Pre-processing of Social Media Remarks for Forensics

Xuhao Gao
University of Warwick
Warwick, UK

Man Qi
Canterbury Christ Church University
Canterbury, UK

Abstract— **The Internet's rapid growth has led to a surge in social network users, resulting in an increase in extreme emotional and hate speech online. This study focuses on the security of public opinion in cyber security by analyzing Twitter data. The goal is to develop a model that can detect both sentiment and hate speech in user texts, aiding in the identification of content that may violate laws and regulations. The study involves pre-processing the acquired forensic data, including tasks like lowercasing, stop word removal, and stemming, to obtain clear and effective data. This paper contributes to the field of public opinion security by linking forensic data with machine learning techniques, showcasing the potential for detecting and analyzing Twitter text data.**

Keywords-pre-process, social media remarks, forensic data

I. INTRODUCTION

Text sentiment analysis involves using algorithms to analyze and extract specific emotions expressed in text, such as sentiments expressed in articles or blogs. By automatically analyzing large volumes of text data, computers can identify the emotional polarity (positive, negative, neutral) of the text. This saves time and enables the collection of extensive emotional data, which has significant implications for decision-making and analysis.

In sentiment analysis, text data can be categorized into three levels: document level, sentence level, and phrase level. Document-level analysis is used when evaluating the overall sentiment polarity of a blog post or similar content. Sentence-level analysis is applied when assessing the sentiment polarity of a paragraph or multiple paragraphs within an article or tweet. Phrase-level analysis, on the other hand, focuses on sentiment analysis at the phrase level, often involving the counting of positive and negative words in an article.

According to Mayur et al. (2022), sentiment analysis employs various methods, including dictionary-based methods, machine learning methods, and hybrid methods. Figure 1 provides an overview of the different approaches used in the field of sentiment analysis. This research explores both traditional and modern analysis methods, combining the strengths of each approach.

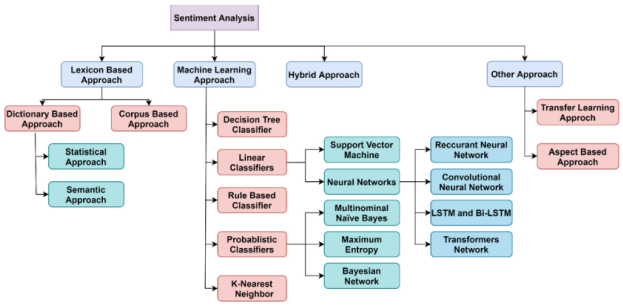

Figure 1. Approach of sentiment analysis (Mayur et al., 2022)

The field of sentiment analysis is continuously evolving alongside technological advancements. Traditional sentiment analysis methods typically rely on sentiment dictionaries, which can be constructed manually or generated automatically. Manual construction of sentiment dictionaries involves multiple rounds of data screening and annotation, where words are classified based on their positive/negative or strong/weak emotional expressions. Manual construction allows for flexible expansion of dictionary entries and quick adjustments based on specific needs. However, it also incurs higher labor, time, and energy costs. Moreover, it has limited applicability in interdisciplinary research.

Automatic construction of sentiment dictionaries is an extension of the manual method. It enhances the comprehensiveness of the sentiment thesaurus by including nouns, verbs, and adverbs. It leverages diverse corpora and calculation rules from relevant fields to automatically identify sentiment words and their polarity, facilitating automatic dictionary construction. Table 1 visually summarizes the advantages and disadvantages of these two types of lexicon-based sentiment analysis methods.

Table 1. Comparison of two sentiment dictionary methods

Emotional Dictionary	Advantage	Disadvantage
Artificial	High scalability High convenience Fast adjustment	High labour cost Time consuming Limited scope Limited in interdisciplinary fields
Automatic	Adapt to variety of situations Quickly classify emotional polarity Wide range	The analysis results of short texts and domain-specific texts are poor

Another modern technique in sentiment analysis is machine learning-based sentiment analysis. These methods employ trained machine learning algorithms to predict the emotional polarity of new texts. Various algorithms can be utilized, including SVM, RNN, DNN, LSTM, and more. Machine learning-based methods offer convenience, speed, and higher automation compared to traditional approaches. Additionally, they exhibit high extensibility and are well-suited for interdisciplinary research. Table 2 provides a clear overview of the advantages and disadvantages associated with several commonly used machine learning algorithms.

