Cybersecurity-Related Tweet Classification by Explainable Deep Learning

Giacomo Iadarola[1], Fabio Martinelli[1], Francesco Mercaldo[2,1], Luca Petrillo[1] and Antonella Santone[2]
[1]*Institute for Informatics and Telematics, National Research Council of Italy (CNR), Pisa, Italy*
[2]*Department of Medicine and Health Sciences "Vincenzo Tiberio", University of Molise, Campobasso, Italy*

Keywords: Unsupervised Classification, X, CVE, Clustering, Neural Networks, Deep Learning.

Abstract: The use of computing devices such as computers, smartphones, and IoT systems has increased exponentially over the past decade. Given this great expansion, it becomes important to identify and correct the vulnerabilities present to ensure the safety of systems and people. Over time, many official entities have emerged that publish news about these vulnerabilities; in addition to these sources, however, social media, such as X (commonly referred to by its former name Twitter), can be used to learn about these vulnerabilities even before they are made public. The goal of this work is to create clusters of tweets, which are grouped according to the description of the vulnerability in the relevant text. This process is accomplished through the use of a combination of two Doc2Vec models and a variant of a BERT model, which allow a text document to be converted into its numerical representation. Once this step was completed, K-means, an unsupervised model for performing clustering, was used, which through this numerical representation obtained in the previous step, groups tweets based on text content.

1 INTRODUCTION

Our daily lives are now constantly influenced by social media due to the instant access and rapid creation and sharing of information. Platforms such as Facebook, Instagram, and X have influenced contemporary society, and over time different types of social media have been created based on the content they offer.

Given the rapid advancement of technology, it is clear that researchers and companies around the world are continuously investigating everything in this field, and one of the most critical aspects is the vulnerabilities of computer systems. Enisa (ENISA, 2022), the European Union's cybersecurity agency, estimates that 60% of affected organizations may have paid ransom demands triggered by a ransomware attack, while 66 zero-day vulnerabilities were revealed in 2021 alone.

For all of these reasons, it is critical to keep track of vulnerabilities that are discovered over time and the Common Vulnerabilities and Exposures (CVEs) system, which provides a reference method for publicly known vulnerabilities and exposures for everything related to cybersecurity; each newly discovered vulnerability has a CVE ID, which provides a reliable way for users to recognize particular vulnerabilities and coordinate the creation of security tools and solutions.

Given these characteristics, it is possible to use the information that social network users exchange to predict the identification of new vulnerabilities or even just to understand how these cybersecurity problems affect people.

With regard to this work, given the motivations previously described, X was used as the main source not only to identify tweets related to cybersecurity topics, but also to create clusters of these tweets by grouping them according to the CVE discussed. To achieve this goal, a large set of tweets was collected from the official X API and a set of CVEs using the API offered by NVD (National Vulnerability Database), which is a database where all newly discovered vulnerabilities are collected. A preprocessing phase was applied to these two sets to facilitate learning tasks. Two variants of a Doc2Vec model (Le and Mikolov, 2014) and a modification of the pre-trained BERT network using Siamese network structures and triplets (SBERT) (Reimers and Gurevych, 2019) were used to perform clustering and produce document em-

Iadarola, G., Martinelli, F., Mercaldo, F., Petrillo, L. and Santone, A.
Cybersecurity-Related Tweet Classification by Explainable Deep Learning.
DOI: 10.5220/0012411100003648

In *Proceedings of the 10th International Conference on Information Systems Security and Privacy (ICISSP 2024)*, pages 438-445
ISBN: 978-989-758-683-5; ISSN: 2184-4356

beddings that accurately represent the semantic meaning of a text. Both variants of Doc2Vec were trained using both tweets and CVE descriptions processed in the previous step to clustered a set of unseen tweets.

2 RELATED WORK

The amount of work and study done to extract cybersecurity data from X has significantly increased in recent years. On the basis of their content, tweets were understood and categorized using a variety of models, methods, and datasets.

Using a novelty detection approach, Le et al. (Le et al., 2019) suggested a method for automatically gathering information on cyber threats from X. To achieve this, the authors collected a specially constructed dataset of tweets from 50 influential cybersecurity-related accounts over the course of twelve months (in 2018) and used all CVE descriptions released in 2017 to train their classifier.

