A cyber security data triage operation retrieval system

Article *in* Computers & Security · March 2018

DOI: 10.1016/j.cose.2018.02.011

CITATIONS
120

READS
377

5 authors, including:

Chen Zhong
The University of Tampa
16 PUBLICATIONS **297** CITATIONS
SEE PROFILE

Tao Lin
Amazon
11 PUBLICATIONS **386** CITATIONS
SEE PROFILE

Peng Liu
Pennsylvania State University
177 PUBLICATIONS **3,625** CITATIONS
SEE PROFILE

A Cyber Security Data Triage Operation Retrieval System

Chen Zhong[1] Tao Lin [2] Peng Liu[2] John Yen[2] Kai Chen[34]
[1] School of Science, Indiana University Kokomo
chzhong@iuk.edu
[2] College of Information Sciences and Technology, Pennsylvania State University
{txl78,jyen,pliu}@ist.psu.edu
[3] SKLOIS, Institute of Information Engineering, Chinese Academy of Science
[4] School of Cyber Security, University of Chinese Academy of Science
chenkai@iie.ac.cn

***Abstract*—Data triage is a fundamental stage of cyber defense analysis for achieving cyber situational awareness in a Security Operations Center (SOC). It has a high requirement for cyber security analysts' capabilities of information processing and expertise in cyber defense. However, the present situation is that most novice analysts who are responsible for performing data triage tasks suffer a great deal from the complexity and intensity of their tasks. To fill the gap, we propose to provide novice analysts with on-the-job suggestions by presenting the relevant data triage operations conducted by senior analysts in a previous task. In a previous study, a tracing method has been developed to track an analyst's data triage operations. This paper mainly presents a data triage operation retrieval system that (1) models the context of a data triage analytic process, (2) uses a centroid-similarity matching method to compare contexts, and (3) presents the matched traces to the novice analysts as suggestions. We have implemented and evaluated the performance of the system through both automated testing and human evaluation. The results show that the proposed retrieval system can effectively identify the relevant traces based on an analyst's current analytic process.**

***Index Terms*—Cyber Situational Awareness, Data Triage, Graph Centroid, Retrieval System.**

I. Introduction

Colossal, complex, unpredictable and undetermined vicious threats occur at anytime and anywhere in cyberspace. A lot of automated defense measures have been developed to protect an organization's critical information infrastructure, such as the intrusion detection systems (IDS), firewalls, anti-virus softwares, vulnerability scanning tools, and security information and event management (SIEM). Currently, most organizations have built Security Operations Centers (SOCs) to provide centralized control of the network monitoring and defending issues. However, the SOC operations are heavily human-in-the-loop processes: given the data collected by the cyber defense measures (e.g, the IDS alerts, firewall logs and SIEM reports), cyber defense analysts' main responsibility is to make sense of the data to achieve cyber situational awareness so that the incident response team can take timely and effective actions against the threats [8], [15], [26]. More specifically, three questions need to be investigated to gain cyber situational awareness: (1) Is the network currently under an attack? (2) How did the attack happen? (3) What will the attackers do in the following step?

To answer the above questions, cyber defense analysts need to conduct a series of analysis to gradually generate and refine their comprehension by exploring the collected massive data sources [7]. Data triage is the most fundamental stage in the analytic process, which involves the tasks of ruling out the noise in the raw data, identifying and grouping the data indicating the suspicious events worth of further investigation. Table I lists some typical data triage operations: FILTER operation that filters the data sources by specifying a condition; SELECT operation that identifies a set of data of interest; and SEARCH operation that searches for data with a specific characteristic. Based on these operations, analysts can filter and group the raw data sources to get a subset of network events for further analysis. Figure 1 demonstrates that the data triage is a process where the analysts explore the raw data to identify the suspicious network events.

The dynamic nature of the cyber environment results in the influx of the raw massive data sources with high noise-to-signal ratio. Therefore, data triage, as the first analysis stage, has to be conducted by analysts within minutes. On the other hand, nowadays attackers tend to coordinate multiple steps to attain their ultimate malicious attack goal over a long time span so that their attack activities are more deceptive and concealing and difficult to detect. Therefore, data triage not only involves tedious data analysis, but also needs analysts' intensive analytical reasonings based on their previous experiences and expertise [8], [26], [34].

For this reason, analysts' expertise is critical in data triage. Senior analysts with more experience can have far better performance than novice analysts [6], [24]. However, the status quo is that data triage tasks in SOCs are undertaken by those analysts at entry level whose expertise are expected to acquired and enhanced through on-the-job training with the guidance of senior analysts [2], [12]. As the results, their training periods last too long with the normal span of one or two years [23], [22]. Therefore, a big gap exists between the high demands on the senior analysts with expertise in data triage and the

TABLE I: The data triage operations recorded by ARSCA

Data Triage Operations	Description
<Item Timestamp= 07/31/2014 13:01:41> FILTER(SELECT * FROM IDS Alerts WHERE SrcPort = 6667 </Item>	**FILTER** a set of network events (i.e., IDS Alerts) based on a condition (SrcPort=6667)
<Item Timestamp= 07/31/2014 13:20:29> SELECT (FIREWALL-[4/5/2012 10:15 PM] -[Built]-[TCP] -(172.23.233.57:3484, 10.32.5.58:6667), FIREWALL-[4/5/2012 10:15 PM] -[Teardown]-[TCP]- (172.23.233.52:5694, 10.32.5.59:6667), FIREWALL-[4/5/2012 10:15 PM] -[Built]-[TCP] (172.23.233.57:3484, 10.32.5.58:6667), LINK (Same DstPort) </Item>	**SELECT** a set of network events (i.e., the underlined firewall log entries) with a common characteristics (DstPort=6667)
<Item Timestamp= 08/09/2014 11:08:01> SEARCH(Firewall Log, 172.23.233.52) </Item>	**SEARCH** in the Firewall log based on the condition (SrcIP = 172.23.233.52 OR DstIP = 172.23.233.52).

abundant junior analsyts short of expertise in SOCs.

Therefore, it is necessary and beneficial to utilize the experience of senior analysts to assist novice analysts. To achieve this goal, we propose a data triage support system that provides novice analysts with suggestions based on the historical analytic processes of senior analysts: we first track the senior analysts' analytic processes in accomplishing data triage tasks; with the collection of traces, the system retrieves the relevant traces and suggest them to a novice analyst based on the novice analyst's current analytic process. The following cases are described to further explain this idea.

Use Case 1. Joe, a junior data triage analyst, was presented with a heavy volume of network events reported by the SIEM tools when he got on his shift. He decided to first rule out the normal routine communication events between the workstations and the mail server which were known as accepted activities. The filtering operation led to a small decrease of the data set. However, Joe had no clue how to further narrow down the data set to locate the suspicious network events. It would be very helpful if he could learn what filtering operations have been done by the senior analysts in such cases.

Use Case 2. After performing a series of filtering operations, Joe observed some unusual network events could hardly be interpreted by either the network usage policy or the known network activities. Although Joe realized the observation could be important, he simply could not interpret it owning to his lack of the knowledge in suspicious event implicated by the selected data. In such case, it is desirable for Joe to know the historical operations performed by other senior analysts when they had made a similar observation because the historical operations inform the senior analysts' focus of attention.

Use Case 3. Joe generated a hypothesis about a potential attack activity after drawing out some observations of suspicious network events. He intended to make a further investigation, but he was unable to know what specific evidence was needed to confirm or deny his hypothesis and where he could find the evidence from the data sources. In this case, it could be beneficial to show Joe the data triage operations conducted by the senior analysts after they generated the similar hypotheses.

the potential benefits of suggestions provided based on the relevant traces of senior analysts in.

The above cases demonstrated three typical problems encountered by junior analysts and the potential benefits of suggestions provided based on the relevant traces of senior analysts in (1) how to effectively explore the data to identify suspicious network events, (2) how to interpret the current observation and what hypotheses/conclusions can be drawn, and (3) how to further confirm or deny a hypothesis. In the current SOCs, junior analysts obtain such assistance by directly consulting with the senior analysts or even under the supervision of senior analysts, which means senior analysts have to spend a lot of time to work together with junior analysts to offer timely assistance. Compared with the transitional approach, the proposed system is able to offer a more cost-efficient step-by-step assistance to junior analysts.

The data triage support system is built on two parts: a tracing system that captures an analyst's data triage operations while he/she is performing a data triage task, and a retrieval system that retrieves relevant traces from its trace collection based on an analyst's current analytic process. A tracing system has been developed and evaluated in a previous study [32]. This work mainly focuses on the retrieval system, and we use the term "system" to refer to the retrieval system in the remaining part of this paper.

To sum up, the contribution of this paper is three-fold.

- To our best knowledge, this is the first work that supports data triage analysts by leveraging the senior analysts' experience.
- The proposed trace retrieval method takes the context of the current analyst's data triage analytic process into account. The context is modeled in both the dimensions of network events and the analyst's data triage operations.
- The performance of the retrieval system has been evaluated through both automated testing with the ground truth and human evaluation. The testing samples are constructed from the traces collected from a previous experiment which involves professional analysts.

The remaining parts of the paper are organized as follows. The research problem and data triage model are formulated in Section II. With the problem defined, we propose our data triage trace retrieval approach in Section III, and the evaluation of the retrieval system is described in Section IV. Related studies are discussed in Section VI with the conclusion of the study at the end.

