Intelligence Analyses and the Insider Threat

Eugene Santos, Jr., *Senior Member, IEEE*, Hien Nguyen, *Member, IEEE*, Fei Yu, *Student Member, IEEE*, Keum Joo Kim, *Associate Member, IEEE*, Deqing Li, *Student Member, IEEE*, John T. Wilkinson, Adam Olson, Jacob Russell, and Brittany Clark

***Abstract*—In the intelligence community, the existence of a malicious insider poses a severe threat to information, to the actual analytic process, and, ultimately, to any decision-making process relying on such information and analyses. An analyst with malicious intent can create irreversible short-term, as well as long-term, damage that is hard to detect. In this paper, we propose a novel methodology that detects malicious analysts who attempt to manipulate decision makers' perceptions through their intelligence reports. This detection method relies on each analyst's working style, which we assume to be consistent from task to task. In order to measure an analyst's degree of consistency, we employ a user-modeling technique that automatically builds a computational model of each analyst based on observation of their activities. We hypothesize that inconsistency is mainly caused by malicious actions. Therefore, the detection method evaluates how consistent an analyst is across different tasks and raises an alert if any significantly large inconsistency is detected. A normalization procedure is employed which allows us to compare across a group of analysts and is shown to reduce noise and amplify inconsistency that is due to malicious actions. We show that this improves detection performance. Our experiments demonstrate the effectiveness of our approach in detecting malicious insiders. In the experiments, the percentage of malicious insiders grouped with legitimate ones is varied, and results are collected with and without normalization in order to provide a comprehensive analysis of our approach.**

***Index Terms*—Cognitive styles, decision-making process, insider threat, intelligence analyses.**

I. Introduction

An *INSIDER* is a member of an organization who has access to privileged resources, has knowledge of internal information systems, and may be involved in decision-making processes. A *malicious insider* is an insider who has malicious intent that acts against the best interests of the organization. In the intelligence community (IC), such insider threats are much more dangerous because they potentially threaten human lives and national security. The overall objective of our work is to detect malicious insiders who aim to interfere with decision-making processes in intelligence analyses. While conducting an intelligence analysis, an analyst's actions generally refer to various information-seeking activities. In general, the key to the insider threat problem is to distinguish malicious actions from normal ones. With regard to the insider threat problem, we define normal actions as the ones that are driven with the intent to deliver an analyst's best judgment. In contrast, malicious actions are defined as the actions taken with the intent to bias the decision makers' perceptions toward a different conclusion from the one he would have drawn if he was not malicious. Current approaches assume normal actions to be both legitimate and relevant to one's task, while malicious actions violate either of these two features. However, such assumptions do not always hold. When malicious insiders attempt to manipulate a decision maker's perceptions through their intelligence reports, their actions are both legitimate (such as having privileges for accessing sensitive materials) and relevant to their analysis tasks in the sense that they deal with topics and events that are pertinent to the task.

In this paper, we propose a novel detection method that relies on a psychological indicator with user-modeling techniques to detect anomalies. Our basis is that the fundamental difference between normal and malicious actions rests with whether they follow an analyst's habitual working style. Because one's habitual working style rarely changes or changes very slowly over time, we conjecture that his information-seeking actions lead to conclusions in a consistent manner. On the other hand, the purpose of malicious actions is to form an attack rather than conduct a task. As a result, normal actions are considered to be consistent from task to task, while the existence of malicious actions breaks such consistency maintained with a habitual working style. In this paper, we design a method that looks for *inconsistent* behavior which serves as an indicator of a potential anomaly. An analyst's level of consistency is computed as the discrepancy between how his actions and conclusions correlate for two tasks. The more inconsistent an analyst is, the more likely he is malicious. As such, among a group of analysts, we determine an analyst to be malicious if his discrepancy value (also called inconsistency value) is higher than the average of all the others' discrepancies.

In order to measure the level of correlation between one's actions and his intelligence report (final conclusions), we employ a user-modeling technique that builds user models based on the textual content of the actions over time. The user model captures how one's perceived information evolves, which allows

Manuscript received June 30, 2010; revised December 23, 2010; accepted March 11, 2011. Date of publication August 30, 2011; date of current version February 17, 2012. This work was supported in part by the Air Force Office of Scientific Research under Grants FA9550-07-1-0050 and FA9550-09-1-0716, by the Defense Threat Reduction Agency under Grant HDTRA1-10-1-0096, and by the Office of Naval Research Multidisciplinary University Initiative. This paper was recommended by Associate Editor W. Pedrycz.

E. Santos, Jr., F. Yu, K. J. Kim, D. Li, and J. T. Wilkinson are with the Thayer School of Engineering, Dartmouth College, Hanover, NH 03755 USA (e-mail: eugene.santos.jr@dartmouth.edu; fei.yu@dartmouth.edu; keum.j.kim@dartmouth.edu; deqing.li@dartmouth.edu; john.t.wilkinson@dartmouth.edu).

H. Nguyen, B. Clark, and A. Olson are with the University of Wisconsin–Whitewater, Whitewater, WI 53190 USA (e-mail: nguyenh@uww.edu; clarkbm02@uww.edu; olsonam25@uww.edu).

J. Russell is with the University of Wisconsin–Milwaukee, Milwaukee, WI 53211 USA (e-mail: RussellJA30@uww.edu).

Digital Object Identifier 10.1109/TSMCA.2011.2162500

for measuring the level of correlation between one's perceived information and the information contained in one's report.

We evaluate the effectiveness of our detection method using data for eight legitimate insiders from a data set called APEX '07 plus five additional malicious insiders, each simulated based on one of the legitimate ones. Among a group of 13 insiders, the detection method captures four out of five malicious insiders without misidentifying any legitimate insider as malicious. The possible reason for failing to detect one of the malicious insiders is that a low correlation value between one's actions and conclusions tends to produce a relatively smaller discrepancy. In order to eliminate the impacts of these individual differences, we carry out a procedure that transforms all correlation values to a similar scale. This procedure normalizes the correlation value of each task over the correlation value of all tasks. In the remainder of this paper, we call it the normalization procedure. After applying the normalization procedure, our method is able to identify all five malicious insiders without raising any false alarms. In order to further examine the sensitivity of the method to different group assignments, exhaustive tests are conducted on different combinations of legitimate insiders and malicious ones. The results are compared with those of the exhaustive tests conducted with the normalization procedure. In general, the detection method has shown a robust performance with different group assignments. In addition, the performance is further improved after applying the normalization procedure. The contribution of our research is threefold. First, we propose a detection method based on a psychological indicator that none of the existing methods have explored. Second, we demonstrate that the method performs well in detecting malicious insiders with different group assignments. Third, the results of this research have also indicated that cognitive styles [1] can be quantified using computational models.

After publishing our preliminary results in [2], we noticed that some artifacts had been introduced during the construction of the malicious insiders, which needed to be removed. Thus, we revisited all the data for the malicious insiders and then made substantial changes. New results after the modification are published in this paper, and the changes are explained in Section V.

This paper is organized as follows. In Section II, we discuss related work tackling the insider threat problem. In Section III, we introduce the user-modeling technique used to model actions and reports, while the details of the detection method and the corresponding hypotheses are presented in Section IV. We describe the data set in Section V and detail the hypotheses in Section VI. In Section VII, we present the experiments conducted to evaluate the performance of the detection method. Discussions on the concept of cognitive styles, which are closely related to the correlation measurement, are presented in Section VIII. Lastly, conclusions and future directions can be found in Section IX.

II. Related Work

Our research involves detecting malicious insiders via analyzing the actions of intelligence analysts as they perform their analysis tasks. In this section, we first survey early research efforts on insider threat detection that were inspired by research in external threat detection. Most of the methods determine anomalies by identifying *uncommon behaviors* of masqueraders [3]. Next, we provide an overview of recent approaches for detecting traitors which take contextual information into account. These approaches assume that a traitor's behavior is *irrelevant* to contextual information. Lastly, we describe a type of insider threat problem where assumptions of uncommon or irrelevant behavior no longer hold.

Masqueraders can be either internal threats or external threats to an organization, depending upon whether a masquerader is a member of an organization or not. To detect masqueraders from outside of an organization (it is often referred to as intrusion detection), *monitoring system calls* [4]–[8] is a popular approach. Many systems, such as host-based and network-based intrusion detection systems (IDS) [9] and distributed program execution monitor (DPEM) [10], use these calls as audit data and have shown to successfully prevent, mitigate, and detect various external threats. The earliest attempts to detect masqueraders as insiders are thus inspired by these approaches. For example, Nguyen *et al.* [11] proposed an experimental system called a buffer-overflow detection system that analyzed system call activities to detect internal masqueraders. In this system, two models are built to examine file access patterns: One is user oriented, and the other is process oriented. The user-oriented model does not seem to be a good candidate for insider threat detection due to large individual differences in user file access patterns. On the other hand, the process-oriented model provides better statistical results for profiling user behaviors because most processes have a fixed list of files that users can access. In conclusion, the abnormal file access activities serve as good indicators of insider attacks. Similarly, Liu *et al.* [12], [13] also assessed system call activities for insider threat detection, but they differed in the features that they used to perform detection. These features are the n-gram feature representation [4], [6], histogram-based feature representation [5], and parameter-based representation [14]. Liu *et al.* used a supervised outlier detection algorithm for anomaly detection. Both the n-gram and histogram feature representations perform close to random chance. The authors stated that the features that were effective in detecting external threats were not effective for internal threat detection. Even though both internal and external masqueraders accomplish their attacks by taking advantage of a legitimate user's identity that they have stolen, internal masqueraders have more knowledge about the organization which changes the nature of the attacks.

In addition to assessing system calls, analyzing command line traces issued by users is another popular approach to tackle the insider threat problem. Schonlau *et al.* [15] constructed a data set for general masquerader detection. The data set contains UNIX shell commands from 70 users. Among all the users, 50 users are selected to serve as intrusion targets, while the rest simulate masqueraders. Fifteen thousand commands are collected from each user over a period of time ranging between a few days and several months. Blocks of 100 commands issued by the masqueraders are randomly inserted into 50 users' command sets to simulate intrusion attacks. Many researchers

have proposed and evaluated their methods [15]–[17] using the data set proposed by Schonlau *et al.* [15]. All UNIX commands collected in this data set are *truncated*, with all flags and additional arguments stripped (e.g., `cd` is a truncated command of a full command `cd /etc/`). In order to explore whether using full commands will result in better detection performance, Maxion [18] assembled a masquerader data set based on Greenburg's data [19]. Greenburg's data contain *full* commands from 168 unpaid volunteer users of the UNIX csh system. Maxion selected 75 users out of 168 original users. Among the 75 users, 50 are treated as victims, and 25 are treated as masqueraders. Maxion reported that the hit rate based on the data with full commands is at 82% level, which is 32% higher than the highest hit rates based on data with truncated commands. Both data sets are constructed to aid the detection of masqueraders. They are useful in simulating situations with external masqueraders; however, they are not useful in terms of simulating internal masqueraders. What makes insiders powerful are their privileges, which external intruders do not possess. Thus, the insider attacks are fundamentally different from external attacks. Unfortunately, the injected malicious actions in the data sets do not capture such differences.

