Towards Selecting Informative Content for Cyber Threat Intelligence

Panos Panagiotou*†, Christos Iliou*, Konstantinos Apostolou*, Theodora Tsikrika*, Stefanos Vrochidis*,
Periklis Chatzimisios† , and Ioannis Kompatsiaris*
*Information Technologies Institute
CERTH
Thessaloniki, Greece
{panagiotou,iliouchristos,konapost,theodora.tsikrika,stefanos,ikom}@iti.gr
†School of Science & Technology
International Hellenic University
Thessaloniki, Greece
pchatzimisios@ihu.gr

***Abstract*—Nowadays, there is an increasing need for cyber security professionals to make use of tools that automatically extract Cyber Threat Intelligence (CTI) relying on information collected from relevant blogs and news sources that are publicly available. When such sources are used, an important part of the CTI extraction process is content selection, in which pages that do not contain CTI-related information should be filtered out. For this task, we apply supervised machine learning-based text classification techniques, trained on a new dataset created for the purposes of this work. Furthermore, we show in practice the importance of a good content selection process in a commonly used CTI extraction pipeline, by inspecting the results of the named entity recognition (NER) process that normally follows.**

***Index Terms*—cyber security, cyber threat intelligence, content selection, text classification, machine learning, NER**

I. Introduction

In their continuous efforts to protect organisations against the ever-growing cyber attacks landscape, security professionals have realized that it is insufficient to collect and analyze data solely from the internal networks and hosts in search of threat indicators. Conversely, it is equally important to obtain information about emerging threats and 0-day vulnerabilities from online sources, such as cyber security-related Surface and Dark Web sites, social media, and online forums. In fact, it is the information extracted from such "external" sources that, when combined with internal data, will provide high quality Cyber Threat Intelligence (CTI), corresponding to evidence-based knowledge about potential cyber threats that can be used to inform decisions regarding the response to them.

Open-Source Intelligence (OSINT) in the cyber security field concerns the cyber security-related intelligence collected from publicly available sources. Most of the literature focuses on collecting data from Twitter [1]–[4], while Dark Web forums have also attracted some interest [5]–[7]. In contrast to typically short social media posts, longer articles on sources such as technical blogs or cyber security-related news web sites naturally contain more information about threats, attacks, and vulnerabilities; this could actually facilitate the correlation of information and, thus, lead to better quality of extracted CTI. For example, technical blogs have been considered a good resource for extracting Indicators of Compromise (IOCs) [8], i.e., artifacts of an intrusion (e.g., virus signatures, IPs, etc); blogs have also been part of input sources mapped to advanced representations of threats and vulnerabilities information in graphs for generating alerts [9], [10].

When the external sources of information contain unstructured text, a common real-world CTI extraction framework is the one depicted in Figure 1. Consider the case of an Early Warning System (EWS), where the process starts with web sites crawling and monitoring. At first, a content selection process should be applied in order to filter out web pages that do not contain CTI information. This filtering is naturally needed, considering the fact that, even for cyber security-related sites, not all articles contain CTI-related information. For example, some articles may discuss anti-malware software, while others may advertise cyber-security conferences.

After the content selection step, CTI-related terms and phrases are usually identified with Named Entity Recognition (NER) techniques [6], [11], while relations between them may also be identified with Relation Extraction (RE) techniques, in order to facilitate the production of graph representations [9]. After this process, correlation techniques can be applied in order to correlate information about threats, attacks, and vulnerabilities and produce actionable intelligence including, e.g.,, information about related attacks and attack groups.

In this paper, we focus on the content selection of information from sources such as blogs and cyber security-related news web sites, emphasizing its importance as the first part of a CTI extraction framework. To this end, text classification techniques based on supervised machine learning methods are considered. Such methods require though appropriate datasets with labelled documents in order for a classification model to be learnt; such datasets are though scarce in the cyber security domain. Our goal is to build effective content selection approaches by also constructing an appropriate dataset and show that a good filtering stage improves the relevance of the cyber security-related named entities that are extracted.

