Technological University Dublin
ARROW@TU Dublin

Conference papers | School of Computer Science

2023-6

# Identifying Online Child Sexual Texts in Dark Web through Machine Learning and Deep Learning Algorithms

Vuong Ngo
*Technological University Dublin*, vuong.ngo@tudublin.ie

Susan McKeever
*Technological University Dublin*, susan.mckeever@tudublin.ie

Christina Thorpe
*Technological University Dublin*, christina.thorpe@tudublin.ie

Follow this and additional works at: https://arrow.tudublin.ie/scschcomcon

 Part of the Psychology Commons

## Recommended Citation

Vuong M. Ngo, Susan Mckeever and Christina Thorpe. 2023. Identifying Online Child Sexual Texts in Dark Web through Machine Learning and Deep Learning Algorithms. In APWG.EU Technical Summit and Researchers Sync-Up (APWG.EU-Tech 2023), pp. 1-6, CEUR Workshop Proceedings. DOI: 10.21427/WFN5-RT72

This Conference Paper is brought to you for free and open access by the School of Computer Science at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, gerard.connolly@tudublin.ie, vera.kilshaw@tudublin.ie.



This work is licensed under a Creative Commons Attribution 4.0 International License.
Funder: the Safe Online Initiative of End Violence and the Tech Coalition

# Identifying Online Child Sexual Texts in Dark Web through Machine Learning and Deep Learning Algorithms

Vuong M. Ngo[1,*], Susan Mckeever[1] and Christina Thorpe[2]

[1]School of Computer Science, Technological University Dublin, Grangegorman, Dublin 7, Ireland
[2]School of Informatics and Cybersecurity, Technological University Dublin, Blanchardstown, Dublin 15, Ireland

Abstract

Predators often use the dark web to discuss and share Child Sexual Abuse Material (CSAM) because the dark web provides a degree of anonymity, making it more difficult for law enforcement to track the criminals involved. In most countries, CSAM is considered as forensic evidence of a crime in progress. Processing, identifying and investigating CSAM is often done manually. This is a time-consuming and emotionally challenging task. In this paper, we propose a novel model based on artificial intelligence algorithms to automatically detect CSA text messages in dark web forums. Our algorithms have achieved impressive results in detecting CSAM in dark web, with a recall rate of 89%, a precision rate of 92.3% and an accuracy rate of 87.6%. Moreover, the algorithms can predict the classification of a post in just 1 microsecond and 0.3 milliseconds on standard laptop capabilities. This makes it possible to integrate our model into social network sites or edge devices to for real-time CSAM detection.

**Keywords**

Child sexual exploitation material, CSEM, CSAM, text content, artificial intelligent, forums

## 1. Introduction

In general, Child Sexual Abuse Material (CSAM) includes any visual, written or audio material that depicts or describes sexual abuse of children. This can include photographs, videos, stories, chats, comments, drawings or any other media[1]. The production and distribution of CSAM has negative impacts on victims and society. Victims can live with long psychological, emotional, and physical harm [1]. A high volume of CSAM is created and shared daily on both surface web platforms such as social network sites and dark web forums. It is not viable for human experts to investigate, detect and prevent CSAM manually [2]. However, automatically detecting and analysing online CSA text can be extremely challenging and time-consuming, due to language complexity, contextual ambiguity, dynamic nature of language and large volume of data. This is particularly the case for CSAM shared on the dark web, where privacy and anonymity are prioritized. Moreover, perpetrators often use code words, slang, or other forms of obfuscation to avoid detection and hide their activities.

In this context, we propose a CSAM detection intelligence model based on both classical Machine Learning (ML) and Deep Learning (DL) techniques. Our CSAM detection model can be used to monitor and remove CSA texts on online platforms in real-time and with high accuracy, providing better protection for children. We have also created a manually labelled dataset of CSAM and non-CSAM content that can be used to train and test CSAM detection algorithms. In the future, our model will be able to detect perpetrator behaviours, collect forensic evidence, and extract valuable knowledge for child agencies, hotlines, education programs and policy makers.