Document access activity [20] is another popular audit trace along with system calls and command line traces to detect uncommon behavior of masqueraders. Both system calls and command line traces are usually chosen for audit data on the Linux/UNIX platform due to their clean auditing mechanisms. However, it is not feasible to apply the detection methods that rely on these two types of audit traces from one platform to another directly. Thus, platform-independent approaches are proposed by researchers. Yang and Tzi-cker [21] implemented a display-only file server (DOFS) that employed a remote display mechanism to prevent information leaks. All sensitive material is stored on centralized servers and cannot be stored on local computers. The DOFS restricts user actions so that the users can only read documents using the applications on the centralized servers. Suranjan *et al.* [22] designed security policies that allowed users to share documents with others who had designated access privileges. Both the DOFS and security policy approaches are useful in insider threat prevention and mitigation but not so useful in terms of detecting insider threat.

In summary, analyses of command line traces, system calls, and document access activities are the most popular approaches inspired by research in external threat detection for solving the insider threat problem. The general idea behind these approaches focuses on profiling the accessible observables of users in order to detect possible misbehavior. These observables are chosen because they capture most of the activities on the computer that are related to attacks and reflect user behaviors as well. Certain behavioral patterns are expected to be learned from normal users so that outliers can be determined by identifying mismatches. It may seem intuitive that external and internal masqueraders convey similar behavior as they both intend to steal an insider's identity. However, it has been shown that the user-profiling approaches are not effective when applied to solve the insider threat problem. As opposed to external masqueraders, internal masqueraders have more advantages when launching malicious attacks. The indicators of external masqueraders may no longer be effective when dealing with internal masqueraders. For example, the most common attacks by external masqueraders are buffer-overflow attacks. They induce buffer-overflow errors in order to invoke malicious programs. In contrast, an internal masquerader can easily invoke malicious programs using other insiders' computers while they are away without the need to induce buffer-overflow errors. In addition, an insider can launch an attack without the need to steal another's identity (we define this type of malicious insiders as traitors). Detecting a traitor is unique to the insider threat problem and requires separate treatment. A traitor may exhibit legitimate behavior while still perpetrating malicious actions. User-profiling approaches that detect masqueraders are generally not effective in detecting traitors due to this difference.

To respond to this challenge in detecting traitors, researchers have started to take the *context* of insiders into consideration. While we may not know anything about external intruders (their identities, intent, or even where they are), we can leverage the contextual information of insiders for insider threat detection. The contextual information can be task-specific information about an insider's information access events, content of the accessed information, and communication with other insiders. Maloof and Stephens [23] proposed detecting suspicious activities that were out of an insider's scope of assignments. They focused on analyzing relevance of information access events by tracking information-use events and determining volumetric anomalies, suspicious behaviors, and evasive behaviors based on carefully implemented detectors. Natarajan and Hossain [24] and Symonenko *et al.* [25], [26] also aimed to detect such malicious insiders. Natarajan and Hossain examined whether an insider has irrelevant access to other insiders and resources, while Symonenko *et al.* focused on whether an insider has access to irrelevant textual documents. They analyzed semantics in textual observables where all observables, such as e-mails, logs, and reports, are called *on topic* if they are relevant to an insider's current assignments. Therefore, they trained a clustering model based on the known on-topic documents and assessed whether documents being accessed or created by malicious insiders were significantly far from the on-topic clusters. Natarajan and Hossain suggested building a network model consisting of analysts, roles, and resources as nodes and expected relationships as edges. Unfortunately, the relevance of a data resource to an insider given the knowledge of his current job assignment is not straightforward to determine. Furthermore, employees usually switch from task to task, and these tasks often correspond to different roles in the same organization. Park and Ho [27] introduced the composite-rule-based monitoring approach that assigned different rules to an insider when they were working under different roles.

Taking contextual information into account helps counter traitors by detecting whether their activities are relevant to what they are supposed to be doing. However, a malicious insider can still carry out attacks while behaving legitimately and relevantly. A typical case of such a malicious insider is an analyst working on an intelligence analysis task. He/she aims to manipulate the perceptions of others, particularly the decision makers, by producing reports with false statements

clerics; 0.500000
friday_prayer; 0.500000
prayer; 0.500000
two_month_imar; 0.250000
cooperation; 0.250000
imposed_sanction; 0.250000
uranium_enrichment_program; 0.250000
imar_deadline; 0.250000

Fig. 1. Example of interests set.

and misleading information. It is challenging to capture him/her due to two reasons. First, depending on the scope of the given tasks, it may be very difficult to determine relevance. For example, a task such as "assess the likelihood that country X will participate in a financial bailout for country Y" is very broad and covers a large range of topics. Second, malicious insiders can insert fabricated evidence, deliberately hide critical information, and modify existing evidence, which are all very subtle ways to conduct information manipulation to deliver altered information. Inside an IC, analysts are relied upon to analyze critical situations. They have privileges to access sensitive materials, and their reports have direct impact on decision-making processes. Upon becoming malicious insiders, their attacks can cause both irreversible short-term and unnoticed long-term damages to the IC. Unfortunately, none of the current approaches, such as capturing internal masqueraders who gain access to protected resources or identifying traitors who access irrelevant resources, can directly point to the intentional manipulation of information instead of looking for the cues of malicious action. Therefore, we are motivated to find psychological indicators that may help us acquire more insights into a malicious insider's mind. Detailed discussions and the advocates for psychological indicators to tackle the insider threat problem can be found in [28] and [29].

III. Background

In this section, we provide the description of the model, referred to as the IPC model, which is a base to store and reason over a user's perceived information and analyzed results.

A. IPC User Model

The IPC user model [30]–[32] is designed to capture users' past and present behaviors and to predict users' future behaviors. Interests set (I), Preferences Network (P), and Context Network (C) are the three components of the IPC model. The *Interests* component captures a user's focus (or short-term interest). The *Preferences* component captures how a user makes decisions given alternative choices. The *Context* component provides insight into a user's knowledge base. Details of the *IPC* user model implementation can be found in [33]. Examples of an interests set and a context network are shown in Figs. 1 and 2, respectively. In this paper, we build context networks as representations of analysts' knowledge bases as a basis for consistency computations.

The Context Network is represented as a directed acyclic graph (DAG) and constructed from documents that are used in the analytic process, such as accessed documents, written

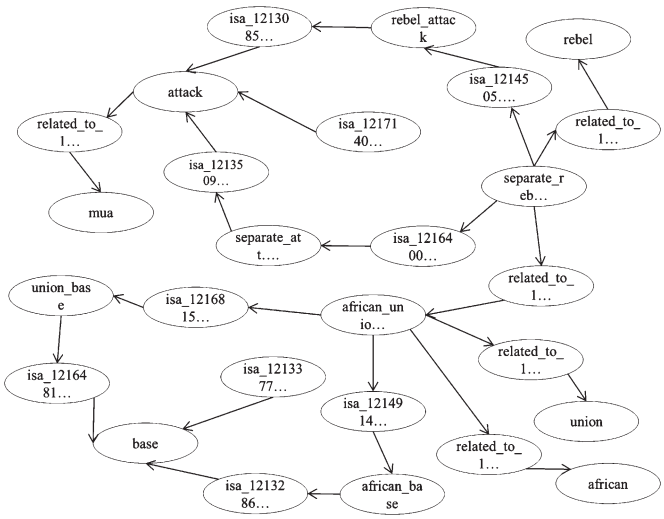


Fig. 2. Example of context network.

reports, and so forth. These documents will be converted to a special type of DAG called a document graph (DG), and we use the DGs to update the Context Network accordingly. Old nodes in the Context Network are faded out once they are not encountered in documents after a period of time. We use Link Parser [34] for processing textual content as a basis for constructing DGs.

B. DG

There are two types of nodes in a DG: *concept nodes* and *relation nodes*. A concept node represents a noun or a noun phrase, and a relation node represents a relationship between two concept nodes. Two kinds of relation node are defined—the "Is a" relation and the "Related to" relation. An "Is a" relation denotes a set–subset relation between two concept nodes which is generated based on a "Noun phrase heuristic." A "Related to" relation links concepts in a sentence according to a "Sentence heuristic," a "Noun phrase heuristic," and a "Prepositional phrase heuristic." Details of these heuristics can be found in [32].

Fig. 3 shows a DG constructed from the sentence "Aya leads Friday Prayer." Two concept nodes "Aya" and "Friday Prayer" are linked by a "related to" relation node, while the concept node "Friday Prayer" has an "is a" relationship with "Prayer." The main reasons for having two types of nodes and relations are twofold. First, noun phrases are content words, and therefore, concept nodes capture and represent the main topic of a text. Second, we aim at a robust method to generate a DG automatically. Therefore, two types of relation nodes allow us to avoid the intractability of the process of understanding natural language semantically while providing sufficient relations between main concepts. DG representation has been used and evaluated in improving a user's performance in information retrieval [30]–[33].

Various similarity measures can be used to compare two DGs, such as the Dice coefficient, the Jaccard coefficient, the cosine similarity coefficient, and so forth [35]. The method that we use in this work is modified from [36]. We essentially

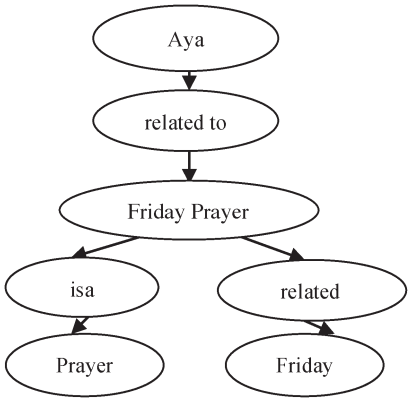


Fig. 3. Example of a DG.

check to see how much of one DG is contained in another DG. This method gives us similarities between zero and one, with one meaning identical and zero meaning totally different. The similarity measure follows:

$$Sim(DG_1, DG_2) = \frac{n}{2N} + \frac{m}{2M} \tag{1}$$

where n is the number of concept nodes shared by DG_1 and DG_2 and m is the number of relation nodes shared by DG_1 and DG_2. N is the total number of concept nodes in DG_1. M is the total number of relation nodes in DG_1.