II. Problem Formulation

A. Framework of the Data Triage Operation Retrieval

Figure 2 demonstrates the framework of the proposed data triage operation retrieval system. The system maintains a

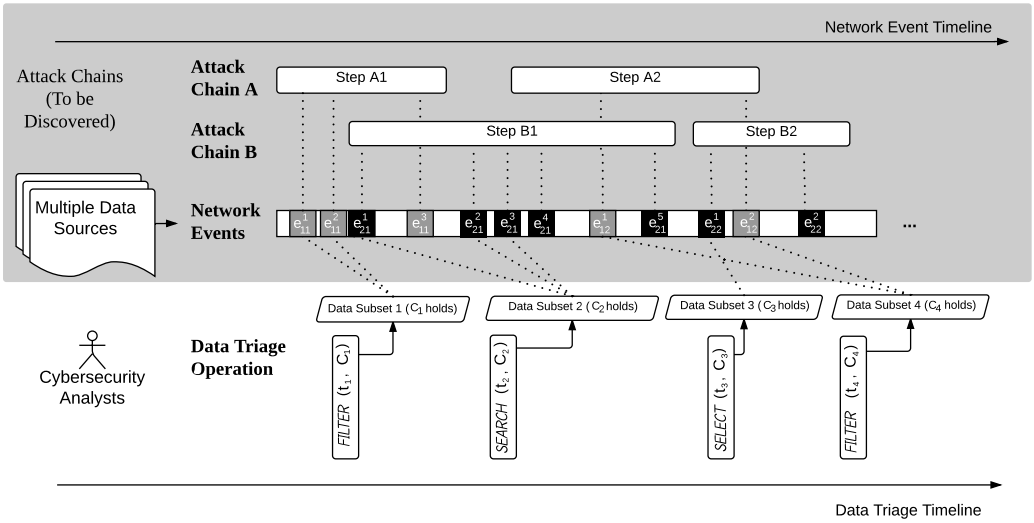


Fig. 1: A series of data triage operations conducted to detect suspicious network events in the raw data. Each data triage operation filters or correlates network events based on a characteristics constraint defined by the analyst.

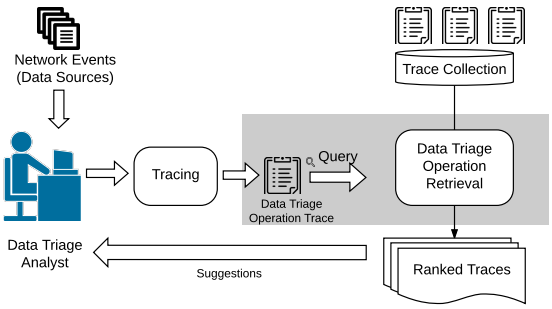


Fig. 2: The framework of data triage operation retrieval

collection of traces which were collected when experts were performing data triage operations. A junior analyst is the current user of the system. The system monitors the analyst's data triage operations to gain awareness of his/her current analysis context. Taking the current context as a query, the system retrieves the relevant traces with the similar analysis contexts from the trace collection. Provided with the retrieved results, the current user can learn how the experts performed data triage under a similar context. Next, we define the data triage operation and analysis context in details.

B. Data Triage Model

Cyber security data triage is targeted at determining whether the incoming data sources are worth of further investigation in a timely and efficient manner. To achieve this goal, security analysts usually conduct a sequence of data triage operations to filter malicious network events and then to group them according to the potential attack chains. Therefore, the unit of data triage analysis is a network event. Network events are the data reported by various network monitoring sensors, including SIEM tools and human intelligence agents, A **network event** can be abstracted as a multi-tuple of its characteristics,

$$e =< t_{occur}, t_{detect}, type, attack_{prior}, sensor, protocol,$$
$$ip_{src}, port_{src}, ip_{dst}, port_{dst}, severity, confidence, msg >,$$

where t_{occur} is the time the event occurred; t_{detect} is the time the event first being detected; $type$ is the type of network connection activity (e.g., Built, Teardown or Deny); $attack_{prior}$ is the attack type of the event being detected by a sensor/agent based on prior knowledge; $sensor$ is the sensor/agent who detected this event; $protocol$ is the network protocol; ip_{src}, $port_{src}$, ip_{dst}, $port_{dst}$ are respectively the source IP, source port, destination IP, and destination port; $severity$ and $confidence$ specify the level of severity and confidence of the event, respectively; msg specifies other characteristics of the event, which depends on the sensor.

Figure 1 illustrates an example of a data triage process where an analyst performs a sequence of data triage operations to identify suspicious network events. Each data triage operation specifies a constraint for the events to narrows down the original data set. As the examples shown in Table I, there are mainly three types of data triage operations.

- FILTER(D, $\mathbb{C}$): to filter a set of events (D) based on a constraint ($\mathbb{C}$).
- SEARCH(D, $\mathbb{C}$): to search a keyword ($\mathbb{C}$) in an event set (D).
- SELECT(D, $\mathbb{C}$): to select a subset of events with a common feature ($\mathbb{C}$) from a set (D).

All these operations result in a subset of events satisfying one constraint. The constraint is defined as follows.

C. Data Triage Operation and Characteristic Constraint

An atomic constraint predicates the value of an event characteristic/attribute,

$$T_i = R_v(char, val), \quad (1)$$

where $R_v = \{=, <>, >, <, <=, >=\}$.

Considering the fact that an event possesses multiple attributes, the constraint can be multidimensional and represented by a predicate in disjunctive normal form, named "**Characteristic Constraint** (CC)",

$$\mathbb{C} = \begin{cases} \bigvee(\wedge T_i)_{i \in \mathbb{N}} \\ T \end{cases} \tag{2}$$

Given the definition of characteristic constraint, a **data triage operation** is defined as follows.

$$O_i =< D, t, \mathbb{C} >, \tag{3}$$

where D is a set of network events; t is the time of being performed; $\mathbb{C}$ is the characteristic constraint specified for data filtering. Recall the examples in Table I, the characteristic constraints of the three operations are "SrcPort = 6667", "DstPort = 6667", and "SrcIP =172.23.233.52 OR DstIP =172.23.233.52" respectively.

D. Trace and Context

A **trace** consists of a sequence of data triage operations performed by an analyst in accomplishing a data triage task. It can be represented by $\mathcal{T} = (O_n)_{n \in \mathbb{N}}$, where $O_i (1 \leq i \leq n)$ is a data triage operation. According to this definition, the boundary of a trace (i.e., operation sequence) is determined by the workload of a data triage task. In a real-world SOC, the tasks of an analyst are usually assigned by a "watch officer" who is responsible for ensuring appropriate procedures taken by all the analysts in their shifts.

Given a data triage operation O_i in a trace $\mathcal{T}$, the **context** of O_i, denoted by $C(O_i)$, is defined by the sequence of the data triage operations that precede O and their relationships. The relationships between data triage operations consist of temporal and logic relationships, which are defined as follows.

Let $O_1 = (D_1, t_1, \mathbb{C}_1)$ and $O_2 = (D_2, t_2, \mathbb{C}_2)$ be two different data triage operations. The temporal relationship between them is determined by t_1 and t_2. We have "happen-before" or "happen-after" relationships, denoted by "$\prec_t$", "$\succ_t$" respectively.

$$\prec_t (O_1, O_2) \Leftrightarrow t_1 < t_2, \quad \succ_t (O_1, O_2) \Leftrightarrow t_1 > t_2 \tag{4}$$

The logic relationship between O_1 and O_2 is determined by their characteristic constraints. Three types of logical relationships, "is-equal-to", "is-subsumed-by", and "is-complementary-with", can be defined as follows.

$$isEql(O_1, O_2) \iff \mathbb{C}_1 \leftrightarrow \mathbb{C}_2, \tag{5}$$

$$isSub(O_1, O_2) \iff \mathbb{C}_1 \rightarrow \mathbb{C}_2, \tag{6}$$

$$isCom(O_1, O_2) \iff \mathbb{C}_1 \rightarrow \neg\mathbb{C}_2 \; and \; \mathbb{C}_2 \rightarrow \neg\mathbb{C}_1, \tag{7}$$

where $\rightarrow$ means "implies". $isEql(\cdot,\cdot)$ and $isCom(\cdot,\cdot)$ are bidirectional relationships but $isSub(\cdot,\cdot)$ is a unidirectional relationship. The "isSub" and "isCom" relationships are demonstrated in Figure 3 (a). In Figure 3 (b), taking the following nodes as an example: $\mathbb{C}_1$ is "DstPort = 6667", $\mathbb{C}_2$ is "DstPort $\neq$ 6667", and $\mathbb{C}_3$ is "DstPort $\neq$ 6667 AND DstPort $\neq$ 80", so we have $isCom(O_1, O_2)$, $isSub(O_3, O_1)$ and $isCom(O_3, O_1)$.

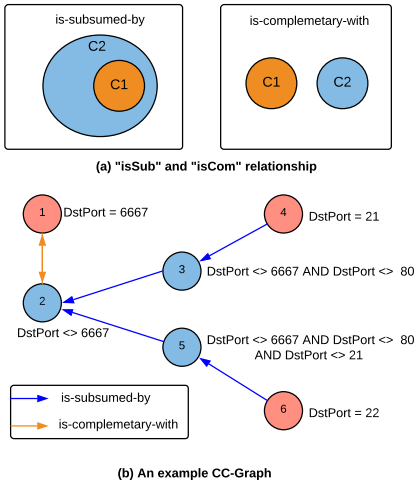


Fig. 3: An example of CC-Graph (defined in Section III) containing the "isSub" and "isCom" relationships. The nodes are data triage operations and the edges are the relationships. (Not all the relationships are demonstrated in the CC-Graph in order to simplify the display.)

Therefore, the context of the data triage operation O_i can be defined as,

$$C(O_i) =< (O_j), \{R_T(O_j)\}, \{R_L(O_j)\} >, \; j < i \tag{8}$$

where $(O_j)_{j<i}$ is the set of data triage operations conducted earlier than O_i; R_T and R_L refer to the temporal and logic relationships among $(O_j)_{j<i}$ respectively.

An analyst make decisions on what data triage operations to perform mainly on the context. The current context of an analyst's data triage process refers to the context of the latest data triage operation, which changes dynamically as long as the analyst performs new operation. Therefore, it is critical to take the context into consideration in order to retrieve the relevant traces.

III. The Trace Retrieval Approach

A. Insights: Context-Driven and Efficient

Our goal is to retrieve the relevant traces based on the current context of an analyst' data triage process and to quickly update the results along with the context change. There have been some approaches to tracking and utilizing human process knowledge. The research on the cognitive model includes Endsley's theory of Situational Awareness (SA) [9], which defines SA as a dynamic process involving perception, comprehension and projection, and the Recognition Primed Decision (RPD) model, which explains how experts match the situation to the previous situations to make decisions [17]. The ability to manage the process knowledge is a key to business process management [16].

