IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 9, 2020, accepted September 21, 2020, date of publication September 28, 2020,
date of current version October 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027321

Quantifying the Significance and Relevance of Cyber-Security Text Through Textual Similarity and Cyber-Security Knowledge Graph

OTGONPUREV MENDSAIKHAN[1], (Student Member, IEEE), HIROKAZU HASEGAWA[2], YUKIKO YAMAGUCHI[3], AND HAJIME SHIMADA[3], (Member, IEEE)

[1]Graduate School of Informatics, Nagoya University, Nagoya 464-8601, Japan
[2]Information Security Office, Nagoya University, Nagoya 464-8601, Japan
[3]Information Technology Center, Nagoya University, Nagoya 464-8601, Japan

Corresponding author: Otgonpurev Mendsaikhan (ogo@net.itc.nagoya-u.ac.jp)

ABSTRACT In order to proactively mitigate cyber-security risks, security analysts have to continuously monitor sources of threat information. However, the sheer amount of textual information that needs to be processed is overwhelming, and it requires a great deal of mundane labor to separate the threats from the noise. We propose a novel approach to represent the relevance and significance of the cyber-security text in quantitative numbers. We trained custom Named Entity Recognition (NER) model and constructed a Cyber-security Knowledge Graph (CKG) to infer the subjective relevance of the cyber-security text to the user and to generate correlation features. In addition, the significance of the given text was analyzed in terms of its textual similarity with different repositories of pre-defined ''significant'' text and the maximum similarities were computed. These analysis results then act as features of the classifier to generate the significance score. The experimental result showed that the overall system could determine the significance and relevance of the text within a controlled environment with 88% accuracy.

INDEX TERMS Cyber-security knowledge graph, cyber threat, text analysis, textual similarity.

I. INTRODUCTION

The digital age has presented various opportunities to society and to business in general. However, these opportunities also bring with them different kinds of risk such as cyber-attacks, data breaches, loss of intellectual property, financial fraud, etc. One approach to mitigate those risks is the sharing of threat information via platforms such as the closed and open information-sharing communities as well as the threat feed generating vendors. The idea of sharing threat information stems from the assumption that an adversary that attacks a certain target is also likely to attack similar targets in the near future. While information-sharing platforms have grown in popularity, the amount of shared threat information has grown tremendously, overwhelming human analysts and undermining the efforts to share threat information. In order to identify the significance of the shared information and relevance to their organizations, the analysts have to process considerable amounts of information and separate the actionable threat information from the noise.

Even though there are approaches that 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 exists. For example, dark-web forums provide valuable threat information, if the noise can be segregated, with less effort. Also, to establish situational awareness, a security analyst has to be able to identify cyber threat-related information specifically applicable to his environment to proactively monitor and prevent the possible intrusion and control the possible risk. For this reason, we are proposing an autonomous system that employs Natural Language Processing techniques to identify the cyber

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks.

[1]https://oasis-open.github.io/cti-documentation/

threat-related information specific to the user and filter out the irrelevant content.

In our earlier work [1], we proposed the overall architecture of this autonomous system to identify user-specific threat information from publicly available information sources. In that work, we proposed to filter the cyber-security specific content from the publicly available text information. As a follow-up, in this paper, we propose a novel approach to identify the user-specific content from those filtered texts. Therefore, this paper focuses on to quantify the significance and relevance of the threat information contained in unstructured text by comparing the vector representation of the text with known important text and identifying the cyber-security entities using a Named Entity Recognizer and by correlating it with an existing Cyber-security Knowledge Graph (CKG). We considered the textual similarity of the text and the correlation of the mentioned entities with the CKG as features of the threat information and fed those features through a classifier to generate a score that quantified the significance and relevance of the text.

According to Harter, the information could have either objective or subjective relevance to the particular situation [2]. The objective relevance measures how well the topic of the information matches the domain, and subjective relevance deals with user-specific situations. In [1] we attempted to identify the text documents that have objective relevance within the domain of cyber-security. In this work our goal is to seek a way to quantify the subjective relevance of the text documents alongside with its potential significance, which can be customized to meet user-specific needs by utilizing existing Natural Language Processing (NLP) techniques and tools.

Identifying the subjective relevance of entities and concepts is a well-studied field of Information Retrieval (IR), where the search engines provide web-page rankings based on the relevance to the user [3]. However, to the best of our knowledge correlating the extracted entity with an existing knowledge base to determine the subjective relevance has not been attempted in the field of cyber-security.

The specific contributions of the paper are as follows:

1) Proposal of a novel approach to analyze text documents to identify the significance and relevance of the text
2) Design for an experiment to prove the viability of this method

The remainder of this paper is organized as follows. Section II will review the related research and highlight how this paper differs in its approach. In Section III we will briefly discuss the conceptual design of the proposed autonomous system along with the implementation of the Natural Language Filter module. Also, the conceptual design of the Analyzer modules will be introduced in Section III. In Section IV the implementation of the proposed Analyzer module will be discussed and in Section V the corresponding experiment to evaluate its viability will be discussed. In Section VI we will compare our work with the industry approach and finally, we will conclude by discussing future work to extend this research in Section VII.

II. RELATED WORK

To the best of our knowledge, currently, there are no published works related solely to determine the subjective relevance and significance of the document by engineering the textual features. However, there have been various approaches to utilize NLP techniques in cyber-security. We categorize the works related to our study as Automated Threat Detection, Cyber-security Knowledge Graph, Cyber-security Named Entity Recognition, and Text classification. Our work could be seen as the amalgamation of these different domains.

A. AUTOMATED THREAT DETECTION

There have been a number of attempts to automatically identify or extract cyber-threat-related information from the unstructured text. Mulwad *et al.* proposed a framework to identify and generate assertions about vulnerabilities, threats, and attacks from web text by using an SVM classifier and Wikitology, an ontology-based on Wikipedia [4]. Joshi *et al.* proposed an information extraction framework that extracts cyber-security entities, terms, and concepts to map them to related web resources and create an open ontology [5]. More *et al.* proposed a knowledge-based approach to intrusion-detection modeling in which the intrusion-detection system automatically fetches threat information from web-based text information and proactively monitors the network to establish situational awareness. Their approach focused mainly on developing a cyber-security ontology that could be understood by intrusion-detecting machines [6]. Jones *et al.* proposed a bootstrapping algorithm to extract cyber-security entities and identify their relationships using Brin's Dual Iterative Pattern Relation Expansion (DIPRE) algorithm, which uses a cyclic process to iteratively build known relation instances and heuristics for finding those instances [7]. Also, Dionísio *et al.* developed a system to detect cyber-threats from Twitter using deep neural networks [8]. Their work has many similarities with our work, e.g. collecting relevant threats from Twitter feeds and identifying the assets with a Named Entity Recognizer. Husari *et al.* developed a system to automate Cyber Threat Intelligence (CTI) analytics that learns attack patterns [9]. They combined NLP and IR techniques to extract threat actions from threat reports based on semantic relationships. These works focused to extract cyber-threat-related information from a text which is similar to our proposed autonomous system. However, the approach we have taken is to first identify the objective relevance (cyber-security domain topic) and then find the subjective relevance (user-specific threat) from the textual information.