IV. Insider Threat Detection

We propose a detection method that examines the level of each analyst's consistency across different tasks. We begin by contrasting "what analysts do" (actions) and "what analysts conclude" (conclusions). In the IC, a typical intelligence analysis task is to analyze a critical issue. To fulfill this goal, analysts take actions such as searching with queries, reading documents, and saving information of interest. In order to reduce cognitive biases, they may adopt different analytical methodologies, such as analysis of competing hypotheses (ACH) [37], while analyzing the issue. Their conclusions are presented in the assessment reports along with a list of supporting evidences. A correlation measurement between perceived information through a series of actions and the report indicates how much of the perceived information is covered in the report. In other words, the correlation value represents the level of dependence of the report on the perceived information. As normal actions tend to follow one's habitual working style, it is intuitive that one's dependence of his report on the perceived information should show consistencies for different tasks. In contrast, malicious actions are taken to form an attack which may break the consistencies maintained with a habitual working style. In Section IV-A, we study whether the common methods that seek uncommon or irrelevant behavior can tackle the type of insider threat problem that we are aiming at. In Section IV-B, we list all the hypotheses on how to differentiate between malicious and normal actions. The detection method is described step by step in Section IV-C.

TABLE I
Examples of Feature Extraction for KNN Method

Action Sequence	Feature Vectors (5-gram)
1 2 3 2 1 2 4	[1 2 3 2 1] , [2 3 2 1 2], [3 2 1 2 4]
1 2 3 2 1 **2(Hide)** 4	[1 2 3 2 1], **[2 3 2 1 4]**
1 2 3 2 1 **2(Fabricate)** 4	[1 2 3 2 1], **[2 3 2 1 2]**, **[3 2 1 2 4]**

TABLE II
Classification Results for KNN and One-Class SVM Methods

	#True Negative	#False Positive	#True Positive	#False Negative
KNN	1590	1933	133	62
One-Class SVM	1213	88	0	14

TABLE III
Recall, Precision, and F-Score for KNN and One-Class SVM Methods (Percentage Scale)

	KNN	One-Class SVM
Recall	68.2051	0
Precision	6.4376	0
F-Score (β=10)	36.4293	0

A. Comparisons With Other Detection Methods

We implement two common approaches found in insider threat literature [12], [13], [25], [26] to verify our conjecture that these approaches might not be effective in tackling the type of insider threat problem that we are aiming at. Both approaches are applied to the APEX '07 data set, the details of which can be found in Section V. The first approach determines insider threat by seeking uncommon behaviors. A K-nearest neighbor (KNN) method is used to classify whether a sequence of five actions is malicious or not. We extract consecutive five-gram action sequences with a sliding window over each analyst's entire action sequence. A five-gram sample is marked as malicious if any of the five actions is malicious. There are three types of malicious actions in the APEX '07 data set. Actions whose type is *Fabricate* refer to the ones that are carried out by a malicious insider but not by his paired legitimate insider. In contrast, actions whose type is *Hide* refer to the ones that are carried out by a legitimate insider but are deliberately skipped by the paired malicious insider. Finally, actions whose type is *Modify* refer to the ones that are carried out by both, but the content of the actions differs. Examples of feature extractions by KNN are shown in Table I. Because a malicious action with type Hide is not carried out by a malicious insider, a five-gram sequence enclosing a Hide action is marked as malicious, but the action itself is excluded in the feature vector. For instance, the extracted feature vector [2 3 2 1 4] in the second row denotes a positive sample, but the sample excludes the Hide action. The other types of malicious actions, such as fabrication and modification, are not excluded (see third row). All the positive samples (five-gram feature vectors) in the training set denote malicious behavior, while negative samples denote nonmalicious behavior. The results of KNN are shown in Tables II and III. Here, a true negative result means that a negative sample (nonmalicious behavior) is classified correctly as negative, while a false negative means that a positive sample

is misclassified as negative. Similarly, a true positive case means that a positive sample (malicious behavior) is classified as positive, while a false positive means that the sample itself is negative. We measure the performance of the detection method using the recall, precision, and F-score metrics depicted in the following:

$$Recall = \frac{\#\,True\ Positive}{\#\,True\ Positive + \#\,False\ Negative} \quad (2)$$

$$Precision = \frac{\#\,True\ Positive}{\#\,True\ Positive + \#\,False\ Positive} \quad (3)$$

$$F\text{-}Score = \frac{(\beta+1) * Recall * Precision}{\beta * Precision + Recall} \quad (4)$$

where β is the weight between precision and recall. Here, we let $\beta = 10$ so that recall weighs ten times as much as precision. As shown in Tables II and III, more than half of the nonmalicious action sequences in the test set are mistakenly classified as malicious, due to which the precision of the KNN method is very low (6.4376%). Finally, the low F-score indicates that the KNN method is not effective in insider threat detection for the APEX '07 data set because manipulation of information can be hidden in nonmalicious actions.

The second approach determines insider threat by seeking information-search behavior that is irrelevant to the topic. The textual information obtained during each action is converted into a feature vector composed of the frequencies of the words from a dictionary that we constructed beforehand. We then implement a one-class support vector machine (SVM) to classify the information as on topic or off topic. The one-class SVM method focuses on semantic content of each action, while the KNN method focuses on the type of each action. Here, the one-class SVM is chosen due to unbalanced data between the numbers of malicious and nonmalicious actions. In this data set, most of the malicious actions are to hide critical information that does not support the opinion that a malicious insider attempts to deliver. Therefore, there are no test samples generated for malicious actions with type Hide due to the fact that these actions are not carried out by a malicious insider. As a result, the training set for the SVM only contains 14 malicious actions which are either with type Fabricate or with type Modify. Out of these 14 positive samples, none of them is classified correctly. The recall, precision, and F-score of this method are all zero. This is reasonable because both the fabricated information and the modified information are used to deliver a different opinion but the semantic content of the information is still on topic. In this case, the one-class SVM method is not effective in tackling the type of insider threat that we proposed because malicious actions can be both on topic and manipulative.

B. Hypothesis

Hypothesis 1—An Analyst's Correlation Measurements Between Normal Actions and Reports Are Similar for Different Tasks: The actions that analysts take vary greatly from task to task. However, we believe that one's working style is an intrinsic characteristic that is unique and stable. For instance, when tackling an analytical question, some analysts prefer dividing a question into smaller pieces and then tackling them one by one. Other analysts may prefer balancing their understanding of different topics and studying many topics at the same time. By stability of one's working style, we mean that it rarely changes in any rapid fashion over time. In other words, any change in working style is slow and/or deliberative. Thus, we hypothesize that the correlations between normal actions and their reports should be similar for different tasks performed in a relatively short time period. In other words, the discrepancy between two correlation values, each computed from one task, should be as small as zero. However, various factors, such as an analyst's task assignments, working environment, collaborative communications, and task deadlines, may influence the consistency of such correlation values. In order to effectively evaluate whether one's behavior is consistent or not, we contrast consistency values for a group of analysts.

Hypothesis 2—An Analyst's Correlation Measurements Between Malicious Actions and Reports Are Dissimilar for Different Tasks: Compared to normal actions, malicious actions are deliberately designed for the purpose of launching malicious attacks. Various types of malicious actions are described in Section V. Because working styles are used to describe patterns found in habitual behaviors, malicious actions are not habitual behaviors and, thus, cannot be explained by one's style. Therefore, we hypothesize that the correlations between malicious actions and the assessment reports are dissimilar for different tasks. When we compare two correlation values, each computed from one task, the discrepancy between them highlights the inconsistency due to malicious actions. If an analyst performs legitimately for one task but performs maliciously for the other task, high discrepancy would reveal the existence of malicious intent. If an analyst is malicious for both tasks, a high discrepancy is still expected to occur due to inconsistency between malicious actions for two tasks.

Hypothesis 3—Higher Inconsistency of an Analyst When Compared Against the Average of All Other Analysts' Inconsistencies Is an Indicator of an Insider Threat: According to the previous hypotheses, the inconsistency of a legitimate analyst should be close to zero, while the inconsistency of a malicious analyst should be significantly nonzero. In order to determine whether an analyst should be suspected of being a malicious insider, we compare his inconsistency value with all other analysts' values. If the value exceeds the threshold which is the averaged inconsistencies of others, we hypothesize that it is an effective indicator of malicious actions.

Hypothesis 4—Normalizing an Analyst's Discrepancy Value Improves the Detection Performance: Our preliminary results [2] show that low correlation values are more likely to produce small discrepancy values and *vice versa*. We want to eliminate the effect of correlation values on the level of inconsistencies in order to improve detection performance. A normalization procedure is thus designed to divide an analyst's discrepancy value by his global correlation value. The global correlation value is the similarity between the textual content of all the actions and the union of reports for all tasks.

C. Detection Method

Our detection method measures the level of inconsistency between two tasks. In particular, we begin with correlation computation for each task. Then, a discrepancy value between these two correlations is calculated to represent an insider's level of consistency. Each discrepancy value is compared against those of all other analysts' to determine malicious insiders. Lastly, we normalize all the discrepancy values to minimize individual differences. The details of the detection methods are illustrated in the following five steps.

Step 1) Organizing the tasks. We denote the ith task as T_i, R_{ij} as analyst j's assessment report for T_i, and U_{ij} as analyst j's user model for T_i.

Step 2) Correlation computation. As introduced in Section III, we build a user model based on the observed actions. The user model aims to capture the dynamics of an analyst's interests, contextual knowledge, and preferences over time for each task. Because the context network represents an analyst's knowledge base when conducting information-seeking actions, we use the context network to represent an analyst's perceived information via a series of actions. The correlation between one's actions and his report is thus measured as the similarity between the context network and the DG created from the report. Because the context network is also in the form of a DG, the correlation measurement turns out to be a similarity measure between two DGs. We compute $Sim(R_{1j}, U_{1j})$ and $Sim(R_{2j}, U_{2j})$ for each analyst j over the two tasks.

