Quantifying the Significance of Cybersecurity Related Text Documents by Analyzing IOC and Named Entities

Otgonpurev Mendsaikhan[1,a)] Hirokazu Hasegawa[1] Yukiko Yamaguchi[1] Hajime Shimada[1]

Abstract: In order to proactively mitigate the cybersecurity risks, the security analysts have to continuously monitor the threat information sources. However the sheer amount of textual information that needs to be processed is overwhelming and requires a mundane labor. We propose a novel approach to automate this process by analyzing and enriching textual cyber threat information using the number of Indicator of Compromise (IOC) and number of Common Vulnerabilities and Exposure (CVE) associated with the content of the information. By fine-tuning the pre-trained Named Entity Recognition model in cybersecurity domain and utilizing various Open Source Intelligence sources to validate the IOCs found in the text we were able to perform experiments and obtain preliminary results.

Keywords: Cyber Threat, Text Analytic, IOC, NER

1. Introduction

The digital age has enabled various opportunities to society and businesses in general. However these opportunities also impose various risks such as cyber attacks, data breach, loss of intellectual property, financial fraud etc. One of the approach to mitigate such risk is the threat information sharing platforms such as closed and open information sharing communities as well as threat feed generating vendors. The idea of threat information sharing stems from the assumption that adversary attacking certain target is also entitled to attack similar target in near future. While the information sharing platforms grew in popularity, the amount of shared threat information grew tremendously that floods the human analyst, thus undermining the threat information sharing effort. In order to identify the significance of the shared information and relevance to their organization the analysts have to process considerable amount of information to separate the actionable threat information from the noise. Even though there are approaches to automatically share information between machines through structured information sharing such as Structured Threat Information Expression (STIX)[*1] and its corresponding protocol Trusted Automated Exchange of Intelligence Information (TAXII), the need to process unstructured text reports that might be shared via email, or forums still exist. For example the darkweb forums provide valuable threat information if the noise can be segregated with less effort.

In our previous work we proposed a system to identify the threat information from publicly available information source[1]. As a successive work in this paper we propose a novel approach to quantify the significance and relevance of the threat information in unstructured text format by counting the Indicators of Compromise (IOC) and identifying the IT assets through Named Entity Recognition method. We've considered the number of IOCs and number of entities and associated known vulnerabilities as features of the threat information in text format and fed to Support Vector Machine (SVM) based classifier to generate confidence score that quantifies the significance of the text.

The main objective of this research is to seek the way to quantify the significance of the text document that can be customized to organizational need using existing Natural Language Processing techniques and tools.

The specific contributions of the paper are as follows:

(1) To propose a novel approach to analyze the text documents to identify its significance.
(2) To prove the viability of the method by doing preliminary experiments
(3) Custom train the Named Entity Recognition model to recognize IT related products.

The remainder of this paper is organized as follows. Section 2 will review the related researches and how this paper differs in approach. In Section 3 we will overview our previous work of autonomous system to generate cyber threat related information. In Section 4 the Analyzer module of the proposed system will be discussed and in Section 5 the experiment conducted using the Analyzer module will be explained. Finally we will conclude by discussing the future works to extend this research in Section 6.

2. Related work

There have been number of attempts to identify or extract cyber threat related information from the dark webs or any other publicly available information source automatically. Mulwad et al. described a prototype system to extract information about security vulnerability from web text using SVM classifier and then extract the entities and concepts of interests from it[2]. They tried to spot the cybersecurity entities using the knowledge from

[1] Nagoya University
[a)] ogo@net.itc.nagoya-u.ac.jp
[*1] https://oasis-open.github.io/cti-documentation/

Wikitology, an ontology based on Wikipedia. Our approach differs to utilize common algorithms of Named Entity Recognition trained on a bigger dataset. Joshi et al. proposed a Cybersecurity entity and concept spotter that uses the Stanford Named Entity Recognizer (NER), a Conditional Random Field (CRF) algorithm based NER framework[3]. They focused to develop more comprehensive data structure identified as cybersecurity related entities whereas our approach is to identify a single label that is associated with known IT product.

