Cyber Information Retrieval Through Pragmatics Understanding and Visualization

Nan Sun, Jun Zhang, *Senior Member, IEEE,* Shang Gao, Leo Yu Zhang, *Member, IEEE,* Seyit Camtepe, *Senior Member, IEEE,* and Yang Xiang, *Fellow, IEEE*

Abstract—The amount of cybersecurity-related information is extraordinarily increasing, given the fast-growing number of cybersecurity attacks and the significant influence brought by them. How to efficiently obtain and precisely understand the relevant knowledge in the sea of information on cybersecurity becomes a challenge. In this paper, we propose an innovative cybersecurity retrieval scheme that supports automatic indexing and searching of cybersecurity information based on semantic contents and hidden metadata. The proposed scheme leverages a customized neural model that incorporates new linguistic features and word embedding by identifying the entities related to cybersecurity incidents from the text. We implement a novel cybersecurity search engine to demonstrate effective, understandable and pragmatic cybersecurity information retrieval based on the proposed schema. Comprehensive performance evaluation over real-world datasets has been conducted to validate the new algorithms and techniques developed for cybersecurity information retrieval. The new engine makes it possible to conduct augmented search, cybersecurity analytics, and visualization, with the ultimate goal of providing direct and efficient results to help people obtain and truly understand cybersecurity information.

Index Terms—Cybersecurity Events, Data-driven, Information Retrieval, Pragmatics Understanding, Search Engine, Visualization.

1 Introduction

With the explosive growth of cybersecurity incidents, governments, enterprises, and organizations begin to collect, store, and analyze unprecedented amounts of data to cope with cybersecurity issues [1]. A cybersecurity incident is defined as "any type of computer network attack, computer-related crime, and the misuse or abuse of network resources or access" by the Australian Computer Emergency Response Team (AUSCERT) [2]. Proactively predicting and discovering cybersecurity incidents plays an increasingly critical role in the last decade. Along with the trend that people attach importance to cybersecurity, more and more people, from individual security professionals or hackers to governments or security sectors, discuss, update and release information about cybersecurity on the Internet. Large volumes of data related to cybersecurity incidents not only offer an opportunity to understand and protect cybersecurity better but also bring a challenge on how to utilize this vast and complex information effectively [3].

Thanks to the invention of search engines that provide users with a way to carry out web-based search and information retrieval via queries. Among them, the Google search engine has become predominant. Every time we google, there are thousands, sometimes millions of web pages returned with relevant information [4]. Besides of the traditional search function, Google also offers multiple response forms based on research findings to help people quickly locate answers to their queries, such as knowledge graphs, featured snippiest and rich lists. Although it is one of the most powerful search engines in the world, Google has limitations when searching cybersecurity-related information. On one hand, a large number of results are hard to read or understand. The amount of information obtained by users generally depends on how much the results being understood rather than the number being retrieved. On the other hand, domain-specific knowledge provided is limited. Due to the lack of cybersecurity domain knowledge in the general search results, human factors are usually required to get involved before users interpreting the information.

Custom Google search engine and Shodan are two typical state-of-the-art cybersecurity search engines [5], [6]. The former [5] can index cybersecurity-related websites and query them. However, the custom search only returns the links of websites containing query keywords instead of multiple interactive responses produced by the general Google search engine. The latter [6] is the world's first search engine for Internet-connected devices. Shodan scans the Internet to find open ports on a given IP address. However, Shodan only emphasizes on device scan information and ignores people without strong cybersecurity background. It does not suit for ordinary people who are expecting to search and learn from the retrieved information.

Motivated by the above observations, we propose a novel cyber information retrieval scheme and design a new cybersecurity search engine. To embrace the large-

- *Nan Sun is with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia. E-mail:nansun.research@gmail.com*
- *Jun Zhang is the corresponding author, and he is with School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, VIC 3122, Australia. E-mail:junzhang@swin.edu.au*
- *Shang Gao and Leo Yu Zhang are with the School of Information Technology, Deakin University, Waurn Ponds, VIC 3216, Australia.E-mail: {shang.gao, leo.zhang}@deakin.edu.au*
- *Seyit Camtepe is with Data 61, CSIRO, Sydney, NSW 2122, Australia. E-mail:seyit.camtepe@data61.csiro.au*
- *Yang Xiang is with the School of Software and Electrical Engineering, Swinburne University of Technology, Hawthorn, VIC 3122, Australia. E-mail: yxiang@swin.edu.au*

scale cybersecurity incidents information, we use heterogeneous information sources instead of a single data source for indexing and retrieval. To emphasize the fact that the amount of information absorbed by users is not depending on the number returned, but relying on the level of understanding of the returned, we leverage a deep neural network with word embedding to identify the entities that are highly relevant to a cybersecurity incident, and index them together with their cybersecurity pragmatic meanings before retrieval. In summary, the proposed schema begins with indexing information from multiple data sources in the form of cybersecurity event tagging. Then, it retrieves the associated results that match a user's search token, followed by representing the results in the form of data analytics and visualization. The contributions of the paper are as follows:

- An innovative cybersecurity-oriented indexing approach is proposed. It stores text with pragmatics about category or contents on cybersecurity incidents, and integrates pragmatic information in conjunction with metadata related to cybersecurity incidents.
- Based on the proposed indexing approach, a novel cyber information retrieval scheme is designed to facilitate people who, with or without security-related domain knowledge, are concerned about ingestion (collection) and digestion (comprehension) of security information.
- New interactive schemes are developed to exhibit search results. Rather than providing a long list of results for users' selection, the retrieved information is delivered in a straight-forward and friendly way with the application of automated data analytics and visualization techniques.

Based on the scheme, we implement a cyber search engine prototype and carry out a large set of experiments on real-world datasets to evaluate the proposed algorithms and techniques. The empirical study demonstrates that the new cyber information retrieval scheme is effective, understandable and pragmatic.

This paper is organized as follows. Section 2 introduces the background and related work. Section 3 summarizes the design and implementation of the index and search engine. Section 4 presents a detailed view of the implemented functions and evaluates the system. Section 5 discusses the remaining unsolved problems as well as future improvements. Section 6 concludes the work.

2 Background and Related Work

2.1 Cybersecurity Information Collection

Cyber Threat Intelligence (CTI) is defined as "evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject's response to that menace or hazard" [7]. Driven by the increasing number of publicly and privately generated CTI data, research communities and industries begin to utilize different kinds of data sources to improve cyber resilience [1]. There are different cybersecurity data sources used in previous studies. Data crawled from web pages provides text as well as meta-information such as author, publication date and HTML format, which is popularly used to discover threats [8]–[10]. Also, social media platforms that generate a steady flow of information ensure the innovative strength to contribute to CTI. Previous work made full advantage of social media data to discover indicators of compromises (IOCs) [11], [12] to detect malicious mobile applications [13] and to find cyber attacks [14]. Besides, Qamar et al. [15] made use of Advanced Persistent Threats (APT) data to gain insights into cyber attacks. Some authoritative datasets published by government and security sectors make CTI more reliable. For example, Common Vulnerability and Exposures (CVE) is the world's leading organization to provide vulnerability information to predict real-world vulnerability exploits. Sabotke et al. [16] and Sun et al. [17] latest work combined Tweets and CVE information in their datasets.

The investigation on previous work suggests that cybersecurity information can be collected from more than one source. To obtain comprehensive, reliable and innovative information on cybersecurity, we consider combining and incorporating multiple data sources in this work.

2.2 Extracting Security Semantic Text

Due to the large volume of security data, it is nearly impossible to review, organize and monitor the tremendous growing amount of information manually. Hence, one challenge is raised how to automatically extract security-related entities from text to avoid "garbage in, garbage out".

There exist sophisticated methods to extract entities from common events, including StanfordNLP, Natural Language Toolkit (NLTK), spaCy, OpenNLP and Gate [18]–[20]. According to the comparison study [20], StandfordNLP [19] usually performs the best. Event entities, such as location, organization and person from a sentence can be automatically labelled. However, these entity recognizers target at annotating common events due to a lack of security domain knowledge.