Step 3) Discrepancy computation. As stated in *Hypothesis 1*, we hypothesize that two correlation values, one for each task, should be consistent for normal actions. Thus, the smaller the discrepancy between two correlations, the more consistent an analyst is. The discrepancy value of each analyst j serves as the key indicator of malicious actions, which is calculated as

$$Diff_j(T_1, T_2) = |Sim(R_{1j}, U_{1j}) - Sim(R_{2j}, U_{2j})|. \quad (5)$$

Step 4) Normalization of discrepancy values. If two correlations are low, the discrepancy value tends to be low and *vice versa*. As a result, high discrepancy values due to high correlation values may result in false positive alarms. In order to minimize the impact from the correlation values on the discrepancies, we first compute a global correlation value $Sim(R_j, U_j)$ between the joint user model and joint report of two tasks where $R_j = R_{1j} \cup R_{2j}$ and $U_j = U_{1j} \cup U_{2j}$. The normalized discrepancy value between T_1 and T_2 is calculated as

$$Norm\_diff_j(T_1, T_2) = \frac{Diff_j(T_1, T_2)}{Sim(R_j, U_j)}. \quad (6)$$

Step 5) Anomaly detection. In order to determine whether a discrepancy value is large enough to suspect an analyst, we calculate the averaged discrepancy value of all other analysts as a baseline value. We identify an analyst to be malicious if his discrepancy value satisfies the criteria shown in the following:

$$Norm\_diff_i(T_1, T_2) > \frac{1}{n-1} \sum_{j \neq i}^{n} Norm\_diff_j(T_1, T_2). \quad (7)$$

V. Test Bed

Our test bed for insider threat detection is constructed based on the APEX '07 data set. The APEX '07 data set was collected by the National Institute of Standards and Technology in 2007. The purpose of the experiment was to evaluate the Intelligence Advanced Research Projects Activity (IARPA) Collaboration and Analyst/System Effectiveness (CASE) program's tools. Eight analysts participated in the experiment, and all of them worked on an analysis task independently. It is important to mention that each analyst was required to conduct the task following the ACH method [37]. The experiment consisted of the following three different stages:

1) problem assignment;
2) information gathering;
3) report production.

Stage 1—Problem Assignment: Each analyst was asked to assess two hypothetical problems with regard to the possible development of a nuclear program in Imar. We refer to the first problem as Q_1 and the second problem as Q_2 (proper names have been removed/replaced for the purposes of our discussion).

Q_1: "Where does the Imar's clerical community stand on Aya and President Amar's policies with regard to Imar's civilian and military nuclear program?"

Q_2: "Are there fissures in the clerical community, and do they represent a deepening divide among the clerics loyal to the Imar's revolution?"

Each analyst was also given a document with scripted questions to answer. The purpose of the scripted questions was to evaluate the analysts' overall understanding of the problems. (For example, how many Grand Aya live in Imar? What are their names?)

Stage 2—Information Gathering: After the problems were assigned, the analysts started searching for information with queries and evaluating the retrieved documents. This is referred to as the information-gathering stage. The goal of the analysts is to get a better understanding of the involved topics, form analytical conclusions, and collect evidence to support their opinions. Analysts' activities are recorded and saved in the form of analysis log events (ALEs). The four types of ALEs used in this paper are as follows.

1) *Search* ALEs that contain search queries.
2) *Retain* ALEs that contain the documents or the snippets of documents saved by the analysts.
3) *Access* ALEs that contain the documents read by the analysts.
4) *Delete* ALEs that contain the documents deleted after they are saved.

Other types of ALEs, such as Start Application ALEs, Assess ALEs, Make Hypothesis ALEs, and Associate Evidence ALEs, are not used because they are either not fully implemented or not contributing to the main goal of this study. Examples of ALEs and detailed statistics of the APEX '07 data set are shown in Tables IV and V.

Stage 3—Report Production: Analysts produced two reports in the third stage: an assessment report and a final report. In the assessment reports, they first provided their analytical decisions/recommendations, entertained the two problems from different perspectives, and finally listed all the evidence that they gathered during the second stage organized in an ACH matrix. The ACH matrix is a table containing pieces of evidence each associated with a rating. An example of an ACH matrix is shown in Table VI. When an assessment report is converted to a DG, only the evidence descriptions in the ACH matrices are included, but the scores and URLs are excluded. Our detection method does not assess either analysts' opinions or the quality of the reports. Analysts may be biased when they collect and evaluate evidences. As long as they are biased consistently across tasks, they are still considered to be legitimate. The final report includes the answers from analysts to the scripted questions. Because the scripted questions are tailored for the CASE program only, these final reports do not represent the reports that analysts typically produce in a realistic situation. As such, we only consider the assessment reports in this paper.

Design of Malicious Insiders: In order to evaluate our detection method, we simulated five malicious insiders, each based on one of the original eight analysts. Out of the five malicious insiders, three are categorized as *expert* malicious insiders, two of which were created by Ph.D. students and the third of which was created by an assistant professor. Each member independently came up with a detailed scenario, including the motivation of the attack (Wood [38] lists four major motivations: profit, provoke change, subversion, and personal motive), what is the alternative conclusion to be drawn in the report, and the malicious actions that each would take to accomplish the attack. Our goal was to provide a clear strategy in simulating the malicious insiders as realistically as possible. The other two malicious analysts are categorized as *novice* malicious insiders, each of which was constructed by an undergraduate intern. The novice insiders' behaviors are overt and can be detected through human observation, while the expert insiders conduct actions that are more subtle and more difficult to detect. During the creation of the malicious insiders, every one was allowed to take any type of malicious action that helped him/her deliver altered information to the readers of their reports. Malicious actions are simulated during both the information-gathering and report-production stages.

We summarize the types of malicious actions that were used to simulate the malicious insiders:

1) misrepresentation
   a) fabrication of evidence;
   b) writing of false statements;
2) omission
   a) use of more supporting queries than nonsupporting queries;
   b) use of more constraints on nonsupporting queries;
   c) ignoring nonsupporting documents;
3) irrelevance
   a) use of outdated documents when supporting documents are not sufficient;
   b) use of irrelevant queries;
4) exaggeration
   a) overcitation of the same evidence;
   b) exaggeration of evidence ratings in the ACH matrix.

We simulate malicious insiders based on legitimate analysts to guarantee that all analysts have similar contexts. More specifically, malicious insiders should work on the same problems with similar specifications (e.g., the same total time allowed for task completion), access the same database, and have the same organizational context as other analysts. This allows us to avoid introducing uncommon or irrelevant behavior into the data set. After publishing our preliminary results in [2], we noticed that some oddities were introduced during the construction of malicious insiders. If the creator of a malicious insider removes a search event from the paired legitimate insider's data, it simulates a malicious action that the malicious insider attempts to ignore some documents. Therefore, the consequent events to read retrieved documents returned by the search query should also be removed consistently. Unfortunately, we found some inconsistencies in the data of a few malicious insiders. We revisited the data set and fixed the occurrences of these oddities.

We note that there are two major artifacts that we needed to introduce with regard to the APEX '07 data set in order to study our approach. The first artifact is the simulation of two tasks by splitting each assessment report into two (sub-) task reports. As discussed earlier, we hypothesize that assessing an analyst's level of consistency from task to task helps reveal misbehavior. However, a data set that collects data for multiple tasks has been extremely rare. In real-life situations, it is possible that an analyst launches malicious attacks for his very first assignment where no previous profile has been stored. It is also possible that an analyst might work on multiple tasks from time to time (intertwining them) which make it hard to determine which task each action belongs to. As such, the detection method needs to be flexible in terms of the amount of data available for detection.

In an analytical process, analysts often must assess several problems. Sometimes, these are subproblems of an overall larger problem (as in APEX '07), multiple independent problems, or some combination of the two [37]. During the information-seeking stage, each analyst often seeks information with all or some subset of the problems in mind. The problems are evaluated and analyzed, respectively, in their report(s). Since we are working with APEX '07, in order to address these issues, we propose to separate out the two subtasks (questions) from the data set. (Recall that the two questions (subtasks) are conducted simultaneously by the analysts and a single assessment report is produced.) *This allows us to actually detect malicious insiders from just a single task/analysis session as long as we can identify subtasks*. Another advantage of splitting a task is that it is easier to study the effects of one's style with the minimal influences from the group assignments, task

TABLE IV
STATISTICS OF ANALYSTS IN APEX '07 AND SIMULATED MALICIOUS ANALYSTS

Analyst	Num of ALEs	Paired Malicious Analyst	Num of ALEs	Task Begins At	Task Ends At
APEXB	642	N/A	N/A	2007-12-10	2007-12-14
APEXF	482	APEXF NOVICE	482	2007-12-07	2007-12-14
		APEXF EXPERT	482	2007-12-07	2007-12-14
APEXK	762	APEXK EXPERT	762	2007-12-07	2007-12-14
APEXC	896	N/A	N/A	2007-12-10	2007-12-14
APEXH	535	APEXH NOVICE	535	2007-12-07	2007-12-14
APEXL	614	N/A	N/A	2007-12-07	2007-12-14
APEXE	474	N/A	N/A	2007-12-07	2007-12-14
APEXP	548	APEXP EXPERT	587	2007-12-07	2007-12-14

TABLE V
EXAMPLE OF ALEs (ABRIDGED)

ALE Type	Num of ALEs	Example
Search	903	Description: verb='execute', noun='query', cause='mouse', location='TRIST', context='LCC PowerAnswer', parameters='', element='How many Grand Ayas live in Imar', neighbour='' User: APEXE Report time: 2007-12-07T15:44:49.611-05:00 Search string: How many Grand Ayas live in Imar
Access	2386	Description: verb='open', noun='document', cause='mouse', location='TRIST', context='', parameters='', element='495cce1bX11635035eb9XY69d6', neighbour='' User: APEXE Report time: 2007-12-07T15:45:49.407-05:00 Accessed: 495cce1bX11635035eb9XY69d6
Retain	548	Description: verb='save', noun='snippet', cause='keyboard', location='Sandbox', context='', parameters='http://129.6.84.47:18080/cmsRest/cms_rest?id=495cce1bX11635035eb9XY2978', element='FragmentIcon …" User: APEXE Report time: 2007-12-11T11:20:28.933-05:00 Resource: Criticism of Amar On January 22, 2007, Grand Aya Mri criticized Imar's President Amar for his nuclear and economic policies[35][36] even though Amar does not formulate Imar's nuclear policy. The Imar's National Security Council determines the nuclear policy. Its decisions must be approved by the Supreme Leader according to Imar's constitution and the president is only a member of the Council. While agreeing that nuclear energy is Imar's right, he criticized Amar's aggressive approach to the issue, saying, "One has to deal with the enemy with wisdom, not provoke it, ... his (provocation) only creates problems for the country."[36] …
Assess	123	Description: verb='changeRelevance', noun='document', cause='mouse', location='TRIST', context='documentDimension', parameters='Maybe Relevant', element='495cce1bX11635035eb9XY4ec0', neighbour='' User: APEXE Report time: 2007-12-11T11:27:08.168-05:00

TABLE VI
EXAMPLE OF AN ACH MATRIX

Evidence	Source	Q_1	Q_2
Prayers and fatwa dictated by a council	url	+2	-2

assignments, and working environment. If data for multiple tasks are available, the detection methods can be directly applied to examine one's consistency.