B. CYBER-SECURITY KNOWLEDGE GRAPH

There have been several proposals to extract the relationships of cyber-security entities and to build Cyber-security

Knowledge Graph from unstructured text. Pingle *et al.* proposed a system called RelExt that would extract possible relationships and create semantic triples over cyber-security text, using a deep-learning approach [10]. Consequently, Piplai *et al.* developed a system to extract information from malware After Action Reports (AAR) that can be merged to create a Cyber-security Knowledge Graph using RelExt [11]. In addition, Jia *et al.* proposed an approach to build a cyber-security knowledge base and deduction rules based on a quintuple model [12]. Both works focused on developing a comprehensive approach to effectively extract cyber-security entity relationships and build Cyber-security Knowledge Graph, whereas our focus is to utilize existing Cyber-security Knowledge Graph to infer correlations between entities.

C. CYBER-SECURITY NAMED ENTITY RECOGNITION

The latest trends in Named Entity Recognition (NER) has been in deep neural network architecture. Yadav *et al.* surveyed the recent advances in Named Entity Recognition focused on neural architectures and compared them to previous feature-based systems [13]. The paper's finding has shown that incorporating the characteristics of feature-engineered models into modern neural network architectures could yield better results. Another development in the NER field is the constituent-based tagging scheme in which a conventional tagging scheme to denote entities is replaced by a more constituent specific tagging scheme. Zhong *et al.* proposed TOMN and UGTO tagging schemes to better indicate the time expression and compared the performance with state-of-the-art models [14]. The experimental results demonstrated that the proposed models trained with a constituent-based tagging scheme perform equally or more effectively than the representative state-of-the-art models indicating the potential in the approach. These advances have been attracting some research interest in the cyber-security field, especially to utilize deep-learning architectures. Simran *et al.* proposed a deep-learning-based framework for NER in cyber-security and evaluated various deep-learning architectures [15]. Gasmi *et al.* proposed an LSTM model for NER and Relation Extraction tasks [16]. Even though, not a deep learning approach Yi *et al.* also proposed cyber-security NER model based on regular expressions and known-entity dictionary [17]. Their proposed models achieved competitive performance when compared with feature engineered models. Since achieving these state-of-the-art results for NER tasks was not the objective of this work we utilized a simpler, but efficient, Conditional Random Fields (CRF) model for the cyber-security entity identification task.

D. TEXT CLASSIFICATION

In general, our work can be viewed as a text classification task by engineering the features of the text to identify its significance and relevance to the user. In this regard, there have been numerous works that reviewed and compared the performances of various methods of text classification. Minaee *et al.* did a comprehensive review of 150 deep learning-based models for the latest state-of-the-art text classification methods [18]. In the paper, the authors reviewed the performance of various language models on different text classification tasks from which the News Categorization task is the most similar task to our approach. In the News Categorization task Transformer based Pre-trained Language Model XLNet has shown the highest performance in the AG News dataset. Yang *et al.* introduced XLNet in [19] as a generalized autoregressive pre-trained model to overcome the limitations of the state-of-the-art language model BERT. Even though XLNet is a state-of-the-art model, its computationally expensive nature makes it difficult for customization such as identifying the relevant text to the user. Our approach differs in which we seek to develop a domain-specific text classification model that can be easily customized to classify the significant and relevant text to the user.

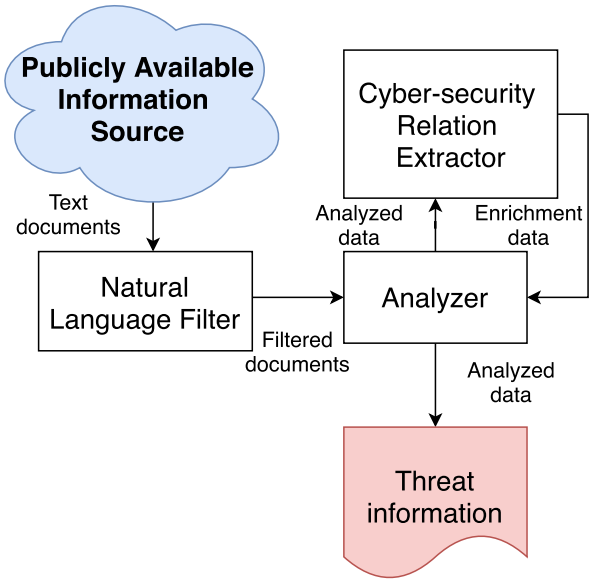

FIGURE 1. Overview of proposed system architecture.

III. PROPOSED AUTONOMOUS SYSTEM

Since this research is the extension of our previous work, the autonomous system architecture proposed in previous work and the implementation of Natural Language Filter are briefly discussed in this section. Also, the theme of the current research, a general architecture of the proposed Analyzer module will be briefly introduced as well.

A. PROPOSED SYSTEM ARCHITECTURE

In our earlier work [1] we proposed a system to identify threat information from publicly available information sources. With some modification to the original design, the proposed system architecture would be as shown in Fig. 1.

The proposed system would scan the publicly available information sources on the Internet to create situational awareness and to assist security analysts in identifying risks and threats posed to their organizations. The Natural

Language Filter module classifies and filters the cyber-security-related text documents. The collected and filtered documents are analyzed by an Analyzer module to determine the significance and relevance of the cyber-security text, thus feeding only the useful threat information to the user. The significant documents are also fed into a Cyber-security Relation Extractor to extract the useful information that will enrich the Cyber-security Knowledge Graph and thus improve the correlation features of the subsequent document.

B. NATURAL LANGUAGE FILTER MODULE

The Natural Language Filter module is a language model that is trained to identify and filter the security-related text documents from publicly available information sources. In [1] we experimented with the Doc2Vec language model to utilize as Natural Language Filter by training it with over 1 million security-specific text documents. The model would compare the cosine similarity of the vector representation of any incoming text document with its training document and filter out the documents that have less than 70% similarity. With custom preprocessing of the text documents, we were able to achieve 83% accuracy. Subsequently, we experimented with the state-of-the-art model Bidirectional Encoder Representations from Transformer (BERT) and improved this result to 90% in [20].

C. ANALYZER MODULE

The purpose of the Natural Language Filter module is to identify cyber-security-related text documents from publicly available information source for further analysis, whereas the purpose of the Analyzer module is to determine the significance and relevance of the text document to the user in order to reduce the workload of the human operators by filtering out information that is insignificant or non-relevant to the organization. We believe that a text document's significance and relevance could be determined by identifying textual similarities with pre-defined significant texts and the correlation between the cyber-security entities mentioned in the text and the terms we are interested in. These features from the text documents could be used to generate a unique number that could represent the significance and relevance of the text document. This could be achieved with the following structure as depicted in Fig. 2.

The Analyzer module would consist of following components.

1) Similarity Analyzer
2) Cyber-security Knowledge Graph Analyzer
3) Significance Score Calculator

Each component is discussed in the subsequent section.

