy was 0.9353. The validation metrics were 0.1998 for loss and 0.9232 for accuracy. We can see that overfitting starts quickly on the models with embedding layer. To avoid it we are saving a checkpoint at the time when the last improvement was seen on the validation set while the training continues for the next 5 epochs. The red dashed line on each image show an epoch when the model was saved.

4.2. Testing and interference

The training results would indicate that the Bidirectional LSTM model with GloVe embeddings was the best performing one, however, it is necessary to test the models on unseen data to ensure

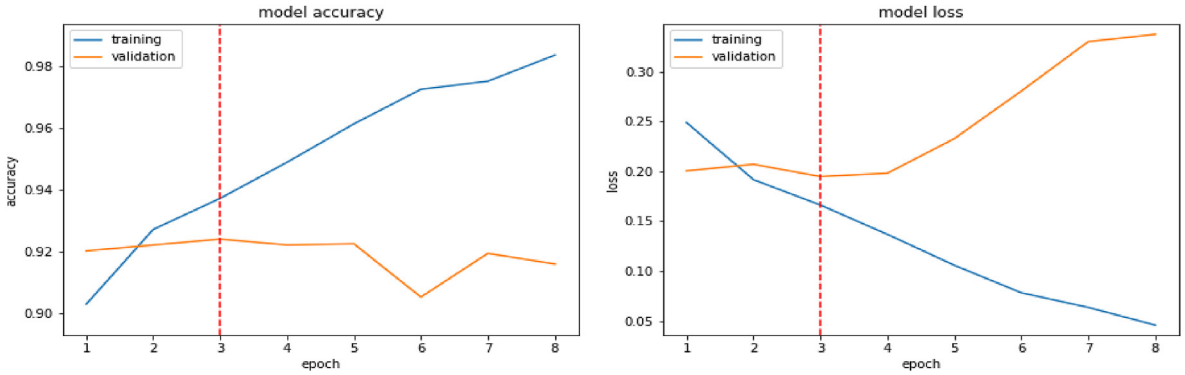

Fig. 6. Bidirectional LSTM training accuracy (left) and loss (right).

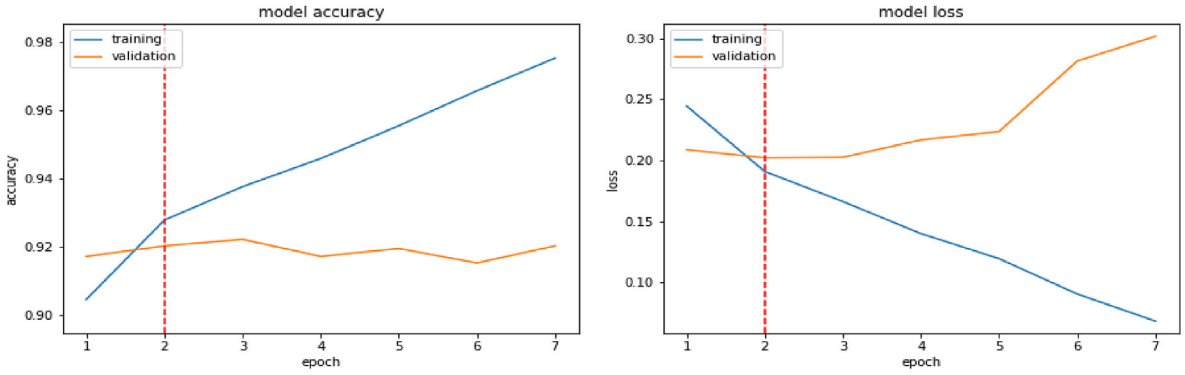

Fig. 7. GRU + GloVe embeddings training accuracy (left) and loss (right).