More et al. proposed a knowledge based approach to intrusion detection modeling in which the intrusion detection system automatically fetches threat related information from web based text information and proactively monitor the network to establish situational awareness. Their approach focused mainly on developing the cybersecurity ontology which can be understood by the intrusion detecting machines[4]. Our research focused on building an autonomous system that assists the human operators by raising the situational awareness.

Bridges et al. did the automatic labeling for entity extraction in cybersecurity corpus consisting of 850,000 tokens[5]. Their dataset was an inspiration to prepare similar dataset from the whole archive of CVE descriptions. Jones et.al attempted to extract cybersecurity concepts using Brins Dual Iterative Pattern Relation Expansion DIPRE algorithm, which uses a cyclic process to iteratively build known relation instances and heuristics for finding those instances[6]. Our approach differs by deploying off the shelf software solution instead of creating a new method.

Dionsio et al. developed system to detect cyberthreat from Twitter using Deep Neural Networks[7]. Their work has many similarities with our work in which collecting a relevant threat from twitter feed and identifying the assets through Named Entity Recognition. Our approach differs by extracting cybersecurity related documents and then prioritizing the relevance of the document.

There have been attempts to classify text using the features contained in the text mainly in the application of spam detection. Mohammad et al. proposed a system that predict phishing website based on self structuring neural networks[8]. They extracted 17 different features such as using IP address in the URL or containing abnormal URL in the text etc. to classify if given email is spam or not. We are proposing to use similar approach by extracting the features of text to classify it as whether significant to the organization or not.

3. Overview

3.1 Proposed system overview

In our previous work we've proposed a system to identify the threat information from publicly available information source[1]. The proposed system architecture is depicted in **Fig. 1**.

The proposed system would scan the Publicly Available Information Sources on the Internet to create the situational awareness and to assist the security analyst to identify risks and threats imposed to his organization. It does it through the Natural Language Filter module to identify and filter the security related text documents. The organization specific textual data is collected as initial training data and fed to Training Document Generator module to prepare the training documents. Natural Language Filter module is trained by these documents and filters the security related documents. The collected and filtered documents are analyzed by the Analyzer module to generate a meaningful analyzed threat information to the human operators. The documents that have been marked as true positive by the Analyzer module are fed back to the Training Document Generator to improve the performance of the Natural Language Filter.

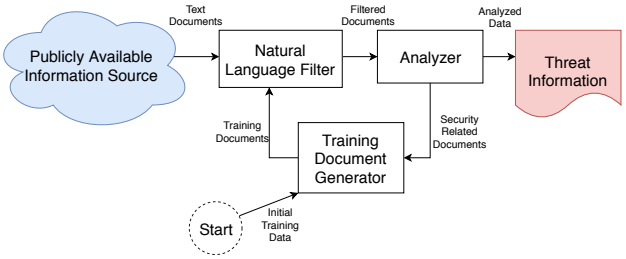

Fig. 1 Overview of proposed system

3.2 Previous work

In our previous work we implemented the Natural Language Filter module of the proposed system[9]. We proposed to utilize neural embedding method called doc2vec[10] as a natural language filter for the proposed system. With the cybersecurity specific training data and custom preprocessing we were able to train a doc2vec model and evaluate its performance. According to our evaluation the Natural Language Filter was able to identify cybersecurity specific natural language text with 83% accuracy. Also we have evaluated the various hyper-parameter settings of the doc2vec model, thus determining the best performing mode and settings of the Natural Language Filter.

As a continuation of our previous work, this paper focuses on the implementation of the Analyzer module as described in the subsequent sections.