1) SIMILARITY ANALYZER

The semantic analysis of the text document refers to extracting the lexical meaning of a text independent of its written language. Since computers can work only with numbers, computational linguistics achieves semantic analysis by representing text in vector space and assigning different meanings of the text in different dimensions of the vector. For example, the word ''bank'' could mean a financial institution as well as geographical terrain adjacent to a river (as in river bank). When the word ''bank'' is represented in vector space, each meaning would be represented by different components of the same vector, depending upon the context. Once the text is represented in vector space, one way of performing the semantic analysis on the text document is to compare its vector representation with another vector. Comparing the vector representations of different texts is called textual similarity and the distance between the vectors represent the closeness of their semantic meanings.

Kenter *et al.* demonstrated the effectiveness of computing textual similarity through vector embeddings of short text in [21] and we believe the textual similarity could be used to define the significance of the text by comparing vector representations of the given text with a pre-defined ''significant'' text.

2) CYBER-SECURITY KNOWLEDGE GRAPH ANALYZER

We propose to utilize existing knowledge sources to identify the subjective relevance of a text document to the user. A similar approach to utilize external knowledge source to analyze the textual features has been proposed by Nguyen *et al.* in [22]. They proposed to identify a short-text semantic similarity through word embeddings and external knowledge sources. Their approach of determining the degree of semantic similarity between pairs of a short text by exploiting the semantic relatedness between concepts based on an external source of knowledge and word embeddings has outperformed state-of-the-art systems in short text semantic similarity task on three different datasets. Similarly, we believe with the right setting, the subjective relevance of the text could be inferred with high confidence using an existing knowledge graph.

According to Ehrlinger *et al.*: ''A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge'' [23]. Based on this definition, we propose to utilize domain-specific knowledge graph to efficiently store the structured information that evolves within the ever-changing field of cyber-security. As shown in Fig. 1, the Cyber-security Relation Extractor would constantly enrich the CKG with new information that could be useful for subsequent analysis.

In a broad sense, a Cyber-security Knowledge Graph is a graph representation of a semantic triple that comprises of a pair of cyber-security entities and the relationship between them. For the purpose of this paper, a CKG is used to determine if the given text document has any relevance to the user. To do that, we defined two types of entities, namely Entities of Interest and Mentioned Entities. The Entities of Interest are the user-specific terms that indicate any hardware/software vendors or product names as well as Common Vulnerabilities and Exposures (CVE) ID. The Mentioned Entities are the entities that have been extracted from the given text document through the Named Entity Recognizer.

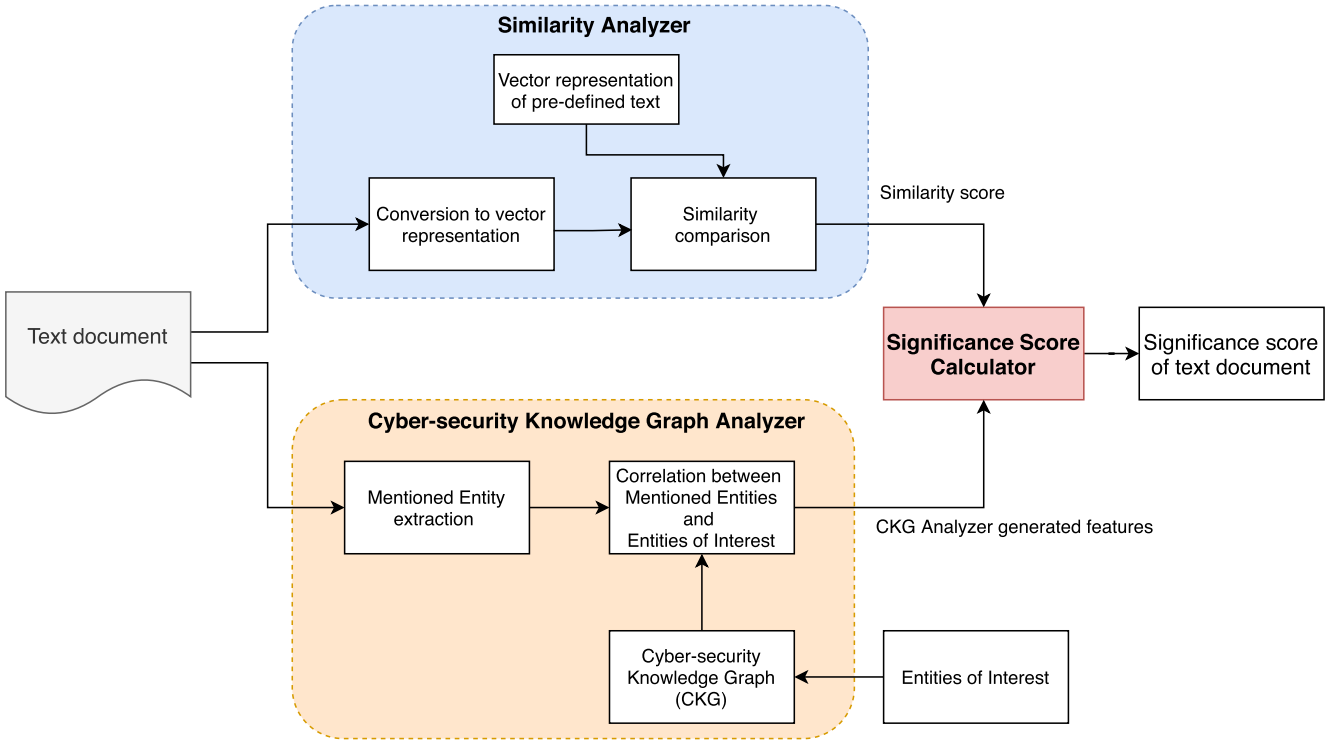

FIGURE 2. Overview of proposed analyzer module.

In our previous work [24], we tried to infer the relevance of the text through the number of subjective named entities mentioned in the text document. However, we concluded it was not a good approach since the Named Entity Recognizer could not account for the semantically related terms (e.g., "desktop" could mean "computer" depending on the context. But the Named Entity Recognizer would not be able to infer this relationship) unless specifically trained on them. Therefore, to overcome this drawback we deployed the Cyber-security Knowledge Graph to infer the correlation between the Entities of Interest and Mentioned Entities.

3) SIGNIFICANCE SCORE CALCULATOR

The Significance Score Calculator (SSC) is a function that outputs a fixed range of numbers based on the given inputs. The inputs consist of the following items.

- Scores with closest similarity to the pre-defined significant text repositories
- Features generated by the correlations between Entities of Interest and Mentioned Entities

These inputs would serve as features to be extracted from the threat-information documents to classify whether the document is significant or not. Ideally, SSC would be a classifier that produces a quantitative number which represents the probability of specific item belonging to a significant class.

IV. IMPLEMENTATION

To verify the viability of the proposed Analyzer module, we implemented the proposed components using common open-source libraries. The details of the implementation and experimental environment are discussed in subsequent sections.