Thus, in terms of the APEX '07 data set, each analyst is asked to assess two problems that concern the development of a nuclear program in Imar. We split every analyst's assessment report into two component reports manually, each of which contains everything reported on one subproblem/question. The conclusion/analysis of each problem is presented in separate sections/paragraphs in the assessment report. Therefore, it is straightforward to split. In terms of the actions, it is not as clear which problem is being addressed by any single action. Therefore, we take the conservative approach where the actions are duplicated rather than split. Moreover, splitting all actions into two components may also unduly impact the patterns of working style. As such, all the U_{ij}'s are the same for each analyst $U_{1j} = U_{2j}$. It is reasonable to believe that each subtask still encompasses elements of a general task without introducing too many artifacts to the data set for the following two reasons. First of all, all the data of each subtask are complete from ALEs to its final conclusions. Second, two subtasks can be treated as two independent tasks with similar topics that are analyzed at the same time.

Another artifact that we have introduced is the injected activities by the creator of each malicious insider. These malicious actions are carefully designed so that they are a sound simulation of real-life situations. First of all, all malicious actions are inserted with a goal of delivering false information. In other words, we attempt to simulate not only the malicious actions but the malicious intent of a possible malicious insider as well.

Second, there is a potential danger that the detection method detects a malicious insider because the injected activities capture the style of the creator rather than the style of the original analyst. In Table X, we have shown that analysts who are malicious for both tasks still show significantly high inconsistency. Thus, even though the injected activities belong to the creator rather than to the original analyst, the malicious actions are shown not to follow either the creator's or the original analyst's style.

VI. Hypotheses Verification

In this section, we conduct various statistical analyses based on the APEX '07 data set to verify the four hypotheses described in Section IV.

Hypothesis 1—An Analyst's Correlation Measurements Between Normal Actions and Reports Are Similar for Different Tasks: An analyst follows his/her working style while conducting different tasks. Thus, we hypothesize that the measured correlation values for normal actions should be similar for an analyst. Each analyst completed two subtasks; therefore, we have two variables, each of which represents the correlation value between the report and the context network for one subtask. In order to verify Hypothesis 1, we compute the correlation coefficient of these two variables as well as the mean difference. The correlation coefficient of these two variables is 0.9883 ($Sig. < 0.0001$ using a T-test) which indicates that the two variables are significantly correlated. The mean of paired differences between the two variables is 0.0098, which shows that the two variables are not only dependent but are very similar to each other as well. Hypothesis 1 is thus verified. The paired difference between these two variables is likely to be caused by the variation in one's working style, by the fitness of the model, or by the influences from other factors, such as task assignments and working environment. Because we simulate two subtasks (T_1 and T_2) by separating out two problems (Q_1 and Q_2), we assume that the influences from group assignments, task assignments, and working environment are minimized. In order to test whether this assumption is valid, we conduct a paired T-test to test the null hypothesis: The mean of differences between these two variables is zero. We chose to conduct a paired T-test because of the significant correlation between the two variables. The null hypothesis is not discredited by the paired T-test. The two-tailed significance level is 0.4542 with 7 degrees of freedom (DOF) ($mean = 0.0098$ and $stderr = 0.0123$) which shows that there is little suspicion that the mean difference is not zero. More specifically, impacts from group assignments and such are shown to be minimized, and thus, the correlation measurements solely reflect impacts from one's working style.

Hypothesis 3—Higher Inconsistency of an Analyst When Compared Against the Average of All Other Analysts' Inconsistencies Is an Indicator of an Insider Threat: After verifying Hypothesis 1, we verify Hypothesis 3 whose results will be involved in the verification of Hypothesis 2. Hypothesis 3 proposes to compare an analyst's inconsistency value against all the other analysts' inconsistency values to determine whether an analyst should be suspected of being malicious. We verify Hypothesis 3 empirically rather than analytically because the average of all other analysts' inconsistencies varies for different group assignments. To evaluate our method, we conducted three experiments based on the APEX '07 data set. The first evaluation demonstrates that our detection method succeeds in detecting four malicious analysts out of five without raising any false positive. The second experiment evaluates the detection method for all combinations of the different group members. The last experiment assesses the performance when a different analytical strategy is adopted. Our detection method demonstrates promising performance in all the evaluations. Details of all the experiments are separately presented in Section VII as a thorough performance evaluation of the detection method.

Hypothesis 2—An Analyst's Correlation Measurements Between Malicious Actions and Reports Are Dissimilar for Different Tasks: As Hypothesis 1 states, a legitimate analyst performs consistently from task to task which can be explained by a stable working style. The verification of Hypothesis 3 shows that an analyst no longer performs consistently when he/she carries out malicious actions. Hypothesis 2 concerns the source of inconsistency for malicious analysts. Here, we study the nature of malicious actions. If an analyst only launches an attack during one subtask, it is intuitive that the existence of malicious actions of that subtask produces inconsistency between two subtasks. However, if an analyst carries out attacks for both subtasks, can he still be caught? This is important to address because the malicious analyst might have already succeeded in several sabotage attempts without being caught. If an insider threat detection system profiles an analyst who is always malicious, it is critical that the system should still be able to detect his/her malicious intent. The design of our malicious insiders allows us to quantify how much a correlation value has changed due to malicious actions because each simulated malicious insider is built based on a legitimate analyst. Equation (8) calculates the impacts of malicious actions, and (9) calculates how inconsistent the impacts of malicious actions are between two tasks. Such inconsistency value is caused by malicious actions. It is intuitive that the correlation value for malicious actions is zero if an analyst does not conduct malicious actions for that subtask. Results of correlation values for malicious actions are presented in the second and third columns in Table VII. The information about whether an analyst conducted attacks in each subtask is shown in the fourth and fifth columns. In (8), shown below, j represents a malicious insider, k represents the paired legitimate insider of j, and i represents subtask T_i:

$$Mal(R_{ij}, U_{ij}) = Sim(R_{ij}, U_{ij}) - Sim(R_{ik}, U_{ik}) \quad (8)$$

$$Diff\_mal_j(T_1, T_2) = |Mal(R_{1j}, U_{1j}) - Mal(R_{2j}, U_{2j})|. \quad (9)$$

We want to study whether the existence of malicious actions has made changes to one's correlation value for a subtask that cannot be explained by natural variance in correlation values. Therefore, we carry out a single-value T-test for each inconsistency due to malicious actions. The null hypothesis is that the discrepancy of malicious actions is zero. We use the standard deviation (stdev) of paired differences for normal actions as an estimate of the stdev of paired differences for malicious actions. The results in Table VII show that the discrepancy values for all malicious insiders are significantly nonzero

TABLE VII
CORRELATION VALUES FOR MALICIOUS ACTIONS AND RESULTS FOR T-TESTS

Analyst	$Mal\ (R_1,U_1)$	$Mal\ (R_2,U_2)$	Malicious for T_1	Malicious for T_2	$Diff\_mal_j\ (T_1,T_2)$	Sig. (2-tailed)
APEXK EXPERT	0.031620	0.006149	YES	NO	0.025471	>0.2
APEXH NOVICE	-0.211083	0	YES	NO	0.211083	<0.0001
APEXP EXPERT	-0.299371	-0.395605	YES	YES	0.096233	<0.05
APEXF NOVICE	-0.260195	-0.511410	YES	YES	0.251215	<0.0001
APEXF EXPERT	-0.144431	-0.370026	NO	YES	0.225596	<0.002

except for APEXK EXPERT. Their discrepancy values are large enough that they cannot be explained by natural variation in discrepancies between two tasks. The high inconsistency values imply that one does not consistently carry out malicious actions from task to task. In other words, malicious actions are not likely to follow one's working style. This is an important feature that differentiates normal actions from malicious actions. Our results also show that APEXK EXPERT's discrepancies are not large enough to reject the null hypothesis. The descriptive statistics of the paired differences as well as the individual correlation values reveal some possible reasons. First of all, the stdev of paired difference (0.0349) is much larger than the mean difference (0.0098). This indicates that the discrepancy values have large spread while the values themselves are comparatively small. Furthermore, the correlation values for two tasks have impacts on their differences. Large correlation values (e.g., APEXF NOVICE) tend to produce large difference values, while small correlation values tend to produce small difference values (e.g., APEXK EXPERT). Without reducing large individual differences, the detection method may fail to determine malicious intent due to low inconsistency values. APEXK EXPERT is an example of the first situation. APEXK EXPERT carried out malicious actions in T_1 but did not in T_2. However, $Mal(R_1,U_1)$ tends to be small (0.0316) because $Sim(R_1,U_1)$ for APEXK EXPERT is relatively small (0.2149) when compared to the averaged correlation value of 0.4420 for subtask T_1. It is hard to tell whether such small value is caused by malicious actions or can be explained by reasonable variation. On the other side, the detection method may capture legitimate insiders due to their high inconsistency when this analyst's report and user model have a high correlation. The problem of large individual difference motivates us to apply a normalization procedure which we believe will improve the detection performance. We will show in the verification part of Hypothesis 4 that the detection rate increases after all the correlation values are normalized.

In order to assess whether there is significant difference between the correlation values for two subtasks, we also conducted a paired T-test. We use the standard error of the mean difference computed during the verification of Hypothesis 1 as an estimate of the standard error of the mean for malicious actions. The null hypothesis is that the mean difference is zero. The standard error of the mean difference which is 0.0123 is an estimate for σ_1

$$t_1 = \frac{\overline{d_1} - \delta_1}{\sigma_1} = 13.1269 \text{ with } 5 \text{ DOF.} \tag{10}$$

The null hypothesis is thus rejected $(Sig. < 0.0001)$.

TABLE VIII
RESULTS OF T-TESTS FOR NORMALIZED VALUES

Analyst	$Norm\_Diff\_mal_j\ (T_1,T_2)$	Sig. (2-tailed)
APEXK EXPERT	0.198396	>0.1
APEXH NOVICE	0.944427	<0.0001
APEXP EXPERT	0.457666	<0.01
APEXF NOVICE	0.545214	<0.005
APEXF EXPERT	0.463132	<0.01

Hypothesis 4—Normalizing an Analyst's Discrepancy Value Improves the Detection Performance: Verification of Hypothesis 4 involves two parts. First, we examine whether the detection method with the normalization procedure is still consistent with Hypotheses 1–3. Second, we validate whether the detection method with the normalization procedure performs better than the method without the normalization procedure. The second part of the validation is presented as part of the method evaluation section.