A framework for the unsupervised classification and data mining of tweets about cyber vulnerabilities was presented by Alperin et al. (Alperin et al., 2021). The authors evaluated two unsupervised machine learning techniques LDA and BART to filter tweets based on cybersecurity relevance using labelled datasets of tweets.

Deep neural networks (Huang et al., 2021), (Huang et al., 2022), (Huang et al., 2023), (Zhou et al., 2021) are used in a new tool created by Dionísio et al. (Dionísio et al., 2019) to process cybersecurity data obtained from X. Specifically, they used a convolutional neural network (CNN) that identifies tweets containing security information about the assets of an IT infrastructure, while the BiLSTM (bidirectional long short-term memory network) extracts named entities from these tweets to form a security alert or compiles a compromise indicator, with a pipeline formed by these two models to classify the tweets.

Previously described works aim to classify tweets based on the relevance of the cybersecurity topic, while this study aims to create clusters where tweets are grouped based on similarity to a given CVE. Moreover, the latter uses a labelled dataset using both supervised and unsupervised models, in contrast to our work where a dataset is constructed specifically for this task that does not require labeling.

3 THE METHOD

As mentioned earlier, the goal of this work is to analyze a collection of tweets to extract vector representations of them. These were obtained through the use of NLP models for representing text in document embeddings. Two variants of the Doc2Vec (Le and Mikolov, 2014) model and one variant of the BERT (Devlin et al., 2018) model were used. Once these representations were obtained, K-means, an algorithm for performing clustering, was used to create groups of tweets based on their similarity and from these extract only those groups of tweets in which a description of a vulnerability is present. Figure 1 shows a simplified schematic of the workflow.

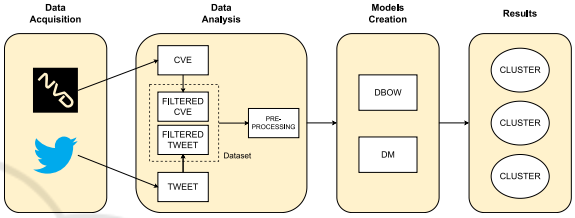

Figure 1: General framework architecture.

3.1 Data Acquisition

In the tweet collection phase, the public API provided by X [1] was used, which allows tweets to be collected daily up to a maximum of 100.000.

3.2 Data Analysis

3.2.1 Filtering Tweets

Once collected, the tweets were divided into relevant and irrelevant. Specifically, only those tweets that contained a keyword representing a specific ***CVE-ID*** (e.g., CVE-2021-41819) were grouped together. This choice was driven primarily by two reasons:

1. Through this filter it was possible to create a robust dataset on which to train two different versions of Doc2Vec. Furthermore, this keyword search made it possible to collect only those tweets that actually contained an explicit description of a vulnerability. In this way, it was possible to exclude those ambiguous texts. An example that provides a better understanding of the issue is the word "*virus*" which can refer to both the medical and cybersecurity fields;
2. Through this phase, in addition, all ***CVE-ID*** were collected to find the related vulnerability descriptions in a second phase. The motivation behind

[1] https://developer.twitter.com/en/products/twitter-api

```
{
  "tweet text": "NEW: CVE identified
  a deserialization issue that was
  present in Apache Chainsaw. Prior
  to Chainsaw V2.0 Chainsaw was a
  component of Apache Log4j 1.2.x
  where the same issue exists.
  https://t.co/edQocRcw9W"
  ------------------------------------
  "CVE description": "CVE-2020-9493
  identified a deserialization issue
  that was present in Apache Chainsaw.
  Prior to Chainsaw V2.0 Chainsaw was a
  component of Apache Log4j 1.2.x where
  the same issue exists"
}
```

Listing 1: Comparison between a tweet containing the official description of a vulnerability in Apache and the official description of "CVE-2022-23307" assigned to it.

this choice was driven by a preliminary analysis in which it was noticed that some X accounts publish tweets containing official vulnerability descriptions; an example can be seen in Listing 1. So by collecting and training models with these official descriptions as well, the goal was set to detect these types of tweets.

3.2.2 CVE Acquisitionn

As mentioned earlier, during the analysis of the tweets, all CVE-IDs identified within the tweets examined were collected. Through the use of NVD's public API [2], official descriptions related to the CVE-IDs just mentioned were retrieved.