The remainder of the paper is organised as follows. In the Section 2 we review the related work. Section 3 presents our system architecture and the machine learning and deep learning algorithms. The evaluation methodology and experimental results for the system are shown in Section 4. Finally, we conclude and give some future directions in Section 5.

## 2. Related Work

Research works [3], [4], [5], [6] and [7] applied deep convolutional neural network models or deep perceptual hashing algorithms with the goal of removing CSAM from social media sites. With the exception of [3], papers [4], [5], [6] and [7] used datasets from third-parties to train and test their models. However, these papers only considered CSA images and not text. Similar to our work, research works [8], [9], [10], [11] and [12] applied ML and DL models to process CSA text. In [8], Naive Bayes (NB), Support Vector Machine (SVM) and Random Forest

*APWG.EU-Tech'23: APWG.EU Technical Summit and Researchers Sync-Up, June 21 & 22, 2023, Dublin, Ireland*
*Corresponding author.
Vuong.Ngo@tudublin.ie (V. M. Ngo);
Susan.Mckeever@tudublin.ie (S. Mckeever);
Christina.Thorpe@tudublin.ie (C. Thorpe)
0000-0002-8793-0504 (V. M. Ngo); 0000-0003-1766-2441 (S. Mckeever); 0000-0002-2359-883X (C. Thorpe)
© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

[1]https://www.inhope.org/EN/articles/child-sexual-abuse-material, https://www.hotline.ie/what-to-report/csam, https://www.rainn.org/news/what-child-sexual-abuse-material-csam