In terms of verifying Hypothesis 1 for the detection method with the normalization procedure, we also compute a coefficient correlation as well as the mean difference between these two variables, each of which represents the normalized correlation value for a subtask. The correlation coefficient of these two variables is 0.9414 $(\text{Sig.} < 0.001)$ which indicates that the two variables are dependent. The mean of paired differences between the two variables is 0.0156 which shows that they are very similar to each other. Thus, Hypothesis 1 still holds with the normalization procedure. A paired T-test is conducted on the normalized correlation values to contrast the results for the nonnormalized values. The null hypothesis of the paired T-test is that the mean of differences between two normalized correlations is zero. The significance value of the paired T-test is 0.6923 with 7 DOF ($mean = 0.0156$ and $stdev = 0.1068$) which indicates that there is no suspicion that the mean of differences of the normalized correlation values should be nonzero. After the normalization procedure is applied, the consistency of normal actions is still preserved, and Hypothesis 1 still holds.

In terms of Hypothesis 2, we conducted T-tests on the normalized discrepancy values for malicious actions. As discussed when verifying Hypothesis 2, APEXK EXPERT's inconsistency value is not large enough to reject the null hypothesis. Similar results are obtained with the normalization procedure (see Table VIII).

In terms of Hypothesis 3, we applied our detection method including a normalization procedure on the APEX' 07 data set. With this, our detection method captures all malicious analysts without raising any false alarms. In other words, the impacts from individual differences are reduced using the normalization

procedure, and the inconsistency caused by malicious actions is amplified. One interesting observation is that the two novice malicious insiders show more inconsistencies than the expert ones (see Table X). However, the number of malicious actions conducted by the two novice insiders (18 for APEXH NOVICE and 56 for APEXF NOVICE) is far fewer than most of the expert ones (e.g., APEXK EXPERT has 162 malicious actions and APEXF EXPERT has 99). It seems rather counterintuitive at first, but the results have shown that an inconsistency value corresponds to how much one's working style has been violated by malicious actions rather than how many actions are being carried out. In this case, an expert insider's working style may receive less influence since all malicious actions are carefully designed.

We also conducted a T-test on the mean difference. The null hypothesis is that the mean difference between normalized correlations is zero. The standard error of the mean difference for normalized correlation values which is 0.0519019 is an estimate for σ_2

$$t_2 = \frac{\overline{d_2} - \delta_2}{\sigma_2} = 9.1302 \text{ with 5 DOF.} \tag{11}$$

$\Pr(t_2 > 9.1302) < 0.0001$ provides strong evidence to reject the null hypothesis with only 5 DOF.

In summary, the normalization procedure amplifies the inconsistency caused by malicious actions, while it does not amplify the variation in normal actions. The detection performance is also shown to be improved.

VII. Evaluation Results and Discussions

This section evaluates the detection performance with respect to the impact of the normalization process, the impact of the group size, and the impact of the ACH matrices.

A. Detection Results

We apply the detection methods to the APEX '07 data set, and the results are shown in Tables IX and X. The performance of the detection method is evaluated using both the recall rate and the false positive rate. The recall rate is the ratio of the number of malicious insiders caught by the detection method to the total number of malicious insiders within a group of analysts. The false positive rate is the ratio of the number of legitimate analysts identified as malicious insiders by the detection method to the total number of legitimate analysts within a group.

The performance of the detection method without the normalization procedure is shown in Table IX. All of the five simulated malicious insiders are highlighted in the table with their code names (e.g., APEXK EXPERT) displayed in column 1. A discrepancy value (in column 2) is bold faced in the table if it exceeds the corresponding baseline value (the averaged discrepancy values of all other analysts) in the third column. The results show that four out of five malicious insiders are flagged by the detection method; therefore, the recall rate is 80%. We then applied the detection method again with the normalization procedure. In the results, shown in Table X, all of the malicious insiders are flagged by the detection method; therefore, the recall rate is 100%. For both cases, none of the legitimate analysts is flagged, and thus, the false positive rate is 0%. With the normalization procedure, the performance of our detection method is improved.

TABLE IX
Results for the Detection Method Without Normalization Procedure

Analyst	$Diff_j(T_1,T_2)$	Baseline
APEXF	0.037894	0.093194
APEXK	0.009944	0.095523
APEXC	0.010586	0.095470
APEXH	0.014297	0.095160
APEXL	0.071972	0.090354
APEXE	0.024999	0.094269
APEXP	0.031203	0.075127
APEXP EXPERT	**0.117560**	0.086555
APEXF NOVICE	**0.289109**	0.072260
APEXF EXPERT	**0.264563**	0.053913
APEXH NOVICE	**0.225380**	0.077570
APEXK EXPERT	0.035415	0.093401

TABLE X
Results for the Detection Method With the Normalization Procedure

Analyst	$Norm\_diff_j(T_1,T_2)$	Baseline
APEXF	0.046985	0.286531
APEXK	0.075116	0.284187
APEXC	0.015404	0.289163
APEXH	0.030603	0.287897
APEXL	0.211879	0.272790
APEXE	0.097643	0.282310
APEXP	0.034211	0.287596
APEXP EXPERT	**0.491878**	0.249457
APEXF NOVICE	**0.592199**	0.241097
APEXF EXPERT	**0.510117**	0.247937
APEXH NOVICE	**0.975030**	0.209195
APEXK EXPERT	**0.273512**	0.267654

B. Impact of Group Size

We note that, as the number of malicious insiders increases, their impacts on the averaged discrepancy may be overwhelming. Since we use the averaged discrepancy as the baseline to detect malicious insiders, the detection method may not be as effective as when malicious analysts dominate. Therefore, we examine whether the number of malicious insiders in the group impacts detection performance. Will a larger number of malicious insiders result in a lower recall rate? If the group is small, is the detection method still capable of detecting malicious insiders? We conducted an exhaustive test described as follows: We created groups consisting of all possible combinations of the malicious and legitimate analysts from one to 13 analysts. There are 13 analysts in total, so we have 8192 different groups. We applied the detection method to each group and computed the results both by group size and by the number of malicious analysts. For each group, we calculated both the recall rate and the false positive rate.

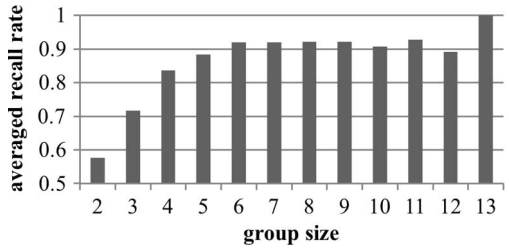


Fig. 4. Average recall rate for different group sizes with normalization procedure.

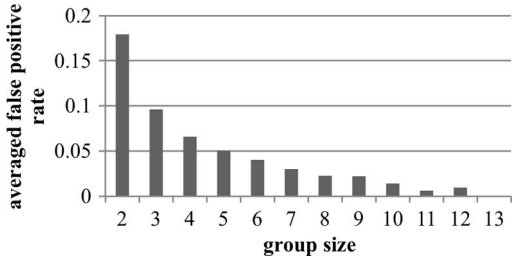


Fig. 5. Average false positive rate for different group sizes with normalization procedure.

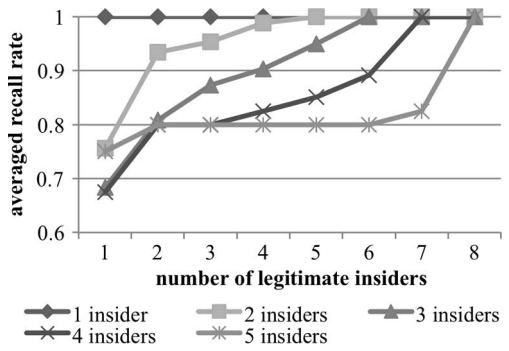


Fig. 6. Average recall rate with different numbers of legitimate and malicious insiders with normalization procedure.

As shown in Fig. 4, as the number of analysts in the group increases, the recall rate also steadily increases. When the group size is relatively small, the recall rate is not dramatically reduced. On the other hand, the false positive rate remains relatively high for a small group size (as shown in Fig. 5), but it decreases sharply as the number of analysts in the group increases and becomes as small as 0.05 when the size of the group reaches five. The results show that a larger group size helps the detection of malicious insiders and reduces false alarms, but our detection method is still robust when the group size is small.

Figs. 6 and 7 show the results of the detection performance with different numbers of malicious and normal analysts. When there is only one malicious insider in the group, the recall rate remains high no matter how many normal analysts are in the group. The false positive rate increases as the number of normal analysts increases for different numbers of malicious insiders. In general, a large number of legitimate analysts in the group lead to more precise detection but also raise more false alarms. The more the malicious insiders in a group, the lower the recall rate and the false alarm are.

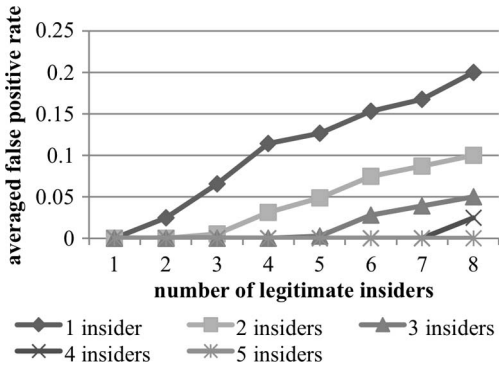


Fig. 7. Average false positive rate with different numbers of legitimate and malicious insiders with normalization procedure.

C. Impacts of Normalization With Regard to Different Group Sizes

In Section VII-A, we have shown that the detection method achieves better performance when the normalization procedure is applied. We want to examine whether the detection method with normalization performs better than that without normalization in terms of different group sizes. We conducted an exhaustive test based on the correlation values that are not normalized.

Compared with the detection results based on normalized correlation values (see Figs. 4–7), the detection performance using nonnormalized values is not so good. The highest recall rate is 0.635498 when $n = 6$. The recall rate is only about 0.6 even though the group size is 12. In contrast, the averaged recall rate is 0.8 when values are normalized (see Fig. 4). Increasing group size neither helps increase the recall rate nor helps reduce the false positive rate when the correlations are not normalized. In addition, the number of malicious insiders does not have a large impact on the recall rate. The recall rates for two, three, four, and five malicious insiders show no significant difference.

D. Impacts of ACH Matrices

All analysts that participated in the APEX '07 experiment used the ACH method to conduct their analyses. We want to evaluate whether the performance of the detection method will degrade if the ACH method is not adopted. We excluded the ACH matrix from the assessment report and applied the detection method with the normalization procedure.

The performance for the detection method without the ACH portion is not as good as the previous results but is still promising. The only difference is that analyst APEXP ($0.2601 > 0.2346$) is now suspected, and the method fails to detect APEXP EXPERT ($0.1502 < 0.2431$).

