2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)

RankAOH: Context-driven Similarity-based Retrieval of Experiences in Cyber Analysis

Chen Zhong, Deepak Samuel, John Yen, and Peng Liu

Robert Erbacher, Steve Hutchinson, Renee Etoty, Hasan Cam, William Glodek

***Abstract*— In cyber analysis, it is highly desirable to support the analysis of junior analysts by leveraging the experiences of experts. But, there are two major challenges to achieve this goal. First, it is very costly to capture the experience of experts for the complex task of cyber analysis using traditional approaches such as protocol analysis. Second, it is difficult to identify previous experiences of experts that are relevant to the dynamic context of an analyst's cyber analysis task. To address the first challenge, a system has been developed to capture non-intrusively the analytical reasoning processes of analysts. To tackle the second challenge, this paper presents an effective and efficient approach for retrieving relevant experiences based on the dynamically changing context of cyber analysis. We define an experience as a process of analytical reasoning and adopt an Action-Observation-Hypothesis (A-O-H) model to represent the processes in cyber analysis. Based on this model, a tool for capturing and supporting the analytical reasoning processes is shown to be able to support the elusive cognitive process in dynamic cyber situations. The experience retrieval approach of this paper supports the efficient experience retrieval, and dynamically updates the results as the context of analysis evolves. The experience retrieval approach is evaluated, based on the precision and recall with respect to the ground truth. The evaluation results suggest that the proposed approach supports significantly the analytical reasoning of analysts by leveraging the experiences of experts.**

***Index Terms*—Decision support, Knowledge management, Context-based retrieval, Intrusion detection.**

I. Introduction

Cyber-attacks have become a serious threat to enterprise network. Attackers tend to achieve their malicious goal by compromising more hosts in the organizational network over a longer period of time. Due to the severity, longevity and subtlety of these attacks, cyber analysis, aimed to detect abnormal signals and ascertain malicious events, has become a big challenge for network protection. Analysts are required to be able to process large amounts of data generated from the monitoring sensors in organizational networks and to possess good analytical reasoning capability in order to make a correct judgment of the situation in a short time. However, a big gap exists between the rapidly increasing complexity of cyber analysis and analysts' limited capability of data processing and analytical reasoning.

Expert analysts tend to be much more successful during analysis because they have gained much experience unlike novice analysts. Therefore, the experience of expert analysts should be in full use in cyber analysis. The basic idea of leveraging experience in cyber analysis is to capture and retain the experience of experts in an experience database and to provide analysts with the relevant previous experience. In this way, an analyst's experience in analyzing an attack may also help other analysts quickly and effectively deal with the similar situation. Besides, novice analysts can learn much from experts' experience through "on job training". A system has been developed that can capture the analytical reasoning process of analysts in a non-intrusive way [1].

Besides experience solicitation and representation, experience retrieval is critical for successfully leveraging experience. Firstly, we need to retrieve experiences that are relevant, considering the fact that irrelevant experiences may waste analysts' time or even be misleading. Secondly, the retrieved results need to be updated when the analysts gain new observations. It enables the support system to provide analysts with relevant previous experiences as guidance in a step-by-step manner. Moreover, because of the dynamic nature of cyber situation, experiences need to be retrieved rapidly in order to help analysts make judgments within a short time.

Faced with the above challenges, the goal of this research is to develop an effective and efficient approach for experience retrieval. Experience retrieval, like most information retrieval problem, largely depends on how experience is represented. Therefore, how to represent analysts' experience in cyber analysis becomes the second focus of our research.

We first narrow the meaning of experience down to the analysts' analytical reasoning process in which we are interested, and adopted the A-O-H model to represent it [1]. The A-O-H representation provides two main benefits. First, the A-O-H model, informed by the sense making theory in cognitive science, includes three key cognitive constructs: Action, Observation and Hypothesis. These constructs captures not only the

This is a draft. To copy, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

C.Z., D.S., J.Y., P.L. Authors are now with College of Information Sciences and Technology, Pennsylvania State University, University Park, State College, PA 16801 USA (email: {czz111, dok5181, jyen, pliu}@ist.psu.edu)

R.E., R.E., H.C., and W.G. Authors are now with Army Research Lab (email: {robert.f.erbacher.civ, renee.e.etoty.civ, hasan.cam.civ, william.j. glodek.civ} @mail.mil). S.H., Author is now with ICF International Inc (email: steve.e.hutchinson.ctr@mail.mil)

context information but also the physical behaviors and mental work of analysts in their analytical reasoning processes. Second, the A-O-H-based representation is semi-formal, combining formal structural Observations and Actions with informal free-text description of Hypotheses. The formally-represented components facilitate the experience retrieval, and meanwhile the free-text Hypotheses makes the representation flexible enough to fit with the unexpected nature of human reasoning in the dynamic cyber situation. Thus, this representation is both computable to retrieval systems and digestible to analysts.

In this paper, we propose an experience retrieval approach (called "**RankAOH**") based on the A-O-H representation. We mainly consider three questions: (1) How to define "similarity" between experiences? (2) How to support the human-centric interactive analysis process? (In other words, how to easily provide analysts with relevant experiences in a step-by-step manner?) (3) How to improve the performance of the experience retrieval?