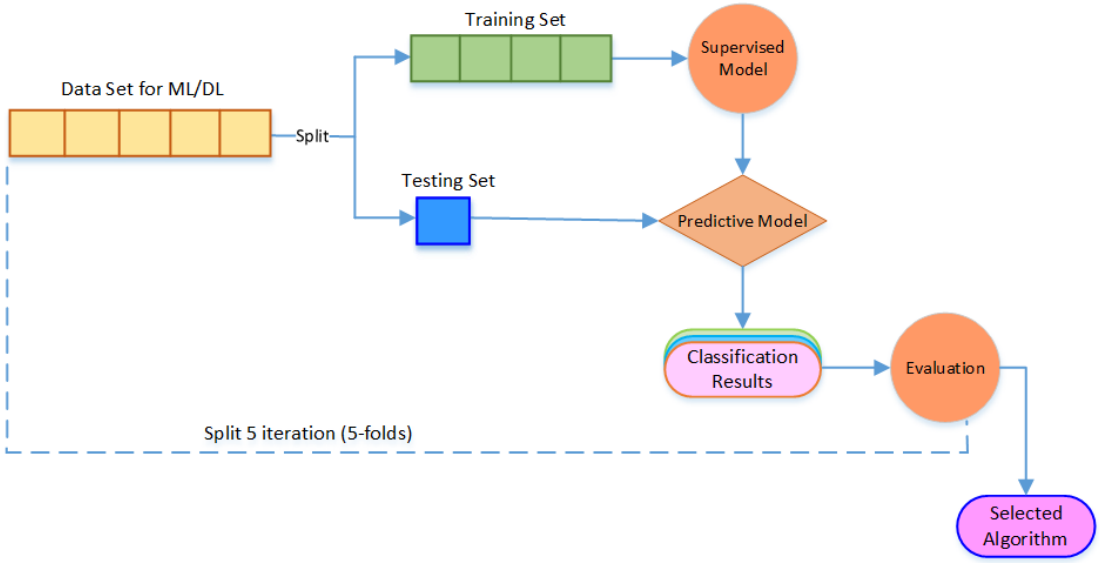



**Figure 1:** The system architecture for CSAM classification algorithm

(RF) were applied to detect online abusive and bullying comments on Facebook and Twitter. In [9], the histogram gradient boosted decision trees were exploited for predatory chat conversation detection. In [10], Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) were applied for YouTube comments. In [11], deepWalk model and graph embedding representations were used to detect abuse chat logs in French on the SpaceOrigin game. In [12], Logistic Regression (LR), XGBoost and Multi Layer Perceptron (MLP) were exploited to detect sexual predatory chats in social networks. To train and test ML/DL models, the papers [8] and [11] created their own datasets and the papers [9], [10] and [12] used datasets of third-parties. However, these papers considered the clear web not the dark web.

Dark web data was also processed in [13], [14], [15], [16] and [17]. However, the approaches did not automatically detect CSA text on the dark web by using post contents and artificial intelligence. In [13], the 450 authorised hidden service sites were manually classified. In [14], the authors analysed seven popular dark web sites to monitor the sites by using their metadata, e.g. the number of users, site names and common users in sites. In [15], the authors statistically analysed some simple metadata e.g. victim ages and the number of CSAM reports per year. In [16], K-Means algorithm was applied to cluster the forum comments into the selected seven labels, i.e. breach, financial, drug, vendor, account, product and other. In [17], the authors manually analysed transcripts of 53 anonymous suspects in United Kingdom to understand suspects' interaction behaviors and sexual interests.

# 3. System Architecture and Algorithms

## 3.1. System Architecture

Supervised learning in classical ML and DL is a popular method for text classification based on learning patterns from labelled training samples [18, 19]. Every supervised learning algorithm has its strengths and weaknesses. Therefore, to find a suitable algorithm to classify CSAM post contents, we apply the two most popular classical ML algorithms, NB and SVM, and the two most popular DL algorithms, LSTM and BERT (Bidirectional Encoder Representations from Transformers). More details can be found in Section 3.2.

Figure 1 shows our system architecture used to design and implement our novel algorithm for CSAM text classification. In that, the Supervised Method component implements NB, SVM, LSM and BERT algorithms. The algorithms tokenizes the post texts and transforms them into vector representation using TF.IDF[2] (in NB and SVM) or embedding layers[3] (in LSTM and BERT).

The Evaluation component is used to determine the execution times (i.e., training time and prediction time) and the classification performance metrics (i.e., precision, recall and accuracy) of each combination of algorithms. To avoid overfitting, we apply 5-fold cross-validation of our dataset. Then the algorithm uses 4 folds for the train-

[2] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
[3] https://www.tensorflow.org/api_docs/python/tf/keras/layers/Embedding

ing set and the remaining fold for the testing set. This process is repeated until every fold serves as the testing set. The average of the recorded classification measures of five rounds are the classification performance measures for the algorithm. Finally, we analyse experimental results on the data set to recommend the best algorithm for CSAM text classification in the dark web.

### 3.2. Machine Learning and Deep Learning

NB is a specific type of probabilistic classifier that relies on applying Bayes' theorem with certain simplifying assumptions. NB is widely used in natural language processing, spam filtering, and other applications where it is necessary to classify items into different categories based on probabilistic features. It assumes that the features are strongly independent to simplify computation. We used the Gaussian Naive Bayes algorithm implemented in [20], with parameters: $alpha$= 1 and $fit_prior$= True. Where, $alpha$ is the additive smoothing parameter and $fit_prior$ determines whether to learn class' prior probabilities or not.

SVM represents patterns as points in space and divides the data points by a clear gap. It constructs a maximum margin separator and can perform a non-linear classification by using the so-called kernel trick. We used the C-support vector classification algorithm implemented in [21], with parameters: $C$= 1.0, $kernel$= 'linear', $degree$= 3 and $gamma$= 'auto'. Where, $C$ is the regularization parameter. $kernel$ is the used kernel type. $degree$ is the degree of the polynomial kernel function and $gamma$ is kernel coefficient.