Table 2. Comparison of machine learning methods

Technique	Advantage	Disadvantage
SVM	The most common SA algorithm is relatively suitable for the unified processing of large data sets with good accuracy	The adjustment is difficult and the training time is long
RNN	The order of data input can be obtained, with higher accuracy and wider application scope	The training time is long and the calculation cost is high
DNN	The model is simple and easy to use	The overfitting problem
LSTM	Compared with RNN, it is more efficient and can record long-term data relationships	The model is complex and the training time is very long

Additionally, on the basis of sentiment analysis, attention should also be paid to hate speech. Generally speaking, hate speech is a form of offensive language expressed towards a group or an individual based on specific characteristics such as religion, race, origin, sexual orientation, gender, and appearance. On the one hand, hateful content and anti-social propaganda can lead to social unrest, resulting in instability in society and public opinion. On the other hand, such remarks can reinforce society's incorrect perception of certain groups and individuals, leading to more severe discrimination and oppression.

For example, according to Gover et al.'s 2020 paper, there has been a significant increase in anti-Asian and Chinese-exclusive content on social media during the COVID-19 pandemic. This misrepresentation, blaming Asian people for the pandemic, reflects an unequal bias and discrimination. Currently, both academia and industry are utilizing machine learning approaches to address the issue of hate speech on social platforms. Currently, various methods such as vector machines, Bayesian logistic regression, and LSTM models are primarily used for identifying this type of speech.

II. Related Work

A. *Sentiment analysis*

Data processing and sentiment analysis play a crucial role in the field of semantic analysis. This type of research involves processing and analyzing datasets to determine the sentiment polarity of the data. Currently, there is extensive research being conducted in this field.

In earlier studies, Warner and Hirschberg (2012) utilized features such as unigram, part-of-speech, and templates to train a linear kernel model based on support vector machines to classify content into different emotions. Using existing datasets for research purposes is an effective method to obtain the necessary content, which can provide valuable insights for future research. However, this approach is based on traditional machine learning methods, which may result in lower efficiency and accuracy compared to models like LSTM.

Kumar et al. (2019) proposed a machine learning approach to analyze customer satisfaction in airline-related social media messages. The researchers collected and cleaned data from Twitter, converting the text content into a digital vector format using the N-gram method. They compared and discussed various training models and found that artificial neural network models outperformed SVM, CNN, and other architectural models in terms of accuracy. They also highlighted that a hybrid model combining CNN and convolutional neural network models demonstrated better performance when analyzing textual content in the context of images. This research serves as a valuable reference for the prediction of text sentiment.

In 2014, Tang Duyu et al. developed three neural network models to learn word vectors from Twitter data that already contained positive and negative emoticons. They conducted emotion prediction and data statistics using a hybrid method with a linear neural network. This research approach has a positive impact on exploring the integration of image text and emotion polarity.

Malmasi and Zampieri (2017) proposed a method to train SVM classifiers using N-gram features, similar to individual character and word datasets. Their word embedding method utilized a numerical vector format similar to neural network models. In their study, text data were transformed into multidimensional vectors during the preprocessing stage. This data processing method holds significant reference value for this project.

Chen Xingming et al. (2019) introduced a system for emotion classification that incorporates dense emotional supplementary information and negative data. They utilized a reverse LSTM model to analyze the potential emotional reversal effect of negative words on subsequent analysis. This model was integrated into three neural networks: LSTM, CNN, and Char SCNN, to evaluate their effectiveness. This system significantly addresses the semantic composition issue of current deep neural network models in sentiment classification.

It is important to note that the aforementioned methods may encounter overfitting issues due to the use of single-call model methods. Additionally, traditional training methods may introduce certain imperfections in the training process.