Our contributions can be summarized as follows.

- We propose a similarity-based experience retrieval approach that retrieves experiences in which the context is similar to the analyst's current context. Besides, the retrieval approach can efficiently update the retrieval results along with the change of current observation of the cyber situation.
- We evaluate our retrieval approach based on precision and recall of the retrieved results with respect to the ground truth. The results show that this retrieval approach can well support analysts' reasoning processes by providing them with previous experiences in a timely manner.

The rest of the paper is organized as follows. We first discuss related previous research. Then, we introduce the A-O-H model and experience representation. In Section 3, we define similarity between contexts based on the A-O-H model. Section 4 describes the detail of our context-driven experience retrieval approach RankAOH and its evaluation in detail. Finally, we discuss future research.

II. Related Work

An experience knowledge base is required for experience retrieval. Therefore, the way we capture the experience of analysts and represent the experience determines how we design and develop our approach for experience retrieval.

Rule-based logic representations are often used to represent knowledge. For example, based on specified rules, a logical attack graph can be generated, given a network and its vulnerabilities [3]. Chen et al. use Horn Rules to represent analysts' experience patterns [4]. Although rule-based representations are powerful for inference, the highly-formalized structure makes them limited for the problems in the dynamic cyber environment. In order to improve the coverage of the logic rules, Chen et al. proposed the concept of "relaxation" that improves the coverage of the logic rules by relaxing the conditions of the rules. However, the rule-based approach still has its limitation for dealing with new attack strategies and unknown vulnerabilities.

Our idea of leveraging previous experience in cyber analysis in new situations is similar to the general vision of Case-Based Reasoning (CBR). CBR approaches utilize the specific knowledge gained from previous situations ("cases") to solve problems [5]. Cases can be structured, semi-structured or unstructured. Retrieval is an important construct in the CBR framework and directly depends on how cases are represented. Therefore, in the light of CBR, a piece of experience gained by an analyst in accomplishing a task is a specific case. In order to achieve the goal of leveraging analysts' experiences to help future analysis, a particular retrieval approach that can effectively retrieve experience cases in a short time is needed.

There is an important difference between CBR and proposed context-based experience retrieval. The AOH-based experience representation captures both positive experience and negative experience (i.e., rejected hypotheses); whereas traditional CBR only retrieves solutions of similar cases. Retrieving relevant negative experience can help cyber analysts to avoid wasting time on pursuing irrelevant hypotheses.

TABLE 1
Relation between A-O-H and OODA

OODA	A-O-H	Explanation
Observation	**Observation**	The observation in OODA refers to the raw information which is presented before analysts' involvement. This data is captured in the Observation component in A-O-H model.
Orientation		Orientation in OODA is "fusing information to build situational awareness)" [2] This essentially incorporates the observation and through hypothesis cycles.
	Action	The action performed to explore the monitoring data, to prove or disprove each hypothesis. These actions will result in new observations.
	Observation	The observation of some interesting data resulted from actions. The collection of data may trigger analysts' new hypotheses.
	Hypothesis	The thoughts generated based on the current observation. It could be an interpretation of current situation, questions in mind, or attempt to future actions.
Decision	**Hypothesis**	The results of analyzing all hypotheses will result in a final decision. This is essentially the confirmed hypotheses of the A-O-H model.
Action	**N/A**	Occurs after analysts' analytical reasoning and is thus not within the scope of the A-O-H model.

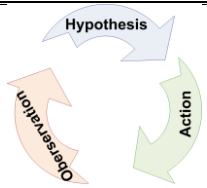

Fig. 1. An analytical reasoning process can be modeled as an iterative cycle involving Action, Observation and Hypothesis (A-O-H Model) [1].

Previous research has also shown that network analysts identified their process as the OODA loop (Observation, Orientation, Decision, Action) [2, 6]. We adopted the A-O-H model to represent the iterative analytical process, which we specifically focus on the iterative process in which the analysts perform filtering, searching, and selection actions on the large volume of data, select observations of interests, describe their

hypotheses, which can lead to additional observations. The relationship between the A-O-H model and the OODA loop is explained in Table 1.

III. Experience Representation And Requirements for Experience Retrieval

A. A-O-H Model and Experience Definition

Analysts' analytical reasoning processes while analyzing cyber-attacks can be modeled as the Action-Observation-Hypothesis model (A-O-H model), which is an iterative cycle involving three components: Action, Observation and Hypothesis (Fig. 1) [1]. Actions are taken by the analysts when they explore the network monitoring data; Observations are those data that the analysts identified to be of interest; Hypotheses refer to the analysts' thoughts, such as assumptions or conclusions in a certain situation.

In order to make the meaning of "Experience" in this case clear and unambiguous, we define experience as follows: a piece of experience gained by analysts in accomplishing a particular task is an analytical reasoning process which can be modeled by the A-O-H model, where each Action lead to a new Observation, which prompt the analyst to generate one or more Hypotheses, which lead to an additional Action, hence the cycle continues [1].