LSTM is a special kind of Recurrent Neural Network (RNN). RNN is a type of neural network commonly used to develop natural language processing models. RNN remembers the sequence of the data and exploits data patterns and feedback loops for prediction. LSTM was applied to avoid the long-term dependency problem in regular RNN. We used the Bidirectional-LSTM algorithm implemented in [22], with parameters: $Embedding$= (1000, 128, input_length=200), $Bidirectional$(LSTM(64)), $Dropout$(0.5) and $Dense$(1, activation='sigmoid')

BERT is a language model using the transformer encoder architecture to process tokens in text. BERT applies pre-training and fine-tuning. Pre-training is an unsupervised way on a general large corpus of text to create BERT model. Fine-tuning is a supervised training BERT model on a specific downstream task with relatively few labels, because the general linguistic patterns have already been learnt during pre-training. We used BERT algorithm implemented in [23], with parameters: $KerasLayer$(bert_en_uncased_preprocess_3, bert_en_uncased_L-12_H-768_A-12_4), $Dense$(1, activation='sigmoid') and $optimizer$='adam'.




**Figure 2:** sexual abuse single words in dark web forums

### 3.3. Training and Testing Datasets

Our first step is to create a labelled dataset that can be used for training or fine-tuning our classifier. The labelled dataset used for our study was collected and supplied by the company Web-IQ, which provided us with over 352,000 posts from 8 dark web forums in 2022, of which approximately 221,000 were in English.

Using a dictionary of 12,628 Sexual Abuse Phrases (SAPs) extracted from THORN project[4] and Web-IQ dark web forums[5], we were able to detect approximately 177,000 English posts with no SAP and approximately 44,000 English posts with at least one SAP. This provides us with a high level grouping of posts, but with refinement required to allow for CSAM posts that does not contain any SAPs, and vice versa. Figures 2 and 3 show the word clouds of single words and two-word-phrases related to sexual abuse, extracted from post contents in dark web forums. The size of each word in the clouds represents its frequency in the forums.

From the group of 177,000 posts with no SAP, experts randomly selected 2,000 non-CSAM posts and 500 CSAM posts. From the group of 44,000 posts with at least one SAP, experts randomly selected 2,000 CSAM posts and 100 non-CSAM posts. Ultimately, our manually labelled dataset contains 4,600 posts from the dark web, including 2,500 CSAM posts and 2,100 non-CSAM posts.

## 4. Experiment and Results

### 4.1. Experiment Setup and Quality Measures

The algorithms were implemented using Python 3.10, scikit-learn library 1.2.2 (for NB and SVM), keras library 1.1.2 run on top of tensorflow library 2.10.0 (for LSTM and BERT). All experiments were run under Windows 10

[4] https://www.thorn.org/
[5] https://web-iq.com/solutions/osint-on-premises

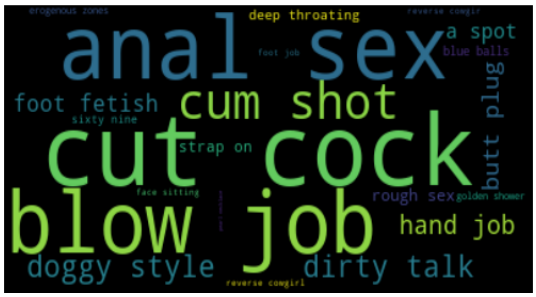



**Figure 3:** sexual abuse two-word-phrases in dark web forums

(64-bit) on a Dell laptop with an Intel Core i7 CPU (3.00 GHz) and 16 GB memory.