Draw on the conceptual models, multiple knowledge representations have been proposed. Chen et al. used Horn logic rules to capture and represent cyber security experts'

experience by defining the event patterns and alert patterns [5]. Association rule is also commonly used to extract knowledge from behavior data [14]. The main limitations of rule-based representation are the lack of flexibility of the pattern matching and the difficulty of abstracting data into effective rules. Chen et al. proposed a rule relaxation method to improve the coverage of the rule patterns [5]. However, there is a tradeoff between the matching accuracy and the level of relaxation. Structure-based knowledge representation has been proposed to better represent the contextual information. Zhong, et al. proposed a tree structure to represent the key observations and their relationships in an analyst's analytical reasoning processes [30]. However, this work is limited by ignoring the operation details. We observed that an analyst's operations and the temporal and logical relationships between the operations contain rich information indicating the analyst's strategies and process knowledge used to accomplish a data triage task. Therefore, our approach is developed on two main insights.

- Insight 1: Graph structure can be used to represent the context information because it can perfectly capture the course of data triage operations and the dependencies between these operations.
- Insight 2: The retrieval results need to be updated dynamically along with the changes of the current context. It requires the graph-based approach to be efficient enough for timely updates. Therefore, we need to avoid graph isomorphism analysis.

Inspired by these two insights, we adopted the centroid idea used by Chen et al.[4]. The concept of centroid originally comes from physics, which represents an object without its structural details. Based on this concept, we construct centroid from a graph-structured context and compare the corresponding centroids of different contexts for trace retrieval.

B. Context Representation: CC Graph

Considering the temporal and logical relationships between data triage operations, the context of a data triage operation can be represented by a directed graph with all the previous data triage operations as nodes and their relationships as the edges. It is defined as "Characteristic Constraint Graph (CC-Graph)",

$$G =< V, \{R_l\} >,\ V = \{O_1, \ldots, O_n\},$$

$$R_l \subseteq V \times V, l \in \{isEql \succ_t, isSub \succ_t, isSub \prec_t, isCom \succ_t\}.$$

The vertexes of the graph are the data triage operations, and a directed edge between two vertexes represents a conjunction of a constraint-related logical relationship and the temporal relationship. Let $< n_i, n_j >$ be an edge in G,

- $isEql \succ_t (n_i, n_j)$ represents that n_i has a "is-equal-to" and a "happen-after" relationship with n_j.
- $isSub \succ_t (n_i, n_j)$ represents that n_i has a "is-subsumed-by" and a "happen-after" relationship with n_j.
- $isSub \prec_t (n_i, n_j)$ represents that n_i has a "is-subsumed-by" and a "happen-before" relationship with n_j.
- $isCom \prec_t (n_i, n_j)$ represents that n_i has a "is-complementary-with" and a "happen-after" relationship with n_j.

An example of CC-Graph is demonstrated in Figure 3. The edges of CC-Graph represent both the temporal and logical relationships among data triage operations. The reason why the temporal relationships matter is that we are mainly interested in how a data triage operation is related to the ones that precede it. Along with the temporal relationships, the logical relationships among data triage operations imply how an analyst switches his/her attention of focus from one subset of network events to another subset while performing these data triage operations. Therefore, an analyst's data triage strategy may be implied by the temporal and logical relationships.

C. Edge-Induced CC Subgraph

Considering the four types of logical relationships, "is-subsumed-by", "is-complementary-with", "is-equal-to" and "subsume", we can extract four edge-induced subgraphs from a CC-Graph. Let $G =< V(G), E(G) >$ be a CC-Graph. Four induced subgraphs are defined as follows.

- "isEql" Subgraph: $G_E =< V(G_E), E(G_E) >$, where $V(G_E) \subseteq V(G)$, $E(G_E) \subseteq E(G)$, $\forall < n_i, n_j > \in E(G_E)$, $isEql \succ_t (n_i, n_j)$.
- "isSub" Subgraph: $G_{IS} =< V(G_{IS}), E(G_{IS}) >$, where $V(G_{IS}) \subseteq V(G)$, $E(G_{IS}) \subseteq E(G)$, $\forall < n_i, n_j > \in E(G_{IS})$, $isSub \succ_t (n_i, n_j)$.
- "Sub" Subgraph: $G_S =< V(G_S), E(G_S) >$, where $V(G_S) \subseteq V(G)$, $E(G_S) \subseteq E(G)$, $\forall < n_i, n_j > \in E(G_S)$, $isSub \prec_t (n_i, n_j)$.
- "isCom" Subgraph: $G_C =< V(G_C), E(G_C) >$, where $V(G_C) \subseteq V(G)$, $E(G_C) \subseteq E(G)$, $\forall < n_i, n_j > \in E(G_C)$, $isCom \succ_t (n_i, n_j)$.

D. Centroid of Subgraph

For the purpose of CC-Graph comparison, we propose a multi-dimensional measure of the four induced subgraphs. As we mentioned above, both the characteristic constraints and the temporal and logical relationships between data triage operations imply an analyst's data triage strategies. Therefore, the proposed measure not only considers the characteristic constraints but also investigates the geometry structure of the graphs. First of all, we project the subgraphs of a CC-Graph to multiple dimensions according to the characteristics of network events. Each node in the CC-Graph has a unique coordinate, which is a vector $< vi_1, ..., vi_m >$, where m is the number of network event characteristics in the data sources, and vi_i refers to the value intervals of the network event characteristic $char_i$. This vector is defined as **MD-Vector**. The MD-Vector indicates the focus of attention of an analyst while conducting the corresponding data triage operation.

Drawn on the centroid concept in physics, we define the centroid of an induced subgraph as a MD-Vector that represents the value intervals of multiple network event characteristics that most interested the analyst. Next, we are ready to define the centroids of "isSub", "Sub" "isCom", and "isEql" subgraph respectively.

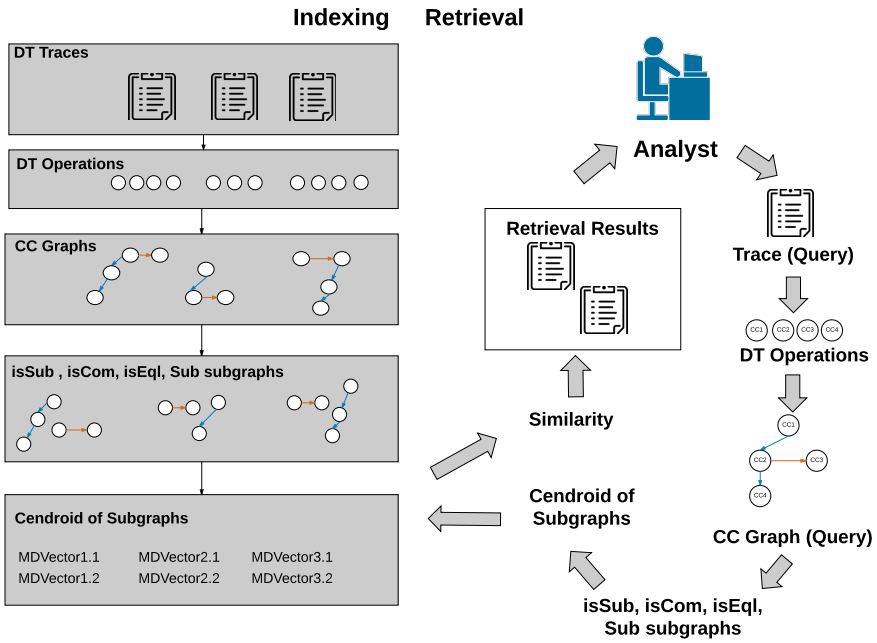


Fig. 4: Our approach for data triage operation retrieval

1) Centroid of "isSub" Subgraph: Let $G_{IS} =< V(G_{IS}), E(G_{IS}) >$ be a "isSub" graph of a CC-Graph G. An edge $e =< v_n, v_p >, e \in E(G_{IS})$ indicates that v_n further narrows down the search space compared with v_p. Therefore, v_n needs to be given more value than v_p when the centroid of G_{IS} being calculated. To achieve this goal, we assign "isSub" weights to the nodes in G_{IS} as follows:

- Step 1: We start with the root nodes in G_{IS} whose in degree is 0, and set the root nodes' weights to 1.
- Step 2: Starting from each root node, we conduct breadth-first search to traverse a graph. During the traversal, we will make sure (1) each node has a same weight as its neighbors, (2) the weight of a node's child nodes is the weight of the node plus 1.

The centroid of G_{IS} is a MD-Vector $\vec{c}(G_{IS}) = < c_1, ..., c_m >$, where $c_k (1 \le k \le m)$ is a value interval that

$$c_k = \bigcup_{e(v_p, v_n) \in E(G_{IS})} f_S(v_p, v_n, k)$$

$$f_S(v_p, v_n, k) = \begin{cases} vi_k^n, & \xi(n) >= \theta; \\ vi_k^p, & \xi(p) >= \theta, \xi(n) < \theta. \\ \emptyset, & \xi(p) < \theta, \xi(n) < \theta. \end{cases}$$

$$\xi(p) = \frac{w_p}{\max\limits_{v_i \in V(G_{IS})} w_i}$$

where $e(v_p, v_n)$ is an edge between two nodes v_p and v_n, which correspond to the MD-Vectors $< vi_1^p, ..., vi_m^p >$ and $< vi_1^n, ..., vi_m^n >$. m is the number of network event characteristics. $vi_k^p (1 \le k \le m)$ refers to the value interval of the kth characteristic specified in node v_p. w_p and w_n are the weights of v_p and v_n respectively and $w_p > w_n$. θ is a threshold in the range of [0,1]. The time complexity of computing the "isSub" subgraph centroid is $O(|V(G_{IS})| + |E(G_{IS})|)$.

The centroid of "Sub" subgraph G_S are calculated in the similar way. But the semantic of an edge $e =< v_n, v_p > isdifferent.e \in E(G_S)$ indicates that v_p further relax the constraint to extend the search space compared with the previous operations v_n.