In 2021, Mohammad et al. proposed a bidirectional CNN-RNN deep model for sentiment analysis detection. This model was designed to address the issue of gradient disappearance and explosion that can occur during text training. The researchers created a new deep learning architecture specifically for sentiment analysis and ensured that the model could adapt to different types of social media texts, including long comments and short sentences. However, a limitation of this study is that the training data was relatively narrow, with a large number of samples but insufficient diversity.

In a study by Chetanpal et al. (2022) that focused on COVID-19 social media comments, an LSTM-RNN conformance model trained with publicly available datasets was utilized to analyze the sentiment of COVID-19 related comments on social networks. This approach demonstrated an improvement in classification accuracy during the machine learning process, which holds some reference significance. However, due to the presence of noise in the data processing stage, there may be some errors in the classification results.

B. Hate speech

On the basis of data processing and sentiment analysis, another important aspect of this study is the further screening and identification of hate speech in the database. Combining cybersecurity-related content with data preprocessing and sentiment analysis can effectively identify potential dangerous public opinions and enable timely response.

In 2016, Waseem and Hovy open-sourced a 16K Twitter benchmark dataset containing hate speech. They used a 4-gram feature training classifier to distinguish hate content from ordinary tweets. The study also incorporated location information, gender characteristics, and the combination of N-gram features for analysis. This dataset and analysis approach have significant reference value in hate speech recognition research using machine learning methods.

Given that identifying hate speech features is a manual and time-consuming task, academic research has increasingly focused on deep learning and artificial intelligence-based approaches. Park and Fung (2017) proposed a hybrid model that combines logistic regression and CNN architecture to distinguish hate speech from ordinary tweets. Their study found that the hybrid model outperformed single machine learning models.

Kamble and Joshi compared three classical deep learning models, CNN, LSTM, and BiLSTM, and found that domain-specific word embeddings yielded better performance than conventional pre-trained word embeddings. This highlights the importance of domain-specific embeddings in hate speech analysis.

In 2022, Shakir et al. proposed a deep learning model based on BiLSTM combined with CNN for identifying hate speech in social networks. Their model achieved high accuracy in detecting hate speech in text data. This innovative approach provides new insights for hate language recognition models. However, the potential issue of inaccurate detection results when using a different dataset should be considered due to the reliance on single detection data.

Overall, these studies contribute to the advancement of hate speech recognition and provide valuable insights for developing effective detection models.

III. RESEARCH METHOD

The research applies natural language processing techniques to preprocess forensic text data. The first step is text pre-processing, where various methods are applied to clean and prepare the text data. This includes techniques such as removing special characters, normalizing text, tokenization, removing stop words, and performing lemmatization or stemming. These pre-processing steps help in standardizing the text and preparing it for further analysis.

The subsequent stage involves word embedding using GloVe (Global Vectors for Word Representation) as the method. GloVe offers pre-trained word embeddings that effectively capture semantic relationships among words. By utilizing these pre-trained embeddings in the experiment, the need for training embeddings from scratch is eliminated, ensuring a high level of accuracy.

IV. VALIDATION AND SPECIFICATION OF DATASET

Prior to the pre-processing of forensic data, it is important to identify and analyze any suspicious remarks that may be present on the forensic device. Therefore, a brief analysis is provided.

By examining the network's web cookie information, it is evident that Twitter is the most widely used and frequently accessed social media platform among various social platforms. Figure 2 displays some of the results obtained from conducting a keyword search on Twitter.

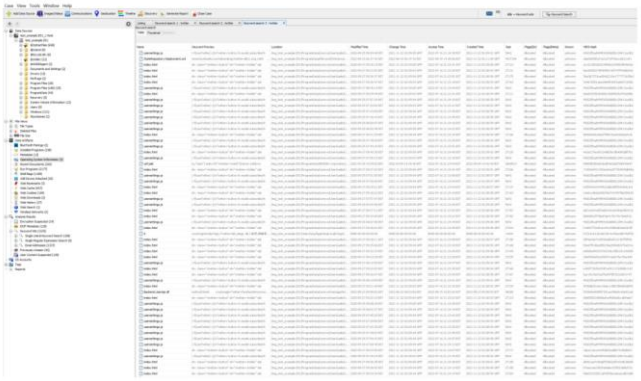

Figure 2. Keyboard search of twitter

Within a package, Twitter-related keywords and information content were discovered, including posts. Selected examples

of posts can be seen below.