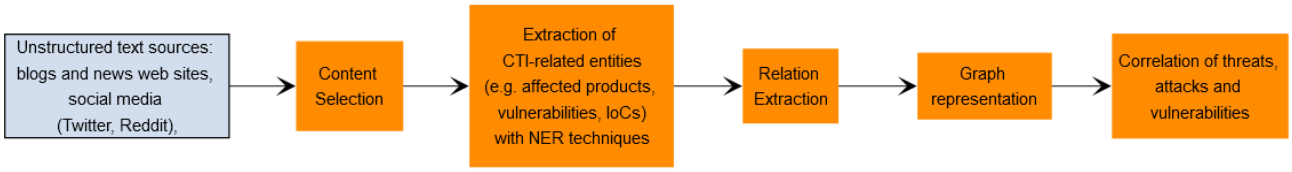

Fig. 1. A common CTI extraction pipeline.

The main contributions of this work are as follows. First, we build a new dataset of annotated web site articles for the purposes of text classification in order to facilitate the content selection process. To the best of our knowledge, this is the first published dataset[1] that is specifically tailored to content selection for articles longer than typical social media posts. Furthermore, the labelling has been performed for the specific purposes of CTI extraction, since the articles have been labelled by the annotators into three classes: (i) articles that are not related to cyber security, (ii) those that are cyber security-related but do not contain CTI-related information, and (iii) CTI-related articles. Next, we perform an experimental study on the effectiveness of text classification as a filtering step; to this end, we apply and evaluate supervised machine learning-based techniques on this newly constructed dataset and illustrate how the classification process helps in improving the quality of the extracted information. To this end, we apply state-of-the-art NER techniques on the classified documents to investigate the relevance of the named entities extracted from the different classes of web documents. We illustrate that those entities that are extracted from the documents that are not classified as CTI-related are also not related to CTI.

II. Related work

The implementation of cyber security-specific filters using text classification models has attracted some interest in recent years. A cyber security-related classification model [12] based on the BERT language representation model [13] reportedly achieved a precision of 0.92 and a recall of 0.90, in a test set comprising of multiple types of textual data, such as Reddit and Stack Exchange discussions, as well as more formal sources such as security news outlet RSS feeds. However, the part of their test set that contains no cyber security-related data consists only of Reddit discussions. In [14], where classification was performed for data from social media sources (Stack Exchange, Reddit and Twitter), this was done separately for each source and preliminary experiments showed that the classification performance degraded when a classifier was trained on one corpus and tested on another.

Among the published research work that concentrates on content selection from hacker forums, [15] reports good results with both Support Vector Machine (SVM) and Convolutional Neural Network (CNN) based classification solution, although their labelling was mainly based on the presence of keywords. Moreover, in [16], the detection of hacker communication is performed with machine learning methods (SVM), for deep web hacker forums and Twitter separately, while they also measure the extent to which the concept drift problem affects the performance of the classifiers in the long term.

For Twitter, content selection was typically performed by simple filtering approaches with keywords and identified security-related accounts (such as security analysts, vendors, researchers); recently machine learning-based classification techniques have also been applied [1], [17], [18]. In [17], a dataset is provided, while in [18], the supervised learning approach is novelty classification, where examples of only one class (the cyber security-related) are used in the training phase.

Among previous work that considers technical blogs as resources for CTI-related analysis, [9] and [10] do not mention any selection - filtering process, while in [8], the selection of pages that are likely to contain IoCs is performed with Natural Language Processing (NLP) techniques, namely content term extraction. We apply machine learning-based classification techniques for content selection and show the importance of a good-quality content selection stage. The existence of appropriately annotated text datasets that could be used for training classification models is currently sparse in the domain of cyber security. To the best of our knowledge, our work is the only one that provides a dataset for text classification for articles like those found in blogs and news web sites.

III. Data collection and annotation

We model content selection as a text classification problem. The lack of available datasets for training supervised machine learning classification techniques for the case of blogs and news web sites has lead us to construct our own dataset.