4. Analyzer module

We believe the number of Indicators of Compromise (IOC) present in the text, relevant Named Entities mentioned and associated publicly known vulnerabilities could determine the potential significance of the threat information in text format. The idea is if given text document contains certain number of IOC and also mentions about specific company or product (Named Entities) that has known vulnerability the text might be relevant and significant to the organization. In order to prove if our approach is valid we've designed a system that consists of following components.

(1) IOC Analyzer
(2) NER Analyzer
(3) Significance Score Calculator

The high level overview of Analyzer module is depicted in **Fig. 2**.

The collected threat information is analyzed concurrently by IOC Analyzer and NER Analyzer. IOC Analyzer extracts the IOCs in the text document and validates it using various Open Source Intelligence (OSINT) services. The result of OSINT analysis enriches the document and passes it to Significance score

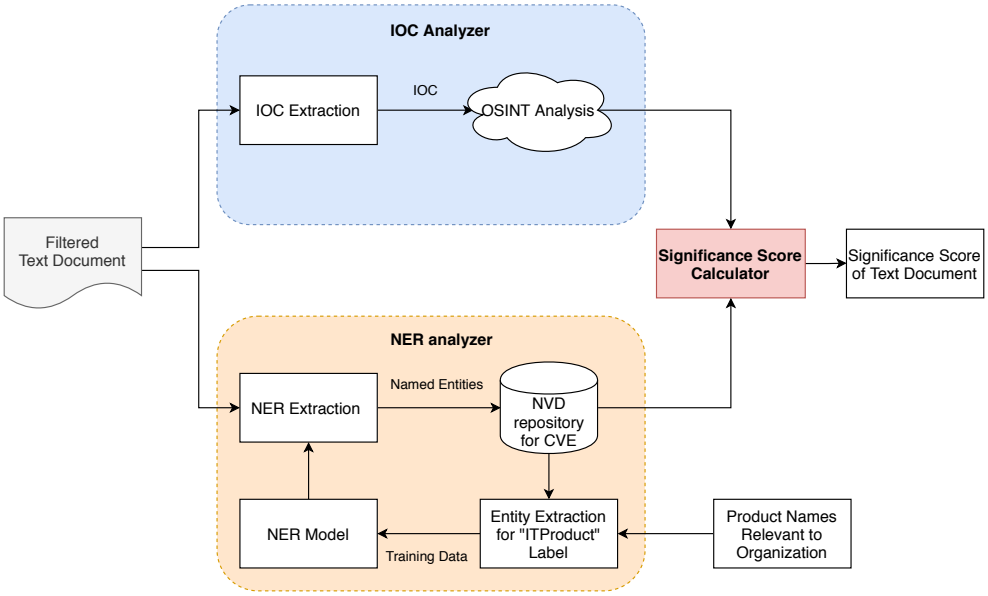

Fig. 2 Overview of Analyzer module

calculator. Meanwhile NER Analyzer extracts the entities that is relevant to the organization and enrich it with associated known vulnerabilities and pass it to Significance score calculator. Finally the Significance score calculator computes the score that the human analyst could decide whether to consider it for further analysis.

Each component of the Analyzer module is discussed in detail in subsequent sections.

4.1 IOC Analyzer

Indicators of Compromise (IOC) are the piece of forensic artifact that can uniquely identify the type and attributes of cyber threats in specific environment. The goal of IOCs is to effectively describe, communicate, and find artifacts related to an incident[11]. IOC may include various artifacts depending on the incident type and environment but in the process of threat information sharing, most commonly used IOCs include *IP address*, *domain name* or *URL*, *email address*, *file hash*, *registry changes* etc.

We believe having a number of IOC in the threat information may increase the possibility of that text information being significant. Hence the IOC Analyzer would extract and count the IOCs contained in the treat information. Once the IOCs are extracted it will be validated against various Open Source Intelligence (OSINT) services to see if the extracted IOC is registered as malicious. Open Source Intelligence or OSINT in the Internet era is the source of collective information that can be used for various purpose. There are many OSINT sources for cyber threat investigation such as Shodan[*2], VirusTotal[*3] or Hybrid-Analysis[*4] etc. where the IOC can be validated for any maliciousness. This validation information is combined with the original document along with the extracted IOC and passed to the Significance Score Calculator.