For the purpose of measuring the quality of the predicted classes of posts compared to the correct classes, we apply the most commonly used metrics namely accuracy, precision and recall ([24, 25]). The metrics are derived from four categories in the confusion matrix: True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) as follows:

- TP: Posts in which the algorithm predicted CSAM and the correct class was also CSAM.
- FP: Posts in which the algorithm predicted CSAM, but the correct class was non-CSAM.
- TN: Posts in which the algorithm predicted non-CSAM and the correct class was non-CSAM.
- FN: Posts in which the algorithm predicted non-CSAM, but the correct class was CSAM.

Accuracy (ACC) in binary classification is defined as a ratio between the correctly classified samples to the total number of samples: $ACC = \frac{TP+TN}{TP+FP+TN+FN}$. The accuracy puts the same emphasis on all these factors. However, when categorising pairs, there is usually a bias: it is much easier to identify true negatives correctly, due to their large number. So, we also look at the the precision $P = \frac{TP}{TP+FP}$ and the recall $R = \frac{TP}{TP+FN}$.

### 4.2. Results

Using the 5-fold cross-validation methodology, each experimental round includes a training set of 3,680 posts (2,000 CSAM and 1,680 non-CSAM) and a testing set of 920 posts (500 CSAM and 420 non-CSAM). Table 1 presents the average training time, average prediction time, average precision, average recall and average accuracy of four algorithm combinations as follows:

- NB: The training time and prediction time were 0.5 and 0.001 seconds, respectively. The precision was 76.1%, recall was 89% and accuracy was 78.8%.

**Table 1**
Average execution time and binary classification performance of the algorithms

| Results | Algorithms | | | |
|---|---|---|---|---|
| | NB | SVM | LSTM | BERT |
| Training time[1] | **0.5** | 1.8 | 32.5 | 4,261 |
| Prediction time[1] | **0.001** | 0.27 | 1.01 | 215.3 |
| True Positive | 445 | 421 | 428 | 415 |
| False Positive | 140 | 35 | 46 | 68 |
| True Negative | 280 | 385 | 374 | 352 |
| False Negative | 55 | 79 | 72 | 85 |
| Precision | 76.1% | **92.3%** | 90.2% | 86% |
| Recall | **89%** | 84.2% | 85.5% | 83% |
| Accuracy | 78.8% | **87.6%** | 87.1% | 83.4% |

[1] second.

- SVM: The training time and prediction time were 1.8 and 0.27 seconds, respectively. The precision was 92.3%, recall was 84.2% and accuracy was 87.6%.
- LSTM: The training time and prediction time were 32.5 and 1.01 seconds, respectively. The precision was 90.2%, recall was 85.5% and accuracy was 87.1%.
- BERT: The training time and prediction time 4,261 and 215.3 seconds, respectively. The precision was 86%, recall was 83% and accuracy was 83.4%.

The combination of the NB algorithm has the fastest execution time, taking only about 1 microsecond to detect a post on our laptop's capabilities. The second best performing algorithm is SVM, which takes about 0.3 milliseconds. These fast prediction times make our models well-suited for processing CSA text in real-time on social networks. Additionally, our models can run on edge devices with limited computational resources and power supply.

In terms of classification precision, the SVM combination performs the best with 92.3%, followed by LSTM and BERT as the second and third-best performers, respectively. Meanwhile, the NB combination has the highest recall rate of 89%, followed by LSTM as the second-best performer. When it comes to accuracy, SVM is the best with 87.6% which is slightly higher than LSTM with 87.1%. The BERT algorithm has long training and prediction times, and it is not suitable for binary classification of CSAM posts in dark web.

## 5. Conclusion and Future Work

We proposed and implemented a novel algorithm based on machine learning and natural language processing to automatically detect and classify CSAM text post content in dark web. In the experimental evaluation on the

dataset of 4,600 CSAM and non-CSAM posts with 5-fold cross-validation, the combination of NB algorithm performed the best in terms of classification recall and execution time. On the other hand, the SVM combination performed the best in terms of classification precision and accuracy, and was the second-best in execution time. The choice of NB and SVM depends on the specific goals and requirements of the CSAM classification task. NB is maximize the number of true positives which could be useful identifying and removing CSAM posts from online platforms to protect potential victims. On the other hand, SVM is minimize false positives which could be useful for identifying CSAM posts to extract information about potential predators and victims for investigative purposes.