Our dataset consists of a total of 920 web pages collected from nine web sites (see Table I), using the respective sitemaps or via crawling. Among them, there are six cyber security-related web sites, two technology-related, and one containing generic news and articles. The technology-related sources were included because they contain a lot of content about software and hardware and, thus, the language used is closer to the language expected in a cyber security-related article; so these articles are expected to be very informative for the training of a machine learning-based text classifier. Furthermore, if the content selection stage is performed as part of a crawling process, it is quite possible that such sources will be encountered.

Due to the high number of pages that were initially downloaded, only a subset is finally included in the dataset, selected via random sampling. For those sites that already categorise their articles into topics, most topics are represented in the

[1]This dataset will be publicly released upon publication of this work.

TABLE I
WEB SITES AND NUMBER OF PAGES INCLUDED IN THE DATASET.

Source	Pages
https://www.govcert.ch/	26
https://thehackernews.com/	92
https://securitynews.sonicwall.com/	34
https://securelist.com/	205
https://www.auscert.org.au/	100
https://www.cbronline.com/	115
https://us-cert.cisa.gov/	24
https://www.zdnet.com/	224
https://edition.cnn.com/	100

dataset, by sampling pages from each topic separately (or from groups of similar topics). In general, we preferred to include articles from a variety of sites, rather than using a lot of articles from fewer sources, in order to provide the learning algorithms with more diversified data (in terms of writing styles), for the algorithms to be able to generalize better.

The next step was to decide the number of classes in which the articles would be classified. One option would be to consider two classes similarly to [12]. As though our goal is to select only those pages that contain information useful for CTI extraction purposes, we opted to use the following three classes. The first one contains articles that are not related to cyber security. The second contains cyber security-related articles that are too generic to be of interest for CTI extraction; these are commonly found in technology-related web sites that mainly target a wider audience. The third class contains the articles that we are interested in – those that contain useful information for CTI. The intuition is that, because of the difference in the language of a cyber security-related text with a common news article, this choice would have a positive effect on the classification potential of our algorithms.

Some examples of articles in the intermediate class include:

- Generic articles about cyber security trends or speculation about types of attacks in a technology domain (e.g. IoT, connected vehicles), possibly proposing high-level policies (such as the existence of digital certificates).
- Security-related advice targeting a wider audience, such as concerning the importance of using good passwords
- Discussions about an Operating System, possibly including some of its security aspects.
- Analysis/marketing articles about anti-malware software.
- Presentation or advertisement of cyber security conferences or courses.

For the labelling, three annotators were employed; all are cyber security experts. Two of them labelled the whole dataset and in case of disagreements the third annotator gave his own verdict. The Cohen's Kappa coefficient between the first two annotators is 0.706. In addition, there were 11 cases in which all 3 annotators have initially disagreed. For those cases, the final decision was made after a discussion among them. In total, the dataset consists of 460 non-cyber security-related, 131 cyber security-related, and 329 CTI-related articles.

IV. METHODS

In this section, we first describe the text classification process, which involves the feature engineering techniques and machine learning algorithms used in our experiments. Next, we describe NER, which is the next stage of the common CTI extraction pipeline, and how we have used it in order to evaluate the effects of content selection with text classification.

A. Text classification

Each web page is initially represented as a numerical vector and then a machine learning algorithm can be applied on this set of vectorized documents in order to learn a model that accurately describes the relationship between the documents' features and their classes. The appropriate representation of the documents is the feature extraction process.

The first step in our feature extraction process is to keep only the main article from each web page, removing any irrelevant content such as navigational elements (e.g., side bars), advertisements, forms, footers and templates, which are referred to as boilerplate content [19]. For this we chose the Readability tool[2] , after a comparison of the quality of a total of six available options[3] in a separate experiment.

For the vectorization of the documents, we use the typical Bag of Words (BoW) model [20] and then apply the popular TF-IDF weighting scheme [21]. After tokenization of the documents, some preprocessing steps may be useful, depending on the domain and problem in hand. We have experimented with a number of them and their combinations: stemming, stop words removal, as well as the normalization of some domain-specific terms. The first two are commonly considered in text classification. For the latter, we tested two techniques: the substitution of all mentioned IPs in the corpus with a unique placeholder term (namely "specifiedip") and the conversion of all mentioned CVE ids into another placeholder term (namely "specifiedcve"). The intuition of the last two techniques was to reduce the vocabulary, while also emphasizing the importance of those domain-specific terms.