Some recent work has begun to pursue approaches to extract cybersecurity events [21]–[24]. Syed et al. [21] proposed the first cybersecurity ontology that mapped the general world ontologies to security use cases. Lim et al. [24] introduced an annotation framework based on APT reports for defining malware characteristics. Bridges et al. [22] developed a highly precise method to label cybersecurity relevant text automatically. The latest work [23] is the state-of-the-art entity extraction system in the cybersecurity domain. It refines the security entity categories by describing how security events could be depicted via semantic schemes. With the development of natural language processing (NLP), the evolution of NLP evolves from "bag-of-word" to "bag-of-concepts", and eventually turns out to "bag-of-narratives" [25]. When it comes to security information extraction, getting the message or story from the text seems more valuable than accessing the word itself.

2.3 Search Engine

Information retrieval is a science of searching for information from text, images or sounds, and also searching for information from metadata that describes data [26].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2022.3151148, IEEE Transactions on Dependable and Secure Computing

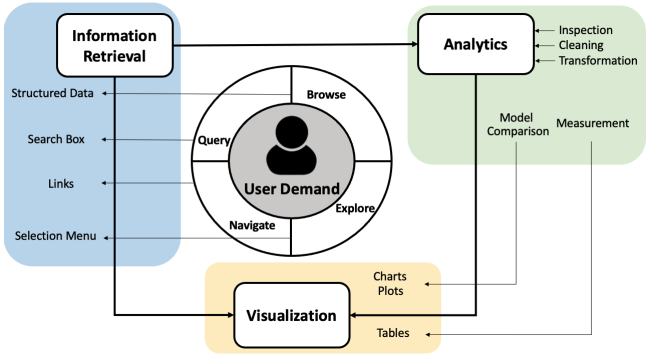


Fig. 1: Pragmatic cyber information retrieval

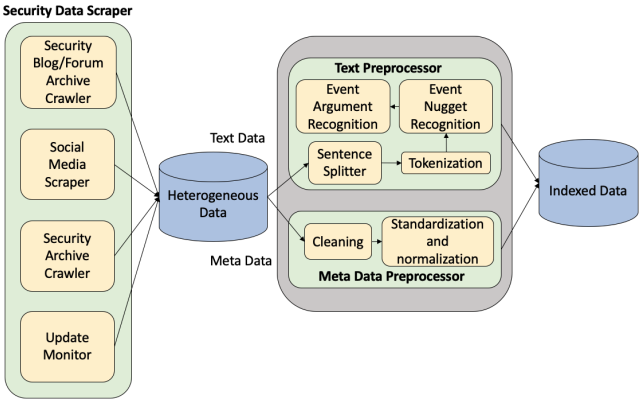


Fig. 2: The process of data indexing

Information is found on the Internet by leveraging search engines.

Usually, a search engine keeps a copy of extensive collections of web pages and related information using URLs as row keys and various aspects of web pages as columns. Search algorithms sort through hundreds of billions of web pages to locate the most relevant results and present them in multiple formats. After reviewing the general-purpose and existing cybersecurity domain search engines in Section 1, we set our ultimate goal: design a search engine that learns from the existing search algorithms and provides interactive features like the traditional search engines; meanwhile, it balances between the scarce cybersecurity domain knowledge and the overloaded awaiting ingestion information, and provides direct and efficient results processed by rich data analytics and visualization techniques.

3 Methodology and Scheme

In this section, we present the methodology and scheme of pragmatics cyber information retrieval, and the design of new search engine. We put users at the center of the system and design the system around users' demands. The first and upmost question to address is how the data from multiple data sources should be indexed as preparation for the search engine. During the process of data indexing, the approaches to extracting, allocating and annotating critical cybersecurity-related tokens are illustrated.

3.1 Pragmatic Retrieval Methodology

The ultimate goal of our cybersecurity search engine is to provide direct and effective results to users, helping them easily capture the information they are looking for. If provided with a long list of ranked contents and links, users will have to select information based on their comprehensive ability and the level of cybersecurity expertise. How to deliver the search results efficiently and help users easily get the information and knowledge behind is challenging. In this subsection, we introduce the presentation of search results from two aspects: (1) security information retrieval; (2) analytics and visualization. As summarized in Figure 1, we take users' demand as the central point to design the functions by means of information retrieval.

We start from considering users' behavior during and after receiving search results to deliver the result representation. Usually, users input query keywords in a search box based on their needs. Once the results are returned, users browse and filter them. To organize the results in a more structural way, we embed structured information (e.g. pragmatic text and metadata) in the index. If users have substantial interests in a known target, the links leading to the corresponding data sources are provided for their further navigation. Moreover, a selection menu is designed to assist those users without a specific search target but who are interested in exploring hot cybersecurity topics. Our search engine automatically summarizes the returned search results in virtue of data analytics. During the indexing process, data is inspected, cleaned and transformed. As to the result presentation, data is modeled, compared and analyzed before being delivered to users.

At the same time, data visualization conveys both concrete and abstract ideas to users in an effective way [27]. Complex data becomes more accessible, usable and understandable to users when we make use of efficient graphic representation of data. Data visualization based on analytics is a branch of descriptive statistics. To make comparisons, understand causality and reveal relationships, patterns and trends existing in data, we deploy multiple types of visual presentations, such as frequency charts, relationship networks and word cloud. Furthermore, structured information saved in spreadsheets facilitates further evaluation and exploration in relevant cybersecurity fields.

3.2 Deep Cyber Index

3.2.1 *The Process of Data Indexing*

The search index is the fundamental component of a search engine. It is the body of structured data that the search engine can refer to when looking for information based on a specific query. The process of data indexing is shown in Figure 2. It indexes heterogeneous data from multiple data sources.

To obtain representative and reliable cybersecurity data, we collect data from a variety of data sources. Security blogs embrace articles composed by security specialists all over the world. Security enthusiasts, professionals or even hackers post their thoughts and remarks on social media platforms. Also, security archives published by security organizations, such as CVE, provide authoritative and definitive information. An update monitor is set up to keep the information up to date.

TABLE 1: An example of metadata index

Document #	Link	Author	Date	Comments #	Title	Text
881	https://www.bleepingcomputer.com/news/security/over-67-000-websites-defaced-via-recently-patched-wordpress-bug/	Catalin Cimpanu	7/2/2017	1	Over 67,000 Websites Defaced via Recently Patched WordPress Bug	WordPress sites that haven't been updated to the most recent version, v4.7.2, released last week, are under attack as four hacking groups are conducting mass defacement campaigns...

TABLE 2: An example of text index with/without cybersecurity entities

Event #	Word	Nugget	Argument
881	According	O	O
	To	O	O
	Web	O	O
	Security	O	O
	Firm	O	O
	Sucuri	O	B-Organization
	,	O	O
	Who	O	O
	Detected	B-Discover Vulnerability	O
	The	O	O
	Attacks	O	O
	After	O	O
	Details	O	O
	Of	O	O
	The	O	B-Vulnerability
	Vulnerability	B-Discover Vulnerability	I-Vulnerability
	Became	I-Discover Vulnerability	O
	Public	O	O
	Last	O	B-Time
	Monday	O	I-Time
	.	O	O

After collecting data from multiple data sources, the challenge now is how to structure the heterogeneous data with a high diversity of data types and in different formats. We extract and split each collected data item into two layers. One layer stores the pragmatic text only, including descriptions from security archives, paragraphs from articles and post contents from forums and social media. After splitting data items into sentences and then sentences into tokens, an entity recognizer applicable to the cybersecurity domain is applied to tag security event nuggets and arguments of each sentence. Each sentence token is tagged by the BIO schema, which completes text indexing. BIO schema is a lightweight, but high-efficiency tagging format for tagging tokens in named entity recognition, based on the comparative experiments from the study of Reimers et al. [28]. An example of text index with or without cybersecurity entities is demonstrated in Table 2, where "B-", "I-" and "O" represent the beginning of an entity, the continuation of an entity and no entity, respectively. The detail of how to extract and tag cybersecurity-related information is described in Section 3.2.2. Through pragmatic text indexing, the security-focusing information is stored. Another layer stores metadata that describes the pragmatic text reserved by the first layer, including the publication date, author, source link, frequency of comments and likes. The process of metadata index creation is shown as Algorithm 1. After steps such as data cleaning, standardization and normalization, metadata is indexed as structured fields ready for search as shown in Table 1.