2) Centroid of "isCom" Graph: Let $G_C =< V(G_C), E(G_C) >$ be a "isCom" graph of a CC-Graph G. An edge $e =< v_n, v_p >, e \in E(G_C)$ indicates a switch of focus of attention from the search space of v_p to v_n. Analysts' observations of suspicious network events result from the bounded value intervals. Therefore, we define the centroid of a "isCom" graph based on the bounded intervals.

A centroid of G_C is a MD-Vector $\vec{c}(G_C) = < c_1, ..., c_m >$, where $c_k (1 \le k \le m)$ is a value interval that

$$c_k = \bigcup_{e(v_p, v_n) \in E(G_C)} f_C(v_p, v_n, k)$$

$$f_C(v_p, v_n, k) = \begin{cases} vi_k^p \cup vi_k^n, & vi_k^p, vi_k^n \text{ are bounded;} \\ vi_k^n, & \text{only } vi_k^n \text{ is bounded.} \\ \emptyset, & \text{otherwise.} \end{cases}$$

where the nodes v_p and v_n correspond to the MD-Vectors $< vi_1^p, ..., vi_m^p >$ and $< vi_1^n, ..., vi_m^n >$. We focus on the bounded value intervals when merging two intervals (vi_k^p, vi_k^n) of a network event characteristic k. Besides, the temporal relationship between two nodes is also considered when there is an unbounded interval: (1) if the unbounded interval occurred before the bounded interval (i.e., only vi_k^n is bounded), we will take the later interval (vi_k^n) because the later bounded interval indicates the analyst's new focus; (2) otherwise, if the unbounded interval occurred after the bounded interval, we will consider neither because it indicates the analyst lost his/her focus. The time complexity of computing the centroid of G_C is $O(|E(G_C)|)$.

3) Centroid of "isEql" Graph: The centroid of "isEql" subgraph can be represented by any node of this subgraph. Let $G_E =< V(G_E), E(G_E) >$ be a "isEql" subgraph. The centroid is a MD-Vector $\vec{c}(G_E) = < c_1, ..., c_m >$, where $c_k (1 \leq k \leq m)$ is a value interval that $c_k = vi_k^p, v_p \in V(G_E)$. The time complexity of computing the centroid of G_E is $O(1)$.

E. Centroid Similarity

Given the subgraph centroids, the similarity between two graphs can be determined by comparing the centroids of their subgraphs. Algorithm 1 describes the overall centroid-based retrieval algorithm. The centroid similarity is calculated as follows.

Algorithm 1 Data Triage Operations Retrieval

Input: Trace Collection $\{Tr\}$, the current trace Tr, a threshold θ
Output: A collection of trace pieces $\{Tr\}_R$ that matches Tr
1: The traces in $\{Tr\}$ are indexed by their centroids
2: Parse Tr and construct its CC Graph G
3: l = trace length
4: Get the four types subgraphs of G: $G_{sub} = \{G_{IS}, G_S, G_E, G_C\}$.
5: **for** $each\ Tr'$ (length(Tr')=l) in $\{Tr\}$ **do**
6: **for** $each\ G_i$ in G_{sub} **do**
7: Calculate the centroid of G_i
8: **end for**
9: Calculate the CSD similarity between Tr and Tr'
10: **if** CSD similarity $\geq \theta$ **then**
11: Add Tr' to $\{Tr\}_R$
12: **end if**
13: **end for**
14: Return $\{Tr\}_R$

Let $\vec{c}$ and $\vec{c'}$ be two centroids. The Centroid Similarity Degree (CSD) between them are defined as

$$CSD(\vec{c}, \vec{c'}) =< s_1, ..., s_m >, \quad s_k = \frac{||c_k \cap c'_k||}{||c_k \cup c'_k||}$$

Let G_1, G_2 be two different CC-Graphs. Their "isSub", "isCom" and "isEql" subgraphs are G_{S_1}, G_{C_1}, G_{E_1}, G_{S_2}, G_{C_2}, G_{E_2} respectively. Two types of CC-Graph similarity can be calculated as follows.

1) Max CSD Similarity:

$$Sim(G_1, G_2) = max(\ CSD(\vec{c}(G_{S_1}), \vec{c}(G_{S_2})),$$
$$CSD(\vec{c}(G_{C_1}), \vec{c}(G_{C_2}),\ \ CSD(\vec{c}(G_{E_1}), \vec{c}(G_{E_2})))$$

2) Weighted CSD Similarity:

$$Sim(G_1, G_2) = \frac{1}{w_S + w_C + w_E}[w_S * CSD(\vec{c}(G_{S_1}),\ \vec{c}(G_{S_2}))$$
$$+w_C * CSD(\vec{c}(G_{C_1}),\ \vec{c}(G_{C_2})) + w_E * CSD(\vec{c}(G_{E_1}),\ \vec{c}(G_{E_2}))]$$

IV. Implementation and Evaluation

We prepared the trace collection by using the traces collected from a previous experiment. In that experiment, we recruited thirty professional analysts and asked them to complete a cyber data triage task [32]. Each analyst's data triage operations were tracked automatically by a tool named ARSCA [31]. A partial trace is shown in Table I. It has been shown that these collected traces can represent the analysts' analytical reasoning process [32].

We built a Java prototype to retrieve relevant traces from the collection traces. JGraphT, a Java graph library, was used to implement the CCGraph. The subgraph centroid computation and matching algorithms were implemented based on the CCGraph implementation. Given the input of a current context (i.e., a sequence of trace operations), the system outputs the matched traces in the trace collection.

Our method has been evaluated by answering the following questions. (E1) Can the retrieval system retrieve the matching traces for cyber security analysts? (E2) How the retrieval system scale given large-scale historical context data sets of triage operations? (E3) What factors may influence the performance of the retrieval system? (E4) Do cyber security analysts find the retrieval output useful?

A. Testing Cases ("Trace Slices") from Subsampling

Although the subjects are professional analysts, their task performance turned out to be quite different. In order to eliminate the interference of the quality of the traces on the performance of the retrieval system, the traces were evaluated carefully with the consideration of task performance, trace quality and the diversity of traces: (1) all the selected traces come from the analysts who had successfully revealed the attack events in the task; (2) each analyst had conducted a series of data triage operations during their exploration that are sufficient for understanding the analysis strategies of the analyst; and (3) these traces embody various analysis strategies used by the analysts. At last, we selected five traces in this way. The average number of data triage operations in the selected traces is 23.2.

Considering the abundance of a single selected trace, we use subsampling to generate a large set of representative trace slices. To maintain the important temporal relationships between the data triage operations, we set a sliding window and extract a sequence of successive data triage operations as a slice, and then move the window to a next l operation (l is called the hop length). Let m be the window length, l be the hop length, given a trace with n data triage operations ($n > m$), $\lceil (n - m + 1)/l \rceil$ slices can be generated. To generate trace slices, we set the window length 8 and the hop length 3 in consideration of three aspects: (1) a trace slice should contain sufficient number of DT operations and most slides involving multiple DT relationships have a length no less than 8; (2) however, a trace slice can't be too complicated for the slice analysts, otherwise, a controversial understanding of the real analytic process may occur; and (3) most frequently, 3 successive DT operations can be highly related and similar, and therefore we use 3 hops to increase the distance between the slices.

Altogether we extracted 29 trace slices. 330 pairs were further generated by combining each two slices. An exclusion is that the slices extracted from a same trace cannot be paired as it will be meaningless to consider the similarity of the slices extracted from the same trace. 330 pairs in total are considered as the testing cases. Although the volume of the testing cases is not large, the testing set overall contains sufficient information because each testing case was an informative and self-contained piece implying a data triage process.

TABLE II: The retrieval performance in terms of different threshold values

Threshold	Precision	Recall	ACC	F_1 Score
0.0	0.5	1	0.5	0.667
0.1	0.523	1	0.544	0.687
0.2	0.527	0.991	0.552	0.688
0.3	0.562	0.914	0.6	0.696
0.4	0.588	0.807	0.621	0.68
0.5	0.679	0.706	0.685	0.692
0.6	0.744	0.483	0.658	0.585
0.7	0.812	0.253	0.597	0.386
0.8	0.906	0.145	0.565	0.25
0.9	0.905	0.04	0.517	0.076

B. Ground Truth

The ground truth refers to the fact that whether two slices match. To determine the ground truth, we have two experts manually label the testing cases, named Coder A and Coder B. The experts were asked to make decisions based on their interpretations of the data triage processes. To guarantee the accuracy of their decisions, we made sure that the two experts have sufficient expertise in cyber security analysis and were familiar with the trace representation and the data triage task ground truth. Besides, two rounds of manual analysis have been conducted to ensure the quality of their decisions. During each round of analysis, Coder A and B analyzed and labeled the pairs independently. At the end of the first round of analysis, Coder A labeled 129 pairs as positive (i.e., matched) and 202 pair as negative (i.e., not matched). Coder B labeled 163 positive pairs and 168 negative pairs. They agreed on 205 pairs among the 330 pairs. During the second round of analysis, Coder A and B went through the inconsistent pairs independently without any information of their previous labels or the other's labels. At the end of the second round, Analyst A and B agreed on 304 pairs in total, including 126 positive ones and 178 negative ones. The inconsistent pairs were discarded.

C. Random Selection

To ensure the randomness, we shuffled the positive and negative pairs, and randomly select 100 pairs respectively from each set (positive pairs and negative pairs). In this way, we can construct a balanced testing set with 100 positive pairs and 100 negative pairs. In the following testing, we repeated the random selection for 10 times and evaluated the average performance of our method.