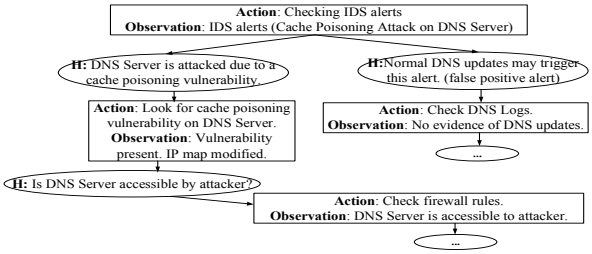

Fig. 2. An example of E-Tree [1].

B. Experience Representation based on the A-O-H Model

1) E-Tree and H-Tree

Each captured analytical reasoning process is structured as an Experience Tree ("**E-Tree**") [1]. Therefore, an E-Tree represents a single instance of experience. An E-Tree consists of nodes modeled as Experience Units ("**EU**"s) and their corresponding Hypotheses ("H"s) resulting from each EU. An EU consists of an Action and the resulting Observation. The Hypothesis nodes which are children of an EU indicates these Hypotheses are generated based on the EU. One or more Hypotheses result from each EU, each of which could in turn trigger a new EU. Fig. 2. is an example E-Tree for analysis involving a DNS attack on the DNS server.

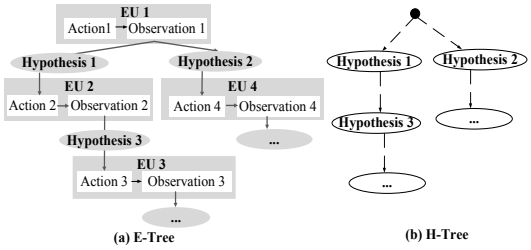

Fig. 3. An E-Tree and the corresponding H-Tree [1].

An abstract E-tree structure is shown in Fig. 3 (a). To facilitate the hypothesis management and capture the flow of thoughts, the Hypotheses in E-Tree are extracted into a tree called "**H-Tree**" (Fig. 3 (b)).

IV. RankAOH: Experience Retrieval Approach

The experience representation based on the A-O-H model enables us to capture experts' analytical reasoning processes as experience pieces in a database while they are accomplishing their tasks. In order to use the captured experiences to guide the current task of an analyst, we need an approach to retrieve the matched experience pieces from the database based on the context of current analysis. Any update of the current context (i.e. gaining more observations) will trigger the updates of the matching results.

A. Context-Driven Retrieval

The "context" of analysis is the basis for our experience retrieval approach. By creating and manipulating Actions, Observations, and Hypotheses, an analyst's analysis process can be viewed as a constructing process of an E-Tree and a corresponding H-Tree. Each Hypothesis is generated based on a context. It means a context is defined regarding a particular Hypothesis. Hypotheses capture the thoughts of the analyst, and the Actions and Observations (i.e. **EU**s) record the context information of Hypotheses. Therefore, we first define "**EU-path**" of a Hypothesis as a list of EUs in the unique path from the root of the E-Tree to this Hypothesis. Based on the definition of E-path, we have,

Definition [Context]: The Context of a Hypothesis is its unique EU-path.

As an example, the context of Hypothesis 3 (in Fig. 3) refers to the EU-path "EU 1 → EU2". In order to distinguish the current E-Tree under construction from the E-Trees in the database, the current E-Tree can be referred as "**Query- E-Tree**". The Query-E-Tree should only have one **current hypothesis**, considering that an analyst only has one focus at a time. Given our definition of context, the **current context** is an EU-path of the current hypothesis in the Query-E-Tree.

The goal of experience retrieval is to search in the database for EU-paths which matches the current context and to rank the matched EU-paths according to the degree of similarity.

TABLE 2
Monitoring Data Sources and Their Attributes

Data Source	Fields/Attributes
IDS Alerts	Time, Src IP, Src Port, Des IP, Des Port, Signature , Description.
Vulnerability Reports	IP, Vulnerability ID, Risk, Description
Package Dumps	Time, Src IP, Des IP, Protocol, Src host type, Des host type, Data Package.
Port Information	IP, Port, Service, Version, Status.
Authorization Logs	Time, IP, Information, Authorization type.
Server Logs (Web Servers, DB, File Servers)	Time, IP, Activity, Host type, Description
Anti-virus Logs	Time, IP, Virus, Description, Status, Priority.
Firewall Logs	Time, Src IP, Src Port, Des IP, Des Port, Priority, Operation, Protocol, Message, Des Server, Direction.
Network Status	IP, Status, Connectivity

*Src: Source, Des: Destination

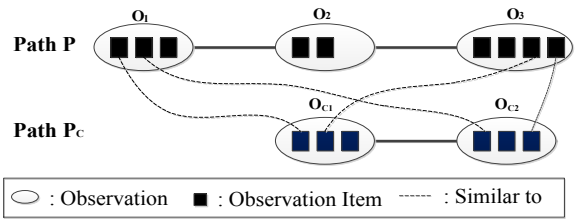

Fig. 4. Computing similarity of Path P and P_C.