Figure 3. Suspicious text content section

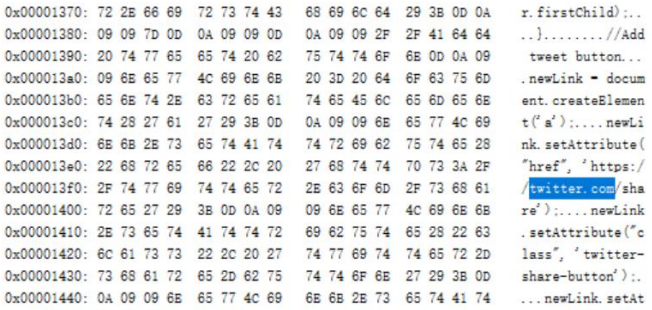

Figure 4. Hex text of twitter

The next step is to export and store the text data for further pre-processing. The figure below displays the top 10 Twitter posts obtained after extracting the relevant files and performing string interception. Please note that the following text may contain highly offensive statements and words. It is important to emphasize that these statements are intended for research purposes only and are not directed towards any specific individual or organization. Additionally, sensitive words in the images have been obscured for privacy and appropriateness.

Cleaned data:

	text
0	@@@@@@@@shut up, you are a liar!!!!!!!!!!
1	At least im not a nixxxr:)
2	im so happy!!!
3	today is a good day.xymahlodmagndmsd
4	im a jewish woman who is blind?!?
5	@fxxk u bixxh@
6	.....see you next time
7	white pxx------0...
8	@today ...
9	thank you mother fk!

Finger 5. Forensic text data fields

The dataset used in the model training has been briefly described previously. Here we use the Sentiment140 dataset from Kaggle, which contains about 1.6 million tweets. The dataset is a .csv file that contains six major fields, as shown in Table below.

Table 3 Dataset information

Data source	Data file type	Dataset size	Data fields number	Data fields
Kaggle	.CSV file	1.6 million	0	Polarity of tweets
			1	ID of tweets
			2	Date of tweets
			3	Query
			4	User of tweets
			5	Text of tweets

Then a simple code is used to do a pre-check on the data set, import the data set and output the data, as shown in Figure 6 below.

	0	1	2	3	4	5
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
5	0	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kwesidei not the whole crew
6	0	1467811592	Mon Apr 06 22:20:03 PDT 2009	NO_QUERY	mybirch	Need a hug
7	0	1467811594	Mon Apr 06 22:20:03 PDT 2009	NO_QUERY	coZZ	@LOLTrish hey long time no see! Yes.. Rains a...
8	0	1467811795	Mon Apr 06 22:20:05 PDT 2009	NO_QUERY	2Hood4Hollywood	@Tatiana_K nope they didn't have it
9	0	1467812025	Mon Apr 06 22:20:09 PDT 2009	NO_QUERY	mimismo	@twittera que me muera ?

Figure 6. Original data

The first column in the dataset represents the target column, indicating the sentiment of the tweet. The dataset uses the values 0 for negative, 4 for positive, and 2 for neutral emotions. The next column displays the unique ID of each tweet. It is followed by the date the tweet was posted and the username of the tweet's author. Finally, the tweet text itself is provided.