For text classification, two of the most prominent algorithms are the SVM and the Random Forest. SVM [22] is known to be suitable for data representations with feature vectors of high dimensionality, as is the case with text classification [20]. Thus, it has been extensively used in recent research on text classification (e.g., [23]), as well as in the context of cyber security [1], [15], [16]. On the other hand, Random Forest is also found to perform well on text classification, e.g., on sentiment analysis [24] and clinical text classification [25].

In our experiments, we compare the two algorithms, as well as the different preprocessing techniques and combinations. In addition, we evaluate our decision to set the problem as multiclass classification instead of binary.

[2]https://pypi.org/project/readability-lxml/

[3]These include two versions of the boilerpipe tool (https://pypi.org/project/boilerpipe3-fix/ tool, the Goose extractor (https://github.com/goose3/goose3), jusText (https://pypi.org/project/jusText/), and Dragnet (https://github.com/dragnet-org/dragnet).

B. Named Entitity Recognition

As we discused, in order to leverage textual content that comes in unstructured format in order to produce CTI, it is very common to extract entities of interest from it, such as vulnerability enumerations (CVEs) and IoCs. We show the effects of using a good filtering approach in the CTI extraction process not only by evaluating the classification itself, but by inspecting the outputs of the NER process that follows. Specifically, we compare the quality of information provided in the sentences that contain named entities among the articles of the different classes, with respect to CTI. We define the quality of the sentences as the extent to which they concern CTI-related concepts, such as CVEs, IoCs and Tactics, Techniques and Procedures (TTPs) of threat groups.

To this end, we need an appropriate NER model. As we will show in Section VI, we have experimented with several state-of-the-art general purpose deep learning-based NER architectures, training and evaluating them on the MalwareTextDB dataset [26], in order to use the best performing NER approach to answer our research question. MalwareTextDB is a dataset that consists of annotated malware reports, in which the annotated tokens may be *Actions* (referring to events such as "registers", "exploiting", "downloads", etc.), *Entities* (either the initiator of the action such as "the dropper" or the recipient of the action such as "a service") and *Modifiers* (that are just linking words related to the action, such as "to" and are not of major interest for our purposes). By training NER models on a cyber security-related dataset like this, it is expected that they are tuned in order to produce cyber security-related entities.

V. Text classification experiments

The experiments presented in this section evaluate the performance of the Random Forest and SVM classification algorithms on our dataset.

A. Experimental setup

In our experiments, we use the linear version of the SVM algorithm. We also experiment with the feature extraction processes described in Section IV (stemming, stop-words removal and CVE and IP normalization), and their combinations.

The performance is assessed with three well-established evaluation metrics: (i) the overall accuracy, (ii) the precision of the CTI category, and (iii) the recall of the CTI category. Considering the potential applications of the classification task, the last two metrics are the most important ones. For example, in an EWS we are mostly interested in avoiding getting many irrelevant news, thus we should consider precision as the most important metric, but recall is also significant if we do not want to lose any potentially useful information.

Additionally, we evaluate the effects of our choice to use a three-classes setting on the performance of the classification, by performing an experiment where the no-cyber security-related and the cyber security-related classes were merged into one, transforming the problem into binary classification. We compare the best model from the previous experiments to a model with the same settings (algorithm and preprocessing) applied on the binary classification setting, focusing only on the precision and recall of the CTI class.

For these experiments, we estimate the classification performance using 5x5 nested cross-validation. For SVM, we tune parameter C. For Random Forest, we tune the *max_features* parameters of the *scikitlearn* implementation in Python.

B. Results

The evaluation results for text classification are presented in Table II. Our results illustrate that the SVM performs better than the Random Forest for all the metrics considered and the difference is more evident when both the overall accuracy and precision of the CTI category are considered. As far as the preprocessing steps are concerned, the CVE normalization, as well as the combinations of the CVE and IP normalizations yield better results for the Random Forest case. On the other hand, a preprocessing that combines the CVE normalization and the removal of stopwords only marginally improves the performance of the SVM classifier, when the CTI-focused metrics (i.e., precision and recall) are concerned. In general, Random Forest seems more volatile by the preprocessing procedures than the SVM classifiers.