B. Similarity

We use **Similarity** to determine the degree of similarity between an EU-path in the database and the current context. Let $\boldsymbol{P}$ be an EU-path and $\boldsymbol{P_C}$ be the current context, the similarity is denoted as $\mathbf{Sim}(\boldsymbol{P}, \boldsymbol{P_C})$. Each EU-path is a set of EUs, therefore we use **Jaccard Similarity** to compute $\mathrm{Sim}(\mathrm{P}, \mathrm{P_C})$:

$$\mathrm{Sim}(\mathrm{P}, \mathrm{P_C}) = \frac{|P \cap P_C|}{|P \cup P_C|} \tag{1}$$

Although P and P_C are sets of EUs, given an EU, we only need to consider the Observation in this EU[1], considering the fact that an Action and the following Observation always come in pairs in each EU. Moreover, each Observation is a set of ***Observation Items (OIs)***. An OI is a particular "signal" from a data source. Table 2 shows nine monitoring data sources and their fields, and an OI could be a data entry from a particular data source. For example, an OI could be a data entry from the IDS Alerts: "*09-10-2010 22:42:05 50.100.1.31 50.100.1.31 (17324) SHELLCODE x86 Linux reverse connect shellcodet*". Essentially, an OI is a set of fields contained in a data entry from a data source. Therefore, **the type of an OI** is defined by the data source.

Fig. 4 shows an example of computing the similarity of two paths. Given the path structure, we have

$$\begin{gathered}\mathrm{Sim}(\mathrm{P}, \mathrm{P_C}) = \frac{|P \cap P_C|}{|P \cup P_C|} = \frac{|P(O) \cap \mathrm{P_C}(\mathrm{O_C})|}{|P(O) \cup \mathrm{P_C}(\mathrm{O_C})|} \\ P(O) = \{O_x | O_x = \{oi_x\}, O_x \in P\}, \\ P(O_C) = \{O_y | O_y = \{oi_y\}, O_y \in P_C\}\end{gathered} \tag{2}$$

where oi_x and oi_y are OIs in Observation O_x and O_y.

1) Similarity between Observations

To calculate (2), we further define the similarity between Observations. Let $\mathrm{O_x}, \mathrm{O_y}$ be two observations, their similarity is

$$\mathrm{Sim}(\mathrm{O_x}, \mathrm{O_y}) = \sum_{oi_x \in O_x, oi_y \in O_y} f(oi_x, oi_y) \tag{3}$$

where oi_x and oi_y are OIs, and $f(oi_x, oi_y)$ refers to their similarity. We don't model Observations as ordered sequences of OIs because the order of two Observations in a path is mainly determined by the analysis sequence but can hardly indicate any logic relationship. Hence, comparing two Observations, we need to consider all the possible combinations of their OIs. For $\mathrm{O_3}$ and $\mathrm{O_{C1}}$ in Fig. 4, 12 (3*4) comparisons need to be done.

2) Similarity between Observation Items (OIs)

Each OI is taken from a data source (Table 2). Let $\mathrm{oi_x}, \mathrm{oi}_y$ be two OIs, the similarity between them, denoted as $f(oi_x, oi_y)$, is determined by matching the field values of their data sources. The matching includes:

- **Base Matching (BM).** BM refers to the minimum matching criteria, that is, if BM of oi_x and oi_y is violated, $\boldsymbol{f(oi_x, oi_y)}$ equals to 0. One basic BM criterion is that oi_x and oi_y should correspond to the same data source. Experts can define other BM criteria by identifying the fields in certain data sources that must have the same value.
- **Weighted Matching (WM)**. Once BM is satisfied, WM is used to calculate the degree of matching. Each individual field of various data sources in Table 2 is assigned a weight (based on domain knowledge). Given oi_x and oi_y, the WM score equals to $\sum \boldsymbol{w_i} * \boldsymbol{Match}\left(\boldsymbol{Field_i}(\boldsymbol{oi_x}), \boldsymbol{Field_i}(\boldsymbol{oi_y})\right)$, where $Match()$ comparing the two field values (equals to 1 if they are equal, otherwise equals to 0).

Taking BM and WM together, the similarity between two OIs (i.e. oi_x and oi_y) is defined as:

$$f(oi_x, oi_y) = \begin{cases} 0, & BM\ violated \\ \sum w_i * Match\left(Field_i(oi_x), Field_i(oi_y)\right), & Otherwis \end{cases} \tag{4}$$

3) Similarity between Paths

Given the definition of the similarity between Observations and OIs, we have

$$\begin{gathered}|P(O) \cap \mathrm{P_C}(\mathrm{O_C})| = \sum_{\mathrm{O_x} \in \mathrm{P(O)}, \mathrm{O_y} \in \mathrm{P_C}(\mathrm{O_C})} \mathrm{Sim}(\mathrm{O_x}, \mathrm{O_y}) \\ |P(O) \cup P_C(O_C)| = \bigcup_{\mathrm{oi} \in \mathrm{P(O)} \cup \mathrm{P_C}(\mathrm{O_C})} oi\end{gathered} \tag{5}$$

In conclusion, we rebuild Formula (1) as

$$\mathrm{Sim}(\mathrm{P}, \mathrm{P_C}) = \frac{\sum_{\mathrm{O_x} \in \mathrm{P}, \mathrm{O_y} \in \mathrm{P_C}} \sum_{oi_x \in O_x, oi_y \in O_y} f(oi_x, oi_y)}{\left| \bigcup_{\mathrm{oi} \in \mathrm{P(O)} \cup \mathrm{P_C}(\mathrm{O_C})} oi \right|} \tag{6}$$

where $f(oi_x, oi_y)$ is defined in Formula (4). It shows that more common OIs in two paths increases the numerator, while increasing the OI number also results in larger denominator.