To proceed with the analysis, it is necessary to rename each column in the dataset accordingly. Additionally, since only the sentiment polarity information and text content are required for the subsequent steps, the unnecessary columns can be eliminated. The resulting dataset, after removing the irrelevant columns, is displayed in Figure 7.

	sentiment	tweet
0	0	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	is upset that he can't update his Facebook by ...
2	0	@Kenichan I dived many times for the ball. Man...
3	0	my whole body feels itchy and like its on fire
4	0	@nationwideclass no, it's not behaving at all....
5	0	@Kwesidei not the whole crew
6	0	Need a hug
7	0	@LOLTrish hey long time no see! Yes.. Rains a...
8	0	@Tatiana_K nope they didn't have it
9	0	@twittera que me muera ?

Figure 7. Processed data

Prior to further processing, it is crucial to verify if the sentiment polarity distribution in the data is balanced. Imbalance can result in significant bias, making it essential to ensure that the dataset is not heavily skewed during the modeling and training process. To achieve this, visualizing the dataset through graphs or charts can help obtain the distribution of sentiment polarity, as depicted in Figure 8 below.

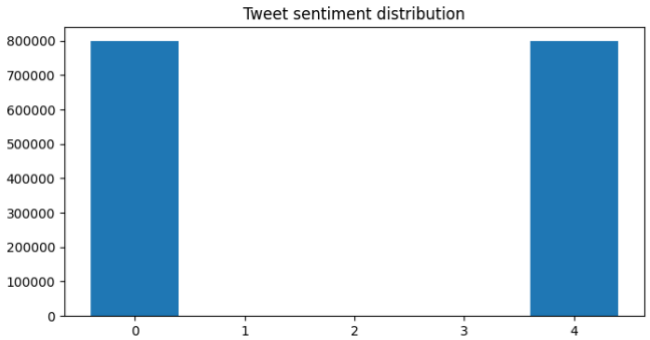

Figure 8. Tweet sentiment distribution

The dataset exhibits an exceptionally balanced distribution of the two types of emotions, with both categories being close to 0.8 million. With the dataset successfully loaded, the next step can be initiated, which involves the clean-up process.

V. PRE-PROCESING

This preprocessing step addresses the two previously mentioned components: forensic data text and training datasets. While the accuracy and other important dataset features have been verified, the original tweet data contains considerable noise and requires cleaning. This cleaning step can be roughly divided into two main stages: redundancy elimination and stemming.

In the redundancy elimination stage, three sub-steps are involved: lowercase conversion, stop word removal, and redundant symbol deletion. Lowercasing is performed to reduce the analysis scope by considering case variations that might lead to data errors. Words in different case formats are treated as distinct vectors in the vector space, potentially causing processing errors. Stop words, such as "the," "and," and "was," which carry little specific meaning and contribute minimally to the model, are removed. The NLTK natural language processing package is utilized in this project to directly employ a stop word set for efficient and straightforward filtering. The removal of redundant symbols involves eliminating irrelevant characters like punctuation from the dataset. Regular expressions are employed for convenient and efficient fulfillment of this requirement.

The second step entails stem extraction, which involves converting tokens into their root forms. Since many words in the text contain suffixes or prefixes that introduce irrelevant content, this redundancy can hinder model training. To address this, words are converted to their root forms. In this project, functions from the NLTK natural language processing library are employed for this purpose. Table 4 illustrates the various tools and methods utilized during this step.

Table 4. Tools and methods in pre-processing stage

Processing steps	Method
Lowercase	Python lower() function
Stop words	NLTK library stopwords.words('english') function
Redundant symbols	Regular expression *"@S+|https?:S+|http?:S|[^A-Za-z0-9]+""*
Stemming	NLTK library stemmer.stem() function

The output below presents the pre-processed forensic text data. Once the data has been cleaned, you can proceed with the subsequent steps of your analysis or task.

```
Cleaned data:
                          text
0        shut up, you are a liar
1       At least im not a nixxxr
2                   im so happy
3             today is a good day
4  im a jewish woman who is blind
5                   fxxk u bixxh
6              see you next time
7                   white pxx
8             today is a good day
9             thank you mother fk
```