For the evaluation of the three-classes setting decision, we present a comparison of the results of the binary with the multiclass classification in Table III. Compared to the three-classes setting, the binary classification setting results in an important degradation in recall (more than 5%), combined with only a small improvement in the precision. This suggests that our decision to use three classes was overall a good one.

VI. NER experiments

Apart from the above experiments, we evaluate the usefulness of our best classification model on the resulting quality of the NER output, as we have discussed in Section IV.

A. Experimental setup

For this experiment, we split our dataset in a training and a test set in a stratified fashion, using $1/5$ of the corpus as test set. We then train the SVM classifier with the best preprocessing procedure (i.e., the "CVE+stop" method according to the above experiments) on the training set and classify the articles in the test set. It should be noted that, for this experiment, we further fine-tune the hyperparameter C of the SVM.

As mentioned in Section IV, for this experiment we first trained and compared the performance of some NER approaches on the MalwareTextDB dataset in order to select the best NER model. The NER approaches that we have used are those in [27], [28], [29] and [30]. For their implementation, we used the Delft tool[4], and the Glove Common Crawl embeddings[5]. The Bi-directional LSTM-CNNs-CRF architecture of [28] performs the best according to all metrics considered.

We perform NER to the test set articles with this model, trained on MalwareTextDB, and inspect the sentences that

[4]https://github.com/kermitt2/delft

[5]https://nlp.stanford.edu/projects/glove/

TABLE II
TEXT CLASSIFICATION EVALUATION USING RF AND SVM FOR DIFFERENT COMBINATIONS OF PREPROCESSING STEPS.

	RF			SVM		
	Accuracy	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
Base	0.830	**0.900**	0.933	0.902	0.934	**0.948**
+ cve	0.838	0.899	0.939	0.901	0.934	**0.948**
+ ip	0.837	0.895	0.942	**0.904**	0.934	**0.948**
+ cve+ip	0.835	0.888	0.945	0.902	0.934	0.945
+ stop	0.837	0.859	0.942	0.897	**0.940**	0.945
+ stem	0.834	0.899	0.936	0.898	0.933	0.936
+ stop + stem	0.834	0.862	0.939	0.899	0.933	0.930
+ cve + stop	0.836	0.863	0.945	0.898	0.937	**0.948**
+ cve + stem	0.828	0.894	0.924	0.901	0.933	0.939
+ cve + stop + stem	0.834	0.877	0.936	0.899	0.934	0.930
+ ip + stop	0.835	0.867	0.939	0.895	0.936	0.939
+ ip + stem	0.830	0.875	0.930	0.901	0.934	0.942
+ ip + stop + stem	**0.848**	0.876	**0.966**	0.900	0.934	0.933
+ cve + ip + stop	0.829	0.868	0.927	0.897	0.937	0.945
+ cve + ip + stem	0.833	0.892	0.939	0.903	0.934	0.945
+ cve + ip + stop + stem	0.837	0.871	0.945	0.901	0.934	0.936

TABLE III
COMPARISON OF THE MULTICLASS AND THE BINARY CLASSIFICATION.

	SVM, cve+stop	
	Precision	**Recall**
binary	0.943	0.893
multiclass	0.937	0.948

TABLE IV
STATISTICS OF THE WEB DOCUMENTS OF THE TEST SET.

Class	Documents	Documents with entities	Sentences with entities
no-csec	95	26	51
csec	19	12	35
CTI (sample)	20	20	54

contain named entities, comparing their quality among the different document classes. As discussed, the quality is measured in terms of how informative they are regarding CTI extraction.

According to our classifier, the test set contains 95 non-cyber security-related, 19 cyber security-related and 70 CTI-related documents. Due to the large number of entities identified in the CTI-related class, we randomly sample 20 of the documents from this class for our inspection. In Table IV we present statistics for this document collection, such as the number of documents and the number of sentences that contain named entities, for each of the document classes.