Algorithm 1 Metadata Index Creation Algorithm

INPUT: Preprocessed JSON, CSV or HTML files, containing metadata of security information crawled from heterogeneous data sources
OUTPUT: $Set_{MetaDataIndex} = (ID_D, L, A, D, C, T_1, T_2)$: A set of metadata index for each input file. ID_D, L, A, D, C, T_1 and T_2 represent document ID, link, author, published date, the number of comments, title and text, respectively.

```
Read data files
for each data file do
    Assign an ID to each document. Start at 1 with increment by 1
    Check metadata of the data file
    for each element in metadata do
        Create metadata index
        Initialize fields of index
        MetaDataIndex = (ID_D, L, A, D, C, T_1, T_2)
        if the file includes the element then
            Store meta data into corresponding field of index with document ID
        else
            Label O in the corresponding field
        end if
    end for
    Update Set_MetaDataIndex = (ID_D, L, A, D, C, T_1, T_2)
end for
return Set_MetaDataIndex
```

3.2.2 Information Extraction

When people describe an event, it is common that they point out the event type in the first place, for example, music festivals or sports. When further describing the event, people add details about the event, including location, time, people evolved and other related elements. To describe an event better, the event types and event details usually provide a direct and clear picture. Similarly, when describing a cybersecurity event, it goes the same way only if the relevant security domain knowledge is provided. Motivated by this idea, we extract the element information associated with event type from the text and encode the text with the extracted information. When searching for a specific topic, results that directly describe the security event can be retrieved.

As the related work reviewed in Section 2.2, CASIE [23] used a deep neural network model to extract semantic information from cybersecurity data. Inspired by the idea of CASIE, we use its latest and most detailed schema to describe a cybersecurity event as follows:

- Event nugget: a word or phrase that can most clearly summarize the event occurrence, including

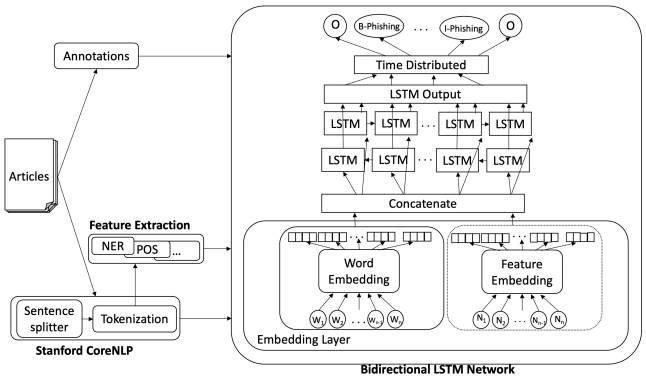


Fig. 3: Neural network based cybersecurity entity recognizer

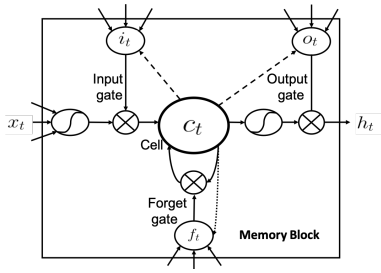


Fig. 4: The structure of an LSTM unit

data breach, phishing, ransom, discovered vulnerability and patched vulnerability.

- Event argument: a word or phrase that describes the participants or properties of an event, including organization, person, device, product, website, capability, file, malware, money, number, purpose geopolitical entity (GPE), time, CVE, vulnerability and version.

Before indexing the text data of a given article, we extract the event nugget and event argument from the text, tag the tokens and index using the cybersecurity entity recognizer as shown in Figure 3. The groundtruth of training data for establishing the neural network model derives from 1000 articles that discuss the five types of events mentioned in the definition of event nugget. The 1000 articles were published by the project [29], each of which was labeled by three experienced computer scientists and a majority vote was adopted to decide the final annotation.

When establishing the neural network-based cybersecurity entity recognizer, we apply Stanford CoreNLP [30] to split sentences from articles with a punctuation mark and then divide tokens from sentences. Each token is tagged with corresponding features, such as Named Entity Recognition (NER) annotated by CoreNLP [30], Part of Speech (PoS) labeled for each word, similar to the process of feature extraction described in the existing work [23]. After that, word embedding and feature embedding are concatenated and fed into the bidirectional Long-Short-Term Memory (LSTM) recurrent neural network (RNN). It uses both the left-to-right and right-to-left information to predict a target word.

For a given input sequence $x_1, x_2, ..., x_T$, RNN computes a hidden state sequence $h_1, h_2, ..., h_T$ from the input, and output $y_1, y_2, ..., y_T$ according to

$$h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h),$$
$$y_t = W_{hy}h_t + b_y,$$

where $t = [1, T]$, W_{xh} (W_{hh} and W_{hy}) represent the weight matrix between input and hidden (hidden and hidden, and hidden and output) states, b_h (b_y) is the hidden (output) bias vector, and $\mathcal{H}$ represents a nonlinear activation function.

Conventional RNN can only access past context and ignore future context. For a security entity recognition system, we expect the network to be able to access both the proceeding and succeeding context. Hence, as shown in Figure 3, we utilize the bidirectional RNN to address the issue. Bidirectional RNN separates each hidden layer into two parts and computes the forward state sequence $\overrightarrow{h}$ and backward state sequence $\overleftarrow{h}$ as follows:

$$\overrightarrow{h}_t = \mathcal{H}(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}})$$
$$\overleftarrow{h}_t = \mathcal{H}(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}})$$
$$y_t = W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y$$

Due to the gradient vanishing problem, conventional RNN can only access a limited range of context, preventing it from modeling the long-span relations in sequential features. The gradients carry information used for the RNN parameter updates. When the gradients become smaller, the parameter updates become less significant, resulting in ineffective learning outcomes. Long short term memory (LSTM) [31], as shown in Figure 4, is designed to overcome the limitation of conventional RNN and address the gradient vanishing problem [32]. With the unique additive gradient structure, LSTM enables the network to encourage desired behavior from the error gradient using frequent gates update on every time step of the learning process. It utilizes memory cells built inside to store information with a mixture of low and high-frequency components. Let i, f, o and c denote respectively the input gate, forget gate, output gate and cell memory, then the recurrent hidden layer function $\mathcal{H}$ is implemented as

$$i_t = \sigma(w_{xi}x_t + w_{hi}h_{t-1} + w_{ci}c_{t-1} + b_i),$$
$$f_t = \sigma(w_{xf}x_t + w_{hf}h_{t-1} + w_{cf}c_{t-1} + b_f),$$
$$c_t = f_t c_{t-1} + i_t\theta(w_{xc}x_t + w_{hc}h_{t-1} + b_c),$$
$$o_t = \sigma(w_{xo}x_t + w_{ho}h_{t-1} + w_{co}c_t + b_o),$$
$$h_t = o_t\theta(c_t),$$

where σ is a sigmoid function and θ is a non-linear activation function, such as tanh. The multiplicative gates enable LSTM memory cells to store and further access data over long periods, whereby relieving the vanishing gradient problem.

Moreover, the time distributed layer applies an operation to every temporal slice of an input to build up the final output sequence. Adaptive Moment Estimation (Adam) [33] optimizer is employed to train the network. It leverages the power of adaptive learning rates to find individual learning rates of each parameter. Adam stores the running average of

past gradients m_t and the running average of past squared gradients v_t as follows:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t,$$
$$v_t = \beta_2 m_{t-1} + (1-\beta_2) g_t^2.$$

Here, g_t is the gradients at iteration t, β_1 and β_2 are decay rates, m_t is the first moment (mean) of gradients and v_t is the second moment (uncentered variance) of gradients. Because m_t and v_t are initialized as 0-vectors, they are biased towards 0. Bias-corrected first and second moments are calculated to update the model parameters by

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t},$$
$$\widehat{v}_t = \frac{v_t}{1-\beta_2^t}.$$

The corrected moments are finally used with the Adam update rule:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\widehat{v}_t} + \epsilon} \widehat{m}_t.$$

When conducting word embedding, we apply ELMo embedding [34]. It captures both meanings of the word in the context and other contextual information. Instead of creating a fixed embedding for each word, ELMo investigates the entire sentence before assigning each word in an embedding. ELMo as a deep contextualized word representation model uses a deep bi-directional LSTM pre-trained on a large text corpus (1 billion word benchmark) to create those embeddings. With 900 articles as training data and 100 articles as testing data, our network with ELMo embedding reaches 0.79 F-1 score and 99.8% accuracy. The evaluation details of the cybersecurity domain entity recognizer is shown in Section 4.2.