C. Similarity-based Experience Retrieval

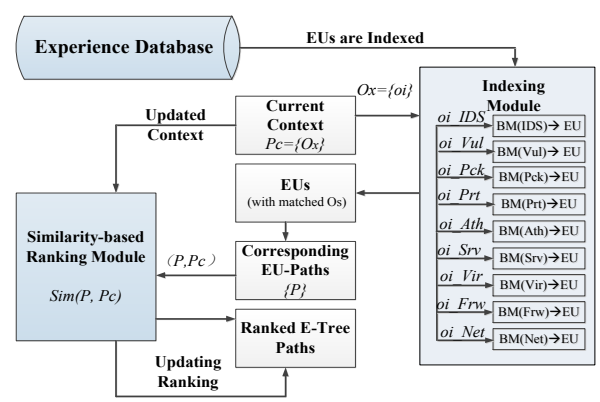

Fig. 5. Similarity-based Experience Retrieval

Fig. 5. shows the framework of the experience retrieval approach, including three components: an experience database, an indexing module, and a similarity-based ranking module.

[1] We may also consider Actions in our retrieval approach in the future. (Please refer to the Discussion Section).

1) Indexing

The indexing module helps efficiently retrieve matched EUs in the database. Given the BM fields predefined by experts, the BM field of each type of OI has a range of possible values. The indexing module maps each BM field value to the EUs that have this particular BM field value. The indexing will be updated whenever a new E-Tree is added in the database. Therefore, we don't need to compare every pair of OIs in the in the current context EUs with those in the other EUs from the database.

Given the current context P_C, we first extract all the BM fields from the EUs in P_C, and then the indexing module identifies all the matched EUs in the database. With the matched EUs using the indexing module, we can get the candidate EU-paths: as long as an EU-path contains at least one matched EU, this EU-path will be counted as a candidate path.

2) Match Propagation (MP) and Ranking

Given the candidate EU-paths, the ranking module ranks the EU-paths from the database, and present the analysts with the complete E-Tree paths (containing both EUs and Hypotheses) corresponding to the top K retrieved EU-paths. We also present the Hypotheses because these Hypotheses records analysts' thoughts in the previous processes so that may help the current analyst understand the retrieved EU-paths and therefore make a wise selection among them.

The **Match Propagation (MP)** algorithm is used by the ranking module to efficiently rank the EU-paths based on the similarity between the EU-paths with the current context. Once the current context is updated (when the analyst gains new observations), an update MP needs to be performed along each retrieved EU-path.

We first describe the structure of E-Tree for MP:

- Each EU could have several child EUs.
- Each EU has a parent EU.
- Each EU is assigned a matching scores **(M-Score)** based on the similarity between the Observation in this EU and the Observations in the current context (initially 0).
- If an EU has child EUs, it is also assigned a list of matching scores, each of which is the M-Score for the subtree with one of its child EU as the root (initially 0). We called the list "**Subtree M-Score List (M-List)**".

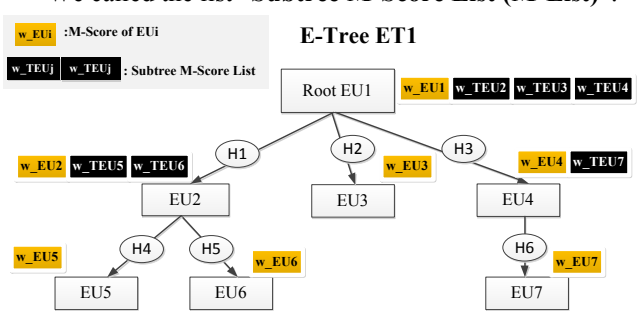

Fig. 6. E-Tree ET1 for ranking

Fig. 6. shows an example E-Tree for ranking. Each EU in ET1 is attached with a M-Score (in yellow). The EUs which have child EUs are also attached with a M-List (in black). The M-List of Root EU-1 is {*w_TEU2, w_TEU3, w_TEU4*}, where *w_TEU2, w_TEU3, w_TEU4* represent the M-Score of the sub-tree with the root EU-2, EU-3, and EU4 respectively.

Given an E-Tree, the MP algorithm propagates the M-Score of each EU in the E-Tree to all its ancestor EUs along the path to the E-Tree root. We define the propagation rule as follows.

Let EU_{parent} be an EU with n child EUs, $EU_{c1}, .., EU_{cn}$. EU_{parent} has a M-List: $\{w_{TEU_{c1}}, .., w_{TEU_{cn}}\}$. For $\forall i \in [1, n]$, the $w_{TEU_{ci}} = w_{EU_{ci}} + \sum w_{TEU_j}$, w_{TEU_j} is in EU_{ci}'s M-List.