B. Results

First, we measured the performance of the classifier on the test set. Its accuracy is 0.918, while precision and recall for the CTI category are 0.928 and 0.984, respectively. This indicates that in this set of documents, the classifier generally performs better than expected judging from the nested cross-validation.

Among the non-cyber security-related documents of our test set, none of the sentences that contain cyber security-related named entities according to the NER model are labelled by our annotators as CTI-related. In addition, among the documents that are classified as cyber security-related, only 3 of the 35 sentences that contained named entities (8% of them) are labelled by the annotators as CTI-related. In contrast, for the documents that are classified as CTI-related, only 15 sentences of the 54 that contained named entities are found during the inspection process to be not informative for CTI purposes.

The amount of sentences in the documents in the first two classes that contain named entities according to the NER model, combined with the extremely low percentage of them that could be relevant to CTI extraction, shows that the absence of a filtering stage (performed by text classification methods in this work), would yield at a later stage of the CTI extraction pipeline many named entities that would be false positives.

We now provide examples of sentences found in documents of the two non-relevant to CTI classes, which would falsely produce CTI-related named entities. In the non-cyber security-related class, an article about an operating system and how to install it contained the following sentences with identified named entities:

- ”As it’s booting up, set it to boot from your USB stick.”
- ”This time around after CloudReady boot ups, we’ll delete Windows 7 and your files” ... ”make sure you’ve backed up your files and install CloudReady as your desktop operating system.”

In the cyber security-related class, an article about an anti-espionage technique contained the following sentences:

- ”Basically, the idea is to profile your attacker(s) and subsequently modify their attack tools to something that you can silently detect”.
- ”In reality, there was an anti-virus program which did detect files crypted using this cryptor”.

Another example in this class is an article covering a

security analysts summit, that yielded the following sentences:

- "I focused on the Samsung Galaxy Gear 2 smartwatch and the ease with which it can be misused by deviants in the 'creepshots' community, as rooting and executing a handful of commands disables camera alerts and recording limitations".
- "Roberto focused on Google Glass whose integrated wifi capability leaves it susceptible to tried-and-true sniffing to expose some of the traffic being relayed to the device".

VII. Conclusions

In this paper, we present our work that involved the collection of a dataset that is appropriate for the content selection problem which, as we have shown with our analysis of the NER output, can play an important role in the quest of achieving high levels of CTI. In contrast to other works that considered content selection from social media sources or hacker forums, we focus on sources such as security- and technology-related blogs and news web sites. For this purpose, we investigated the application of two supervised machine learning algorithms for text classification, focusing on the filtering of pages that are not only related to cyber security, but also contain CTI-related information. Finally, we have shown that modelling the classification problem with three classes had positive effects.

Acknowledgment

This work was supported by the FORESIGHT (H2020 833673) and ECHO (H2020 830943) projects, funded by the European Commission.

References

[1] F. Alves, A. Bettini, P. M. Ferreira, and A. Bessani, "Processing tweets for cybersecurity threat awareness," *Information Systems*, vol. 95, p. 101586, 2021.

[2] Q. Le Sceller, E. B. Karbab, M. Debbabi, and F. Iqbal, "Sonar: Automatic detection of cyber security events over the twitter stream," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, pp. 1–11, 2017.

[3] A. Ritter, E. Wright, W. Casey, and T. Mitchell, "Weakly supervised extraction of computer security events from twitter," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 896–905, 2015.

[4] A. Bose, V. Behzadan, C. Aguirre, and W. H. Hsu, "A novel approach for detection and ranking of trendy and emerging cyber threat events in twitter streams," in *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 871–878, IEEE, 2019.

[5] S. Samtani, H. Zhu, and H. Chen, "Proactively identifying emerging hacker threats from the dark web: A diachronic graph embedding framework (d-gef)," *ACM Transactions on Privacy and Security (TOPS)*, vol. 23, no. 4, pp. 1–33, 2020.