A. DATA USED

During our previous work [1], a significant amount of cyber-security-related text data has been collected from various sources. For the purpose of this research, part of it has been re-utilized. The data sources include

- **MalwareTextDB:** Phandi *et al.* proposed a shared task to classify relevant sentences, predict token labels and relation labels and attribute labels for malware-related text at the International Workshop on Semantic Evaluation 2018 [25]. In the task proposal, they had compiled the largest publicly available dataset of annotated malware reports, which is called MalwareTextDB and consists of 85 Advanced Persistent Threat (APT) reports that contain 12,918 annotated sentences. For the purpose of this paper, this data source will be called MWTDB for short.
- **CVE repository:** Common Vulnerability and Exposures (CVE) descriptions of the National Vulnerability Database (NVD). The NVD is the U.S. government's repository of standards-based vulnerability-management data and is known as the central database of all software security vulnerabilities.[2] For the purpose of this paper, the data source will be referred to simply as CVE for short.

[2]http://nvd.nist.org

- **StackExchange discussions:** StackExchange[3] is a network of question-and-answer websites on various topics. The whole of the discussions among the Security,[4] Cryptography[5] and Reverse Engineering[6] communities since the site was created up until December 2nd 2018 has been collected. For the purpose of this paper, this data source will be called SE for short.
- **Security news outlet RSS feeds:** RSS feed summary for selected cyber-security news outlets during the period of November 1st 2018 to December 2nd 2018. News outlets include DarkReading,[7] NakedSecurity,[8] SecurityMagazine[9] and ThreatPost.[10] For the purpose of this paper, this data source will be called as RSS in short.

From each of the data sources 1,100 text documents have been randomly selected to be utilized for following purposes.

- **Reference text:** 100 documents from each source would act as pre-defined ''significant'' text to be used in the Similarity Analyzer.
- **Test data:** 1,000 documents from each source makes a total of 4,000 text documents for the evaluation of the overall system.

B. SIMILARITY ANALYZER

In order to perform semantic analysis through textual similarity, the given text is converted into numerical vectors, also known as embeddings. Conventionally, vector embeddings were achieved through shallow algorithms such as Bag of Words (BoW) or Term Frequency Inverse Document Frequency (TFIDF). These approaches have been superseded by predictive representation models such as Word2Vec [26], GloVe [27] etc. Since the utilization of deep neural networks has been proven to be superior in different fields, various studies have adopted deep neural models to embed the text into vector space, such as Facebook's InferSent[11] and Universal Sentence Encoder (USE) from Google Research. Perone *et al.* evaluated different sentence embeddings and Universal Sentence Encoder outperformed InferSent in terms of semantic relatedness and textual similarity tasks [28]. Therefore, for the purpose of this research, Universal Sentence Encoder has been utilized to generate the vector embeddings of the text.

1) UNIVERSAL SENTENCE ENCODER (USE)

In the paper by Cer *et al.*, transformer-based and deep averaging network (DAN)-based models for encoding sentences into embedding vectors have been introduced [29]. The USE models take English sentences of variable lengths as input and produce 512 fixed-dimensional vector representations of the sentences as output. Both the models are pre-trained using Wikipedia, web news, web question-answer pages and discussion forums.[12]

Since the sentence embeddings from USE produce good task performance with little task-specific training data, a DAN-based sentence encoder has been employed for this research in order to find textual similarity between the texts in vector space, thereby performing a semantic analysis. The DAN-based sentence encoder model makes use of a deep averaging network whereby input embeddings for words and bi-grams are first averaged together and then passed through a feedforward deep neural network to produce sentence embeddings with minimal computing resource requirements.

TABLE 1. Reference text similarity.

	MWTDB	CVE	SE	RSS
MWTDB	1.0	0.5696	0.5352	0.5730
CVE		1.0	0.5946	0.6523
SE			1.0	0.5459
RSS				1.0

2) IMPLEMENTATION OF SIMILARITY ANALYZER

At first, the USE is utilized to generate vector representations of an initial 256 bytes of every entry in the Reference text and store them in repositories named after the data source. (The choice of the initial 256 bytes is heuristic. Even though it is claimed that USE can work on varying lengths of text, we observed that better textual similarity is obtained when texts of the same length are compared.) In order to illustrate the relative differences between the Reference text repositories, the maximum textual similarities have been computed. Table 1 shows the maximum cosine similarity between the different reference text repositories.

From Table 1 it can be seen that the Reference text repositories are distinct enough to represent different types of cyber-security text. To reflect this distinctive nature in the semantic analysis process, we assigned weights to the repositories. We believe the actions and capabilities of the malware are of utmost importance to the human analysts; therefore, if the given text is semantically similar to pre-defined texts describing malware actions and capabilities as included in the MalwareTextDB, the significance of that text should be considered the highest. The second highest significance is given to a CVE description wherein the details of software vulnerabilities are discussed. The informal discussions that happen around a question-answer system such as StackExchange may be useful when attempting to understand the situation, thus this is third in line. Official news reports have been assigned the lowest significance, based on the assumption that if something is already on a public news outlet, its importance is likely to be outdated. These priorities have been reflected in the test data, as will be discussed in Section V-B.

[3] https://stackexchange.com/
[4] http://security.stackexchange.com/
[5] http://crypto.stackexchange.com/
[6] http://reverseengineering.stackexchange.com/
[7] https://www.darkreading.com/
[8] https://nakedsecurity.sophos.com/
[9] https://www.securitymagazine.com/
[10] https://threatpost.com/
[11] https://github.com/facebookresearch/InferSent
[12] https://tfhub.dev/google/universal-sentence-encoder/2

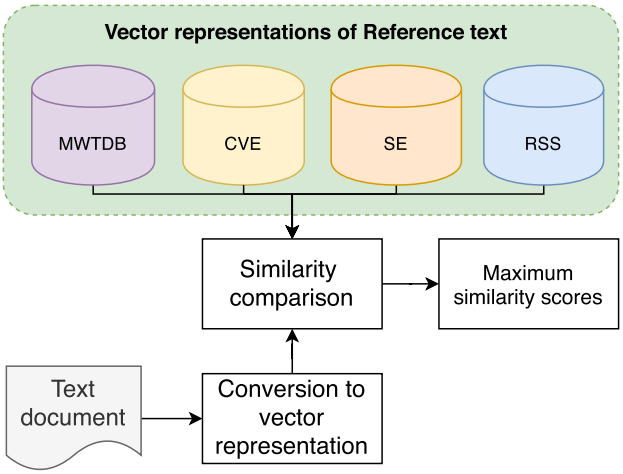

FIGURE 3. Similarity analysis process.

Similar to the Reference text, 256 characters from the start of the input text have been extracted and converted into a vector representation by the USE. Consequently, cosine similarity is computed with each entry in the Reference text and the highest similarity score for each repository is considered as the output of the Similarity Analyzer. The process is illustrated in Fig. 3.

As a result of the process depicted in Fig. 3, the Similarity Analyzer feeds following features to the Significance Score Calculator.

1) Maximum similarity score with MWTDB repository
2) Maximum similarity score with CVE repository
3) Maximum similarity score with SE repository
4) Maximum similarity score with RSS repository

C. CYBER-SECURITY KNOWLEDGE GRAPH ANALYZER

Whether a given text document has any subjective relevance to the user has been analyzed using a Cyber-security Knowledge Graph Analyzer. It uses the custom-trained NER model as a Named Entity Recognizer to extract the Mentioned Entities from a text and the CKG to infer the relationship with the Entities of Interest. The details of these components are discussed in subsequent sections.