With the established neural network-based cybersecurity entity recognizer, when new text comes in, critical information in the text is extracted, labeled and indexed, as the example shown in Figure 2. The process of text index creation is presented as Algorithm 2. Together with metadata, the collected data is indexed with sufficient details and structured in an easy-to-retrieval format.

Algorithm 2 Text Index Creation Algorithm

INPUT: Metadata index $Set_{MetaDataIndex} = (ID_D, L, A, D, C, T_1, T_2)$
OUTPUT: $Set_{TextIndex} = (ID_D, ID_E, ID_W, Word, N, A)$: A set of text index for each document in $Set_{MetaDataIndex}$. ID_D, ID_E, ID_W, $Word$, N and A represent document ID, event ID, word ID, word, event nugget and argument label with BOI format, respectively.

for each record in $Set_{MetaDataIndex}$ **do**
  if ID_D & T_2 != O **then**
    Split T_2 into sentences and tokens using Stanford CoreNLP
    Initialize and update $Set_{Token} = (ID_D, ID_S, ID_W, Word)$
    Extract nugget and argument from Set_{Token} using security entity recognizer
    Update $Set_{Token} = (ID_D, ID_S, ID_W, Word, N, A)$
    if N != O | A != O **then**
      $ID_E \leftarrow ID_S$
      Update $Set_{TextIndex} = (ID_D, ID_E, ID_W, Word, N, A)$
    end if
  end if
end for
return $Set_{TextIndex}$

	ID	word	nugget	argument	Attack
1	4677	Heimdal	O	B-Organization	DiscoverVulnerability
2	4677	Security	O	I-Organization	DiscoverVulnerability
3	4677	also	B-DiscoverVulnerability	O	DiscoverVulnerability
4	4677	claims	I-DiscoverVulnerability	O	DiscoverVulnerability
5	4677	RIG	O	B-Capabilities	DiscoverVulnerability
6	4677	attempts	O	I-Capabilities	DiscoverVulnerability
7	4677	to	O	I-Capabilities	DiscoverVulnerability
8	4677	exploit	O	I-Capabilities	DiscoverVulnerability
9	4677	vulnerabilities	O	B-Vulnerability	DiscoverVulnerability
10	4677	Internet	O	B-Software	DiscoverVulnerability

Fig. 5: An example of nuggets and arguments for cybersecurity event with ID 4677

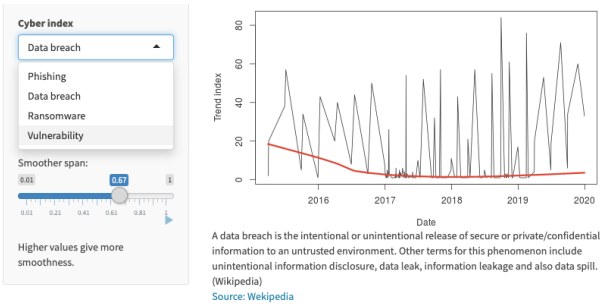


Fig. 6: Cybersecurity trend

3.3 Information Retrieval Approaches

We utilize four information retrieval approaches driven by users' demands, as shown in Figure 1: (1) displaying structured data; (2) selecting from indexed security categories; (3) redirecting to data sources by hyperlinks; (4) searching by keywords. The first three can operate separately or based on the search results of the last approach. Specific information retrieval strategies are described below.

3.3.1 Structured Data

Structured data refers to highly organized data, which is easy to understand by both machine and humans. As shown in Figure 5, all the event nuggets and arguments tagged for one sentence are recorded by a unique ID, which provides an explicit clue to the cybersecurity event for quick browsing. The standardized format classifies the event type and provides descriptive information on the event, such as date, organization or capability. In addition, the returned metadata, such as the number of likes, comments and author details stored in the structured data in the form of a spreadsheet, provides users with an opportunity to explore relevant information further.

3.3.2 Selection Menu

The selection menu is a straight forward function. It is designed to facilitate users to promptly search cybersecurity information with relevant security domain knowledge. If a user prefers investigating the security trend of one particular security threat, they can choose it from the cyber index. The index lists all the common threats as shown in Figure 6. The dropdown selection menu provides possible arguments (e.g. capability, vulnerability, software and others) that provides more convenience when people under-

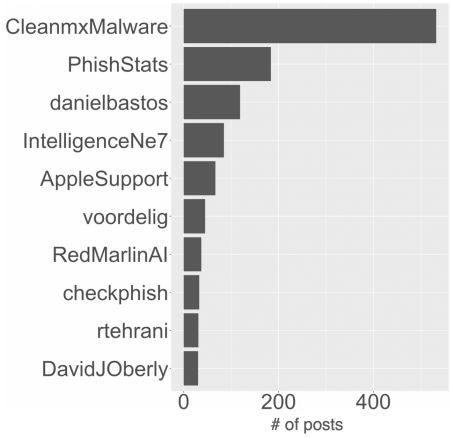


Fig. 7: Sources of top 10 posts about "phishing" between 2015 and 2019

stand a cybersecurity event. It also requires a certain level of expertise to manipulate, as presented in Figure 5. The elements listed in the selection menu provide unambiguous and direct messages to security specialists if they decide to further dig in.

3.3.3 Links

A hyperlink that commonly exists in a web search engine points the specific search result to its data source (e.g. document, website, social media platform and other sources). It redirects users to navigate more detailed information. In our security search engine, links to information sources are provided.

3.3.4 Search Box

The search box provides an interface to conduct the search, where keywords or queries are usually used. In our search engine, multiple search bars are provided to improve the interaction between users and background cybersecurity engine. Figure 5 shows a search box used to locate security event. Besides, relevant resources can be retrieved based on the user's request. Users can search for information based on one or multiple keywords. They can also filter data with specified authors and chosen time ranges, which is demonstrated by our prototype system that has been developed and presented in Section 4.1.2.

3.4 Information Analytics and Visualization

Once information (both pragmatic text and metadata) is retrieved from the database, it is modeled, compared and presented in the form of various tables, charts and plots after rich data analytics and visualization schemes being applied.

3.4.1 Security Trend

Showing particular trends in cybersecurity fields reveals not only the hot topics in the field but also their influence. Based on the statistics, various security trends can be observed. For example, the information flow tendency on cybersecurity is shown in the form of a line chart as Figure 6. The pattern of certain data characteristics (e.g., comments on posts) is represented using the boxplot with quartiles. The details

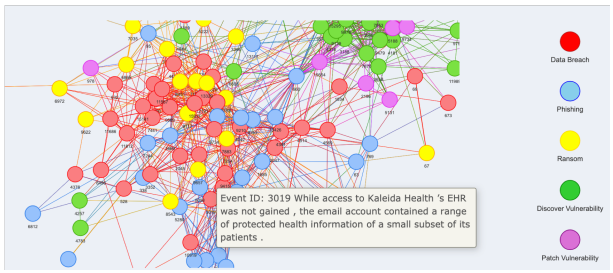


Fig. 8: Security event visualization

on particular items or security components (e.g., trend on the number of posts published by a particular source that enables the investigation on the qualities of data sources [11]) are visualized by histogram as Figure 7.

To aid users in exploring the potential tendency, a smoother is added to fit the data and underline the trend. Extreme trend changes like sharp increases can raise an alarm and advise the severity of an issue. It motivates users to investigate deeper into the irregular period by searching for other information within the selected time range. Here we fit the robust locally-weighted polynomial regression curve proposed by Cleveland [35]. As shown in Figure 6, a smoother is plotted to simulate the trend. As the smoother span is adjusted, the percentage of close points in the plot is changed accordingly, which decides the smoothness at each value based on the selected time range. Normally, the larger value on the smoother span, the smoother the curve.

3.4.2 Security Event Visualization

"A picture is worth a thousand words" [36]. Data visualization is usually applied to engage humans in discovering the pattern in data. It enables storytelling with cybersecurity and facilitates communication among various stakeholders.