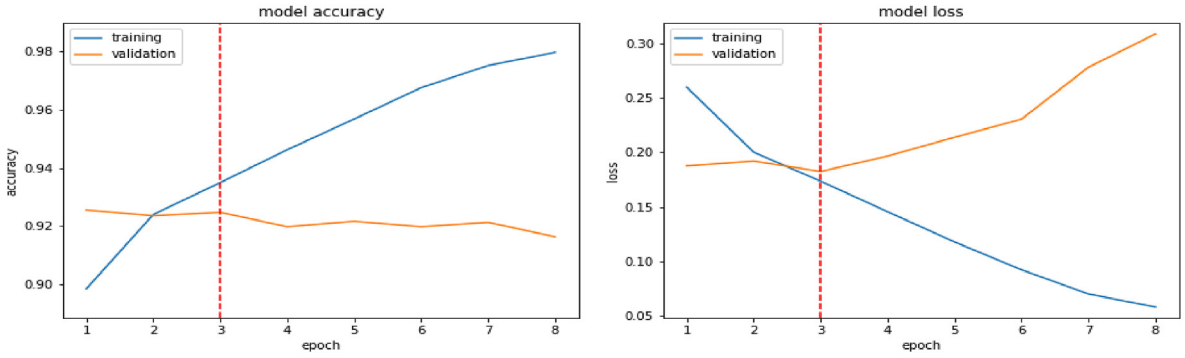

Fig. 8. LSTM + GloVe embeddings training accuracy (left) and loss (right).

unbiased results. The appropriate evaluation of the developed models required a two-stage process. The first models are tested on the selected set of 2617 comments. The comments are randomly selected during each iteration. The results from model testing are shown in Table 2. The training/test process was repeated 10 times and the final results are mean values of all iterations. The testing results show that the Bidirectional LSTM model with embeddings layer has the best performance with 0.921 accuracy and 0.922 F1-measure. The impact of word embeddings on the model's performance is very clear as the results improved by 30% with their implementation. All models with applied word embeddings achieved very good results with over 90% metric scores. Moreover, the number of epochs necessary to train the models dropped to 8.

The second stage of evaluation involved inference performed on the images with the comments from the test set embedded on the images. A tesseract module is used to extract the text from images. Fig. 11 presents a sample of an image with embedded comments and a resulting extracted content that is later passed to the classification module for interpretation. After extraction, all the comments undergo the same processing stages as data used for training. This involves cleaning and tokenization of the comment content. We utilize a saved tokenizer that was used during the training of a particular model.

Table 3 presents the inference results of all trained models. The majority of the results mirror results obtained in the prior test, however, a drop in accuracy is visible for all models. This is invariably tied to the quality of the extracted text from images. Some fonts make it difficult for the tesseract module to extract comments accurately. As a result, the prediction capability of the models is poorer. The reduction of accuracy to around 80% still

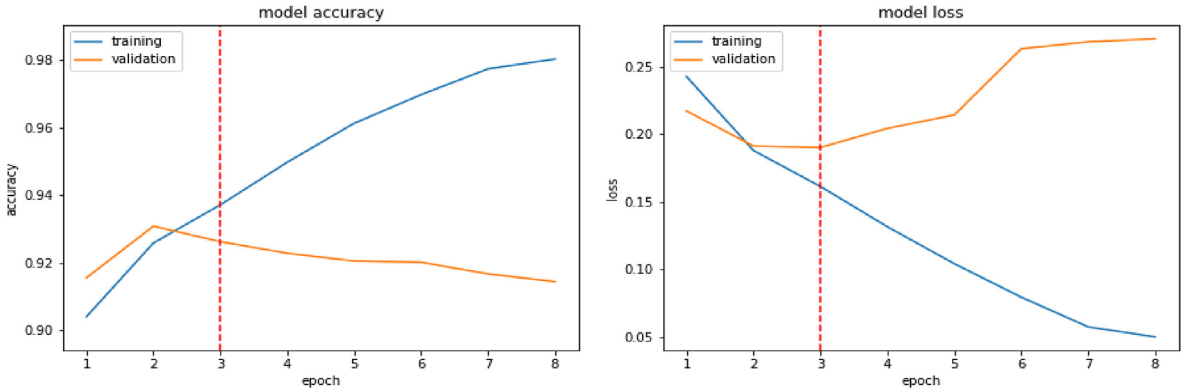

Fig. 9. Bidirectional GRU + GloVe embeddings training accuracy (left) and loss (right).