[*2] https://www.shodan.io/
[*3] https://www.virustotal.com/
[*4] https://www.hybrid-analysis.com/

4.2 NER Analyzer

Named Entity Recognition (NER) is part of information extraction task of Natural Language Processing. NER models are trained to identify real world entities such as People, Location or Organization etc. Commonly known approaches to implement NER model consist of rule or pattern based approach such as identifying the pattern of entity with regular expression or statistics or machine learning based algorithms such as Conditional Random Fields (CRF).

Training a domain specific NER model is difficult due to the lack of annotated training data in the specific domain. Fortunately the National Vulnerability Database (NVD)[*5] of National Institute of Standards and Technology (NIST) provides Common Vulnerabilities and Exposures (CVE) in a structured format that can be used to train the NER model. Also the Common Product Enumeration (CPE) of NVD provides structured naming conventions for commonly used software and hardware products. The CPE dictionary contains Vendor Name, Product Name, Product version and environment etc in Uniform Resource Identifier (URI) format which can be extracted from the CVE description to be used as training data for NER model. As shown in Fig. 2 the organization will specify the IT products that are relevant to them and those products are searched in the CPE dictionary for associated vulnerabilities. The NER model will be explicitly trained to find the entities that are similar to the products specified by the organization.

4.3 Significance Score Calculator

The Significance Score Calculator (SSC) would be function that outputs fixed range of number based on the given inputs. The inputs would consist of following items.

- Number of IOCs present in the document.
- Number of Named Entities labeled as ”ITProduct” that have been found in the document.

[*5] http://nvd.nist.org

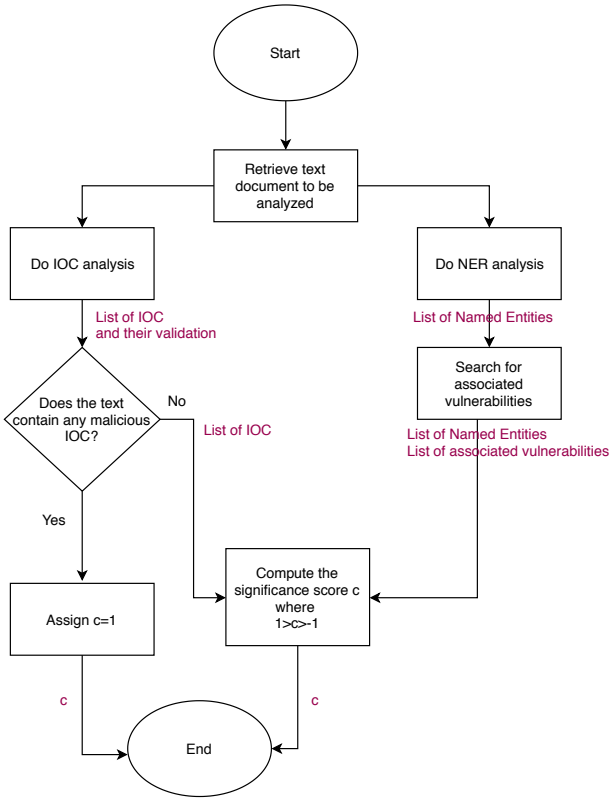

Fig. 3 Workflow of the significant score generation process

- From those Named Entities, the ones that has any associated known vulnerability.

These inputs would serve as features to be extracted from the threat information document to classify if the document is significant or not. Ideally SSC would be linear classification system that produces quantitative number which represents the confidence of specific item belonging to significant class.