Algorithm 3 Event relationship model

INPUT: $Set_{Event} = (ID, N, A)$: A set of events with event ID, phrase of event nuggets and phrases of event argument
OUTPUT: A set of events with event category $Set_{EventCategory} = (ID, C)$, a set of event pairs with the weight between each event pair $Set_{Pair} = (E_1, E_2, W)$.

if Exist(ID & N in Set_{Event}) **then**
  $C \leftarrow N$
  Initialize and update $Set_{EventCategory} = (ID, C)$
end if
while Exist(ID & A in Set_{Event}) **do**
  $Set_{Argument} = (A, ID)$
end while
Full joins A in Set_{Event}
Group by event pair (E_1, E_2)
$W \leftarrow len(unique(A))$
if E_1 != E_2 **then**
  Update $Set_{Pair} = (E_1, E_2, W)$
end if
return $Set_{EventCategory}$ and Set_{Pair}

To show the relationships between cybersecurity events, we visualize the security event network, as shown in Figure 8. Each color in the relationship plot represents a security category, such as phishing, ransomware and vulnerability. Each node represents one security event and the edge denotes the relationship between two events. The algorithm

3 demonstrates the process of constructing the event relationship model. Simply put, during the process of data indexing, each sentence is indexed with an event nugget and corresponding event arguments. If a sentence is tagged with one event nugget, the sentence directly describes a specific categorized cybersecurity event. Besides, if a sentence is deemed as a description of a security event and tagged with one or more event arguments, it covers more details about the event. Hence, if two events share one or more event arguments, the two events are related. As the frequency of repeated arguments increasing, the relationship between two events becomes closer. The relationship weight is calculated and noted as the weight of the edge between the two events.

Other visualization techniques are also leveraged to describe cybersecurity events, which can be found in the implemented prototype introduced in Section 4. When a user receives a bunch of data, word frequency distribution can be used as the weight to conduct visualization in the form of word cloud. Heatmap, as a graphical representation of data, is used to show the popularity of discussed topics, words or specific security arguments based on the statistics. Furthermore, the location map shows the geography information obtained after the analytics of cybersecurity data. These visualization techniques are used to help users gain deeper insights into cybersecurity information. Compared with text-based results retrieved, the representation brought by data analytics and visualization is more effective and meaningful. It enables users to manage information more effectively towards their needs and objectives.

4 Implementation and Evaluation

We implement the proposed search engine using Shiny [37]. Shiny is a robust web framework for building interactive web applications. Our interactive search engine is not data driven, but goal driven and case driven by considering users' demand. In this section, we describe the search engine's implementation in detail, including the datasets used, settings adopted and the prototype modules implemented. Our source code, dataset, and experimental evaluation are released to the community [38].

4.1 System Implementation

4.1.1 Datasets and Settings

Three datasets are utilized in our study to establish the search engine prototype: 1000 security news articles [29] that mentioned five security events and annotated by experienced security experts; vulnerability archives collected from authoritative vulnerability database [39]; tweets from 2015 to 2020 that mentioned security keywords using a Python package called Twitterscraper [40]. During the implementation phase, the parameters adopted are listed as follows:

Pragmatic tags for indexing cybersecurity events: The tags used for indexing sentences contain cybersecurity information. Cybersecurity event nuggets and arguments represent cyber semantic details. They are automatically extracted from text and indexed in a structured format as introduced in Section 3.2. There are five category of event nugget, including "patch vulnerability", "data breach",

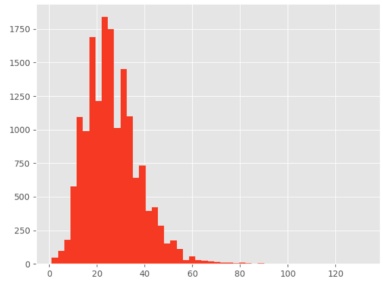


Fig. 9: Length of sentences in groundtruth used to establish the security entity recognition neural network

"discover vulnerability', "ransom" and "phishing". Each event nugget/argument is tagged using the BOI format, which indicates the tokens are not only limited to a single word but also phrases. For those sentences annotated with event nugget, 21 type of arguments are further replenish the details of the event, including "capabilities", "CVE", "data", "device", "file", "GPE", "malware", "money", "number", "organization", "patch", "payment method", "person", "PII", "purpose", "system", "software", "time", "version", "vulnerability" and "website".

Threshold of sentence length: The sentence length distribution of the groundtruth data is used to set up the security entity recognition model. As shown in Figure 9, the lengths of the majority sentences are under 80 words, while the most extended sentence length is 132 words. Based on the performance of the model, the capability is not limited to the long sentences when the number of words is under 132, so we set the maximum length of sentence to 132.

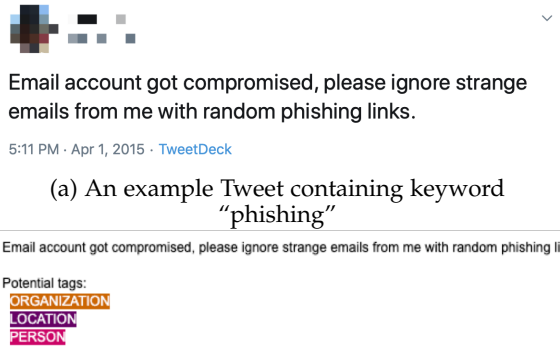
Parameters of neural network: The embedding dimension is the dimensionality of embedding vectors. It is set to 132 according to the threshold of sentence length. The number of bidirectional LSTM units or the number of the Bi-LSTM units per layer is set as 1024. We experimentally tune the batch size, epoch and optimization function for better performance. Based on the results, the batch size, being the number of training samples propagated through the network each time, is set as 32. The monitor is attached to control the training process if the validation accuracy ceases to increase. The epoch, being one cycle of forwarding and backwarding pass for all the training data through the network, is set as 4. The optimizer is Adam [33] that is a method for stochastic optimization, and the loss is sparse categorical cross-entropy [41].

Interactive search engine: An interactive system is a system that highly responses to users' input and reaction. With the aim of supporting user-driven customization, we build a web-based search engine prototype system using Shiny framework [37].

4.1.2 Search Modules

To demonstrate the proposed automatic index and search engine, we apply the indexing approach introduced in Section 3 to our collected dataset. With the indexed text and meta information, four modules are built to manifest the interactive search engine proposed.

The first module *Trend* [42] reflects the security trends using the metadata of the collected cybersecurity data sources.



(a) An example Tweet containing keyword "phishing"

(b) Entities labeled by Stanford named entity tagger [30]

(c) Entities labeled by our cybersecurity entity recognizer

Fig. 10: Examples of cybersecurity entities extraction

The second module *Cybersecurity Event Nugget and Argument* [43] is built based on the security text index. It shows up-to-date, explainable and precise information according to users' demands. The third module *Security Event Visualization* [44] shows the relationship between security events across different security domains. And the fourth module *Security Search and Visualization* [45] aggregates search and presentation, data modeling and visualization, and supports report download for the investigated cybersecurity data. The results of the set of search modules provide insight into cybersecurity through multiple use cases, such as annual cybersecurity report generation, actionable mitigation strategies recommendation and security information interpretation. Please refer to paper [46] for the use cases details.

4.2 Performance Evaluation

Security Orchestration, Automation and Response (SOAR) [49], as an emerging term and a goal pursued in cybersecurity, is used to describe the software capabilities that enable users to collect data about threats from multiple data sources and automatically conduct low-level security tasks (e.g., threat and vulnerability management, security incident response and security operations automation) without human assistance [50]. Designed to support a range of organizations, enterprises and individual users to enhance cybersecurity and efficiency, our proposed system automatically indexes security information and presents compelling security messages. These messages are closely related to the security components based on users' queries and requirements through information retrieval. In this section, we conduct detailed evaluations on our system and its core units by answering the following research questions (RQ1-5).

RQ1: What is the coverage of cybersecurity pragmatics in terms of information extraction? The foundation of our proposed system is indexing the security data. The indexing outcomes are stored into two different buffer pools in the form of semantic information extracted from the text and metadata. By automating the process of extracting semantic information from security data, users are freed from repetitive tasks. Without spending hours compiling security reports, instead, they can now focus on higher valued tasks such as security forensic investigation or decision making. Meanwhile, the automatic cybersecurity information extraction process can bridge the gap between the security experts and beginners to some extent, making the expertise more reachable for general users.