3.3 Preprocessing

Processing natural language is particularly difficult and complex because of its inherent characteristics of ambiguity. Therefore, during this phase, text cleaning and simplification operations were carried out. First, only English-language tweets were analyzed and processed; in addition, for each one, all URLs in the text were removed. Since X allows users to interact with other users through mentions, these were also removed. Finally, the hashtags present were removed. Given the use of different models, it was necessary to perform different preprocessing operations based on them. Specifically regarding the data used for Doc2Vec, all text was converted to lower case and split into tokens. While for the SBERT model, the text was only converted to lower case, without the need to divide it into tokens.

3.4 Models

Once the tweets were divided into relevant and irrelevant, the relevant ones and the official vulnerability descriptions were used to train two different versions of a Doc2Vec model. These are two different strategies for representing text in document embeddings: one using the PV-DBOW (Distributed Bag of Words Version of Paragraph Vector) and one via the PV-DM (Distributed Memory Version of Paragraph Vector).

The PV-DBOW model considers a paragraph as an unordered set of words and disregards the word order within the paragraph. Based on the context words in the paragraph, it guesses the target words, which are randomly selected from the paragraph. In contrast, the PV-DM model takes into account the paragraph's word order. Using the preceding words and the paragraph vector—a distinct vector representation for every paragraph—it attempts to anticipate the following word in a series. Both variants use an additional vector, called Paragraph ID, which is used as additional context for the specific document. This step was designed to make a comparison between the two text representation techniques.

In addition to the two versions just mentioned, we relied on a pre-trained version of the BERT model. Specifically, a Sentence Transformer model was used that maps sentences and paragraphs into a dense vector space of 768 dimensions and can be used for tasks such as clustering or semantic search. It is a MiniLM model tuned to a large dataset with over 1 billion training pairs.

3.4.1 Hyperparameters Tuning

Concerning Doc2Vec models, a hyperparameter tuning step was performed. To train this model it is possible to specify some parameters in addition to the one for the mode of representation of document embeddings. These parameters were obtained through a preliminary testing phase and a customized implementation of the random search approach, taking cues from the work of Jey Han Lau et al (Lau and Baldwin, 2016). Prior to the training phase of these two models, the dataset (consisting of tweets with a vulnerability description and CVE descriptions) was divided into training, testing, and validation set. During this phase, the validation set was used.

[2]https://nvd.nist.gov/developers/vulnerabilities

3.5 Clusters Creation

For the creation of the clusters, as mentioned above, the K-means model was used. As for the Doc2Vec models, once the training and hyperparameters tuning phases were completed they were concatenated into a single model. Through the latter, document embeddings related to the new unseen tweets were obtained. The same tweets were also submitted to the SBERT model to obtain the vector representations. Through these new data, two variants of the K-means were trained (one with the document emdebbings obtained from the concatenation of the two Doc2Vec models and one with the document embeddings obtained from SBERT).

4 EXPERIMENTAL RESULTS

4.1 Data Acquisition and Filtering

During this phase, useful tweets for analysis were retrieved through X's public API. Once this collection of tweets was retrieved, they were divided into two different sets. Specifically, a search was conducted in the text of each tweet for a keyword corresponding to a CVE-ID (e.g., CVE-2020-9493). Each time it is detected in a text the tweet is marked as relevant, with the corresponding CVE-ID.

4.2 Dataset

The work is based on the analysis and extrapolation of a dataset comprising two types of data. The first related to tweets collected through X's public API for a period ranging from 01/11/2021 - 14/11/2022 for a total of 37.308.818 tweets. The second related to the CVEs identified in the tweets resulted in the collection of a total of 32.409 unique descriptions.

After the filtering phase, 227.457 tweets containing a description of a vulnerability and traceable to a CVE-ID were identified. Table 1 provides a summary of these data.

4.3 Preprocessing

After filtering tweets into relevant or not relevant and collecting CVEs based on those identified in the tweets, a preprocessing phase was carried out. For each tweet analyzed, the language was detected and only those in English were analyzed. In addition, any URLs were removed from each text and all characters other than [a-z] were removed. Within text messages, X allows users to interact with other users or brands through the use of the "@" symbol and to use hashtags, i.e., a combination of keywords or phrases preceded by the "#" symbol, excluding spaces or punctuation; during this phase these were also removed. A final operation was to remove in the case of the relevant tweets, the presence of the keywords (CVE-IDs) precendently mentioned.

Table 1: Dataset elements after data collection, filtering and preprocessing.