VIII. Discussions on Styles

Our experimental results have shown that the correlation values from task to task remain stable for each legitimate analyst. We have hypothesized that such consistent behaviors are due to the specific style that each analyst has for conducting analyses. In this section, we consider the following question: What kind of style does the correlation measurement capture?

The quantified correlation values have three characteristics. First, for each legitimate analyst, the correlation value remains stable from task to task. Second, the correlation values themselves vary significantly (ranging from 0.17 to 0.85) for different legitimate analysts. Third, the correlation values denote the percentage of the assessment report that can be found in the information viewed by the analyst during the information-seeking stage. More specifically, the correlation value computes the similarity between the user model created for the analyst and the conclusions in their report. Each user model captures the knowledge base of each analyst, and the report captures the decisions made based on the accessed information. Therefore, the correlation denotes the dependence of the analysts' analysis on the perceived information.

Among all the "style" research (e.g., cognitive style, learning style, thinking style, etc.), cognitive style has the most similar characteristics with the correlations that we measured. Cognitive style refers to a preferred way of thinking, perceiving, and remembering [40]. Cognitive styles are used to describe habitual ways of performing tasks which rarely change over time [39]. In addition, individual differences in style are expected. There is a variety of dimensions of cognitive styles. Wholist–analytic [1], holist-serialist [41], field-dependent-field-independent [42], convergence-divergence [43], and adaptation-innovation styles [44] are popular dimensions frequently used. The wholist-analytic dimension is the most popular dimension for analyzing one's cognitive style. It is conceptualized as an individual's preference for processing information either in complete wholes or in discrete parts. Despite the fact that all dimensions attempt to explain one's cognitive style, they discuss cognitive styles under different contexts, such as learning, information seeking, and so forth. Among all the different labels, the field-dependent-field-independent dimension seems to be the best match to the third characteristic of our correlation value measurement. Field-dependent people are those who have a relevantly higher tendency to rely on the surrounding field. In contrast, field-independent people are less dependent on the surrounding field. In our context of intelligence analyses, the surrounding field is the information perceived by an analyst. The correlation value computes how dependent one analyst is on the surrounding environment.

In recent decades, research on cognitive styles has received much criticism in terms of the overlapping of definitions and style measurements in the field, lack of independent evaluation, and so forth [45]. In this paper, our work indicates that the idea of cognitive style can be applied to solve challenging problems such as insider threat detection for intelligence analysis tasks. Our ideas find support in psychology and, more significantly, might also shed light on further research into cognitive styles. First of all, our approach provides a computational way to study the characteristics of cognitive styles, such as the stability of cognitive styles over time. In our experiments, the averaged discrepancy of correlation values from task to task is as small as 0.0098. The discrepancy values (with the average value of 0.0927) demonstrate that the stability of a cognitive style is still preserved when different strategies are adopted to conducting the analysis (e.g., if the ACH approach is not applied). Second, it provides the possibility of using computational methods to measure an individual's cognitive style. Different computerized tests (e.g., cognitive style analysis, embedded figure tests, etc.) have been used to determine one's cognitive style. These tests require a lot of user effort and are not linked to any cognitive processes, thus making it hard to study the relationships between cognitive style and specific cognitive process. User-modeling techniques model a user's behaviors for a specific cognitive process (e.g., information seeking), which allows the study of cognitive style in different cognitive processes. Finally, user-modeling techniques also provide opportunities to analyze the patterns of behaviors that can be explained by one's determined style.

IX. Conclusion and Future Work

In the IC, analysts are relied upon to interpret critical situations. They can be the first ones to analyze new problems based on incomplete, dynamic, and conflicting information. Once an analyst becomes a malicious insider, they become a severe threat to the decision-making process and the security of the organization. In this paper, we have focused on detecting malicious insiders who aim to interfere with the decision-making process by manipulating decision makers' perception of the situation in question. This type of malicious insiders is hard to capture because their behaviors are both legitimate and relevant to their tasks. Nonverbal behavior, biometric information, and daily activities are common indicators used by traditional insider threat detection. In this paper, we have proposed a detection method to examine how consistent an analyst is from task to task. We conjecture that the correlation measurement, as we have discussed in Section VIII, may capture a psychological indicator—cognitive styles. The evaluation results have demonstrated that the detection method is effective in differentiating malicious insiders from legitimate ones. Without profiling an analyst's past activities, insider threat can still be determined by analyzing the current task that the analyst is involved with. In this paper, we have also proposed two hypotheses with regard to normal actions and malicious actions. We have verified that normal actions follow one's style while malicious actions do not. A key finding in the preliminary version of this paper is that individual differences have a large impact on the discrepancy for each analyst. Unless these differences are factored out, the malicious insiders are hard to be distinguished from normal analysts. The experimental results obtained with normalization showed that the discrepancy caused by individual differences could be mitigated and detection rates could be increased. The same strategy may also be used to minimize individual differences when two tasks are conducted under different contexts. However, data sets that contain multiple tasks (in fact, data sets in general) are very hard to obtain. In the case of the APEX '07 data set, we constructed two subtasks from one task.

If such a data set is available, we hope to evaluate the detection performance as well as the robustness of its performance. Such data sets can also help us further investigate how various factors, such as one's task assignments, working environment, and so forth, may influence the consistency of one's behavior over time. As the size of the current data set is small, we also

hope to evaluate our detection method on a large-scale data set that covers different scenarios of malicious insiders. In addition, it may also help us study how different types of malicious actions may impact the detection performances. Furthermore, we intend to investigate the cognitive styles of analysts both in the information-seeking stage and in the report-production stage. It would be interesting to measure each analyst's cognitive styles during the information-seeking stage and study the correlation between the cognitive style in the information-seeking stage and that in the report-production stage using our techniques. Moreover, we also want to investigate what is factored out by the normalization procedure, which may shed light on studying individuals' cognitive styles. The measure of cognitive styles in different stages and their correlations would help us better detect insider threats. Also, we plan to further expand our existing framework so that it is not only able to identify suspicious analysts but also able to list the abnormal behaviors of the suspected analysts as evidence.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for all the comments that help improve the quality of this paper.

REFERENCES

[1] R. J. Riding and S. Rayner, *Cognitive Styles*, S. Rayner and R. J. Riding, Eds. Westport, CT: Greenwood, 2000.

[2] E. Santos, Jr., H. Nguyen, F. Yu, K. J. Kim, D. Li, J. T. Wilkinson, A. Olson, and R. Jacob, "Intent-driven insider threat detection in intelligence analyses," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, Sydney, Australia, 2008, pp. 345–349.

[3] M. B. Salem, S. Hershkop, and S. J. Stolfo, "A survey of insider attack detection research," in *Insider Attack and Cyber Security*. New York: Springer-Verlag, 2008, pp. 69–90.

[4] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *J. Comput. Security*, vol. 6, no. 3, pp. 151–180, Aug. 1998.

[5] D.-K. Kang, D. Fuller, and V. Honavar, "Learning classifiers for misuse detection using a bag of system calls representation," in *Proc. 6th Annu. IEEE SMC IAW*, 2005, pp. 118–125.

[6] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for Unix processes," in *Proc. IEEE Symp. Security Privacy*, Oakland, CA, 1996, pp. 120–128.

[7] D. Mutz, W. Robertson, G. Vigna, and R. Kemmerer, "Exploiting execution context for the detection of anomalous system calls," in *Proc. Int. Symp. RAID*, Gold Coast, Australia, 2007, pp. 1–20.

[8] M. S. Sharif, K. Singh, J. Giffin, and W. Lee, "Understanding precision in host based intrusion detection," in *Proc. Int. Symp. RAID*, 2007, pp. 21–41.

[9] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE Netw.*, vol. 8, no. 3, pp. 26–41, May/Jun. 1994.

[10] C. Ko, "Execution monitoring of security-critical programs in distributed systems: A specification-based approach," in *Proc. IEEE Symp. Security Privacy*, 1997, pp. 175–187.

[11] N. Nguyen, P. Reiher, and G. H. Kuenning, "Detecting insider threats by monitoring system call activity," in *Proc. IEEE Syst., Man Cybern. Soc. Inf. Assur. Workshop*, 2003, pp. 45–52.

[12] A. Liu, C. Martin, T. Hetherington, and S. Matzner, "A comparison of system call feature representations for insider threat detection," in *Proc. 6th Annu. IEEE SMC IAW*, 2005, pp. 340–347.

[13] A. Liu, C. Martin, T. Hetherington, and S. Matzner, "AI lessons learned from experiments in insider threat detection," in *Proc. AAAI Spring Symp.*, 2006, pp. 49–55.

[14] C. Kruegel, D. Mutz, F. Valeur, and G. Vigna, "On the detection of anomalous system call arguments," in *Proc. ESORICS*, 2003, pp. 326–343.

[15] M. Schonlau, W. DuMouchel, W.-H. Ju, and A. F. Karr, "Computer intrusion: Detecting masquerades," *Stat. Sci.*, vol. 16, no. 1, pp. 58–74, Feb. 2001.

[16] R. A. Maxion and T. N. Townsend, "Masquerade detection using truncated command lines," in *Proc. Int. Conf. DSN*, 2002, pp. 219–228.

[17] J. Seo and S. Cha, "Masquerade detection based on SVM and sequence-based user commands profile," in *Proc. 2nd ACM Symp. Inf., Comput. Commun. Security*, 2007, pp. 398–400.

[18] R. A. Maxion, "Masquerade detection using enriched command lines," in *Proc. Int. Conf. DSN*, San Francisco, CA, 2003, pp. 5–14.

[19] S. Greenberg, "Using Unix: Collected traces of 168 users," Dept. Comput. Sci., Univ. Calgary, Calgary, AB, Canada, Tech. Rep. 88/333/45, 1988.

[20] M. Kirkpatrick, E. Bertino, and F. Sheldon, "An architecture for contextual insider threat detection," cspurdueedu, 2009, pp. 1–11.

[21] Y. Yang and C. Tzi-cker, "Display-only file server: A solution against information theft due to insider attack," in *Proc. ACM Workshop Digital Rights*, 2004, pp. 31–39.

[22] P. Suranjan, S. Vidyaraman, and U. Shambhu, "Security policies to mitigate insider threat in the document control domain," in *Proc. Comput. Security Appl. Conf.*, 2004, pp. 304–313.

[23] M. Maloof and G. D. Stephens, "ELICIT: A system for detecting insiders who violate need-to-know," in *Proc. Recent Adv. Intrusion Detection*, 2007, pp. 146–166.

[24] A. Natarajan and L. Hossain, "Towards a social network approach for monitoring insider threats to information security," in *Proc. 2nd NSF/NIJ Symp. Intell. Security Informat.*, Tucson, AZ, 2004, pp. 501–507.