Cybersecurity data downloaded from various data sources are stored in the database through which search engine employs data indexing. Through the cybersecurity entity recognizer, the semantics of text is precisely extracted with the aid of the predefined schema. With metadata providing critical information about the source data, the indexing process is crucial for later information retrieval. As discussed before, when people receive a message, we understand what happened effectively by integrating the information structure and its specifications. Inspired by the way that the human brain works when acquiring knowledge, we extract the cybersecurity event categories and corresponding security details from the text and index the text in a semantic structure. Figure 10 shows an example of cybersecurity entities labeled by our entity recognizer. The recognizer precisely tags the cybersecurity nuggets and event arguments. Compared to the entities labeled by a traditional named entity recognizer as shown in Figure 10(b), the semantics of text associated with cybersecurity are extracted and tagged with domain knowledge by our recognizer.

Some existing work has been done to extract the security event categories and corresponding arguments. The event categories identify the main security components mentioned in sentences, paragraphs or documents. The categories include security attacks, threats and vulnerabilities. Furthermore, security arguments refine the semantics of the security component by supplementing details, such as the security event involved people, location, time and other elements that uniquely represent it. To evaluate the coverage of our information extraction method, we quantify the number of derived event types and arguments in the existing work, as shown in Table 3.

According to the statistics of extracted cybersecurity information, some work focused on collecting information from one particular type of cyber attack. Work [47] specialized in picking up the four-dimensional key security factors from the text when it comes to ransom. Besides, work [24] concentrated on extracting security components related to malware. Moreover, some existing work [14], [48] conducted a proof-of-concept to demonstrate that their approaches could cover more than one security event type. However, the security detail hidden in the text was not specified, which is hard to facilitate users' further genuine understanding of the security component. From the quantitative measurement shown in Table 3, it is clear that our approach covers the most extensive scope and support the complexity of cybersecurity events.

RQ2: What is the performance of entity recognizer with different embedding models? We evaluate and compare the performance of our proposed entity recognizer with LSTM ELMo embedding and fine-tuned BERT embedding

TABLE 3: Cybersecurity information extraction comparison

	Data breach	Phishing	Ransom	Discover vulnerability	Patch vulnerability	DDoS	Account hijacking	Malware
[14]	0	No	No	No	No	0	0	No
[24]	No	No	No	No	No	No	No	12
[47]	No	No	4	No	No	No	No	No
[48]	The event is annotated to "cyber attack" or "other". Event arguments are not specified.							
Ours	15	14	13	10	11	No	No	No

TABLE 4: Performance comparison of the entity recognizer with different embedding models

Embedding	Event category	Precision	Recall	F-1 Score	Avg F-1 score
LSTM + ELMo	Phishing	0.82	0.74	0.78	0.79
	Ransomware	0.77	0.76	0.76	
	Patch Vulnerability	0.84	0.89	0.86	
	Discover Vulnerability	0.76	0.77	0.76	
	Data Breach	0.80	0.77	0.78	
Fine-tune BERT	Phishing	0.58	0.61	0.59	0.62
	Ransomware	0.56	0.54	0.55	
	Patch Vulnerability	0.72	0.72	0.72	
	Discover Vulnerability	0.66	0.60	0.63	
	Data Breach	0.61	0.63	0.62	

TABLE 5: Statistics of event nuggets in a sample of labeled data

Event type	Number	Avg nugget length	Avg nugget length in groundtruth
Phishing	2270	1.77	1.95
Discover Vulnerability	205	1.4	1.57
Data Breach	113	1.75	1.83
Ransom	84	2.76	2.29
Patch Vulnerability	41	1.46	1.57

TABLE 6: Statistics of event arguments in a sample of labeled data

Event argument	Number	Avg argument length	Avg argument length in groundtruth
Person	600	1.44	1.6
File	384	2.98	2.2
Organization	206	2.7	1.8
Vulnerability	93	2.91	2.4
System	91	2.34	2.2

by using metrics including precision, recall, F1 score and accuracy. When training the cybersecurity entity recognizer neural network, we first use Stanford CoreNLP [30] to split paragraphs into sentences and further divide sentences into tokens. Then, we apply the ELMo embedding and the BERT embedding to create contextualized word embeddings to establish the neural network. The former is one of the state-of-the-art word embedding models and the latter can achieve excellent performance in many natural language processing tasks [23], [51], [52]. The neural network with both the ELMo embedding and BERT embedding achieves nearly perfect accuracy, which is 99.8% and 99.2%, respectively. However, when considering F1 score that combines precision and recall to evaluate the performance of per category classification, as the result shown in Table 4, it can be seen that the adapted security entity recognizer with ELMo embedding outperforms the neural network with fine-tune BERT embedding for labeling event nuggets. Among the various categories of security events, events related to vulnerability achieve relatively better performance due to the fact that their used terms are in a specific format, such as CVE ID consisting of "CVE prefix-year-digit number".

To further evaluate the performance of our proposed security entity extraction approach, we randomly select a bunch of tweets containing the keyword "phishing" from our dataset. These Tweets as input data are automatically split into tokens and have the event nuggets and arguments extracted from the text through the security entity recognizer. By splitting the tweets into sentences and sentences into tokens, 204,306 tokens and 16,800 sentences are identified in the 7,550 tweets posted from 2015 to 2019. Statistics of the extracted event nuggets and the top five arguments are listed in Table 5 and Table 6. Among the 7,550 tweets, 2,270 phishing and 443 other types of events are indexed. Although all of the randomly selected Tweets contain the keyword "phishing", some Tweets discuss other categories of security events instead of "phishing". As the Tweet example in Figure 10 shows, the event type of the Tweet is actually "data breach" and labeled as "data breach" in our index, though the keyword "phishing" exists in the Tweet. Another reason is that sometimes Twitter users prefer to add multiple hashtags across multiple specific domains, which may not be very relevant to the current topic discussed. Compared with the event annotations in the groundtruth, the mean squared deviations are 0.06 and 0.34 for the average length of the event nuggets and the top five event arguments, respectively. The deviation difference between the estimated length and the actual length of the labeled entities further demonstrates the performance of

TABLE 7: Existing cybersecurity search engines

		Google custom search	Shodan	Our work
Open source				✓
Extendable by user defined			✓	✓
Query function	Search	✓	✓	✓
	Navigation		✓	✓
	Browsing		✓	✓
	Report generation		✓	✓
Indexing	Web pages	✓		✓
	Semantic text			✓
	Metadata		✓	✓
	Links and contents	✓	✓	✓
Results presentation	Structured data		✓	✓
	Map and bar chart		✓	✓
	Other visualization			✓
Scalability		Security related web pages	Connected devices data	Crowdsourced heterogeneous data
Total No. of results for queries (on keyword "phishing"/ "vulnerabiliry"/"data breach")		59,100/343,000 /49,500	428/1,297 /220	22,772/38,345 /14,425

our information extraction approach, besides the traditional evaluation methods listed in Table 4.

RQ3: Are there any traditional criteria useful for measuring the functionality of a search engine? We implement the search engine prototype as demonstrated in Section 4.1 to present the cyber information retrieval through pragmatic understanding and visualization. To evaluate the functionality of the search engine, we follow the criteria raised by Search Technologies [53]. As shown in Table 7, we compare our work with Google customized search engine and cybersecurity domain search engine Shodan as introduced in Section 1. By following the instructions from Google Custom Search, we build a custom cybersecurity search engine [54]. It covers the top 10 cybersecurity intelligence sources investigated and measured by [11]. Besides, several public cybersecurity websites (such as CVE [39]) are included in the customized cybersecurity search engine. Follow the criteria that measure the functionality of a search engine, we outline the features of our search engine prototype in the aspect of publicly accessible, query functions, indexing contents, search results representation and scalability, as listed in Table 7. In line with RQ4 and RQ5, the features mentioned above are discussed as follows.

RQ4: What functions should be provided to help users understand cybersecurity-related data? Our search engine is open source and written in the R programming language. It provides a higher degree of elasticity for customized features. The system is extendable to incorporate other data that privately belongs to users, and can easily integrate new data analytics and visualization techniques. In the long term, especially for organizations and enterprises, users can better enhance their own security by means of human argumentation that is constantly motivating employees to equip and empower with security knowledge, instead of seeking for external assistance.