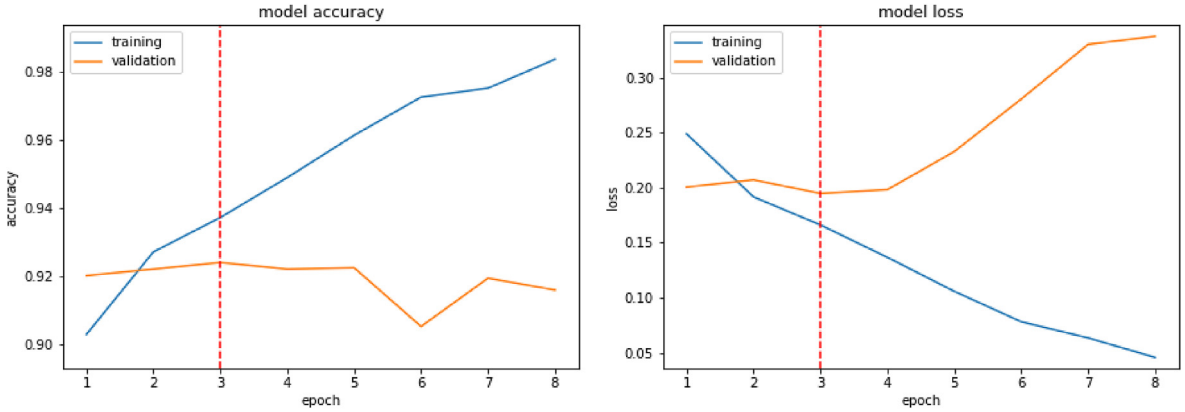

Fig. 10. Bidirectional LSTM + GloVe embeddings training accuracy (left) and loss (right).

Table 2
Model testing results.

Model	Accuracy	Precision	Recall	F1-score
GRU	0.662	0.746	0.493	0.593
LSTM	0.657	0.736	0.490	0.587
Bi-GRU	0.670	0.756	0.506	0.605
Bi-LSTM	0.673	0.767	0.497	0.602
GRU + GloVe	0.921	0.928	0.913	0.920
LSTM + GloVe	0.919	0.924	0.913	0.919
Bi-GRU + GloVe	0.917	0.931	0.900	0.915
Bi-LSTM + GloVe	0.921	0.918	0.925	**0.922**

Table 3
Model inference results.

Model	Accuracy	Precision	Recall	F1-score
GRU	0.615	0.445	0.679	0.537
LSTM	0.623	0.480	0.679	0.560
Bi-GRU	0.619	0.462	0.678	0.548
Bi-LSTM	0.620	0.454	0.688	0.544
GRU + GloVe	0.884	0.841	0.921	0.879
LSTM + GloVe	0.886	0.846	0.920	0.881
Bi-GRU + GloVe	0.879	0.822	0.929	0.872
Bi-LSTM + GloVe	0.886	0.857	0.912	**0.883**

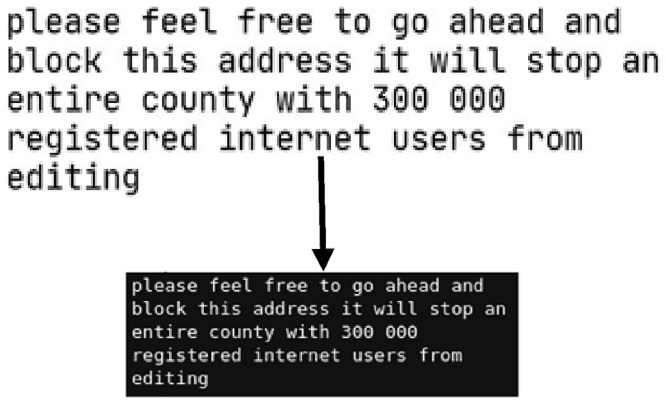

Fig. 11. Sample of an inference image with embedded text (top) and extracted content (bottom).

grants good predictions, however, it also shows that better work can be done when it comes to the implementation of the extraction module. The 10-fold cross-validation was applied to ensure the correct evaluation of results. The difference in metric scores is not significant, thus we do not think it is necessary to repeat the experiment, however, future work may involve an increased number of evaluation tests.