D. Performance Measurement

To answer the first evaluation question (E1), we evaluate the performance of the retrieval system. We counted the True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), and used them to determine the following measurements: (1) Precision, the percentage of the pairs predicated as positive that actually are matched; (2) Recall, the percentage of the matched pairs that are predicated as positive; (3) Accuracy (ACC), the percentage of correct predictions; and (4) F-1 Score (F_1), the harmonic mean of precision and recall. They are calculated as follows.

$$Precision = \frac{TP}{TP + FP}$$

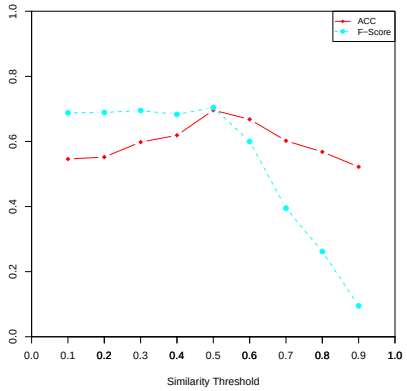


Fig. 5: The Accuracy (ACC) and F1-Score with respect to the value of the threshold

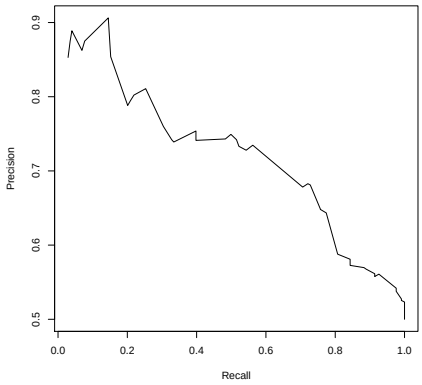


Fig. 6: The Precision-recall curve

$$Recall = \frac{TP}{TP + FN}$$

$$ACC = \frac{TP + FN}{TP + TN + FP + FN}$$

$$F_1 = 2\frac{Precision * Recall}{Precision + Recall}$$

The threshold value of similarity was increased from 0.1 to 0.9, and 10 rounds of sampling and testing were conducted. The average results are shown in Table II. With the increase of the threshold, precision raises from 0.5 to 0.905 while recall dropping from 1 to 0.04 with the increase of the threshold. ACC first increases, and then drops when the threshold exceeds 0.5. F_1 remains stable until the threshold reaches 0.5. The F_1 and ACC values changing with the threshold are demonstrated in Figure 5. Both ACC and F_1 deteriorate when the threshold value becomes larger than 0.5. The peak could be explained by the use of a balanced dataset. With the threshold value 0.5, we can achieve the best ACC and a stable F_1.

To further investigate the trade-off between the precision and recall, we show the RP-Curve in Figure 6. The Recall, as shown on the x-axis, is identical with sensitivity of our algorithm, and the Precision, as shown on the y-axis, is identical with the positive predictive rate. We notice that

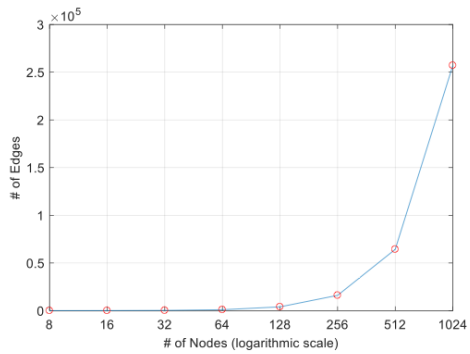


Fig. 7: The average edge number of the CC-Graphs generated from the trace slices with different number of operations.

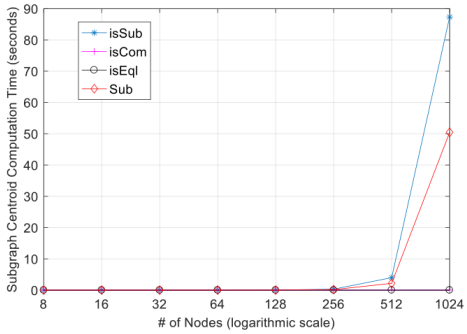


Fig. 8: The average time of computing the subgraph centroids with respect to the number of nodes.

precision remains relatively stable when recall is in the range roughly between 0.4 to 0.7. Combined the results in Table II, an appropriate balance between precision and recall can be drawn at threshold value 0.5.

E. Scalability

A scalability test has been conducted to answer the second evaluation question (E2). To evaluate the time complexity of the centroid computation, we first simulated several sets of trace slices with different orders of magnitude (in terms of the number of nodes and edges of the CC-Graphs). These trace slices were simulated as follows: given the 29 trace slices we used in the above tests, we generated 7 additional sets of slices by duplicating the operations in each slice from twice to 128 times. With the original set of trace slices, we had 8 sets of slices, each of which contains 29 slices with the node number of 8, 16, 32, 64, 128, 256, 512 and 1024 respectively. We ended with the node number of 1024 because that number is reasonably large enough given that the average number of operations the analysts performed in our one-hour experiment is 23.2. Figure 7 shows the average number of the the CC-Graphs generated from each set of slices.

We ran the subgraph centroid computation using a MAC computer (OS X 10.11) with Intel Core i5 (3.2 GHZ) and 8 GB memory. Figure 8 demonstrates the average time used for computing the subgraphs (i.e., "isSub", "isCom", "isEql", "Sub") with respect to the number of nodes (i.e, operations). We observed that the average computation time for all the subgraphs are less than 1 second when the node number is less than 256. The computation of the "isSub" and "Sub" subgraph centroids are more time-consuming. When the node number reaches to 512, the average computation time of "isSub" subgraph and "Sub" subgraph increase to 3.9942s and 2.1794s respectively, and then further increase to 87.3329s and 50.4592s when the node number is 1024. The result shows that the centroid computing time will not be a concern when the context size is controlled below 512. It is reasonably practical considering that an analyst works for eight hours in one shift and multiple tasks are usually assigned to one analyst within one shift.

TABLE III: The five cases of the weights of "isSub" and "isCom" subgraphs

Case	Description
S1	weight(isSub) = 4 weight(isCom)
S2	weight(isSub) = 2 weight(isCom)
S3	weight(isSub) = weight(isCom)
S4	weight(isSub) = 0.5 weight(isCom)
S5	weight(isSub) = 0.25 weight(isCom)

TABLE IV: The five cases of the weights of network event characteristics (Port and IP)

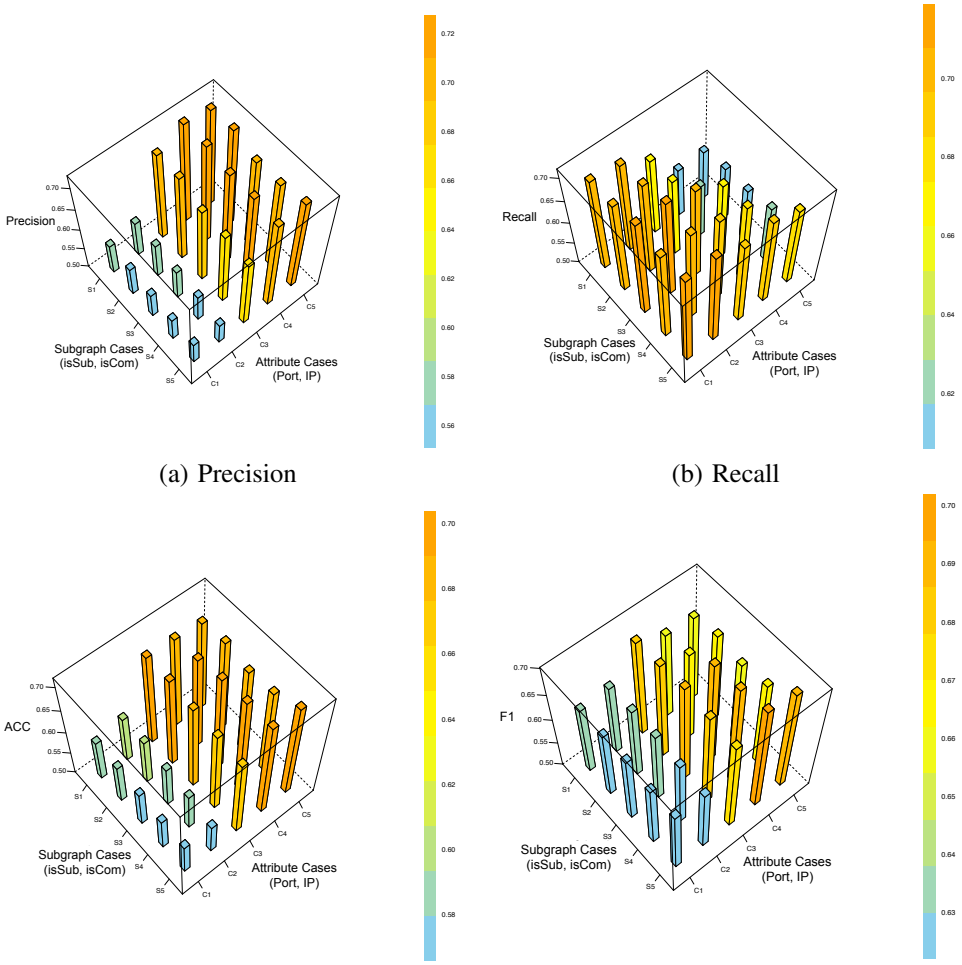
Case	Description
C1	weight(port) = 4 weight(ip)
C2	weight(port) = 2 weight(ip)
C3	weight(port) = weight(ip)
C4	2 weight(port) = weight(ip)
C5	4 weight(port) = weight(ip)

F. Impact Parameter

1) Subgraph Types: To answer the third evaluation question (E3), we first evaluate how the weights of different types of subgraphs influence the overall performance. The same measurements of performance are used (i.e., precision, recall, accuracy and f-measure). We mainly focus on the comparison between the "isSub" and "isCom" subgraphs in consideration of their representativeness of the data triage strategies. Firstly, "isEql" subgraphs can be viewed as a special case of "isSub" subgraph. Besides, "Sub" subgraphs, unlike other subgraphs, imply no strategy of narrowing down the search space. It is because each later data triage operation in the "Sub" subgraph relaxes the characteristic constraint of its previous operations. Therefore, five cases have been listed in Table III to investigate the impact of different subgraph weights on the performance.

2) Network Event Characteristics: Another factor is the weight of network event characteristics. Considering that IP addresses (ip_{src} and ip_{dst}) and port numbers ($port_{src}$ and $port_{dst}$) are the most common characteristics used by the analysts to define data triage constraints, we focus on these two characteristics and assign different weights to the IP address and port number (shown in Table IV).

To investigate the impacts of different weights, we evaluate the 5×5 models considering the cases listed in Table III and Table IV. Based on the previous discussion, the threshold value 0.5 is used to achieve relative good performance. We evaluate the average precision, recall, accuracy and f-measure for these



(a) Precision (b) Recall

(c) Accuracy (d) F_1 value

Fig. 9: The results of comparing the performance of the models with different weights for the "isSub" and "isCom" subgraphs and different weights for the IP address and port number.