As part of our future work, functional APIs will be implemented to create a user-friendly web application. Furthermore, we aim to leverage the metadata associated with CSAM posts to identify the characteristics, conversation and behaviours of perpetrators. This information can be valuable in developing more effective models for preventing and addressing CSA text on social media platforms. We also will recognise named entities in CSA text to supply important concepts for ML models [26].

## Acknowledgments

The paper is an extension of the long abstract [27] being part of the N-Light project which is funded by the Safe Online Initiative of End Violence and the Tech Coalition through the Tech Coalition Safe Online Research Fund (Grant number: 21-EVAC-0008-Technological University Dublin).

## References

[1] V. M. Ngo, C. Thorpe, C. N. Dang, S. Mckeever, Investigation, detection and prevention of online child sexual abuse materials: A comprehensive survey, in: the 16th IEEE International Conference on Computing and Communication Technologies (RIVF-2022), IEEE, 2022, pp. 707–713. doi:https://doi.org/10.1109/RIVF55975.2022.10013853.

[2] H. Lee, T. Ermakova, V. Ververis, B. Fabian, Detecting child sexual abuse material: A comprehensive survey, Forensic Science International: Digital Investigation 34 (2020) 301022. doi:10.1016/j.fsidi.2020.301022.

[3] A. Gangwar, V. González-Castro, E. Alegre, E. Fidalgo, Attm-cnn: Attention and metric learning based cnn for pornography, age and child sexual abuse (csa) detection in images, Neurocomputing 445 (2021) 81–104. doi:10.1016/j.neucom.2021.02.056.

[4] E. Guerra, B. G. Westlake, Detecting child sexual abuse images: traits of child sexual exploitation hosting and displaying websites, Child Abuse & Neglect 122 (2021) 105336. doi:10.1016/j.chiabu.2021.105336.

[5] P. Vitorino, S. Avila, M. Perez, A. Rocha, Leveraging deep neural networks to fight child pornography in the age of social media, Journal of Visual Communication and Image Representation 50 (2018) 303–313. doi:10.1016/j.jvcir.2017.12.005.

[6] C. Laranjeira, J. Macedo, S. Avila, J. Santos, Seeing without looking: Analysis pipeline for child sexual abuse datasets, in: the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT'22), ACM, 2022, p. 2189–2205. doi:10.1145/3531146.3534636.

[7] L. Struppek, D. Hintersdorf, D. Neider, K. Kersting, Learning to break deep perceptual hashing: The use case neuralhash, in: the 2022 ACM Conference on Fairness, Accountability, and Transparency, 2022, pp. 58–69. doi:10.1145/3531146.3533073.

[8] M. M. Islam, M. A. Uddin, L. Islam, A. Akter, S. Sharmin, U. K. Acharjee, Cyberbullying detection on social networks using machine learning approaches, in: the 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), IEEE, 2020, pp. 1–6. doi:10.1109/CSDE50874.2020.9411601.

[9] P. R. Borj, K. Raja, P. Bours, Detecting sexual predatory chats by perturbed data and balanced ensembles, in: the 2021 International Conference of the Biometrics Special Interest Group (BIOSIG), IEEE, 2021, pp. 1–5. doi:10.1109/BIOSIG52210.2021.9548303.

[10] M. Akhter, Z. Jiangbin, I. Naqvi, M. AbdelMajeed, T. Zia, Abusive language detection from social media comments using conventional machine learning and deep learning approaches, Multimedia Systems (2021) 1–16. doi:10.1007/s00530-021-00784-8.

[11] N. Cecillon, V. Labatut, R. Dufour, G. Linares, Graph embeddings for abusive language detection, SN Computer Science 2 (2021) 1–15. doi:10.1007/s42979-020-00413-7.