4.3. Comparison with other works

In comparison with other works in the field, our processing allowed us to reduce training and prediction times by reducing training sets. Moreover, the methodology reduced bias towards clean comments as the number of toxic and non-toxic samples was equal. Furthermore, the implemented GloVe embeddings provide an improvement over the models training without augmentation. We compared our approach for binary detection of toxic comments

Table 4
Comparison to other works.

Model	F1-score
GRU + GloVe	0.920
Bi-LSTM + GloVe	0.922
CNN - Ibrahim et al. (2019)	0.825
Bi-LSTM - Ibrahim et al. (2019)	0.814
Bi-GRU - Ibrahim et al. (2019)	0.817
Ensemble - Ibrahim et al. (2019)	0.828

to Ibrahim et al. (2019). The reason for comparison to only a single work is that other authors' experiments involved the implementation of algorithms to a multi-label problem. What is more, is that our evaluation metric matches the approach used in (Ibrahim et al., 2019). Moreover, we are comparing only testing results as no other work used the Toxicity dataset to develop a model for the classification of image embedded content. The results presented in Table 4 show that augmentation using GloVe embeddings provides much better results in terms of robustness that is evaluated using F1-measure.

The main findings from our results are as follows. Firstly, due to the pre-processing steps undertaken to clean input data the training of all models was conducted on a limited sample of comments. Many of the comments were not toxic and had to be discarded to reduce bias. The results indicate that it was possible to create a well-performing deep learning model for the detection of toxic comments. Extraction of text from an image requires improvement as there is a distinct 5% drop in accuracy and F1 measure. This shows that the tesseract module does not extract the entire content of the embedded message. It is important to note that standard fonts were used in this research and other types may further negatively influence the extraction of text. The curly fonts such as Gigi and Freestyle Script or very wide characters that are commonly used in memes are examples that may prove difficult to extract.

5. Conclusions and future work

Our work explores application of deep learning techniques for the detection of 'trolls' and toxic content shared on social media platforms. The results of the experiments clearly show that utilization of the word embeddings augmentation layer significantly increases the performance of all models. Further improvement has been achieved with the implementation of Bidirectional RNN's, however, the difference between regular networks and two-way networks is not great. Both GRU and LSTM models are very similar in the classification of 'troll'/toxic comments. The inference results, however, showed that all models experienced a drop in accuracy that is caused by the quality of the extracted content from the image. The comparison with similar work in the area showed that GloVe word embeddings grant a significant improvement of model performance when applied as augmentation for NLP classification tasks.

One of the novel contributions of this work is the ability of the developed framework to accurately extract and classify text from images that can be attached to messages sent online. The framework can be used to train models on different data and labels. The example use of such framework could be the detection of 'trolls' and toxic online activity as well as several types of crimes such as online abuse or hate speech. The use of the framework does not have to be limited to Twitter/short character posts but may also involve an application for other social media platforms such as Discord or Facebook. The developed system was evaluated using 2617 images with embedded comments. The tests demonstrate that while some accuracy is being lost when text is extracted from an image, the solution retains a high detection rate. The future research may involve experiments over a better extraction module that would provide more accurate text representation of the embedded content.

Acknowledgements

This study is supported with Research Incentive Funds (R20090) and Provost Research Fellowship award (R20093), Zayed University, United Arab Emirates.

References

Aghababaei, S., Makrehchi, M., 2017. Mining social media content for crime prediction. In: Proceedings - 2016 IEEE/WIC/ACM International Conference on Web Intelligence. IEEE, pp. 526–531. https://doi.org/10.1109/WI.2016.0089. WI 2016.