25 cases by conducting the sampling and testing for 10 rounds. The results are shown in Figure 9.

According to Figure 9 (a) and (b), we found the cases where "isSub" subgraphs with larger weights and "isCom" subgraphs with less weights (i.e., the "1S4C", "1S2C") result in higher precision rates and correspondingly smaller recall rates, given the same weights of the network event characteristics. It may indicate that the operations conducted by the analysts narrow down the search space, which are captured in the "isSub" subgraphs, are more representative for matching traces than other operations. In terms of the different weights of network event characteristics, we found the cases where the IP addresses with larger weights than the port numbers have relatively higher precision rates and smaller recall relates, given the same weights of the subgraphs. It can be explained by the fact that attending to routing, addressing and typology are critical tasks during data triage analysis [11].

Considering both of the weights of subgraphs and network event characteristics, we first selected six cases with good performance according to the accuracy and F_1 measures. Both the accuracy and F_1 values of these selected cases are no less than the third quartiles of the corresponding values of the 25 cases (i.e., accuracy> 0.691, F_1 > 0.683). The 6 selected cases are S1-C3, S2-C3, S3-C4, S4-C4, S5-C4, and S5-C5. Figure 10 shows the performance of the selected cases. We noticed that assigning higher weights to the "isSub" subgraph than "isCom" subgraph resulted in slightly better results, but the weight difference of the subgraphs didn't impact the overall performance as much as the one of network event characteristics did. Besides, assigning higher weights to IP addresses than port numbers improved the performance in our case study. It indicates that, given the same network setting, the observation that two analysts exploring the data involving

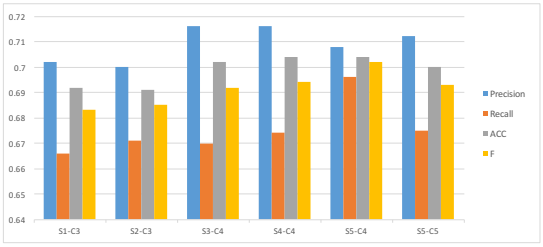


Fig. 10: The performance of the selected 6 cases

TABLE V: The selected subgraph features

Feature	Description
issub_node, iscom_node, iseql_node	The # of nodes of the isSub, isCom, or isEql subgraph
issub_edge, iscom_edge, iseql_edge	The # of edges of the isSub, isCom, or isEql subgraph
issub_degree, iscom_degree, iseql_degree	Maximum total degree
issub_density, iscom_density, iseql_density	Kernel density estimates
issub_island, iscom_island, iseql_island	# of maximal (weakly or strongly) connected components
issub_trans, iseql_trans, iscom_trans	clustering coefficient that measures the probability that the adjacent vertices of a vertex are connected
issub_diameter, iscom_diameter, iseql_diameter	the length of the longest geodesic
issub_003, issub_012, issub_021D, issub_021U issub_021C, issub_030T iscom_003, iscom_012, iscom_021D, iscom_021U iscom_021C, iscom_030T iseql_003, iseql_012, iseql_021D, iseql_021U iseql_021C, iseql_030T	The # of motifs in the subgraph (shown in Figure 11)

the same IP addresses is a stronger indicator that shows they have the similar data triage processes.

G. Relationship between CC Subgraph Features and Retrieval Performance

Our retrieval method is designed based on the similarity of the centroids of the corresponding subgraphs. Therefore, we further investigate how the retrieval performance is related to the features of the subgraphs. The subgraph features selected are described in Table V. Given a query slice and a retrieved slice, we calculated their differences in the selected features and investigate how these differences are related to the matching results (i.e., TP, TN, FP and FN).

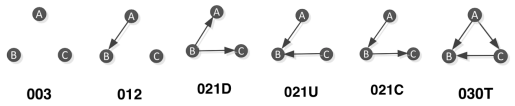
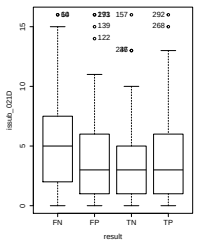
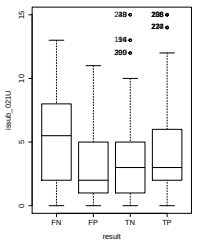
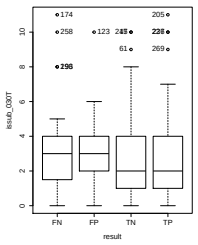
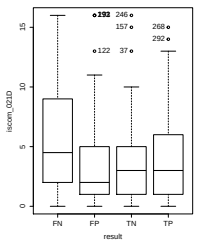
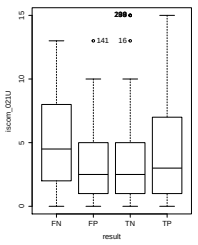
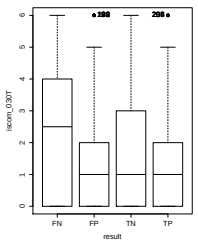


Fig. 11: The motifs of subgraphs

Given the testing pairs of trace slices (n=303), we investigate the distribution of the motif frequency difference of two slices in terms of the four matching results. According to the boxplots in Table VI, the frequency differences of the 021D and 021U motifs in the isSub and isCom subgraphs and the 030T motifs in the isCom subgraphs have similar distributions: the difference of the motif frequency tends to be larger in the false negative cases than the others. An ANOVA (type 3) is conducted to test the difference between the motif difference means in the four results. The ANOVA results (shown in Table VI) indicate that the difference of all the selected motifs other than issub_030T is statistical significant (α = 0.1). It implies that a large frequency difference of these motifs can hurt the recall rate.

TABLE VI: The difference of motif frequency in subgraphs in terms of the 4 matching results.

issub_021D	issub_021U	issub_030T
		
F=2.236, P=0.084	F=4.017, P=0.008	F=0.601, P=0.615
iscom_021D	**iscom_021U**	**iscom_030T**
		
F=2.257, P=0.081	F=2.701, P=0.046	F=2.254, P=0.082

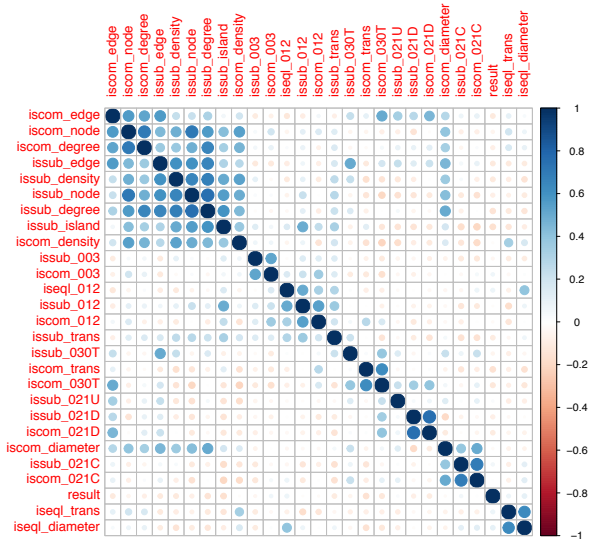


Fig. 12: The correlation matrix of the reduced features and the matching results.

We conducted feature importance ranking based on boosted tree and random forest by using two feature selection packages in R: xgboost [1] and ranger [2]. Before the investigation of the graph features, we remove the highly correlated feature with the consideration of the feature importance: if two features have a strong correlation (i.e., Spearman correlation $\geq$ 0.75), one feature with lower importance ranking will be dropped. Figure 12 visualizes the correlation matrix of the reduced features. It indicates that the slices' differences in the graphical

[1]https://cran.r-project.org/web/packages/xgboost/xgboost.pdf
[2]https://cran.r-project.org/web/packages/ranger/ranger.pdf

features don't have strong correlation of their matching results. It can be explained as follows. The selected features focus on the scalability and structure of a subgraph. The centroid of the subgraph which used for matching is calculated by merging the "important" nodes. Although the structure of the subgraph implies the importance of the nodes, it has nothing to do with the merging process so that it has a rather limited effect on the centroid calculation. Therefore, our method is robust in terms of the subgraph structure change.

H. Human Evaluation

Comparing the outputs with the ground truth, we have shown that the centroid-similarity method can be tuned to get a relatively high precision or recall rates. To further evaluate the usefulness of the retrieval results (E4), we asked two human judges to manually rate the outputs during the process of data triage.

1) Evaluation Protocol: The ideal experiment is to have the human subjects perform a real data triage task with the assistant of the retrieval system and rate the retrieval outputs in each trial. However, it requires the subjects' domain knowledge and expertise. Besides, this step-by-step evaluation takes a lot time per trial (about 30-60 minutes are needed to complete a data triage task). However, professional analysts with the expertise cannot afford such a long time. To make the human evaluation feasible, we made two adjustments. Due to the limited access to professional analysts, we invited two graduate students who has the expertise of cyber security analysis and had participated in our previous experiment as the judges. To reduce the evaluation time in each trial, we prepared a data triage context for the judges in each case so they could situate themselves in a scenario and make a decision without performing the task from the beginning. The details are described in the following.

For each trial, we randomly selected one slice (i.e., a sequence of data triage operations) from the trace collection to be the current context and presented it to a judge. The judge first read through the slice to interpret the current context. The retrieval system took the current context as a query and suggested the retrieved slices. The judge needed to manually go through each retrieved slice to make a decision on the usefulness of the suggestion. A 5-point Likert scale was used: 1 (strongly negative), 2 (negative), 3 (neutral), 4 (positive), 5 (strongly positive).

2) Result: In total, we collected 934 responses from the 2 judges. The average time used by the judges for making a decision per trial are respectively 23.36 seconds (Judge 1) and 53.16 seconds (Judge 2). Besides, the two judges were consistent with their responses: we compared the judge's response in the repeated cases; we found all the responses of both judges were consistent in terms of positive and negative ratings, and the average of the rating difference is 0.329 and 0.307 respectively. The statistical description of the responses is shown in Table VII. It shows that the majority responses are positive (the overall mean is 3.97 and median is 4).