1) CYBER-SECURITY KNOWLEDGE GRAPH

In order to construct a CKG, we utilized the NVD repository of CVE descriptions as of 04th March 2020. From each CVE description, a unique semantic tuple is derived that contains the subject, object and their relationship. We assigned an arbitrary number as a cost for each type of tuple to denote the uniqueness of the relation. The cost has been also used to compute the shortest distance between the nodes. The tuple types and their corresponding costs are shown in Table 2.

In total 221,202 tuples have been extracted from the NVD repository. The subjects and objects of those tuples are represented as nodes for Vendor, Product, and CVE ID in the graph and the relationship between them serves as the edges

TABLE 2. Semantic tuples derived from NVD.

Subject	Relation	Object	Cost
Vendor	has product	Product	15
Product	product of	Vendor	16
Product	is vulnerable	CVE ID	9
CVE ID	vulnerability of	Product	10

TABLE 3. Unique nodes in the graph.

Node type	No. of nodes
Vendor	20,190
Product	36,626
CVE ID	115,551
Total	**172,367**

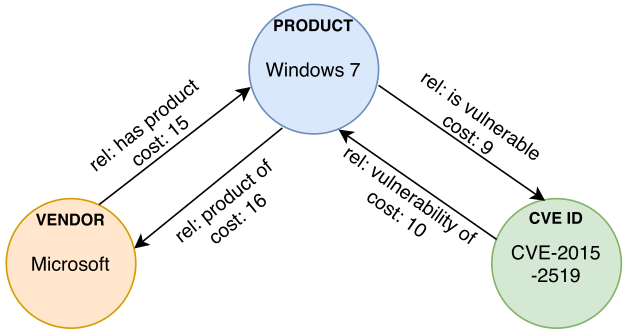

FIGURE 4. Cyber-security knowledge graph example.

of the graph. The total number of unique nodes derived from the semantic tuples is shown in Table 3.

For example, the relationships: {(**Vendor:** Microsoft) (has product) (**Product:** Windows 7)}, and {(**Product:** Windows 7) (is vulnerable) (**CVE ID:** CVE-2015-2519)} is represented in the graph as shown in Fig. 4.

2) NAMED ENTITY RECOGNIZER

The Cyber-security Knowledge Graph Analyzer would utilize a Named Entity Recognizer to identify the Mentioned Entities in the text. Hence, it becomes necessary to train the custom NER model to act as a Named Entity Recognizer that can identify the entities of CKG. Fortunately, as pointed out by [30], the structured text data of the NVD repository provides an easier way to automatically label and annotate the large corpus for custom training the NER model. Hence, the CVE descriptions in the NVD repository have been annotated with the vendor and product information. Since it is extremely difficult to distinguish between an IT product and IT vendor from unstructured text (e.g., the entity "debian" could mean the Debian project i.e., **the Vendor** or Debian Linux **the Product** depending upon the context), we defined both the product and vendor under the same label, **Vendor**. In addition, the structured naming convention of CVE ID (e.g., CVE-YYYY-NNNN) provides an easily identifiable pattern to define a label **CVE ID** from an unstructured CVE description.

For the Named Entity Recognizer we utilized the Stanford NER model, also known as CRFClassifier. The

TABLE 4. Custom trained NER model performance.

Label	Prec	Rec	F1 score
Vendor	0.7473	0.7761	0.7615
CVE ID	1.0	1.0	1.0

Stanford NER model is a general implementation of linear-chain Conditional Random Field (CRF) sequence models that is customizable to user-specific data [31]. Even though there are models that utilize deep neural architecture and demonstrate better performance, the Stanford NER model is efficient enough, as an off-the-shelf model that can be easily retrained. Hence, the Stanford NER Model has been trained for the annotated CVE descriptions consisting of 17,113,939 words and tested on 4,658,262 words. The performance of the custom-trained NER model per label is shown in Table 4.

From Table 4 it could be seen that the Named Entity Recognizer is able to distinguish CVE IDs without any error since it could be identified by a clear pattern.

3) IMPLEMENTATION OF CYBER-SECURITY KNOWLEDGE GRAPH ANALYZER

Rada *et al.* defined a metric of distance in the knowledge base represented as a graph in order to measure the relatedness of the two nodes of a graph [32]. We propose to utilize similar measures such as the number of nodes, number of edges and total cost to define the relatedness or subjective relevance of the cyber-security entities.

In order to know if the given text has any relevance to the user, the Entities of Interest (EI) have been defined as cyber-security entities that concern the user. Entities extracted from the given text are marked as Mentioned Entities (ME) and looked up in the CKG to compute the shortest path with every Entity of Interest. The CKG analyzer follows the process outlined below:

1) Define EIs in the CKG.
2) Extract MEs of label **Vendor** and **CVE ID** from input text using Named Entity Recognizer (As mentioned earlier both the Product and Vendor entities are identified under same label as **Vendor**).
3) For every extracted ME, compute the shortest path with every EI using Dijkstra's shortest path algorithm.
4) Consider number of nodes, number of edges and total cost as output between every ME and EI as per the shortest path.
5) If the ME does not exist in CKG or there is no path that connects ME to a specific EI, then assign -1 for output.
6) For every input text, compute the average number of nodes, edges and average cost for each ME with the label **Vendor** and **CVE ID**.

For example, if the user is interested in the node "Debian Linux" and input text contains an entity "Windows 7", then the features of the text are computed as follows. The shortest path between the node "Debian Linux" and the node "Windows 7" according to Dijkstra's algorithm is through nodes "CVE-2018-15473" $->$ "Enterprise Linux Server" -> "CVE-2019-1125" as shown in Fig. 5.

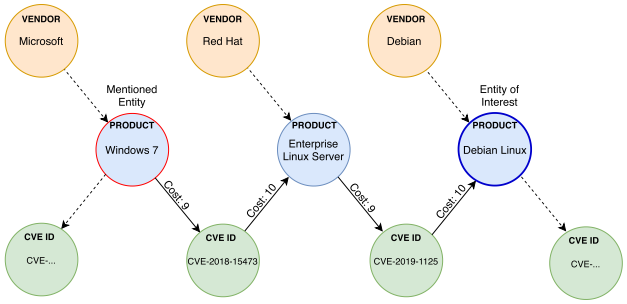

FIGURE 5. Cyber-security knowledge graph example: The correlation between node "Debian Linux" and "Windows 7".

TABLE 5. Features generated by CKG Analyzer.

Vendor			CVE ID		
Nodes	Edges	Cost	Nodes	Edges	Cost
3	4	38	-1	-1	-1

This means Debian Linux was affected by OpenSSH user-enumeration vulnerability CVE-2018-15473 as was the Red Hat Enterprise Linux Server, which also shares Windows Kernel Information Disclosure vulnerability CVE-2019-1125 with Windows 7 OS. This correlation could be represented as the **Vendor** label includes the features of 3 nodes, 4 edges and a total cost of 38. Also, since the text does not contain any **CVE ID** labeled entity, the corresponding features would be all -1. As a result, this input text would generate features as shown in Table 5.