Al-Room, K., et al., 2021. 'Drone forensics: a case study of digital forensic investigations conducted on common drone models', international journal of digital crime and forensics. IGI Global 13 (1), 1–25. https://doi.org/10.4018/IJDCF.2021010101.

Anand, M., Eswari, R., 2019. Classification of abusive comments in social media using deep learning. In: Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019. Institute of Electrical and Electronics Engineers Inc., pp. 974–977. https://doi.org/10.1109/ICCMC.2019.8819734

Arshad, H., Jantan, A., Omolara, E., 2019. Evidence collection and forensics on social networks: research challenges and directions. Digit. Invest. 28, 126–138. https://doi.org/10.1016/j.diin.2019.02.001. Elsevier Ltd.

Bhattacharya, P., 2019. Social degeneration through social media: a study of the adverse impact of "memes". In: ITT 2019 - Information Technology Trends: Emerging Technologies Blockchain and IoT. Institute of Electrical and Electronics Engineers Inc., pp. 44–46. https://doi.org/10.1109/ITT48889.2019.9075096

Boast, K., Harriss, L., 2016. Digital Forensics and Crime. POSTnote 520 March 2016.

Cao, L., et al., 2019. Book spine recognition based on OpenCV and tesseract. In: Proceedings - 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2019. Institute of Electrical and Electronics Engineers Inc., pp. 332–336. https://doi.org/10.1109/IHMSC.2019.00083

Chaffey, D., 2022. 'Global social media statistics research sumary 2022. Available at: https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/.

Chung, H., Park, J., Lee, S., 2017. Digital forensic approaches for Amazon Alexa ecosystem. Digit. Invest. 22, S15–S25. https://doi.org/10.1016/j.diin.2017.06.010. Elsevier Ltd.

Conti, M., et al., 2012. Looking ahead in pervasive computing: challenges and opportunities in the era of cyber–physical convergence. Pervasive Mob. Comput. 8 (1), 2–21. https://doi.org/10.1016/j.pmcj.2011.10.001.

Hoffstaetter, S., 2022. Pytesseract 0.3.9. Available at: https://pypi.org/project/pytesseract/.

Huang, G. Bin, Zhu, Q.Y., Siew, C.K., 2004. Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE International Conference on Neural Networks - Conference Proceedings, pp. 985–990. https://doi.org/10.1109/IJCNN.2004.1380068.

Huang, C.T., et al., 2019. Mobile forensics for cloud storage service on iOS systems. In: Proceedings of 2018 International Symposium on Information Theory and its Applications. IEICE, pp. 178–182. https://doi.org/10.23919/ISITA.2018.8664393. ISITA 2018.

Husnain, M., Khalid, A., Shafi, N., 2021. A novel preprocessing technique for toxic comment classification. In: 2021 International Conference on Artificial Intelligence, ICAI 2021. Institute of Electrical and Electronics Engineers Inc., pp. 22–27. https://doi.org/10.1109/ICAI52203.2021.9445252

Ibrahim, M., Torki, M., El-Makky, N., 2019. Imbalanced toxic comments classification using data augmentation and deep learning. In: Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018. Institute of Electrical and Electronics Engineers Inc., pp. 875–878. https://doi.org/10.1109/ICMLA.2018.00141

Iqbal, F., et al., 2019. Wordnet-based criminal networks mining for cybercrime investigation. IEEE Access 7, 22740–22755. https://doi.org/10.1109/ACCESS.2019.2891694. IEEE.

Jigsaw, Kaggle, 2022. Multilingual toxic comment classification. Available at: https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification.

Li, S., et al., 2019. IoT forensics: amazon echo as a use case. IEEE Internet Things J. 6 (4), 6487–6497. https://doi.org/10.1109/JIOT.2019.2906946.