[12] C. H. Ngejane, J. H. Eloff, T. J. Sefara, V. N. Marivate, Digital forensics supported by machine learning for the detection of online sexual predatory chats, Forensic science international: Digital investigation 36 (2021) 301109. doi:10.1016/j.fsidi.2021.301109.

[13] G. Owen, N. Savage, The tor dark net, Chatham House (2015).

[14] M. Schäfer, M. Fuchs, M. Strohmeier, M. Engel,

M. Liechti, V. Lenders, Blackwidow: Monitoring the dark web for cyber security information, in: the 11th International Conference on Cyber Conflict (CyCon), volume 900, 2019, pp. 1–21. doi:10.23919/CYCON.2019.8756845.

[15] E. Kokolaki, E. Daskalaki, K. Psaroudaki, M. Christodoulaki, P. Fragopoulou, Investigating the dynamics of illegal online activity: The power of reporting, dark web, and related legislation, Computer Law & Security Review 38 (2020) 105440. doi:10.1016/j.clsr.2020.105440.

[16] S. Nazah, S. Huda, J. H. Abawajy, M. M. Hassan, An unsupervised model for identifying and characterizing dark web forums, IEEE Access 9 (2021) 112871–112892. doi:10.1109/ACCESS.2021.3103319.

[17] J. Woodhams, J. A. Kloess, B. Jose, C. E. Hamilton-Giachritsis, Characteristics and behaviors of anonymous users of dark web platforms suspected of child sexual offenses, Frontiers in Psychology 12 (2021) 623668. doi:10.3389/fpsyg.2021.623668.

[18] T. Tran, L. Nguyen, V. Ngo, Machine learning based english sentiment analysis, Journal of Science and Technology 52(4D) (2014) 142–155. doi:https://doi.org/10.48550/arXiv.1905.06643.

[19] S. J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach (4th Edition), Pearson, 2022.

[20] Scikit-learn, Multinomial naive bayes, https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB, 2023. Version 1.2.2, accessed April 01, 2023.

[21] Scikit-learn, C-support vector classification, https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html, 2023. Version 1.2.2, accessed April 01, 2023.

[22] Keras, Long short term memory, https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM, 2023. Version 1.1.2, accessed April 01, 2023.

[23] Keras, Text classsification with bert, https://www.tensorflow.org/text/tutorials/classify_text_with_bert, 2023. Version 1.1.2, accessed April 01, 2023.

[24] V. M. Ngo, T. V. T. Duong, T. B. T. Nguyen, P. T. Nguyen, O. Conlan, An efficient classification algorithm for traditional textile patterns from different cultures based on structures, Journal on Computing and Cultural Heritage (JOCCH) 14(4) (2021) 1–22. doi:https://doi.org/10.1145/3465381.

[25] A. Tharwat, Classification assessment methods, Applied Computing and Informatics 17(1) (2021) 168–192. doi:https://doi.org/10.1016/j.aci.2018.08.003.

[26] V. M. Ngo, G. Munnelly, F. Orlandi, P. Crooks, D. O'Sullivan, O. Conlan, A semantic search engine for historical handwritten document images, in: G. Berget, M. M. Hall, D. Brenn, S. Kumpulainen (Eds.), Linking Theory and Practice of Digital Libraries, LNCS, vol. 12866, Springer, 2021, pp. 60–65. doi:https://doi.org/10.1007/978-3-030-86324-1_7.

[27] S. Mckeever, C. Thorpe, V. M. Ngo, Determining child sexual abuse posts based on artificial intelligence, in: the 2023 International Society for the Prevention of Child Abuse & Neglect Congress (ISPCAN-2023), Edinburgh, Scotland, UK, September 24-27, 2023, 2023, pp. 1–4. doi:https://doi.org/10.21427/S3GQ-3536.