These features are fed into the Significance Score Calculator along with its similarity features to generate the significance score of the text.

D. SIGNIFICANCE SCORE CALCULATOR

Since the objective of the system is to generate quantitative numbers that represent the significance and relevance of the text, we decided to utilize a classifier that could generate numeric output showing the probability of particular element belonging to the significant class. Hence, implementations of the various classification algorithms have been tested using a popular open-source library sklearn.[13]

According to our evaluation, the following classifiers could be utilized as SSC since they could generate probability of items belonging to either class:

- Support Vector Machines (SVM)
- K-Nearest Neighbors (KNN)
- Decision Tree Classifiers (Dec.Tree)
- Gaussian Naive Bayes (GNB)
- Logistic Regression (LogReg)
- Multi-layer Perceptron neural model (MLP)

[13]https://scikit-learn.org/stable/

TABLE 6. Highest degree nodes in CKG.

Node	Degree
Debian Linux	3,686
Android	2,592
Linux Kernel	2,462
Ubuntu Linux	2,334
Mac Os X	2,319
Chrome	1,813
iPhone OS	1,802
Firefox	1,758
HP	1,542
Windows Server 2008	1,506

TABLE 7. Preliminary test results.

Prec	Rec	F1 score	Accuracy
0.5633	0.2649	0.3603	0.5266

Even though the purpose of the SSC is to generate significance score in a form of probability that particular text is significant and relevant to the user, to evaluate the viability of the proposed method, an experiment has been conducted to generate the performance indicators, such as Accuracy and F1 score. Each output of the Similarity Analyzer and CKG Analyzer is inputted into the Significance Score Calculator to generate classification result that can be used to evaluate the overall performance of the system.

V. EXPERIMENT AND EVALUATION

The implementation of the system has been tested by experimenting in different settings. In this section, we will discuss the experiment and the corresponding evaluation results.

A. PRELIMINARY TEST

In order to test the functionality of the proposed system, a preliminary test has been conducted on the test data. The randomly selected 4,000 text documents mentioned in Section IV-A have been split into a balanced set of positive and negative datasets. In addition, the Entities of Interest have to be defined in order to generate correlation features from the CKG. As an arbitrary choice, the highest degree 10 nodes of the CKG are defined as Entities of Interest. Table 6 lists the nodes chosen as Entities of Interest and the corresponding number of edges connected to them as degree.

Once the Entities of Interest are defined, the test data is processed by the Similarity and CKG Analyzers, respectively. The generated features are fed to SVM classifier and 70% of the data is used to train and 30% to test the performance of SVM classifier. The evaluation result is as shown in Table 7.

From Table 7, it can be seen that the proposed system could not work directly, i.e., without a controlled environment it would not be possible to test the full potential of the proposal. Hence, the experimental design has been modified to consider the constraints and simulate a controlled environment. Thus, experimental setup is discussed in the next section.

TABLE 8. Dataset composition.

Dataset	MWTDB	CVE	SE	RSS	Total
Positive	800	600	400	200	2,000
Negative	200	400	600	800	2,000
Total	**1,000**	**1,000**	**1,000**	**1,000**	**4,000**

B. EXPERIMENTAL SETUP

Determining the shortest path of each Mentioned Entity to every Entity of Interest using Dijkstra's shortest path algorithm is a computationally expensive task, hence we decided to conduct the experiment on a limited set of data to prove the viability of the concept. To ease the computational burden, 4,000 text documents mentioned in Section IV-A have been selected and processed using off-the-shelf hardware of 8× Intel i7-7700 CPU 3.60GHz with 16GB of memory.

As mentioned in Section IV-B2, the significance weights of the Reference text repository need to be reflected in the test dataset. Therefore, we constructed the test dataset by allocating more from higher weight repository to the positive dataset. This way, the SSC would be trained to assign higher significance probability to the input text that is similar to the higher weight repository. The dataset composition is shown in Table 8.

Also, as discussed in the previous section, the raw textual data cannot be used to test the full potential of the proposed system. We believe the reasons could include the following:

- The randomly selected test data may not include the cyber-security named entity that could be identified by our NER model.
- The named entities found in the test data may not exist in our CKG; hence, the CKG analyzer is not able to generate correlation data.

Hence, to overcome these constraints, the following assumptions are made to simulate the controlled environment.

- The actual NER model would have perfect performance, so that it can identify any named entity mentioned in the text.
- Every positive data includes at least one named entity that could be identified by our NER model.
- The actual CKG is much larger in scope and contains every entity that is found in the text. In other words, any entity found in the text would give some correlation with the Entities of Interest.

In order to reflect these assumptions, the positive data has been manually manipulated by inserting an entity at the end of the text. In all, 2,000 unique entities with the costliest relationship with chosen Entities of Interest are selected and manually inserted at the end of each of the positive text data.

C. FINAL EVALUATION

For the final evaluation, the classification experiment is conducted using different classifiers on the final dataset, as explained in the previous section. Since the objective of the experiment is not the classification, but rather the viability of

TABLE 9. Different classifier results.

Classifier type	Prec	Rec	F1 score	Accuracy
SVM	0.9938	0.7706	0.8681	0.8792
KNN	0.8767	0.8158	0.8452	0.8458
Dec.Tree	0.8141	0.8207	0.8174	0.8108
GNB	0.9856	0.7722	0.8659	0.8767
Log.Reg	0.9780	0.7916	0.8750	0.8833
MLP	0.9955	0.7092	0.8283	0.8483

TABLE 10. 10-fold cross validation result using Logistic Regression classifier.

Fold	Prec	Rec	F1 score	Accuracy
0	0.9747	0.7857	0.8701	0.8850
1	0.9471	0.7816	0.8564	0.8650
2	0.9808	0.7887	0.8743	0.8900
3	0.9664	0.7461	0.8421	0.8650
4	0.9716	0.8301	0.8953	0.9000
5	0.9770	0.8095	0.8854	0.8900
6	0.9811	0.7393	0.8432	0.8550
7	0.9942	0.8333	0.9067	0.9125
8	0.9615	0.7979	0.8721	0.8900
9	0.9695	0.8281	0.8933	0.9048
Average	**0.9724**	**0.7940**	**0.8739**	**0.8857**

the method of generating the features of the text that could be used by any classifier, we used the default parameter settings for all the classifiers. Table 9 shows the performance differences of using different classifiers.

From Table 9 it could be seen that features generated by the Similarity Analyzer and CKG Analyzer could be used by any classifier to generate significance score. Since the test dataset is small in size, the test result has been validated by conducting 10-fold cross validation. The Logistic Regression classifier shows the highest performance among other classifiers; hence, 10-fold cross validation test has been done using Logistic Regression. The results of the 10-fold cross validation are shown in Table 10.

From Table 10, it could be seen that the proposed Analyzer module could determine the significance and relevance of cyber-security text with an average of 88% accuracy under certain assumptions.