Tweets collection	
Time period	Number of tweets
01/11/2021 - 14/11/2022	37.308.818
Tweets filtering	
Type of data	Number of elements
Relevant tweets	244.364
CVE	32.409
Data preprocessing	
Type of data	Number of elements
Tweets	21.056.076
Relevant tweets	227.457
CVE	32.409

For the CVE descriptions the cases to be considered are different from those of the tweets in that the vulnerability descriptions are reported in more technical language and usually do not contain misleading phrases but more controlled ones. In addition, these were all retrieved in the English language. So the operations were to remove the special characters and any versions of the described packages.

4.4 Dataset Split

In this phase, the dataset was created to carry out the training and evaluation phase of the two Doc2Vec models. As discussed in the previous sections, the dataset consisted of all filtered tweets (with the presence of a keyword CVE-ID) and all collected CVE descriptions, the latter was divided into training, test, and validation set using the ratio of 80%, 10%, 10%, respectively. When this was done so that at least one tweet or CVE referable to a CVE-ID was included in the training set. This was done to prevent the model from having no knowledge of a CVE-ID at the time it will be evaluated in the later stages; despite this operation, it was done so that the division still retains the ratio described above. The Table 2 provides a summary of what has just been described.

Table 2: Dataset elements for the models.

Type	Number of elements
Training set	208.332
Validation set	24.198
Test set	27.336

4.5 Models Creation

During this phase, the two Doc2Vec models mentioned so far were created. The tweets marked as relevant (i.e., those with a certain description of a vulnerability) and the descriptions of the CVEs retrieved in the previous steps were used to perform the training. The same data mentioned above were used for both. All useful data can be found in Section 4.2.

4.5.1 Hyperparameters Tuning

For both models, i.e., the Paragraph Vector Distributed Memory model (PV-DM) and the Paragraph Vector Distributed Word Bag model (PV-DBOW), it is possible to define a number of parameters such as epochs, i.e., the number of iterations that the model goes through on the training corpus, or the negative parameter (a number) that if given triggers negative sampling, i.e., how many ”nonsignificant words” are to be drawn during training and that goes to affect the quality of the document and word vectors learned. In order to choose these values, a preliminary study was carried out on both models and also some indications from the study by Jey Han Lau et al (Lau and Baldwin, 2016) were followed. Starting from these parameters, a customized random search method was implemented to search for the parameters that yielded the best results. During this stage 5 rounds of random search were performed where for each round 10 configurations of Doc2Vec hyperparameters are randomly sampled to search for the best accuracy, the graph 2 reports for each of the 5 rounds the best accuracy obtained. Once both models were evaluated through a customized evaluation technique and described in Section 4.5.2, an accuracy of 41.7% was obtained for the DBOW while 34.4% was obtained for the DM.

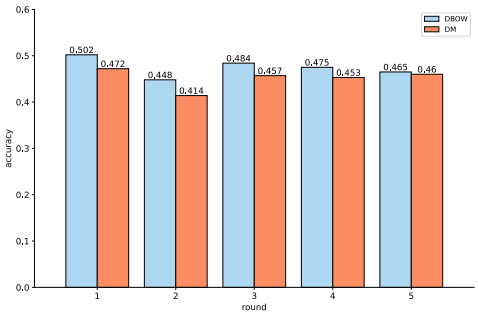

Figure 2: Accuracy of the 5 random sampling rounds of 10 hyperparameter configurations.

4.5.2 Evaluation

In this step, both models created in the previous step were evaluated. To make a prediction using Doc2Vec, an unseen tweet is submitted to the model and it returns the most similar tweet it was trained on, an operation performed by Doc2Vec through the calculation of cosine similarity. To assess whether or not the model provided a correct result, it was verified that the CVE-ID of the new unseen tweet matched the CVE-ID of the tweet returned by the model. This simple expedient made it possible to evaluate performance both after the hyperparameters were tuned and after they were created with the correct hyperparameters.

4.6 Clusters Creation

As mentioned earlier to perform the clustering of these tweets, K-means was used. To perform this operation from the previously collected set of tweets, 176.431 elements were randomly sampled. As for Doc2Vec, it was decided to follow the approach proposed by the work of Dai et al. (Dai et al., 2015) and then concatenate the two versions of the model. Once this concatenation was done the sub-sample of these unseen tweets was submitted to this new model and from which the document embeddings were obtained. The same sample of tweets was submitted to the SBERT model to obtain, again, the document embeddings related to these tweets.