To provide straight forward information to users without tedious reading, summarizing and manual filtering, we index a message using semantics extracted from both the text and metadata of security information. This idea is inspired by the web pages indexing approach by Google and metadata indexing by Shodan, plus our proposed semantic index. In regard to the scalability that refers to the ability of a search engine to scale up data, our search engine sources from websites, archives, databases, social media and various sources that have cybersecurity-related information. At the same time, the contents are processed and represented in the form of structured data, ready for further analysis and visualization. Based on the demand of users, search functions are fulfilled by querying keywords, selecting from index or browsing semantics.

RQ5: Would the functions support a large variety of queries, given different users may have different requirements? Looking into the representation of results, Google customized search engine delivers results in a format of title and corresponding link combinations. Shodan as the first search engine for connected devices manifests search results by taking advantage of structured information on the connected devices and several straight forward visualization techniques, such as bar charts and devices location representation through a world map. Our prototype search engine not only provides the title and link combination but also makes use of multiple visualization techniques. It shows messages from information retrieval on pragmatics understanding and related metadata based on the user's query.

In the meantime, we conduct a proof-of-concept demonstration by querying on keywords to measure the search results from a quantitative perspective as displayed in Figure 7. Taking the term "phishing" as an example, our prototype returns $22,772$ results, including $12,466$ records from the metadata index and $10,306$ instances from the text index. Besides, one of the critical factors for information retrieval is the retrieval speed quantified by how long it takes to return results. In this respect, we count the retrieval time using various keywords (e.g., "phishing", "vulnerability" and "data breach") on platforms (e.g., safari and Google

Chrome browsers on mobile phones and desktops). The average retrieval time using our prototype built on the free Shiny platform [37] is less than 5 seconds. This number is for information only as the retrieval speed may be affected by multiple factors, such as the configuration of the hosting server, service load and network bandwidth, etc. Although it uses less than 1 second to obtain the results using a custom Google Search Engine, the retrieval speed of our proposed search engine holds the line with Shodan, which can be improved by transplanting our search engine to a powerful hosting server. With the proposed search engine, users expecting to see a broader range of results and references for a conclusive understanding can retrieve results from their customized databases. If constrained by time, users can directly render the result summary presented in various forms of statistics, visualizations and pragmatics understanding.

5 Discussion

Besides of the above demonstration and evaluation, there are also a few points that need further elaboration.

5.1 Named Entity Recognition in cybersecurity

In this work, semantic security information is extracted from the text and tagged with corresponding event nuggets and arguments. Meta information as data used to describe the text is indexed together with the semantic information. Compared with the existing indexing approaches [4], [6], our indexing method has obvious advantages.

First of all, the information is stored in messages instead of words. We leverage a pragmatic text index that directly extracts and saves the most concerned entities in cybersecurity events. This further facilitates the search engine to deliver effective results to users. Secondly, meta information is indexed with pragmatics. If a user prefers further exploring the returned messages, they are provided with the original data source links saved in the metadata index. Thirdly, metadata and text pragmatics are structured in a predefined format, making it easy to conduct information retrieval.

Compared to the state-of-the-art work, such as Stanford NER system [30], our entity recognizer can identify entities that are directly related to a cybersecurity attack. As one potential future work, additional event nuggets (such as insider threat) can be integrated to further improve the performance of entity recognizers by means of enlarging the groundtruth data and embedding supplementary features to concatenate with word embedding.

5.2 Result Presentation of Information Retrieval

By putting users at the center, our search engine is implemented as an easy-to-use application. It generates and communicates results to users according to their queries and demands for cybersecurity information. Intending to deliver direct and effective information, without being limited by domain knowledge and understanding capability, the results generated by multiple data analysis approaches offer a straightforward message with security domain knowledge to users. The visualization of distinctive features of the security datasets is also a promising way to convey novel insights to users.

The application is extendable for users and developers: new analysis and visualization components can be easily integrated into the search engine to accommodate the requirement of sustainable development for large scale and high throughput cybersecurity data. One of our future work is to integrate security components to measure security severity, such as threat priority and vulnerability impact on optimizing the retrieved results.

6 Conclusion

This paper describes the cyber information search engine we have implemented. It begins with introducing the proposed indexing approach with the pragmatics and meta information of the collected cybersecurity resources, followed by the description of the design and development of the web-based search engine, including information modeling, presentation and visualization. The R programming language brings descriptive, impressive and professional statistics and visualization to the implementation. Four information retrieval modules are implemented and demonstrated, which are respectively security trend, cybersecurity event nugget and argument, security event visualization, and security search and visualization. They work collaboratively to support the interactive exploration and various cybersecurity data mining.

The knowledge perceived by users is not determined by the volume or length of returned messages but depends on how much they understand. Users expect to obtain straight forward and effective results polished with security domain knowledge per query rather than long paragraphs that may contain helpful information but need longer time to render and filter. We have demonstrated that the proposed search engine can be used to search, model and visualize security datasets under a broad security context by answering the defined research questions. Through our proposed search engine, plentiful presentations are provided based on users' queries and demands. We expect that the output generated by our search engine be helpful to advance users insights into a specific cybersecurity question.

References

[1] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, and Y. Xiang, "Data-driven cybersecurity incident prediction: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1744–1772, 2019.

[2] A. C. E. R. Team, "Auscert is a leading cyber emergency response team (cert) in australia and the asia/pacific region," accessed on 03/04/2018. [Online]. Available: http://www.auscert.org.au/

[3] L. Liu, O. De Vel, Q.-L. Han, J. Zhang, and Y. Xiang, "Detecting and preventing cyber insider threats: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1397–1417, 2018.

[4] "How google search works," https://www.google.com/search/howsearchworks/, 2019, accessed on 16/3/2020.

[5] "Google custom search," https://www.seobility.net/en/wiki/Google_Custom_Search, 2019, accessed on 5/5/2021.

[6] "Shodan," https://www.shodan.io/, 2019, accessed on 2/4/2020.

[7] R. McMillan, "Open threat intelligence," https://www.gartner.com/en/documents/2487216, 2013, accessed on 31/3/2020.

[8] K. Soska and N. Christin, "Automatically detecting vulnerable websites before they turn malicious," in *23rd USENIX Security Symposium*, 2014, pp. 625–640.

[9] K. Borgolte, C. Kruegel, and G. Vigna, "Delta: automatic identification of unknown web-based infection campaigns," in *Proceedings of the ACM SIGSAC conference on Computer & Communications Security*, 2013, pp. 109–120.

[10] H. Yang, X. Ma, K. Du, Z. Li, H. Duan, X. Su, G. Liu, Z. Geng, and J. Wu, "How to learn klingon without a dictionary: Detection and measurement of black keywords used by the underground economy," in *IEEE Symposium on Security and Privacy*, 2017, pp. 751–769.

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

[12] J. Zhao, Q. Yan, J. Li, M. Shao, Z. He, and B. Li, "Timiner: Automatically extracting and analyzing categorized cyber threat intelligence from social data," *Computers & Security*, vol. 95, p. 101867, 2020.

[13] D. Kong, L. Cen, and H. Jin, "Autoreb: Automatically understanding the review-to-behavior fidelity in android applications," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 530–541.

[14] R. P. Khandpur, T. Ji, S. Jan, G. Wang, C.-T. Lu, and N. Ramakrishnan, "Crowdsourcing cybersecurity: Cyber attack detection using social media," in *Proceedings of the ACM on Conference on Information and Knowledge Management*, 2017, pp. 1049–1057.

[15] S. Qamar, Z. Anwar, M. A. Rahman, E. Al-Shaer, and B.-T. Chu, "Data-driven an2.1alytics for cyber-threat intelligence and information sharing," *Computers & Security*, vol. 67, pp. 35–58, 2017.

[16] C. Sabottke, O. Suciu, and T. Dumitraș, "Vulnerability disclosure in the age of social media: exploiting twitter for predicting real-world exploits," in *USENIX Security Symposium*, 2015, pp. 1041–1056.

[17] N. Sun, J. Zhang, S. Gao, L. Y. Zhang, S. Camtepe, and Y. Xiang, "Data analytics of crowdsourced resources for cybersecurity intelligence," in *International Conference on Network and System Security*. Springer, 2020, pp. 3–21.

[18] H. Ji and R. Grishman, "Refining event extraction through cross-document inference," in *Proceedings of Association for Computational Linguistics*, 2008, pp. 254–262.

[19] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of Association for Computational Linguistics*, 2005, pp. 363–370.