Next, we describe the MP process using the example in Fig. 6. Before MP, all the M-Scores are initialized to 0 (Fig. 7 (a)). Suppose the current context is P_C, and we have $O_{C1}, O_{C2}, O_{C3} \in P_C(O_C)$, (i.e. the EU-path P_C contains three Observations O_{C1}, O_{C2}, O_{C3}). If $O_{EU5} \in EU5$, $Sim(O_{EU5}, O_{C1})$ =0.6, EU-5 is assigned a M-Score 0.6. Then, this score will be propagated according to the propagation rule. The updated M-Lists are shown in Fig. 7 (b). If $O_{EU6} \in EU6$, $Sim(O_{EU6}, O_{C1})$ =0.5, and $O_{EU7} \in EU7$, $Sim(O_{EU7}, O_{C2})$ =0.2 we further assign EU6 with a M-Score=0.5 and EU7 with M-Score=0.2, and propagate them. Fig. 7 (c) shows the result of the propagation.

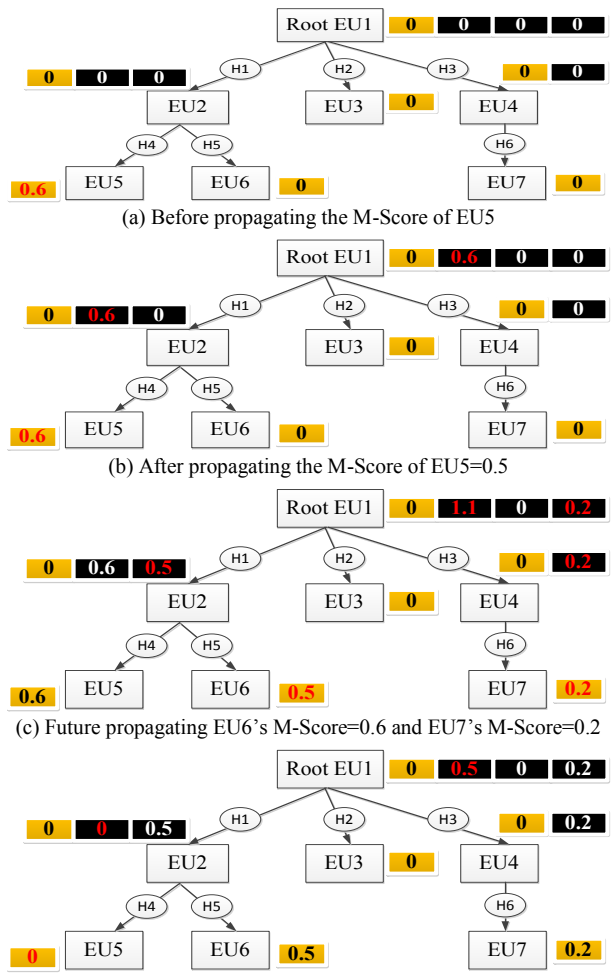

Fig. 7. M-Score propagation cases

The MP process is carried out for all the E-Trees that contains the matched EUs in the database. After MP, each root of the E-Tree that contains at least one matched EU has its M-list updated. Then, we can rank these E-Tree paths based on these M-Lists. The time complexity of the MP algorithm is O (# of matched EUs * average length of the matched EU-paths).

Once the current context is changed, we need to redo MP to update the M-Scores of the affected EUs, in order to provide analysts with continuous and immediate retrieval results. For example, if $\mathrm{Sim}(\mathrm{O}_{\mathrm{EU5}}, O_{C1})$ becomes 0 due to the change of Pc, the M-Score of EU5 is updated to 0. The update MP result is shown in Fig. 7(d). The time complexity of the update algorithm is ***O (Length of the matched path)***.

V. EVALUATION

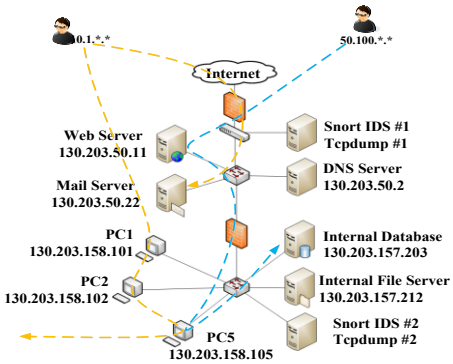

Fig. 8. Network Topology and Ground Truth.

We evaluate the RankAOH retrieval approach based on the precision and recall of the results of search with respect to the ground truth. We have nine types of monitoring data (in Table 2) regarding the network in Fig. 8[2]. The ground truth of attack implied by the monitoring data involves two multi-step attack chains targeting the database server and mail server respectively. We describe the ground truth as below,

- **(Ground Truth 1)** ***Attacker → Web server → PC → Database server.***
  The attacker attempts to exfiltrate packets from the database server. Remote code execution attack takes place at the web server. Malicious page served to PC3 resulting in RAT installation on PC3. DB records are ex-filtrated via PC3.
- **(Ground Truth 2)** ***Attacker → PC → Mail server.***
  Remote Admin Tool (RAT) installation on PC2 takes place via spam e-mail, followed by cracking stolen password hashes from PC2 for mail server login. Mail server privilege escalation attack takes place and packets/emails are exfiltrated.