To perform K-means training, the number of clusters to be created by the algorithm must be specified. Since it is not possible to know this value regardless, there are some techniques for choosing it: silhouette analysis and the elbow method.

The silhouette analysis measures the separation distance between clusters and provides a way to visually assess the number of clusters. It calculates a silhouette coefficient for each data point, ranging from -1 to 1. Higher values indicate better defined clusters, while lower values indicate overlapping clusters or poorly classified points. The elbow method calculates the sum of squares within the cluster (WCSS) for different values of k (the number of clusters). It plots the WCSS against the number of clusters and looks for the ”elbow” point at which the rate of decrease in WCSS slows down significantly. This point is considered the optimal number of clusters.

Initially, given the number of tweets, K-means was tested for both models and with both methods with a number of clusters equal to 100. In Figures [3, 4] it is possible to observe the results for the Doc2Vec model obtained from the combination of the Distributed Memory model of Paragraph Vec-

tors (PV-DM) and the Distributed Word Bag model of Paragraph Vectors (PV-DBOW). While in Figures [5, 6] the results with the vectors obtained through the SBERT model can be consulted.

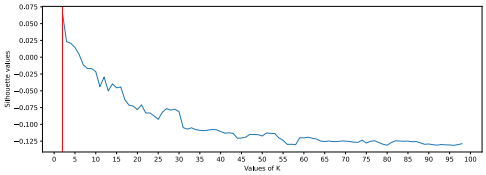

Figure 3: Silhouette values obtained from K-means by document embeddings extrapolated from Doc2Vec for 100 clusters.

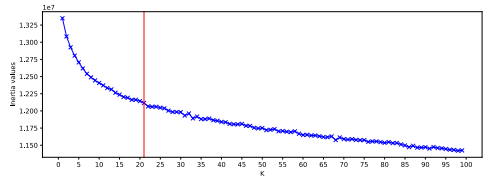

Figure 4: Values of WCSS obtained from K-means by document embeddings extrapolated from Doc2Vec for 100 clusters.

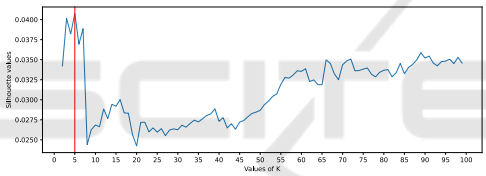

Figure 5: Silhouette values obtained from K-means by document embeddings extrapolated from SBERT for 100 clusters.

Regarding the Doc2Vec results obtained through Silhouette analysis and shown in Figures [3, 7], given the number of tweets (176.431) it was not deemed useful to create 2 clusters, as there would not be a clear distinction in the topics covered. Therefore, the training of the K-means model with 21 clusters was directly carried out as emerged from the Elbow method analysis and present in Figure 4.

Instead as revealed by the results through the two proposed methods (Silhouette analysis and Elbow method) regarding the SBERT data it was decided to make two attempts: one by creating a number of clusters equal to 5 as visible in Figure 5 and one with a number of clusters equal to 18 as visible in Figure 6. The results obtained during this phase can be found in Section 4.7.

4.7 Results

This section discusses the results obtained at the conclusion of this work. Table 3 presents the results ob-

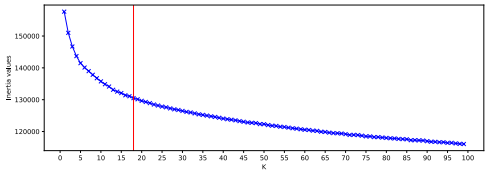

Figure 6: Values of WCSS obtained from K-means by document embeddings extrapolated from SBERT for 100 clusters.

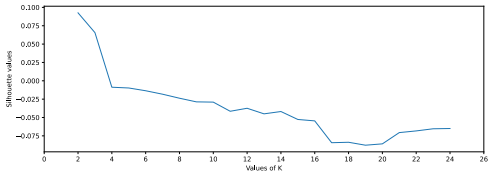

Figure 7: Silhouette values obtained from K-means by document embeddings extrapolated from Doc2Vec for number of clusters restricted between 2 and 25.

tained from training the K-means model with different numbers of clusters that emerged during the analysis described in Section 4.6. As visible from the low coefficient of the Silhouette, it is understood that the clusters obtained through Doc2Vec are overlapping and not well defined. This was also evident from a manual analysis conducted on the results. Nevertheless, some clusters were identified in which a macOS malware was described, this is to make it clear that some clusters are distinct well despite the noisiness of the tweets. Table 4 shows some examples of tweets found.