5. Implementation and Evaluation

To verify the viability of the proposed system the experiment has been conducted by implementing the proposed components using common open source libraries. The overall significant score generation process of the experiment is depicted in **Fig. 3**

The threat information in text format would be analyzed by IOC Analyzer and NER Analyzer concurrently. The IOC Analyzer would output a list of IOCs and their respective validation. If any of the extracted IOC is found to be malicious by OSINT analysis we treat the text document as significant and assign the highest significance score to the document. Concurrently the NER Analyzer would extract the named entities with ITProduct label from the text document and search for the associated known vulnerabilities per entity. All these information is fed to Significance Score Calculator which returns the significance score between -1 to 1. The significance score should indicate the importance and relevance of the threat information to the organization. The specifics of each system module has been explained in subsequent sections. Once the text document is passed through IOC Analyzer and NER Analyzer, the corresponding features such as extracted IOCs and their respective maliciousness, extracted entities and their associated vulnerabilities will be stored in structured


```
{
    "Original Text": "An anonymous reader writes: A few weeks ago, Google announced its
new Nexus phones &mdash; the 5X built by LG, and the 6P built by Huawei. The phones are
starting to ship, and reviews for both devices have landed. So far, they're largely
positive. Ars Technica calls them the Android phones to beat, though criticizes them for
having fairly large bezels and no wireless charging. Android Police says the 6P's form
factor is an improvement over the Nexus 6, being slightly narrower and taller. Meanwhile,
most publications report that the 5X does a good job at carrying on the legacy of the
excellent Nexus 5. It's their lower end phone, and most reviews mention that it feels that
way in the hand &mdash; but battery life is reportedly excellent. The Nexus 6P's battery
is capable, but doesn't last as long. Fortunately, the worries about overheating with its
Snapdragon 810 chip seem overblown. <p><a href=\"http://mobile.slashdot.org/story/
15/10/19/1539254/nexus-5x-and-nexus-6p-reviews-arrive?
utm_source=rss1.0moreanon&amp;utm_medium=feed\">Read more of this story</a> at Slashdot.",
    "IoCs": {
        "url": [
            {
                "Domain": "http://slashdot.org/slashdot-it.pl?
op=discuss&amp;id=8193263&amp;smallembed=1",
                "Malicious": "No"
            },
        ],
        "email": [],
        "hash": [],
        "IP": [],
        "yara rules": []
    },
    "Entities": [
        {
            "Entity": "huawei",
            "Entity Type": "ITProduct",
            "Related CVE's": [
                {
                    "ID": "CVE-2016-8759",
                    "Description": "Video driver in Huawei P9 phones with software
versions before EVA-AL10C00B192 and Huawei Honor 6 phones with software versions before
H60-L02_6.10.1 has a stack overflow vulnerability, which allows attackers to crash the
system or escalate user privilege."
                },
                {
                    "ID": "CVE-2019-5215",
                    "Description": "There is a man-in-the-middle (MITM) vulnerability on
Huawei P30 smartphones versions before ELE-AL00 9.1.0.162(C01E160R1P12/C01E160R2P1), and
P30 Pro versions before VOG-AL00 9.1.0.162 (C01E160R1P12/C01E160R2P1). When users
establish connection and transfer data through Huawei Share, an attacker could sniff,
spoof and do a series of operations to intrude the Huawei Share connection and launch a
man-in-the-middle attack to obtain and tamper the data. (Vulnerability ID:
HWPSIRT-2019-03109)"
                }
            ]
        },
        {
            "Entity": "android",
            "Entity Type": "ITProduct",
            "Related CVE's": [
                {
                    "ID": "CVE-2018-9357",
                    "Description": "In BNEP_Write of bnep_api.cc, there is a possible out
of bounds write due to an incorrect bounds check. This could lead to local escalation of
privilege with User execution privileges needed. User interaction is not needed for
exploitation. Product: Android Versions: Android-6.0 Android-6.0.1 Android-7.0
Android-7.1.1 Android-7.1.2 Android-8.0 Android-8.1 Android ID: A-74947856."
                }
            ]
        }
    ],
    "Tokens": 164
}
```


Fig. 4 Example data in JSON format

JSON format as shown in **Fig. 4**.