Liu, C., Singhal, A., Wijesekera, D., 2017. Identifying evidence for cloud forensic analysis. In: IFIP Advances in Information and Communication Technology. Springer, New York LLC, pp. 111–130. https://doi.org/10.1007/978-3-319-67208-3_7.

Lwin, H.H., Aung, W.P., Lin, K.K., 2020. Comparative analysis of android mobile

forensics tools. In: 2020 IEEE Conference on Computer Applications. IEEE, pp. 1–6. https://doi.org/10.1109/ICCA49400.2020.9022838. ICCA 2020.
MacDermott, A., et al., 2019. Forensic analysis of wearable devices: fitbit, garmin and HETP watches. In: 2019 10th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2019 - Proceedings and Workshop. https://doi.org/10.1109/NTMS.2019.8763834.
MacDermott, Á., et al., 2020. The internet of things: challenges and considerations for cybercrime investigations and digital forensics. Int. J. Digital Crime Forensics (IJDCF) 12 (1), 1–13. https://doi.org/10.4018/IJDCF.2020010101.
Maipradit, R., Hata, H., Matsumoto, K., 2019. Sentiment classification using N-gram inverse document frequency and automated machine learning. IEEE Software. IEEE Computer Society 36 (5), 65–70. https://doi.org/10.1109/MS.2019.2919573.
Mielly, M., 2017. Meme wars: how the internet has given vent to the anger fuelled by globalisation. Available at: https://scroll.in/article/843911/how-memes-have-made-it-easier-to-spread-anger-and-hate.
Pascual-Ferrá, P., et al., 2021. Toxicity and verbal aggression on social media: polarized discourse on wearing face masks during the COVID-19 pandemic. SAGE Publications Ltd Big Data & Society 8 (1). https://doi.org/10.1177/20539517211023533, 205395172110235.
Poornima, A., Priya, K.S., 2020. A comparative sentiment analysis of sentence embedding using machine learning techniques. In: 2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020. Institute of Electrical and Electronics Engineers Inc., pp. 493–496. https://doi.org/10.1109/ICACCS48705.2020.9074312
Powell, A., Haynes, C., 2020. Social media data in digital forensics investigations. In: Zhang, X., Choo, K.-K.R. (Eds.), Digital Forensic Education: an Experiential Learning Approach. Springer International Publishing, Cham, pp. 281–303. https://doi.org/10.1007/978-3-030-23547-5_14.
Quick, D., Choo, K.K.R., 2017. Pervasive social networking forensics: intelligence and evidence from mobile device extracts. J. Netw. Comput. Appl. 24–33. https://doi.org/10.1016/j.jnca.2016.11.018. Elsevier Ltd.
Quick, D., Choo, K.K.R., 2018. IoT device forensics and data reduction. IEEE Access 6, 47566–47574. https://doi.org/10.1109/ACCESS.2018.2867466. IEEE.
Rahul, et al., 2020. Classification of online toxic comments using machine learning algorithms. In: Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020. Institute of Electrical and Electronics Engineers Inc., pp. 1119–1123
Ravindran, R., et al., 2019. Traffic sign identification using deep learning. In: Proceedings - 6th Annual Conference on Computational Science and Computational Intelligence, CSCI 2019. Institute of Electrical and Electronics Engineers Inc., pp. 318–323. https://doi.org/10.1109/CSCI49370.2019.00063
Salminen, J., et al., 2020. Topic-driven toxicity: exploring the relationship between online toxicity and news topics. In: Chen, P.-Y. (Ed.), Public Library of Science vol. 15, e0228723. https://doi.org/10.1371/journal.pone.0228723, 2, PLOS ONE.
Sulke, A., et al., 2019. Classification of Online Pernicious Comments Using Machine Learning. IJSRD-International Journal for Scientific Research & Development|.
Zhang, X., Zheng, X., 2017. Comparison of text sentiment analysis based on machine learning. In: Proceedings - 15th International Symposium on Parallel and Distributed Computing, ISPDC 2016. Institute of Electrical and Electronics Engineers Inc., pp. 230–233. https://doi.org/10.1109/ISPDC.2016.39