The results obtained through SBERT are slightly better in both cluster attempts made. This indicates that this model is better able to represent the text of tweets in document embeddings. During a manual analysis, it was found that the model was able to optimally cluster tweets regarding descriptions of some CVEs as shown in Table 5 and malware that plagued one of the largest propane distributors in North America. In addition, it was noted that numerous tweets from users reporting a phishing scam carried out via Telegram were clustered in one cluster.

5 DISCUSSIONS

The analysis carried out in this work showed that the concatenation of the two Doc2Vec models, manage to correctly identify tweets that contain a description of a vulnerability, even those that do not explicitly contain the keyword CVE, this is because the latter was removed through a pre-processing step. Regarding the SBERT model, it was definitely better than the Doc2Vec model built in this work, as it is a MiniLM

Table 3: Results obtained through the K-means model with document embeddings extracted through Doc2Vec and SBERT.

Model	Number of clusters	Silhouette coefficient	Inertia value
Doc2Vec	21	-0.07	12168827.45
SBERT	5	0.04	141530.94
SBERT	18	0.03	131214.25

Table 4: Tweets obtained via Doc2Vec and clustered in the same cluster reporting a description of malware that has sharpened MacOs with related article links.

Tweet
I will take Apple Christmas bug for $100. Expert Details macOSBug That Could Let Malware Bypass Gatekeeper Security https://t.co/vFTqwUQTRb
Expert Details macOS Bug That Could Let Malware Bypass Gatekeeper Security https://t.co/gGGL391DzD https://t.co/XaQtorXne4

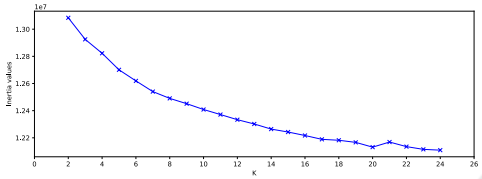

Figure 8: Values of WCSS obtained from K-means by document embeddings extrapolated from Doc2Vec for number of clusters restricted between 2 and 25.

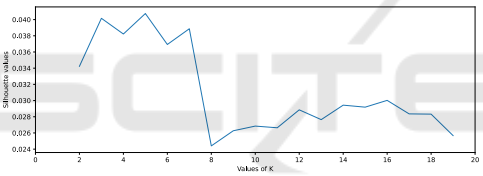

Figure 9: Silhouette values obtained from K-means by document embeddings extrapolated from SBERT for number of clusters restricted between 2 and 20.

model tuned on a large dataset with more than 1 billion training pairs. This factor ensured more accurate results comparing with the latter, which, however, was trained on a fairly small dataset (259.866 tweets and CVE descriptions). Despite these considerations the ability to create clusters appears to be very promising and has ample room for improvement. As described in the previous section in some clusters, tweets containing keywords such as "malware," "ransomware," or "CVE" were clustered correctly. In many other cases the clusters created had correctly clustered tweets but which had no relevance to the theme researched in this paper. An example of a tweet placed in these clusters is one containing the word "spam," which, however, offers no cybersecurity information: *"timeline is dead, i have to spam, i think"*.

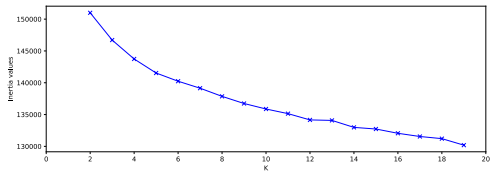

Figure 10: Values of WCSS obtained from K-means by document embeddings extrapolated from SBERT for number of clusters restricted between 2 and 20.