34 pieces of experiences in the database is generated by two graduate students. They simulated the analytical reasoning processes in analyzing the network data and represented them using the A-O-H model. The experiences in the database are categorized into 3 types based on whether their analysis results are relevant to the ground truth (Table 3).

The analyzed attack scenarios that are partially related (PR) to the Ground Truths are shown below,

- **(PR1)** ***Attacker → PC → File server***
  PC installed with RAT with spam mail and file server vulnerability exposed resulting in access to shared files and directories.
- **(PR2)** ***Attacker → DNS server → Web server → PC***
  DNS cache poisoning attack on DNS server resulting in web server serving malicious pages to PC installing RAT.
- **(PR3)** ***Attacker → Web server → Database server***
  Remote code execution attack takes place on web server, followed by exploitation of DB SQL injection vulnerability on database server with a crafted request.
- **(PR4)** ***Attacker → Web Server → File Server → PC***
  Remote code execution attack takes place on web server, followed by privilege escalation on File server leading to ex-filtrating files from PC via shared file server.

The analyzed scenarios that are irrelevant to the Ground Truths are shown below,

- **(NR1)** ***Attacker → Web Server***
  Denial of Service attack takes place on the web server.
- **(NR2)** ***Attacker → Mail Server***
  Denial of Service attack takes place on the mail server.

TABLE 3
TYPES OF EXPERIENCE

Category	Description	# of Experiences
Relevant **(R)**	Related to the ground truth.	4
Partially Relevant **(PR)**	Have one or more steps in common with the ground truth.	15
Irrelevant**(NR)**	Not related to the ground truth.	15

A. Systematic Evaluation based on Precision and Recall

Precision and Recall have been used extensively as the evaluation metric for retrieval systems, which are defined as,

$$\text{Precision} = \frac{|\{\text{relevant EU_paths in the retrieved EU_paths}\}|}{|\{retrieved\ EU_paths\}|}$$

$$\text{Recall} = \frac{|\{\text{relevant EU_paths in the retrieved EU_paths}\}|}{|\{relevant\ EU_paths\}|}$$

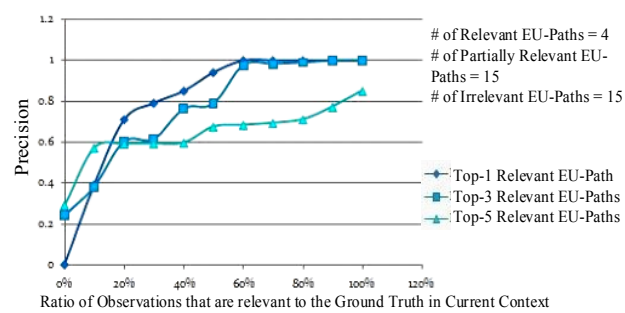

Fig. 9. Average Precision over 100 Randomized Test Cases for each of the Different Ratios of Relevant EU-Paths.

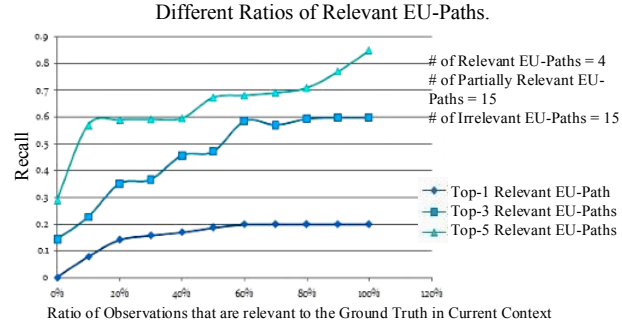

Fig. 10. Average Recall from 100 Randomized Test Cases for each of the Different Ratios of Relevant Observations

We first study how precision and recall are affected by the Observations. Fig. 9 shows that the precision increases with

[2] The data set is collected by S.Oh (http://yenlab5.ist.psu.edu/cybersa/).

increasing number of relevant observations. For relevant observations < 20%, it can be seen that, the top-5 results have better precision than top-1 and top-3 results. This is because the relevant results get ranked lower than some irrelevant paths when the ratio of relevant observation is less than 20%. In all other cases, top-1 and top-3 results have higher precisions since the relevant results get higher rank. Fig. 10 shows that recall improves on increasing ratio of relevant observations.

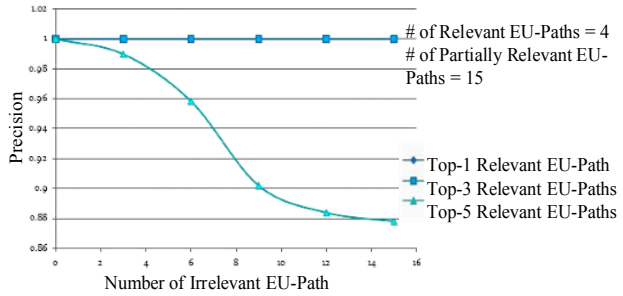

Fig. 11. Effect of Irrelevant Experiences on Precision

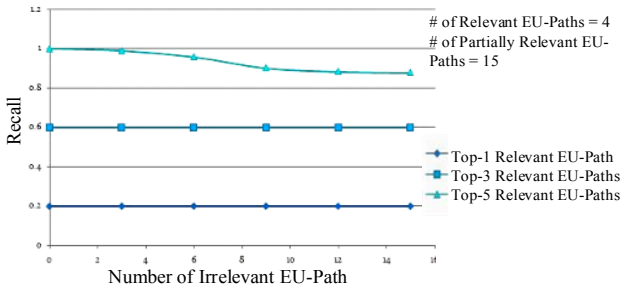

Fig. 12. Effect of Irrelevant Experiences on Recall.