Figure 9. Pre-processing step result of forensic text data

The following results display the state of the Twitter dataset before undergoing the cleaning process, which is used for model training. It is evident that the impact of the cleaning process is substantial, indicating a noticeable improvement in the dataset's quality.

```
Cleaned data:
   target                                               text
0       0      awww bummer shoulda got david carr third day
1       0  upset update facebook texting might cry result...
2       0  dived many times ball managed save 50 rest go ...
3       0                  whole body feels itchy like fire
4       0                                behaving mad see
5       0                                      whole crew
6       0                                        need hug
7       0  hey long time see yes rains bit bit lol fine t...
8       0                                            nope
9       0                                        que muera
```

Figure 10. Pre-processing step results of training data

VI. WORD EMBEDDING

Tokenization involves breaking down a given sentence or paragraph into smaller parts, such as fixed word collocations or individual words. This process results in the generation of a list of token sequences and an associated index. During tokenization, a sequence of characters is transformed into a token, and each word is assigned a unique value. For the implementation phase of the project, I utilized classes provided by TensorFlow and Keras. These frameworks offer a text pre-processing module that includes a convenient function called Tokenizer(). This function enables quick and accurate tokenization while also allowing for the specification of split conditions and the setting of a maximum number of words.

In preparation for future work, which will involve the use of the LSTM model that requires inputs of the same length, the padding function pad_sequences() is employed. This function ensures that each text is of uniform length. For this project, a fixed length of 50 words is set.

Once the tokenization process is complete, the size of the data is outputted to verify if it aligns with expectations. To clarify, 80% of the data is chosen for training the model, while the remaining 20% is used for testing. The verification results are presented in Figure below.

```
x_train shape:  (1280000, 50)
x_test shape:  (320000, 50)
y_train shape: (1280000, 1)
y_test shape: (320000, 1)
Vocab size:  290714
Max text length:  50
```

Figure 11. Tokenization step output

It is important to note that in the output provided above, the variable "x" represents the Twitter text in the dataset. Each text has a length of 50 words. The training dataset consists of 1.28 million samples, while the testing dataset consists of 320,000 samples. On the other hand, the variable "y" represents the sentiment polarity in the dataset. Since it is a binary

classification task, each sentiment polarity is represented by a single bit.

After completing the series of pre-processing steps, the final step is word embedding. Word embedding is the process of representing text in a vector space through computation. It allows words with similar meanings or related representations to have similar vector space values, capturing contextual and semantic relationships. In this project, instead of using complex and uncertain pre-trained word embeddings, a pre-trained word embedding model with proven performance is utilized. The word embeddings used are Twitter-trained files downloaded from the GloVe website, which employ an unsupervised learning algorithm for obtaining word vectors. These downloaded files have high confidence as they have been trained on word co-occurrence statistics from a reliable source corpus. Table 5 below presents some of the tools and methods employed during this step.

Table 5 Tools and methods in word embedding stage

Word Embedding steps	Method
Tokenization	Keras library Tokenizer() function
Extended character length	Keras library pad_sequences() function
Word embedding	GloVe from https ://nlp.stanford.edu/projects/glove/

The next step involves creating the embedding layer, which transforms the input sequence (i.e., the cleaned text data) into dense vectors. Given that the previously utilized GloVe training model employed a dense layer represented by 200 vectors, an embedding dimension of 200 is selected here. Additionally, the input length is set to 50. The resulting vector is then passed as input to the subsequent model, which is a recurrent neural network (RNN).

VII. Conclusion and Future Work

The focus of this research lies in the field of media digital forensics, driven by the increasing generation of data on social platforms due to the advancements in technology. Targeted analysis of this data allows for the extraction of sentiment data for public opinion analysis. Furthermore, the prevalence of hate speech on social media platforms necessitates its analysis to identify potential social unrest and develop appropriate solutions.When it comes to sentiment analysis, there are two main categories of methods. The first category is based on sentiment dictionaries, which offer convenience and expandability. However, these methods are associated with high labor costs and time consumption, making them less suitable for interdisciplinary research. The second category consists of machine learning-based methods. These methods leverage automated analysis and research techniques, allowing for broader application, higher efficiency, and better accuracy. Automated methods are particularly adept at handling complex analysis situations.