6 CONCLUSIONS AND FUTURE WORKS

The goal of this work was to collect and create clusters of tweets based on the described vulnerability. To achieve this goal, 37.308.818 tweets were collected through the X API. Through a filtering step, tweets that contained an explicit mention of the CVE keyword were identified. For each extracted keyword, the description of the related vulnerability was retrieved from the NVD API to form a consistent dataset consisting of the filtered tweets and the CVEs themselves. Through this dataset, two different versions of a Doc2Vec model were trained. These two models were concatenated into a new model to extract vector representations of the data. In addition, a variant of the BERT model (SBERT) was used to obtain the document embeddings and make a comparison between the two models. To create the tweet clusters, the K-means model trained with the document embeddings extracted from the concatenation of the two versions of Doc2Vec and the document embeddings extracted from the SBERT model was used. The results of this work show that currently the SBERT model performs better than the ad-hoc created model. This is because models like Doc2Vec require much larger datasets, as demonstrated by the work of Andrew M. Dai et al. (Dai et al., 2015). The authors used a corpus taken from the online encyclopedia Wikipedia composed of

Table 5: Tweets obtained through SBERT showing how tweets containing a description of a CVE were merged into a cluster.

Tweet
CVE-2022-30161 : #Windows Lightweight Directory Access Protocol LDAP Remote Code Execution Vulnerability. This CVE ID is unique from CVE-2022-30139.... https://t.co/pQY3uvtcJH
Attackers could exploit a now-patched spoofing vulnerability (CVE-2022-35829 aka FabriXss) in Service Fabric... https://t.co/LoyRYEmnXZ https://t.co/YTUo4gssFH

4.490.000 article-text corpus and one of 886.000 full arXiv papers. The filtering applied in this work ensures consistent data that surely includes a text that mentions a CVE. However, the model would also need to be trained with texts that are more general but still related to the vulnerability domain. This improvement would guarantee a broader set of results.

In addition, the creation of the clusters using the K-means model should be explored in depth, optimally considering the initialization parameters of the model. Choices could fall on selecting the initial centroids of the clusters by sampling based on an empirical probability distribution of the points' contribution to the overall inertia, rather than choosing the clusters randomly from the data for the initial centroids.

Also since in this specific case the initial number of clusters is not known a priori, hierarchical clustering could be considered. In fact, this type of algorithm returns as the result of the analysis a dendrogram that starts with each data point as a separate cluster and then proceeds to join the closest cluster pairs until all data points belong to a single cluster, thus allowing the optimal number to be reached.

ACKNOWLEDGEMENTS

This work has been partially supported by EU DUCA, EU CyberSecPro, SYNAPSE, PTR 22-24 P2.01 (Cybersecurity) and SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the EU - NextGenerationEU projects.

REFERENCES

Alperin, K., Joback, E., Shing, L., and Elkin, G. (2021). A framework for unsupervised classificiation and data mining of tweets about cyber vulnerabilities. *arXiv preprint arXiv:2104.11695*.

Dai, A. M., Olah, C., and Le, Q. V. (2015). Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dionísio, N., Alves, F., Ferreira, P. M., and Bessani, A. (2019). Cyberthreat detection from twitter using deep neural networks. In *2019 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

ENISA (2022). Enisa threat landscape 2022. In *https://www.enisa.europa.eu/publications/enisa-threat-landscape-2022*.

Huang, P., He, P., Tian, S., Ma, M., Feng, P., Xiao, H., Mercaldo, F., Santone, A., and Qin, J. (2022). A vit-amc network with adaptive model fusion and multiobjective optimization for interpretable laryngeal tumor grading from histopathological images. *IEEE Transactions on Medical Imaging*, 42(1):15–28.

Huang, P., Tan, X., Zhou, X., Liu, S., Mercaldo, F., and Santone, A. (2021). Fabnet: fusion attention block and transfer learning for laryngeal cancer tumor grading in p63 ihc histopathology images. *IEEE Journal of Biomedical and Health Informatics*, 26(4):1696–1707.

Huang, P., Zhou, X., He, P., Feng, P., Tian, S., Sun, Y., Mercaldo, F., Santone, A., Qin, J., and Xiao, H. (2023). Interpretable laryngeal tumor grading of histopathological images via depth domain adaptive network with integration gradient cam and priori experience-guided attention. *Computers in Biology and Medicine*, 154:106447.

Lau, J. H. and Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*.

Le, B. D., Wang, G., Nasim, M., and Babar, A. (2019). Gathering cyber threat intelligence from twitter using novelty classification. *arXiv preprint arXiv:1907.01755*.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Zhou, X., Tang, C., Huang, P., Mercaldo, F., Santone, A., and Shao, Y. (2021). Lpcanet: classification of laryngeal cancer histopathological images using a cnn with position attention and channel attention mechanisms. *Interdisciplinary Sciences: Computational Life Sciences*, 13(4):666–682.