D. ANALYSIS ON THE EVALUATION RESULTS

Based on the results shown in Table 9 and Table 10, we can conclude that the experimental evaluation proves the practical implication of the approach. However, we should acknowledge that these results came as the outcome of several assumptions and simulated environment. The results of Table 7 have shown that the proposed system is not able to function as intended, without the simulated environment.

Also, the poor performance of our Named Entity Recognizer influences the final experiment result. Since achieving a state-of-the-art result in domain-specific NER itself is a separate research problem, we utilized the off-the-shelf NER model that yielded an F1 score of 0.76. We believe by improving the NER performance, not only the experiment result will improve, but also the need to simulate the test data will diminish.

TABLE 11. Classification results on the individual analyzers.

Analyzer	Prec	Rec	F1 score	Accuracy
CKG Analyzer	0.9759	0.7851	0.8702	0.8792
Sim. Analyzer	0.6565	0.6236	0.6396	0.6375

In addition, we did an analysis to see the contribution of the individual Analyzer modules on the overall classification scheme. The features generated by the CKG Analyzer and Similarity Analyzer are fed into the best performing classifier Logistic Regression separately and classification results are shown in Table 11.

From Table 11 it could be seen that the features generated by CKG Analyzer produce an almost identical result as Table 9, hence majorly contributing to the final system performance, whereas features generated by Similarity Analyzer are less likely to influence the final result. One of the reasons for this outcome is that the CKG Analyzer contributes 6 features and Similarity Analyzer 4 features to the final result. Also, the effect of the Similarity Analyzer in the experiment is through the dataset composition mentioned in Table 8. We believe another reason for such asymmetric contributions by the Analyzers is due to the poor distinction between positive and negative examples. Hence such deficiencies should be tackled in the consequent experimental designs.

VI. DISCUSSION

As listed in Section II, there have been various approaches to utilize NLP techniques in cyber defense. The works discussed in Section II-B Cyber-security Knowledge Graph and Section II-C Cyber-security Named Entity Recognition are the examples of specific use cases of NLP in cyber-security. Our work in the scope of this paper was not to achieve the best results in these specific tasks, rather utilize the most optimal implementations of them to prove the viability of the proposed system. The works listed in section II-A Automated Threat Detection have some similarities with our proposed autonomous system. But the main difference in our proposal is to narrow down the textual information into the cyber-security domain and then identify user-specific information by utilizing CKG and NER.

The ultimate goal of the Analyzer module could be seen as a custom version of the text classifier that engineered the textual features to classify the significant and relevant text to the user. However, our focus is to customize the classification result to the user-specific content whereas the traditional text classification studies mentioned in section II-C Text Classification focus on the performance of the classification.

In general, the main aspect of our approach is the attempt to quantify the subjective relevance of the text documents along with its potential significance, which can be customized to meet user-specific needs. To the best of our knowledge, this approach of correlating the extracted entity with an existing knowledge base to determine the subjective relevance has not been attempted in the field of cyber-security. Given the classification accuracy of 88%, we believe this approach may have a practical application.

In addition to primarily academic researches mentioned, the industry has also created a curated knowledge base called MITRE ATT&CK® that enlists adversary behaviors including their tactics and techniques based on real-world observations [33]. It is a powerful framework commonly used as a threat model in adversary emulation, red and blue teaming, and cyber threat intelligence practices. MITRE ATT&CK generalizes the adversary attack techniques and tactics based on the common weaknesses of the systems without mentioning specific product or vulnerability. Our approach for the scope of this paper is to associate and correlate specific vulnerabilities and products thus identifying the user-specific interest and relevance of the cyber-security text. Even though there is no direct relationship between the two approaches, we believe the generalized approach of MITRE ATT&CK could be more practical in the ever-changing and dynamic cyber-security environment. Hence, we plan to extend the research by mapping the vulnerabilities of the Cyber-security Knowledge Graph to the adversary Techniques and Sub-Techniques of the MITRE ATT&CK framework using the relationship extraction methods.

VII. CONCLUSION

In this paper we proposed a novel approach to quantify the relevance and significance of unstructured text that represents subjective importance to the user. We believe our approach could address the problem of processing a massive amount of unstructured text for cyber-security situational awareness. We propose to do it, through textual similarity with pre-defined important documents that the significance of the text can be determined and that by utilizing existing Cyber-security Knowledge Graph to correlate the named entities, the subjective relevance of the cyber-security text could be found. For that, we trained a custom Named Entity Recognition model using over 17 million words and constructed a Cyber-security Knowledge Graph with 221,202 semantic tuples in order to generate features that would represent their significance and subjective relevance. Combining these features, the significance and relevance of the text document could be represented in quantitative numbers.

Due to the constraints such as a probable lack of identifiable cyber-security named entity in test data and the uncertainty of identified Mentioned Entities to exist in CKG the effectiveness of the proposed architecture could not be proven directly on the raw test documents; however, by simulating the controlled environment by manipulating the test document achieved a classification accuracy of 88% using the logistic regression classifier. Since it is impossible to expect the controlled environment in a real-life situation, the experiment has to be improved to reconcile the simulated dataset with real-life data. We believe by improving the NER performance and extending the scope of CKG the experiment would come closer to producing production-grade results.

In addition to addressing these flaws, in the future we will extend the research by extracting the relations from the cyber-security text and building a more comprehensive Cyber-security Knowledge Graph. With the relation extractor that constantly enriches the Cyber-security Knowledge Graph, the whole system would be able to run a full cycle as depicted in Fig. 1.

REFERENCES

[1] O. Mendsaikhan, H. Hasegawa, Y. Yamaguchi, and H. Shimada, "Identification of cybersecurity specific content using the Doc2Vec language model," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Milwaukee, WI, USA, Jul. 2019, pp. 396–401.

[2] S. P. Harter, "Psychological relevance and information science," *J. Amer. Soc. Inf. Sci.*, vol. 43, no. 9, pp. 602–615, Oct. 1992.

[3] A. M. Rinaldi, "An ontology-driven approach for semantic information retrieval on the Web," *ACM Trans. Internet Technol.*, vol. 9, no. 3, pp. 1–24, Jul. 2009.

[4] V. Mulwad, W. Li, A. Joshi, T. Finin, and K. Viswanathan, "Extracting information about security vulnerabilities from Web text," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, Lyon, France, Aug. 2011, pp. 257–260, doi: 10.1109/WI-IAT.2011.26.

[5] A. Joshi, R. Lal, T. Finin, and A. Joshi, "Extracting cybersecurity related linked data from text," in *Proc. IEEE 7th Int. Conf. Semantic Comput.*, Irvine, CA, USA, Sep. 2013, pp. 252–259, doi: 10.1109/ICSC.2013.50.

[6] S. More, M. Matthews, A. Joshi, and T. Finin, "A knowledge-based approach to intrusion detection modeling," in *Proc. IEEE Symp. Secur. Privacy Workshops*, May 2012, pp. 75–81, doi: 10.1109/SPW.2012.26.

[7] C. L. Jones, R. A. Bridges, K. M. T. Huffer, and J. R. Goodall, "Towards a relation extraction framework for cyber-security concepts," in *Proc. 10th Annu. Cyber Inf. Secur. Res. Conf. (CISR)*, 2015, pp. 1–4, doi: 10.1145/2746266.2746277.