5.1 Implementation of IOC Analyzer

To count the number of IOCs contained in the text document we've extracted email address, URL, file hash, IP address and YARA rules from the given text using python iocextract[*6] module.

For the purpose of this paper the most commonly extracted IOCs such as URL and IP address have been validated using free OSINT tools. The URL have been validated for the maliciousness by Google's SafeBrowsing API[*7] in which the extracted URL is sent to Google's API to check if it has been listed as malicious. For the IP address the AlienVault's OpenThreatExchange (OTX)[*8] platform has been used in which the researchers share various indicator information. The IP address has been checked through OTX's API if it has been ever shared as malicious.

Once the OSINT analysis has been done the extracted IOCs and their maliciousness information is saved in the JSON format along with the original text.

5.2 Implementation of NER Analyzer

For the purpose of this research we've utilized off the shelf open source library spaCy as a NER model. spaCy's Named

[*6] https://pypi.org/project/iocextract/
[*7] https://transparencyreport.google.com/safe-browsing/search
[*8] https://otx.alienvault.com/

Table 1 Comparison of NER models

Model	Precision	Recall	F1 Score
Default en-core-web-lg model	87.03	86.20	86.62
Custom trained on "ITProduct"	79.91	70.49	74.91

Entity Recognition system features a sophisticated word embedding strategy using subword features and "Bloom" embeddings, a deep convolutional neural network with residual connections, and a novel transition-based approach to named entity parsing[*9].

As mentioned in Section 3, by utilizing CVE descriptions and CPE dictionary of NVD we were able to retrieve in total of 109,635 CVE descriptions as of 8th July 2019. For the purpose of this paper we didn't specify any specific product and all the products and companies of CPE dictionary have been used to train the model. The 90% of total collected data has been used to train the spaCy[*10] NER model and remaining 10% of documents have been used to test the trained model.

SpaCy's en-core-web-lg model which have been trained on OntoNotes in English and also contains GloVe vectors trained on Common Crawl[*11] has been utilized as pre-trained model. We've further trained en-core-web-lg model with the training data obtained from NVD and added a label of "ITProduct" to identify the IT related companies and products. After training the model we've evaluated the performance of it on the test dataset of 10,962 documents that contains total of 19,452 entity with the label "ITProduct". The trained model identified 13,713 entities correctly (True Positive) and missed 5,739 entities (False Negative) and misidentified 3,446 entities (False Positive). Using those numbers the performance have been compared with default en-core-web-lg model as shown in **Table 1**.

The extracted ITProducts are searched in the CVE database of NVD and any known vulnerability that mentions the identified ITProduct is saved as an enrichment to the original file.

5.3 Implementation of Significance Score Calculator

To generate significance score Support Vector Machine algorithm has been used. Support Vector Machine (SVM) is one of the most popular supervised Machine Learning algorithm mostly used in classification and regression problems. SVM classifier separates the datasets by finding a line or hyperplane that is found by computing the closest points to both the classes of data. These points are called Support Vectors and distance between the line to dataset is called margin. And the SVM algorithm maximizes the margin, thus giving the most optimal classification between datasets.

The number of IOCs contained and number of entities and its associated known vulnerabilities extracted from the threat information have been considered as features of the text. These features have been fed to sklearn[*12]'s implementation of SVM classifier. The sklearn's SVM classifier have different functions as its kernel depending upon the shape of hyperplane. Preliminary experiment revealed that Radial Basis Function (RBF) has the highest performance in our dataset hence the RBF mode of the SVM kernel has been chosen and every other hyperparameters are kept as default. Also sklearn's SVM classifier can have three modes of output namely:

- Binary classification into two classes
- Probability of item belonging to either classes
- Confidence score of the item to belong either class

Since the objective of the research is to generate quantitative number that represents significance of the text this implementation suits our need. But for the purpose of this experiment binary classification has been conducted to generate the performance indicators such as Accuracy and F1 score.