Fig. 11 and Fig. 12 shows that both precision and recall of the top-5 results decrease marginally on increasing number of irrelevant experiences while the top-1 and top-3 results are not affected.

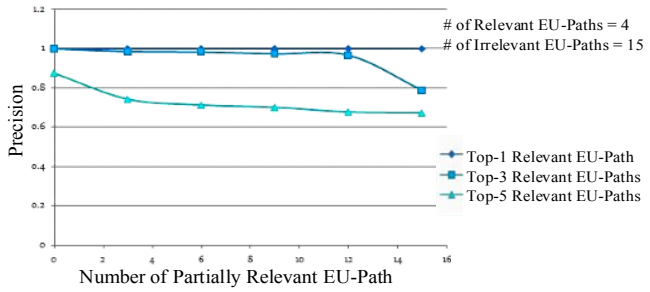

Fig. 13. Effect of Partially Relevant Experiences on Precision.

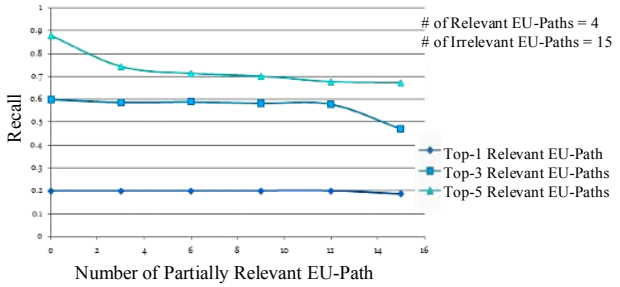

Fig. 14. Effect of Partially Relevant Experiences on Recall.

Fig. 13 shows that the precision of top-3 and top-5 results are more affected by the increase in the number of partially relevant experiences. The increase in partially relevant experiences affects precision more adversely compared to the effect of the increase in irrelevant experiences. Fig. 14 shows that the recall of the top-3 and top-5 results is more affected by the increase in the number of partially relevant experiences. The increase in partially relevant experiences affects recall more adversely compared to the effect of increase in irrelevant experiences.

It can be concluded that with increase in the number of irrelevant and partially relevant experiences in the experience base, the ranking of the results gets affected when the current context includes a small ratio of relevant observations. The precision and recall of the system are improved when the ratio of relevant observations is over 20%.

VI. Conclusion And Discussion

We propose an experience retrieval approach by defining the concept of context and similarity between contexts. The benefit is that the experience retrieval approach can significantly support the analytical reasoning of analysts by providing analysts with the experts' experiences in the contexts that are similar to their current context.

This work can be improved in the following aspects. First of all, we are doing experiments with analysts to gather their experience in accomplishing cyber analysis tasks, so that we can enrich our experience database to better evaluate the experience retrieval approach. Secondly, we only consider the Observations in the EU-paths when defining similarity. Actions and Hypotheses could also be considered when comparing two paths. For example, analysts could use linking actions in the A-O-H model to describe the reason that multiple OIs are identified together in an Observation (e.g., they involve the same destination IP). This linking action also contains rich information of analysts' insights at that stage of analysis. Finally, the retrieval also can be extended to support keyword-based retrieval of hypotheses from the experience database.

Acknowledgment

Supported by ARO Grant W911NF-09-1-0525 (MURI).

References

[1] C. Zhong, D. S. Kirubakaran, J. Yen and P. Liu, "How to Use Experience in Cyber Analysis: An Analytical Reasoning Support System," in *Proc. of IEEE Conf. on Intelligence and Security Informatics (ISI),* 2013, pp. 263–265.

[2] G. Vandenberghe, "Visually Assessing Possible Courses of Action for a Computer Network Incursion," in *SANS Institute, InfoSec Reading Room,*2007.

[3] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation," in *Proc. of the 13th ACM Conf. on Computer and Communications Security (CCS)*, 2006, pp. 263–265.

[4] P.C. Chen, P. Liu, J. Yen and M. Tracy, "Experience-based cyber situation recognition using relaxable logic patterns," in *Proc. IEEE International Multi-Disciplinary Conf. on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA),* 2012, pp. 243-250.

[5] A. Aamodt, and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI communications,* vol. 7, no. 1, pp. 39-59, 1994.

[6] R. F. Erbacher, D. A. Frincke, P. C. Wong, S. Moody and F. Glenn, "A multi-phase network situational awareness cognitive task analysis," *Information Visualization,* vol. 9, no. 3, pp. 204-219, 2010.