[20] X. Schmitt, S. Kubler, J. Robert, M. Papadakis, and Y. LeTraon, "A replicable comparison study of ner software: Stanfordnlp, nltk, opennlp, spacy, gate," in *International Conference on Social Networks Analysis, Management and Security*, 2019, pp. 338–343.

[21] Z. Syed, A. Padia, T. Finin, L. Mathews, and A. Joshi, "Uco: A unified cybersecurity ontology," in *AAAI Conffference on Artificial Intelligence*, 2016.

[22] R. A. Bridges, C. L. Jones, M. D. Iannacone, K. M. Testa, and J. R. Goodall, "Automatic labeling for entity extraction in cyber security," *arXiv preprint arXiv:1308.4941*, 2013.

[23] T. Satyapanich, F. Ferraro, and T. Finin, "Casie: Extracting cybersecurity event information from text," in *AAAI Conffference on Artificial Intelligence*, 2020.

[24] S. K. Lim, A. O. Muis, W. Lu, and C. H. Ong, "Malwaretextdb: A database for annotated malware articles," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 1557–1567.

[25] E. Cambria and B. White, "Jumping nlp curves: A review of natural language processing research," *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, pp. 48–57, 2014.

[26] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge university press, 2008.

[27] A. Kirk, *Data visualisation: a handbook for data driven design*. Sage, 2016.

[28] N. Reimers and I. Gurevych, "Optimal hyperparameters for deep lstm-networks for sequence labeling tasks," *arXiv preprint arXiv:1707.06799*, 2017.

[29] T. Satyapanich, T. Finin, and F. Ferraro, "Extracting rich semantic information about cybersecurity events," in *IEEE International Conference on Big Data*, 2019, pp. 5034–5042.

[30] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of Association for Computational Linguistics*, 2014, pp. 55–60.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[32] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of North American Chapter of the Association for Computational Linguistics*, 2018.

[35] W. S. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *Journal of the American Statistical Association*, vol. 74, no. 368, pp. 829–836, 1979.

[36] O. L. I. W. A. Thousand, "Words," *Piqua Leader-Dispatch*, p. 2, 1913.

[37] W. Chang, J. Cheng, J. Allaire, Y. Xie, J. McPherson *et al.*, "Shiny: web application framework for r," *R package version*, vol. 1, no. 5, 2017.

[38] N. Sun, "Cyber information retrieval through pragmatics understanding and visualization repository," https://github.com/nansunsun/Cyber-Information-Retrieval-Through-Pragmatics-Understanding-and-Visualization, 2020.

[39] "Common vulnerabilities and exposures," http://cve.mitre.org/, 2018, accessed on 11/3/2020.

[40] A. Taspinar and L. Schuirmann, "Twitterscraper 0.2. 7: Python package index," 2017.

[41] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data," in *European Conference on Information Retrieval*. Springer, 2016, pp. 45–57.

[42] N. Sun, "Search engine module 1 prototype," https://nansun.shinyapps.io/SearchEngine1/, 2020.

[43] ——, "Search engine module 2 prototype," https://nansun.shinyapps.io/SearchEngine2/, 2020.

[44] ——, "Search engine module 3 prototype," https://nansun.shinyapps.io/SearchEngine3/, 2020.

[45] ——, "Search engine module 4 prototype," https://nansun.shinyapps.io/SearchEngine4/, 2020.

[46] N. Sun, J. Zhang, S. Gao, L. Y. Zhang, S. Camtepe, and Y. Xiang, "My security: An interactive search engine for cybersecurity," in *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2021, p. 6206.

[47] N. Ariffini, A. Zainal, M. A. Maarof, and M. N. Kassim, "Ransomware entities classification with supervised learning for informal text," in *International Conference on Cybersecurity*, 2019, pp. 86–90.

[48] X. Qiu, X. Lin, and L. Qiu, "Feature representation models for cyber attack event extraction," in *IEEE International Conference on Web Intelligence Workshops*, 2016, pp. 29–32.

[49] S. Engelbrecht, "The evolution of soar platforms," https://www.securityweek.com/evolution-soar-platforms, 2018, accessed on 20/3/2020.

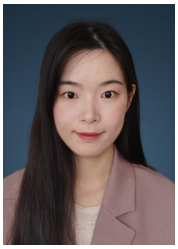
[50] C. Islam, M. A. Babar, and S. Nepal, "A multi-vocal review of security orchestration," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–45, 2019.

[51] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems*, 2019, pp. 5754–5764.

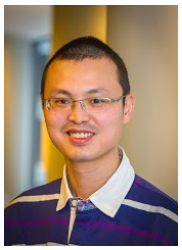
[52] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[53] "Top 10 criteria for evaluating a search engine," https://www.searchtechnologies.com/blog/how-to-evaluate-a-search-engine, 2016, accessed on 16/3/2020.

[54] N. Sun, "Customized google search engine," https://cse.google.com/cse?cx=012104931910478407356:vdp4jcckw2x, 2020.



Nan Sun received the B.S. degree (Hons.) and the Ph.D. degree in Information Technology from Deakin University. She is currently a Lecturer in the School of Engineering and Information Technology at the University of New South Wales (UNSW), Canberra, Australia. Before joining UNSW,she was a postdoctoral research fellow at Deakin University. Her current research interests include cybersecurity and social network security.



Jun Zhang (M'12-SM'18) received the Ph.D. degree in Computer Science from the University of Wollongong, NSW, Australia, in 2011. He is currently a full Professor and the Director of the cybersecurity lab, Swinburne University of Technology, Australia. He was recognized in The Australian's top researchers special edition publication as the leading researcher in the field of Computer Security & Cryptography in 2020. He leaded Swinburne cybersecurity research and produced excellent outcome including many high impact research papers and multi-million-dollar research projects. Swinburne was named in The Australian's 2021 Research magazine, the top research institution in the field of Computer Security & Cryptography. He has served as a steering committee member of the P-TECH program at Melbourne since 2019, which the Australian Government invested in, promoting STEM education. He devotes himself to communication and community engagement, boosting the awareness of cybersecurity.

Shang Gao received her Ph.D. degree in computer science from Northeastern University, Shenyang, China in 2000. She is currently a senior Lecturer in the School of IT, Deakin University, Geelong, Australia. Before joining Deakin, she was a postdoctoral research fellow at UTS and associate lecturer at CQU, Australia. Her research interests include networking, big data processing, cyber security and cloud computing.

Leo Yu Zhang (M'17) is currently a Lecturer with the School of Information Technology, Deakin University, VIC, Australia. He received the bachelor's and master's degrees in computational mathematics from Xiangtan University, Xiangtan, China, in 2009 and 2012, respectively, and the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2016. Prior to joining Deakin, he held various research positions with the City University of Hong Kong, the University of Macau, Macau, China, the University of Ferrara, Ferrara, Italy, and the University of Bologna, Bologna, Italy. His current research interests include applied cryptography and AI-related security, and he has published more than 60 refereed journal and conference articles in these fields.

Seyit Camtepe (Senior Member, IEEE) is a principal research scientist and team leader at CSIRO Data61. He received his PhD from Rensselaer Polytechnic Institute in 2007, he was with the Technische Universitaet Berlin as a senior researcher, and with QUT as a lecturer. He was among the first to investigate the security of Android smartphones and inform society of the rising malware threat. His research interests include ML and cyber security, malware detection and prevention, smartphone security, applied and malicious cryptography, CII security.

Yang Xiang received his PhD in Computer Science from Deakin University, Australia. He is currently a full professor and the Dean of Digital Research, Swinburne University of Technology, Australia. His research interests include cyber security, which covers network and system security, data analytics, distributed systems, and networking. He is also leading the Blockchain initiatives at Swinburne. In the past 20 years, he has published more than 300 research papers in many international journals and conferences. He is the Editor-in-Chief of the SpringerBriefs on Cyber Security Systems and Networks. He serves as the Associate Editor of IEEE Transactions on Dependable and Secure Computing, IEEE Internet of Things Journal, and ACM Computing Surveys. He served as the Associate Editor of IEEE Transactions on Computers and IEEE Transactions on Parallel and Distributed Systems. He is the Coordinator, Asia for IEEE Computer Society Technical Committee on Distributed Processing (TCDP). He is a Fellow of the IEEE.