5.4 Dataset and experimental setup

Balanced dataset of insignificant and significant classes have been utilized to conduct the experiment. The dataset of significant class consisted of 300 text documents that contains threat information. We were able to obtain real threat information shared on the internal network of Mongolian CERT/CC (MNCERT/CC) with their permission. 300 real threat information that has been classified for public disclosure during the time period of July 2017 to October 2017 has been considered and any identifier information has been manually removed.

For the dataset of insignificant class, the collected data from our previous work has been utilized. Random 300 text documents from 4 different data sources are selected and labeled as insignificant. The data sources include:

- **Reddit discussions:** Reddit[*13] is popular social news aggregation and discussion website. As part of our previous work total of 114,391 security related text documents have been collected.
- **StackExchange discussions** StackExchange[*14] is a network of question and answer websites on various topics. As part of our previous work total of 841,311 security related text documents have been collected.
- **Security news outlet RSS feeds** RSS feed summary for select cybersecurity news outlets used for our previous work. As part of our previous work total of 2,077 security related text documents have been collected.
- **Slashdot news** Slashdot[*15] is social news website that features stories on science, technology and politics. As part of our previous work total of 7,751 security related text documents have been collected.

Each of the above data source makes an experiment by mixing with dataset of real threat information, making total data of each experiment as 600 documents respectively. Since the nature of the conversation and text being used may differ depending upon the setting we wanted to test the model with formal type of conversation as well as informal type, to minimize the potential bias caused due to the form of the text. **Table 2** shows the different experiments with corresponding data type and data sources.

The preliminary analysis on the dataset is shown in **Table 3**. The median per each of the features such as Number of IOCs, Number of Entities, Number of Entities that has associated vul-

[*9] https://spacy.io/universe/project/video-spacys-ner-model
[*10] https://spacy.io/
[*11] https://spacy.io/models/en#en_core_web_lg
[*12] https://scikit-learn.org/stable/modules/svm.html#svm-classification
[*13] https://www.reddit.com/
[*14] https://stackexchange.com/
[*15] https://slashdot.org/

Table 2 Data sources of experiments

Exp.no	Significant data	Insignificant data	Type
Exp.1	Real threat information	Reddit	Informal
Exp.2	Real threat information	StackExchange	Informal
Exp.3	Real threat information	RSS feeds	Formal
Exp.4	Real threat information	Slashdot	Formal

nerability is represented along with the median number of tokens per each dataset.

The analysis shows that real threat information i.e. data from Significant class contains more number of features as compared to any of the data source of Insignificant class except Slashdot news.

5.5 Result

As mentioned in previous section each experiment has total dataset of 600 documents that consists of 50% from significant and remaining 50% from insignificant classes. The total dataset of 600 text documents have been randomly split by 420 and 180 as Training and Test dataset. The results are shown per experiment in **Table 4**.

From the result it can be seen that experiments with formal type of data i.e. Exp.3 and Exp.4 has better performance than experiments with informal type of data. The preliminary analysis of Table 3 shows that Slashdot news dataset i.e. Exp.4 has more number of tokens as well as features than rest of the dataset. And the difference between the median numbers of features of Slashdot news and real threat information is closest, hence probability of loss would be higher. But actual result shows the opposite, and having more number of similar features with significant class makes the performance even better.

However overall classification result seems to be overrated. We believe due to the homogeneous nature of the dataset used for significant class the classifier is being too biased for more number of features to be classified as significant.

6. Conclusion

In this paper we proposed a novel approach of quantifying the relevance and significance of the cyber threat information in text format by extracting the features such as number of IOCs and number of relevant Named Entities and their associated vulnerabilities. We also conducted preliminary experiment to analyze the viability of the method. Even though overall experiment results of Table 4 seem to be compelling, we believe the lack of sufficient data and experiment design might have skewed the results. Also using off the shelf software for custom Named Entity Recognition model has shown poor performance as indicated in Table 1. By improving the experiment design we could overcome this limitations.