[25] S. Symonenko, E. D. Liddy, O. Yilmazel, R. Del Zoppo, E. Brown, and M. Downey, "Semantic analysis for monitoring insider threats," in *Proc. 2nd NSF/NIJ Symp. Intell. Security Informat.*, Tucson, AZ, 2004, pp. 492–500.

[26] O. Yilmazel, S. Symonenko, N. Balasubramanian, and E. D. Liddy, "Leveraging one-class SVM and semantic analysis to detect anomalous content," in *Terrorism Informatics*. New York: Springer-Verlag, 2008, pp. 407–424.

[27] J. S. Park and S. M. Ho, "Composite role-based monitoring (CRBM) for countering insider threats," in *Proc. 2nd NSF/NIJ Symp. Intell. Security Informat.*, Tucson, AZ, 2004, pp. 201–213.

[28] C. P. Pfleeger, "Reflections on the insider threat," in *Insider Attack and Cyber Security: Beyond the Hacker*. New York: Springer-Verlag, 2008, pp. 5–16.

[29] J. Hunker and C. W. Probst, "Insiders and insider threats—An overview of definitions and mitigation techniques," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 2, no. 1, pp. 4–27, 2011.

[30] H. Nguyen, E. Santos, Jr., Q. Zhao, and H. Wang, "Capturing user intent for information retrieval," in *Proc. 48th Annu. Meeting HFES*, New Orleans, LA, 2004, pp. 371–375.

[31] E. Santos, Jr., Q. Zhao, H. Nguyen, and H. Wang, "Impacts of user modeling on personalization of information retrieval: An evaluation with human intelligence analysts," in *Proc. 4th Workshop Eval. Adapt. Syst., Conjunction With UM*, 2005, pp. 27–36.

[32] H. Nguyen, "Capturing user intent for information," Ph.D. dissertation, Univ. Connecticut, Storrs, CT, 2005.

[33] E. Santos, Jr., H. Nguyen, Q. Zhao, and H. Wang, "User modelling for intent prediction in information analysis," in *Proc. 47th Annu. Meeting Hum. Factors Ergonom. Soc.*, 2003, pp. 1034–1038.

[34] D. Grinberg, J. Lafferty, and D. Sleator, "A robust parsing algorithm for link grammars," in *Proc. 4th Int. Workshop Parsing Technol.*, 1995, pp. 111–125.

[35] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 2002.

[36] M. Montes-y-Gómez, A. Gelbukh, and A. Lópes-López, "Comparison of conceptual graphs," in *Proc. 1st MICAI*, 2000, pp. 548–556.

[37] R. J. Heuer, Jr., *Psychology of Intelligence Analysis*. Washington, DC: U.S. Govt. Printing Off., 1999.

[38] B. Wood, "An insider threat model for adversary simulation," in *Proc. Res. Mitigating Insider Threat Inf. Syst.*, 2000, vol. 2, pp. 41-47.

[39] R. J. Riding and S. Rayner, *Cognitive Styles and Learning Strategies: Understanding Style Differences in Learning and Behaviour*, S. Rayner, Ed. London, U.K.: Fulton, 1998.

[40] H. A. Witkin, D. R. Goodenough, and S. A. Karp, "Stability of cognitive style from childhood to young adulthood," *J. Personality Social Psychol.*, vol. 7, no. 3, pp. 291–300, Nov. 1967.

[41] G. Pask, "Styles and strategies of learning," *Brit. J. Educ. Psychol.*, vol. 46, no. II, pp. 128–148, 1976.

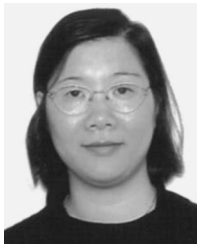
[42] H. A. Witkin, C. A. Moore, D. R. Goodenough, and P. W. Cox, "Field-dependent and field-independent cognitive styles and their educational implications," *Rev. Educ. Res.*, vol. 47, no. 1, pp. 1–64, 1977.

[43] L. Hudson, *Contrary Imaginations: A Psychological Study of the English Schoolboy*. New York: Taylor & Francis, 1966.
[44] K. Michael, "Adaptors and innovators: A description and measure," *J. Appl. Psychol.*, vol. 61, no. 5, pp. 622–629, Oct. 1976.
[45] E. R. Peterson, S. G. Rayner, and S. J. Armstrong, "Researching the psychology of cognitive style and learning style: Is there really a future?" *Learning Individual Differences*, vol. 19, no. 4, pp. 518–523, Dec. 2009.

Eugene Santos, Jr. (M'93–SM'04) received the B.S. degree in mathematics and computer science and the M.S. degree in mathematics (specializing in numerical analysis) from Youngstown State University, Youngstown, OH, in 1985 and 1986, respectively, and the Sc.M. and Ph.D. degrees in computer science from Brown University, Providence, RI, in 1988 and 1992, respectively.

He is currently a Professor of engineering with the Thayer School of Engineering, Dartmouth College, Hanover, NH. He is currently an Associate Editor for the *International Journal of Image and Graphics* and is also on the editorial advisory board for *System and Information Sciences Notes* and on the editorial boards for the *Journal of Intelligent Information Systems* and the *Journal of Experimental and Theoretical Artificial Intelligence*. His areas of research interest include artificial intelligence, intent inferencing, social and cultural modeling, computational social science, automated reasoning, decision science, adversarial reasoning, user modeling, natural language processing, probabilistic reasoning, knowledge engineering, verification and validation, protein folding, virtual reality, and active user interfaces.

Dr. Santos has served on many major conference program committees from intelligent agents to evolutionary computing. He is currently the Editor-in-Chief for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: PART B.



Hien Nguyen (M'10) received the Ph.D. degree in computer science from the University of Connecticut, Storrs, in 2005.

She is currently an Assistant Professor with the Department of Mathematical and Computer Sciences, University of Wisconsin, Whitewater, where she is actively involved in supervising undergraduate research. She is a member of the Special Reviewers Board of the *User Modeling and User-Adapted Interaction* journal. Her research interests include user modeling, information retrieval, collaborative information retrieval, recommender systems, intent inferencing, and text summarization with a current focus on hybrid user models for improving a user's performance in information retrieval.

Dr. Nguyen has worked in professional services and program committees, including the 2010 User Modeling, Adaptation, and Personalization Conference, the 2008–2010 Florida Artificial Intelligence Research Society Conferences, and the 2006, 2007, and 2011 IEEE International Conferences on Systems, Man, and Cybernetics.

Fei Yu (S'07) received the B.S. degree in computer science from The Hong Kong Polytechnic University, Kowloon, Hong Kong, in 2007. She is currently working toward the Ph.D. degree in computer engineering with the Thayer School of Engineering, Dartmouth College, Hanover, NH.

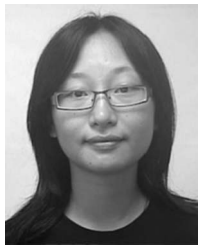
She has published papers at the Web Intelligence and Intelligent Agent Technology Conference, the User Modeling, Adaptation, and Personalization Conference, and the EUROSIM Conference. Her research interests include intent and cultural modeling, information retrieval, and text summarization.

Ms. Yu has also served as a Reviewer for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: PART C, the *Journal of Intelligent Information Systems*, etc.

Keum Joo Kim (A'09) received the B.S. and M.S. degrees in computer science and engineering from Ewha Women's University, Seoul, Korea, and the Ph.D. degree from the Thayer School of Engineering, Dartmouth College, Hanover, NH.

She was a Research Scientist with LG Central Institute Technology, Seoul. She is currently a Research Associate with the Thayer School of Engineering, Dartmouth College. Her research interests include algorithm complexity analysis, evolutionary computation, and knowledge engineering. In recent years, she has been developing an effective and efficient evolutionary algorithm for evolutionary computation. She is also building up a computational framework to assist medical professionals through knowledge engineering for ensuring patient safety.

Dr. Kim has been a member of Sigma Xi and a program committee member for the IEEE International Conference on Systems, Man, and Cybernetics. She has served as a Reviewer for several professional communities, such as the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: PART B, the *International Journal of Image and Graphics*, the *Data and Knowledge Engineering Journal*, the *International Conference on Parallel Processing*, the *Journal of Supercomputing*, etc.



Deqing Li (S'09) received the B.S. degree in electronic and information engineering from The Hong Kong Polytechnic University, Kowloon, Hong Kong, in 2007. She is currently working toward the Ph.D. degree in computer engineering with the Thayer School of Engineering, Dartmouth College, Hanover, NH.

She has published papers in the *Proceedings of SPIE*, the *Proceedings of the User Modeling, Adaptation, and Personalization Conference*, the *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, the *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: PART A, and a book chapter in *E-Government Diffusion, Policy, and Impact: Advanced Issues and Practices*. Her research interests include decision theory, intent modeling, and related applications.

John T. Wilkinson received the B.S. degree in computer science and the B.S. degree in mathematics from the Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, in 2007. Since the fall of 2007, he has been working toward the Ph. D. degree with the Thayer School of Engineering, Dartmouth College, Hanover, NH.

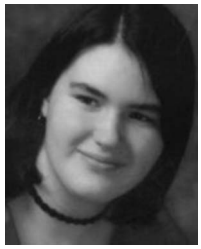
While at Virginia Tech, he began his research career studying parallel computing, social networks, and probabilistic reasoning as an Undergraduate Researcher. Since the fall of 2007, he has also been a Research Assistant with the Thayer School of Engineering, Dartmouth College, where he continues to work in the same areas when he started in as an undergraduate. He has published papers in several conferences, including the International Conference on Artificial Intelligence (2008), the International Florida Artificial Intelligence Research Society Conference (2009), the Intelligent Agent Technology Conference (2008), and the IEEE International Conference on Systems, Man, and Cybernetics (2009).

Adam Olson received the B.S. degree in management computer systems from the University of Wisconsin, Whitewater, in 2010.

He is currently an Interactive Developer with IQ Foundry, Madison, WI, where, just like in research, the key to success is keeping up with the latest technologies and trends.

Jacob Russell received the B.S. degree in management computer systems and mathematics with computer emphasis from the University of Wisconsin, Whitewater, in 2009. He is currently working toward the M.S. degree in computer science with the University of Wisconsin, Milwaukee.

His current research interests include artificial intelligence, knowledge engineering, human factors, and user modeling.



Brittany Clark is currently working toward the B.S. degree in management computer systems with the University of Wisconsin, Whitewater.

She works under Dr. H. Nguyen, doing undergraduate research. Her research interests include ontologies, user modeling, and intent inferencing.

Ms. Clark is a member of the Whitewater Chapter of the Association of Information Technology Professionals.