[8] N. Dionisio, F. Alves, P. M. Ferreira, and A. Bessani, "Cyberthreat detection from Twitter using deep neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Budapest, Hungary, Jul. 2019, pp. 1–8.

[9] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, "TTPDrill: Automatic and accurate extraction of threat actions from unstructured text of CTI sources," in *Proc. 33rd Annu. Comput. Secur. Appl. Conf.*, Orlando, FL, USA, Dec. 2017, pp. 103–115.

[10] A. Pingle, A. Piplai, S. Mittal, A. Joshi, J. Holt, and R. Zak, "RelExt: Relation extraction using deep learning approaches for cybersecurity knowledge graph improvement," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Vancouver, BC, Canada, Aug. 2019, pp. 879–886.

[11] A. Piplai, S. Mittal, A. Joshi, T. Finin, J. Holt, and R. Zak, "Creating cybersecurity knowledge graphs from malware after action reports," Univ. Maryland Baltimore County, Baltimore, MD, USA, Tech. Rep., Nov. 2019.

[12] Y. Jia, Y. Qi, H. Shang, R. Jiang, and A. Li, "A practical approach to constructing a knowledge graph for cybersecurity," *Engineering*, vol. 4, no. 1, pp. 53–60, Feb. 2018.

[13] V. Yadav and S. Bethard, "A survey on recent advances in named entity recognition from deep learning models," in *Proc. 27th Int. Conf. Comput. Linguistics*, Santa Fe, NM, USA, Aug. 2018, pp. 2145–2158. [Online]. Available: https://www.aclweb.org/anthology/C18-1182

[14] X. Zhong, E. Cambria, and A. Hussain, "Extracting time expressions and named entities with constituent-based tagging schemes," *Cognit. Comput.*, vol. 12, no. 4, pp. 844–862, Jul. 2020.

[15] K. Simran, S. Sriram, R. Vinayakumar, and K. P. Soman, "Deep learning approach for intelligent named entity recognition of cyber security," 2020, *arXiv:2004.00502*. [Online]. Available: http://arxiv.org/abs/2004.00502

[16] H. Gasmi, J. Laval, and A. Bouras, "Information extraction of cybersecurity concepts: An LSTM approach," *Appl. Sci.*, vol. 9, no. 19, p. 3945, Sep. 2019.

[17] F. Yi, B. Jiang, L. Wang, and J. Wu, "Cybersecurity named entity recognition using multi-modal ensemble learning," *IEEE Access*, vol. 8, pp. 63214–63224, 2020.

[18] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning based text classification: A comprehensive review," 2020, *arXiv:2004.03705*. [Online]. Available: http://arxiv.org/abs/2004.03705

[19] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Proc. 33rd Conf. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, 2019, pp. 5753–5763.

[20] O. Mendsaikhan, H. Hasegawa, Y. Yamaguchi, H. Shimada, and E. Bataa, "Identification of cybersecurity specific content using different language models," *J. Inf. Process.*, vol. 28, pp. 623–632, Sep. 2020.
[21] T. Kenter and M. de Rijke, "Short text similarity with word embeddings," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, New York, NY, USA: Association for Computing Machinery, 2015, pp. 1411–1420.
[22] H. T. Nguyen, P. H. Duong, and E. Cambria, "Learning short-text semantic similarity with word embeddings and external knowledge sources," *Knowl.-Based Syst.*, vol. 182, Oct. 2019, Art. no. 104842, doi: 10.1016/j.knosys.2019.07.013.
[23] L. Ehrlinger and W. Wöß, "Towards a definition of knowledge graphs," in *Proc. 12th Int. Conf. Semantic Syst.*, Leipzig, Germany, 2016, pp. 1–4.
[24] O. Mendsaikhan, H. Hasegawa, Y. Yukiko, and H. Shimada, "Quantifying the significance of cybersecurity text through semantic similarity and named entity recognition," in *Proc. 6th Int. Conf. Inf. Syst. Secur. Privacy*, Valetta, Malta, 2020, pp. 325–332.
[25] P. Phandi, A. Silva, and W. Lu, "SemEval-2018 task 8: Semantic extraction from CybersecUrity REports using natural language processing (SecureNLP)," in *Proc. 12th Int. Workshop Semantic Eval.*, New Orleans, LA, USA, 2018, pp. 697–706, doi: 10.18653/v1/s18-1113.
[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Workshop Int. Conf. Learn. Representations (ICLR)*, Scottsdale, AZ, USA, 2013.
[27] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1532–1543, doi: 10.3115/v1/d14-1162.
[28] C. S. Perone, R. Silveira, and T. S. Paula, "Evaluation of sentence embeddings in downstream and linguistic probing tasks," 2018, *arXiv:1806.06259*. [Online]. Available: http://arxiv.org/abs/1806.06259
[29] D. Cer, Y. Yang, S.-Y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil, "Universal sentence encoder for English," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, Brussels, Belgium, 2018, pp. 169–174, doi: 10.18653/v1/d18-2029.
[30] R. A. Bridges, C. L. Jones, M. D. Iannacone, K. M. Testa, and J. R. Goodall, "Automatic labeling for entity extraction in cyber security," 2013, *arXiv:1308.4941*. [Online]. Available: http://arxiv.org/abs/1308.4941
[31] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Ann Arbor, MI, USA, Jun. 2005, pp. 363–370, doi: 10.3115/1219840.1219885.
[32] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and application of a metric on semantic nets," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 1, pp. 17–30, Jan. 1989.
[33] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas. (2020). MITRE ATT&CK: Design and philosophy. MITRE. [Online]. Available: https://attack.mitre.org/docs/

OTGONPUREV MENDSAIKHAN (Student Member, IEEE) received the M.S. degree in information security policy and management from Carnegie Mellon University, Pittsburgh, PA, USA. He is currently pursuing the Ph.D. degree with the Graduate School of Informatics, Nagoya University, Japan.

His main research interests include cybersecurity situational awareness, cyber threat intelligence, and knowledge graph.

HIROKAZU HASEGAWA received the Ph.D. degree in information science from Nagoya University, Japan, in 2017.

He is currently an Assistant Professor with the Information Security Office, Nagoya University. His research interests include internet and network security.

YUKIKO YAMAGUCHI received the B.S. degree in information engineering from the Nagoya Institute of Technology, in 1983, and the M.S. degree in information engineering from Nagoya University.

She was with Fujitsu Laboratories Ltd. She joined the Computer Center, Nagoya University, as an Assistant Professor, in April 1991. She is currently an Assistant Professor with the Information Technology Center. Her research interests include network management technology and cyber-security.

HAJIME SHIMADA (Member, IEEE) was born in 1976. He received the B.E., M.E., and D.E. degrees from Nagoya University, Japan, in 1998, 2000, and 2004, respectively.

He was an Assistant Professor with Kyoto University from 2005 to 2009 and an Associate Professor with NAIST from 2009 to 2013. He has been an Associate Professor with Nagoya University, since 2013. His current research interests include cyber-security, network operation, and computer architecture related researches.

Dr. Shimada is a member of IPSJ and IEICE.