In the future we would like to improve our NER model by using different algorithms and extending the scope of the entities further than single label of "ITProduct". Also we would like to extend the feature extraction to have more sentiment analysis using the models trained by SemEval[12] dataset.

Table 3 Median values of features

Data source	Tokens	IOC	Ents	Ents with CVE
Significant	115	6	2	1
Reddit	37	0	1	0
StackExchange	57	0	1	0
RSS feed	47	0	1	0
Slashdot	175	10	3	2

Table 4 Results

Exp.no	Precision	Recall	F1 Score	Accuracy
Exp.1	92.22	96.51	94.31	94.44
Exp.2	91.13	83.72	87.27	88.33
Exp.3	97.75	92.55	95.08	95.00
Exp.4	98.80	96.51	97.64	97.77

Acknowledgement

This work is partially supported by JSPS KAKENHI Grant Number 19H04108 and 19K11961.

References

[1] Otgonpurev Mendsaikhan, Hirokazu Hasegawa, Yukiko Yamaguchi, Hajime Shimada, "Mining for operation specific actionable cyber threat intelligence in publicly available information source," *In Proc. of 2018 Symposium on Cryptography and Information Security*, Jan. 2018.

[2] Varish Mulwad, Wenjia Li, Anupam Joshi, Tim Finin, Krishnamurthy Viswanathan, "Extracting Information about Security Vulnerabilities from Web Text," *In Proc. of the 2011 International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT'11)*, pp. 257-260, Aug. 2011.

[3] Arnav Joshi, Ravendar Lal, Tim Finin, Anupam Joshi, "Extracting Cybersecurity Related Linked Data from Text," *In Proc. 7th International Conference on Semantic Computing*, pp. 252-259, Sep. 2013.

[4] Sumit More, Mary Matthews, Anupam Joshi, Tim Finin, "A Knowledge-Based Approach to Intrusion Detection Modeling," *In Proc. of 2012 IEEE Symposium on Security and Privacy Workshops*, pp. 75-81, May. 2012.

[5] Robert A. Bridges, Corinne L.Jones, Michael D.Iannacone, John R.Goodall, "Automatic Labeling for Entity Extraction in Cyber Security," *arXiv:1308.4941*, Aug. 2013.

[6] Corrine L.Jones, Robert A. Bridges, Kelly M. T. Huffer, John R. Goodall, "Towards a Relation Extraction Framework for Cyber-Security Concepts," *arXiv:1504.04317*, Apr. 2015.

[7] Nuno Dionsio, Fernando Alves, Pedro M. Ferreira, Alysson Bessani, "Cyberthreat Detection from Twitter using Deep Neural Networks," *arXiv:1904.01127*, Apr. 2019.

[8] Rami M. Mohammad, Fadi Thabtah, Lee McCluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Computing and Applications,* pp. 443-458, Aug. 2014.

[9] Otgonpurev Mendsaikhan, Hirokazu Hasegawa, Yukiko Yamaguchi, Hajime Shimada, "Identification of Cybersecurity Specific Content Using the Doc2Vec Language Model," *In Proc. of 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, pp. 396-401, Jul. 2019.

[10] Quoc Le and Tomas Mikolov, "Distributed Representations of Sentences and Documents," *arXiv:1405.4053*, May. 2014.

[11] Jason T.Luttgens, Matthew Pepe, Kevin Mandia, "Incident Response & Computer Forensics," Third ed. *McGraw-Hill Education,* 2014.

[12] Peter Phandi, Amila Silva, Wei Lu, "SemEval-2018 Task 8: Semantic Extraction from CybersecUrity REports using Natural Language Processing (SecureNLP)," *In Proc. of The 12th International Workshop on Semantic Evaluation,* pp. 697-706, Jun. 2018