[6] O. Mendsaikhan, H. Hasegawa, Y. Yamaguchi, and H. Shimada, "Quantifying the significance and relevance of cyber-security text through textual similarity and cyber-security knowledge graph," *IEEE Access*, vol. 8, pp. 177041–177052, 2020.

[7] A. Sapienza, A. Bessi, S. Damodaran, P. Shakarian, K. Lerman, and E. Ferrara, "Early warnings of cyber threats in online discussions," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 667–674, IEEE, 2017.

[8] X. Liao, K. Yuan, X. Wang, Z. Li, L. Xing, and R. Beyah, "Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 755–766, 2016.

[9] S. Mittal, A. Joshi, and T. Finin, "Cyber-all-intel: An ai for security related threat intelligence," *arXiv preprint arXiv:1905.02895*, 2019.

[10] S. N. Narayanan, A. Ganesan, K. Joshi, T. Oates, A. Joshi, and T. Finin, "Early detection of cybersecurity threats using collaborative cognition," in *2018 IEEE 4th international conference on collaboration and internet computing (CIC)*, pp. 354–363, IEEE, 2018.

[11] P. Koloveas, T. Chantzios, S. Alevizopoulou, S. Skiadopoulos, and C. Tryfonopoulos, "intime: A machine learning-based framework for gathering and leveraging web data to cyber-threat intelligence," *Electronics*, vol. 10, no. 7, p. 818, 2021.

[12] O. Mendsaikhan, H. Hasegawa, Y. Yamaguchi, H. Shimada, and E. Bataa, "Identification of cybersecurity specific content using different language models," *Journal of Information Processing*, vol. 28, pp. 623–632, 2020.

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

[14] R. P. Lippmann, J. P. Campbell, D. J. Weller-Fahy, A. C. Mensch, and W. M. Campbell, "Finding malicious cyber discussions in social media," tech. rep., MASSACHUSETTS INST OF TECH LEXINGTON LEXINGTON United States, 2016.

[15] I. Deliu, C. Leichter, and K. Franke, "Extracting cyber threat intelligence from hacker forums: Support vector machines versus convolutional neural networks," in *2017 IEEE International Conference on Big Data (Big Data)*, pp. 3648–3656, IEEE, 2017.

[16] A. L. Queiroz, B. Keegan, and S. Mckeever, "Moving targets: Addressing concept drift in supervised models for hacker communication detection," in *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pp. 1–7, IEEE, 2020.

[17] V. Behzadan, C. Aguirre, A. Bose, and W. Hsu, "Corpus and deep learning classifier for collection of cyber threat indicators in twitter stream," in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 5002–5007, IEEE, 2018.

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

[19] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate detection using shallow text features," in *Proceedings of the third ACM international conference on Web search and data mining*, pp. 441–450, 2010.

[20] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in *Mining text data*, pp. 163–222, Springer, 2012.

[21] M. Lan, C. L. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 4, pp. 721–735, 2008.

[22] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[23] A. Rehman, K. Javed, and H. A. Babri, "Feature selection based on a normalized difference measure for text classification," *Information Processing & Management*, vol. 53, no. 2, pp. 473–489, 2017.

[24] D. P. Kamanksha and A. Sanjay, "A critical analysis of twitter data for movie reviews through 'random forest'approach," in *International Conference on Information and Communication Technology for Intelligent Systems*, pp. 454–460, Springer, 2017.

[25] G. Mujtaba, L. Shuib, N. Idris, W. L. Hoo, R. G. Raj, K. Khowaja, K. Shaikh, and H. F. Nweke, "Clinical text classification research trends: Systematic literature review and open issues," *Expert systems with applications*, vol. 116, pp. 494–520, 2019.

[26] S. K. Lim, A. O. Muis, W. Lu, and C. H. Ong, "Malwaretextdb: A database for annotated malware articles," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1557–1567, 2017.

[27] M. E. Peters, W. Ammar, C. Bhagavatula, and R. Power, "Semi-supervised sequence tagging with bidirectional language models," *arXiv preprint arXiv:1705.00108*, 2017.

[28] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," *arXiv preprint arXiv:1603.01354*, 2016.

[29] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 357–370, 2016.

[30] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.