3) Case Study: Table VIII demonstrates two cases: one rated positive (i.e., Case 1) and the other rated negative (i.e., Case 2) by the judges. In Case 1, the query slice indicates that the analyst first identified a suspicious host and paid attention to the network connections targeting this host via port 6667. After that, he switched his attention to other suspicious network connections via port 21. After conducting the filtering based on port 21, he ruled out the network connections via port 6667 and 80, indicating he was searching for other suspicious activities. Comparing with the query slice, the retrieved slice on the right also indicated the analysts had conducted data

TABLE VII: Human judges' response description

Judge	# of Trials	Mean	Sd	Median
1	527	3.92	0.85	4
2	407	4.03	0.82	4
Total	934	3.97	0.84	4

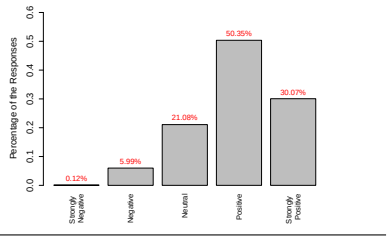


TABLE VIII: The cases of human evaluation

Case 1: Positive Rating (4-5)	
Query Slice	Retrieved Slice
1. *FILTER* (SRCIP = 172.23.235.57) 2. *SELECT* (DSTIP = 172.23.235.57 AND SRCIP >= 10.32.0.0 AND SRCIP < 10.33.0.0 AND SRCPORT = 6667) 3. *FILTER* (Priority <> Info) 4. *FILTER* (Priority <> Info AND DSTPORT = 21) 5. *SELECT* (Priority <> Info AND DSTIP >= 10.32.5.50 AND DSTIP < 10.33.0.0 AND DSTPORT = 21) 6. *SELECT* (Priority <> Info AND SRCIP = 172.23.235.57 AND DSTIP >= 10.32.5.50 AND DSTIP < 10.33.0.0 AND DSTPORT = 21) 7. *FILTER* (DSTPORT <> 80) 8. *FILTER* (DSTPORT <> 80 AND SRCPORT <> 6667)	1. *FILTER* (SRCPORT = 6667) 2. *FILTER* (SRCPORT <> 6667) 3. *FILTER* (SRCPORT <> 6667 AND DSTPORT <> 6667) AND SRCPORT <> 80 AND DSTPORT <> 80) 4. *FILTER* (DSTPORT = 21) 5. *FILTER* (DSTPORT = 21 AND DSTIP >= 10.32.5.50 AND DSTIP < 10.33.0.0) 6. *FILTER* (DSTPORT <> 6667 AND DSTPORT <> 80 AND DSTPORT <> 21) 7. *FILTER* (DSTPORT = 22) 8. *SELECT* (DSTPORT = 22 AND DSTIP >= 10.32.5.50 AND DSTIP < 10.33.0.0)
Case 2: Negative Rating (1-2)	
Query Slice	Retrieved Slice
1. *FILTER* (DSTPORT <> 80) 2. *FILTER* (DSTPORT <> 80 AND SRCPORT <> 6667) 3. *FILTER* (Priority <> Info DSTIP >= 10.32.5.50 AND DSTIP < 10.33.0.0 AND DSTPORT = 21) 4. *FILTER* (DSTIP = 172.23.235.57) 5. *FILTER* (DSTIP = 172.23.235.57 AND SRCPORT = 6667) 6. *FILTER* (DSTPORT = 6667) 7. *FILTER* (DSTIP = 10.32.5.56 AND DSTPORT = 6667) 8. *FILTER* (DSTIP = 10.32.5.51 AND DSTPORT = 6667)	1. *FILTER* (DSTPORT = 445) 2. *SELECT* (DSTIP = 172.23.0.10 AND SRCIP >= 172.23.0.0 AND SRCIP < 172.24.0.0) 3. *FILTER* (SRCPORT = 6667) 4. *FILTER* (DSTPORT = 6667) 5. *FILTER* (SRCPORT = 445) 6. *FILTER* (DSTPORT = 445) 7. *FILTER* (SRCPORT = 6667) 8. *FILTER* (DSTPORT = 6667)

filtering based on the port 21 and 6667 and detected the same set of network connections. Similarly, the analyst later filtered out these network connections and then detected a set of new network connections via port 22 targeting to the IPs in the range of 10.32.5.*. Based on our understanding of these two slices, it is worth considering the retrieved slice for an analyst who has been conducting the operations in the query slice but he has no idea what to do next.

In terms of the negative Case 2, the query slice indicates the analyst first detected the network connections targeting to the IPs in the range of 10/32.5.* via port 21. After that, he identified one suspicious host 172.23.233.57 and further switched his attention back to the connection via port 6667. On the other hand, the retrieved slice suggested that another analyst also filtered the network connections via port 6667. However, the analyst switched his attention to the connections via port 445 back and forth instead of exploring more suspicious activities via port 6667. Therefore, presenting the retrieved slice can hardly help the current analyst.

The results of the case study provide some insights into the improvement of our centroid-based retrieval method. We found in both cases the retrieved slices are relevant to the query slice. The difference between the positive case and the negative case lies in whether the retrieved slice matched the analyst's current focus of attention. Therefore, the analyst's focus of attention is a critical factor for consideration when matching traces. In fact, this factor has been implicitly incorporated in the centroid calculation: we placed more weight on the operations that are more recent when calculating the centroid of the "isSub" subgraph. Such weight allocation can be tuned by a system user to achieve a performance that satisfies the user's needs.

V. Discussion and Limitation

Running the retrieval system on the testing cases, we found the system can complete the retrieval process within a minute, which indicates that the centroid-based approach is time efficient. Adjusting the threshold value of similarity, a satisfactory precision and recall rate can be achieved and the outcome can be affected by the weights placed on the different types of subgraphs and different network event characteristics, which suggests that heuristics can be leveraged to allocate the weights appropriately. The results also demonstrated the benefits of centroid: the centroid-based method is robust in terms of the graph structural change; meanwhile, it takes into account an analyst's focus of attention by putting more weight on the recent operations. Above all, the retrieval objective of context awareness and efficiency has been achieved.

However, there are some limitations in this study. First of all, as the centroid calculation algorithm assigns weights to data triage operations which implicitly incorporate an analyst's focus of attention, it may be inconvenient for the system user to manually tune the weights to achieve the best outcome. We found it worthwhile to realize automatic weight allocation by conducting additional graph structural analysis. The structural analysis can be completed offline for the trace collection in order to guarantee the run-time efficiency of the retrieval system.

Furthermore, the testing cases is limited in volume, compared with the magnitude of the standard test data set for information retrieval. It is mainly because our system specifically focused on the data of analysts' operation traces and considerable manual efforts were required on trace analysis to get the ground truth. Each trace was collected by tracking an analyst's operations in performing a task, and a task normally took an analyst 60 to 120 minutes to complete. Besides, a analytical process conducted by an analyst is seldom linear so that it was a heavy burden to analyze an entire trace considering the complexity and diversity of human cognitive processes. We also need to ensure each each testing case was informative enough to represent a self-contained data triage process. Therefore, tremendous manual efforts were required to get the ground truth for evaluation. Given these challenges, the current testing data set enabled us achieve our evaluation goal.

In addition, having more human judges and conducting statistical analysis can better validate the fairness of the human evaluation results. However, we did encounter some real-world constraints in running the evaluation experiments. First, given the tight schedule of professional analysts, we found that professional analysts have very limited availability. Second, we found that the human judges have to be familiar with the cyber security data triage operations and possess sufficient expertise. Due to this constraint, we could hardly recruit human judges from a crowdsourcing platform (e.g., Amazon Mechanical Turk). Taking the difficulties into consideration, we ended up with having two human judges evaluate hundreds of cases so that we could check two types of consistency within and between the subjects. The results of our two expert judge experiment can provide some valuable insights regarding the usefulness of the proposed retrieval system; in contrast, the inputs from many if not most human judges recruited through Amazon Mechanical Turk are probably incorrect.

Last, the usefulness of the retrieval results is bound to the quality of the collected traces. Considering the difficulty of avoiding low-quality traces once an analyst has been performing a task, it is easier to filter out low-quality trace after a task. Our suggestion for SOC practitioners is to track all the operations of senior analysts and evaluate their performance after each task. The task performance can be evaluated using the existing performance measures, which is out of the scope of this work. Once the task performance being evaluated, the traces of the low performance can be removed. The same trace-quality-control method was used in our experiment: we first evaluated the analysts' performance by comparing their reports with the ground truth, and then selected the traces corresponding to the high task performance.

VI. Related Work

So far we have found no prior work specific to the information retrieval on analysts' data triage operations to facilitate cyber analysis. However, we note several areas of related work that are of interest in our work. Generally, these works can be classified into two categories: cyber defense analysis in SOCs and graph-based information retrieval approaches.

A. Cyber Defense Analysis in SOC

One of the most significant challenges to SOCs is how to process overmuch information flooded into SOCs with vast quantity and variety. Traditional IDS systems are unable to provide a global view of the network security and the number of alerts are too large to manually process. Therefore, most analysts have to spend plenty of time analyzing massive data to detect multistep attacks. Several cognitive task studies have been conducted to explore analysts' cognitive process. Amico and Whitley focused on the process in which the raw network data are analyzed by different roles of analysts and finally transformed into cyber situational awareness [7]. Erbacher et al. mainly investigated the analysts' needs, goals, concerns through interview studies [10]. Paul and Whitley tried to understand the analysts' mental model by conducting a series of interviews, observations and a card sorting activity. They developed a taxonomy of the questions that were asked by the analysts for event detection and orientation [19]. Such cognitive task analysis studies highlight that multiple stages of analysis are needed to transform raw network data into cyber situational awareness. Data triage, as the first stage of data processing, is time-consuming but not simply rule-based, so that it still requires analysts' expertise and experience.