In this research, the pre-processing of hate speech is conducted alongside sentiment analysis. The excessive presence of hate speech can lead to social movements and unrest, posing a threat to social development and progress. Consequently, public opinion analysis plays a role in preventing such situations to some extent. Currently, most research in this field employs automated detection and analysis tools due to their fast processing speed and high efficiency. Common methods which will be utilized in this research domain include vector machines, Bayesian logistic regression, and recurrent neural networks (RNNs).

References

[1] Chow, Jacky. (2017). Analysis of Financial Credit Risk Using Machine Learning. 10.13140/RG.2.2.30242.53449.

[2] Chen, Xingming & Rao, Yanghui & Xie, Haoran & Wang, Fu Lee & Zhao, Yingchao & Yin, Jian. (2019). Sentiment Classification Using Negative and Intensive Sentiment Supplement Information. Data Science and Engineering. 10.1007/s41019-019-0094-8.

[3] Casey E. (2011). Digital evidence and computer crime: forensic science, computers and the internet. Academy Press Publications, London. ISBN 978-0-12-374268

[4] Gover, A. R., Harper, S. B., & Langton, L. (2020). Anti-asian hate crime during the covid-19 pandemic: Exploring the reproduction of inequality. American Journal of Criminal Justice. Advance online publication. https://doi.org/10.1007/s12103-020-09545-1

[5] Garfinkel SL. (2010). Digital forensic research: the next 10 years. In: Proceedings of the 10th annual conference on digital forensic research workshop. Digit Investig 7:S64–S73

[6] Harshith. 2019. Text Preprocessing in Natural Language Processing. [online] Available at: https://towardsdatascience.com/text-preprocessing-in-natural-language-processing-using-python

[7] Hernandez-Suarez, A. & Sanchez-Perez, Gabriel & Martinez-Hernandez, V. & Perez-Meana, Hector & Toscano,

[8] Karina & Nakano-Miyatake, M. & Sanchez, Victor. (2017). Predicting political mood tendencies based on Twitter data. 1-6. 10.1109/IWBF.2017.7935106.

[9] Hosmer C. (2006). Digital evidence bag. Commun ACM 49(2):69–70.

[10] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

[11] Kumar, S., Zymbler, M. 2019. A machine learning approach to analyze customer satisfaction from airline tweets. J Big Data 6, 62.

[12] Kathirgamanathan, A., Patel, A., Khwaja, A.S., Venkatesh, B., Anpalagan, A. (2022). Performance comparison of single and ensemble CNN, LSTM and traditional ANN models for short-term electricity load forecasting, J. Eng. 2022, 550– 565. https://doi.org/10.1049/tje2.12132

[13] Kamal, R., Hemdan, E.ED. & El-Fishway. (2021). N. A review study on blockchain-based IoT security and forensics. Multimed Tools Appl 80, 36183–36214. https://doi.org/10.1007/s11042-021-11350-9

[14] Karina Reyes. (2020). Sentiment Analysis on Tweets with LSTM. [online] Available at: https://medium.com/@karyrs1506/sentiment-analysis-on-tweets-with-lstm-22e3bbf93a61. Konikoff, D. (2021). Gatekeepers of toxicity: Reconceptualizing Twitter's abuse and hate speech policies. Policy Internet, 13, 502– 521. https://doi.org/10.1002/poi3.265.

[15] Mikolov, T., Chen, K., Corrado, G.S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. ICLR.

[16] Mohammad Ehsan Basiri, Shahla Nemati, Moloud Abdar, Erik Cambria, U. Rajendra Acharya. (2021). ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis. Future Generation Computer Systems, Volume 115, 2021, Pages 279-294, ISSN 0167-739X. https://doi.org/10.1016/j.future.2020.08.005.

[17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. 2012. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 15(56):1929-1958, 2014.

[18] Pauly, Leo & Peel, Harriet & Luo, Shan & Hogg, David & Fuentes, Raul. (2017). Deeper Networks for Pavement Crack Detection. 10.22260/ISARC2017/0066.