These studies also provide insight into the necessity for cyber security analysis tools. Etoty and Erbacher investigated the data visualization needs at multiple analysis phases [11]. Cappers et al. proposed a visual analytics approach for exploring the network events in multiple contexts [3]. Lots of automated data analysis tools have been developed to reduce the volume of data that overwhelms analysts. Improving the signature-based approaches, Yen et al. developed a system, named Beehive, to automatically detect suspicious network activities [27]. It extracts the suspicious data and reports them as potential incidents to analysts to reduce the workloads of analysts. With a similar goal of reducing the data volumne, Pecchia et al. leverages different text weighting schemes to filter the alerts generated by a SIEM tool in a Software-as-a-Service (SaaS) cloud [20]. In additional to data filtering, analysts need to manually correlate the evidence from various data sources. Raftopoulos et al. built a decision support tool based on decision tree classifier to show how to combine and correlate evidences from multiple data sources [21]. A gray-box prediction model using statistical properties can predicate attack rates based on honeypot history data retrieval [28], in which it provides a percentage rate on attack to cyber analysts. Zhong et. al developed a conceptual model to represent analysts' experience and pointed out the potential benefits of utilizing the analysts' experience [29]. Follow-up works based on the conceptual model include an experience retrieval system [30] and an automated data triage system [33]. These automated cyber security analysis support tools were developed with the recognition of analysts' needs. The critical part of most of the cases is the abstract model that represents and uses analysts' expertise and experience. Comparing with these works, the proposed approach of this paper mainly focuses on the analysts' expertise and utilizes it in a more explicit way. Therefore, our approach is a good complement to the existing systems in facilitating cyber security analysis in SOCs.

B. Context-Aware Information Retrieval Approaches

Context-aware information retrieval has experience tremendous growth. Graph, as a common representation of the relationships among entities, has been widely used in such information retrieval methods. In this work, we focus on graph-based information retrieval. Query information through graph is time-consuming considering that subgraph isomorphism is worst-case exponential. One approach for achieving efficiency is to avoid subgraph isomorphism. Degree reduction can be applied if the graphs are labeled in a way that the correspondences between vertices and their includer graphs can be restricted [25]. However, the efficiency of degree reduction is uncertain because it relies heavily on the practical implementation [25]. To guarantee the efficiency of our graph-based algorithm, we avoid subgraph isomorphism by introducing centroids as a subgraph representation. Besides, parallelism is a powerful approach for addressing the computing problem. Many parallel graph processing libraries have been developed, such as Parallel Boost Graph Library (PBGL) [13], MultiThreaded Graph Library (MTGL) [1], and Pregel [18]. In practice, the analysts' operations traces will be continuously collected. Given a large graph collection, we can consider parallel graph processing to improve the scalability of our system.

VII. Conclusions

This work aims to facilitate data triage tasks by providing junior analysts with the suggestions of the data triage operations conducted by the senior analysts in a similar context. We developed a retrieval method for matching the traces of analysts' data triage operations based on the centroid measure. This centroid-based measure not only considers the characteristic constraints but also investigates the geometry structure of the graphs. The performance of the retrieval system has been evaluated through both automated testing and human evaluation. To our best knowledge, it is the first effective method for retrieving analysts' data triage operations. In the future, we will continue to improve the performance in terms of the precision and recall rates.

VIII. Acknowledgement

This work is supported by ARO W911NF-15-1-0576.

References

[1] J. W. Berry, B. Hendrickson, S. Kahan, and P. Konecny. Software and algorithms for graph queries on multithreaded architectures. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pages 1–14. IEEE, 2007.

[2] T. Caldwell. Plugging the cyber-security skills gap. *Computer Fraud & Security*, 2013(7):5–10, 2013.

[3] B. C. Cappers and J. J. van Wijk. Understanding the context of network traffic alerts. In *Visualization for Cyber Security (VizSec), 2016 IEEE Symposium on*, pages 1–8. IEEE, 2016.

[4] K. Chen, P. Liu, and Y. Zhang. Achieving accuracy and scalability simultaneously in detecting application clones on android markets. In *Proceedings of the 36th International Conference on Software Engineering*, pages 175–186. ACM, 2014.

[5] P.-C. Chen, P. Liu, J. Yen, and T. Mullen. Experience-based cyber situation recognition using relaxable logic patterns. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2012 IEEE International Multi-Disciplinary Conference on*, pages 243–250. IEEE, 2012.

[6] A. D'Amico and K. Whitley. The real work of computer network defense analysts. In *VizSEC 2007*, pages 19–37. Springer, 2008.

[7] A. D'Amico and K. Whitley. The Real Work of Computer Network Defense Analysts. In J. R. Goodall, G. Conti, and K.-L. Ma, editors, *VizSEC 2007: Proceedings of the Workshop on Visualization for Computer Security*, pages 19–37. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[8] A. D'Amico, K. Whitley, D. Tesone, B. O'Brien, and E. Roth. Achieving cyber defense situational awareness: A cognitive task analysis of information assurance analysts. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 49, pages 229–233. SAGE Publications Sage CA: Los Angeles, CA, 2005.

[9] M. R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human factors*, 37(1):32–64, 1995.

[10] R. F. Erbacher, D. A. Frincke, P. C. Wong, S. Moody, and G. Fink. A multi-phase network situational awareness cognitive task analysis. *Information Visualization*, 9(3):204–219, 2010.

[11] R. E. Etoty and R. F. Erbacher. A survey of visualization tools assessed for anomaly-based intrusion detection analysis. Technical report, DTIC Document, 2014.

[12] E. T. Greenlee, G. J. Funke, J. S. Warm, B. D. Sawyer, V. S. Finomore, V. F. Mancuso, M. E. Funke, and G. Matthews. Stress and workload profiles of network analysis: Not all tasks are created equal. In *Advances in Human Factors in Cybersecurity*, pages 153–166. Springer, 2016.

[13] D. Gregor and A. Lumsdaine. The parallel bgl: A generic library for distributed graph computations. *Parallel Object-Oriented Scientific Computing (POOSC)*, 2:1–18, 2005.

[14] N.-C. Hsieh. An integrated data mining and behavioral scoring model for analyzing bank customers. *Expert systems with applications*, 27(4):623–633, 2004.

[15] S. Jajodia, P. Liu, V. Swarup, and C. Wang. *Cyber situational awareness*, volume 14. Springer, 2010.

[16] J. Jung, I. Choi, and M. Song. An integration architecture for knowledge management systems and business process management systems. *Computers in industry*, 58(1):21–34, 2007.

[17] G. Klein. The recognition-primed decision (rpd) model: Looking back, looking forward. *Naturalistic decision making*, pages 285–292, 1997.

[18] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 135–146. ACM, 2010.

[19] C. L. Paul and K. Whitley. A taxonomy of cyber awareness questions for the user-centered design of cyber situation awareness. In *International Conference on Human Aspects of Information Security, Privacy, and Trust*, pages 145–154. Springer, 2013.

[20] A. Pecchia, D. Cotroneo, R. Ganesan, and S. Sarkar. Filtering security alerts for the analysis of a production saas cloud. In *Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on*, pages 233–241. IEEE, 2014.

[21] E. Raftopoulos, M. Egli, and X. Dimitropoulos. Shedding light on log correlation in network forensics analysis. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 232–241. Springer, 2012.

[22] P. Rajivan, M. A. Janssen, and N. J. Cooke. Agent-based model of a cyber security defense analyst team. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 57, pages 314–318. SAGE Publications Sage CA: Los Angeles, CA, 2013.

[23] S. Stevens-Adams, A. Carbajal, A. Silva, K. Nauer, B. Anderson, T. Reed, and C. Forsythe. Enhanced training for cyber situational awareness. In *International Conference on Augmented Cognition*, pages 90–99. Springer, 2013.

[24] S. C. Sundaramurthy, J. Case, T. Truong, L. Zomlot, and M. Hoffmann. A tale of three security operation centers. In *Proceedings of the 2014 ACM Workshop on Security Information Workers*, pages 43–50. ACM, 2014.

[25] J. R. Ullmann. Degree Reduction in Labeled Graph Retrieval. *Journal of Experimental Algorithmics (JEA)*, 20:1–3, 2015.

[26] J. Yen, R. F. Erbacher, C. Zhong, and P. Liu. Cognitive process. In *Cyber Defense and Situational Awareness*, pages 119–144. Springer, 2014.

[27] T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, and E. Kirda. Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks. In *Proceedings of the 29th Annual Computer Security Applications Conference*, pages 199–208. ACM, 2013.

[28] Z. Zhan, M. Xu, and S. Xu. Predicting cyber attack rates with extreme values. *IEEE Transactions on Information Forensics and Security*, 10(8):1666–1677, 2015.

[29] C. Zhong, D. S. Kirubakaran, J. Yen, P. Liu, S. Hutchinson, and H. Cam. How to use experience in cyber analysis: An analytical reasoning support system. In *Intelligence and Security Informatics (ISI), 2013 IEEE International Conference on*, pages 263–265. IEEE, 2013.

[30] C. Zhong, D. Samuel, J. Yen, P. Liu, R. Erbacher, S. Hutchinson, R. Etoty, H. Cam, and W. Glodek. RankAOH: Context-driven similarity-based retrieval of experiences in cyber analysis. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2014 IEEE International Inter-Disciplinary Conference on*, pages 230–236. IEEE, 2014.

[31] C. Zhong, J. Yen, P. Liu, R. Erbacher, R. Etoty, and C. Garneau. Arsca: a computer tool for tracing the cognitive processes of cyber-attack analysis. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2015 IEEE International Inter-Disciplinary Conference on*, pages 165–171. IEEE, 2015.

[32] C. Zhong, J. Yen, P. Liu, R. Erbacher, R. Etoty, and C. Garneau. An integrated computer-aided cognitive task analysis method for tracing cyber-attack analysis processes. In *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security*, page 9. ACM, 2015.

[33] C. Zhong, J. Yen, P. Liu, and R. F. Erbacher. Automate cybersecurity data triage by leveraging human analysts' cognitive process. In *Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), 2016 IEEE 2nd International Conference on*, pages 357–363. IEEE, 2016.

[34] C. Zhong, J. Yen, P. Liu, R. F. Erbacher, C. Garneau, and B. Chen. Studying analysts' data triage operations in cyber defense situational analysis. In *Theory and Models for Cyber Situation Awareness*, pages 128–169. Springer, 2017.