Multi-source fusion for anomaly detection: using across-domain and across-time peer-group consistency checks

Hoda Eldardiry*, Kumar Sricharan, Juan Liu, John Hanley, Bob Price, Oliver Brdiczka, and Eugene Bart
Palo Alto Research Center
Palo Alto, California, United States

Abstract

We present robust anomaly detection in multi-dimensional data. We describe information fusion across multiple levels in a layered architecture to ensure accurate and reliable detection of anomalies from heterogeneous data. We consider the problem of detecting anomalous entities (e.g., people) from observation data (e.g., activities) gathered from multiple contexts or information sources over time. We propose two anomaly detection methods. The first method seeks to identify anomalous behavior that blends within each information source but is inconsistent across sources. A supervised learning approach detects the *blend-in* anomalies manifested as across-information source inconsistencies. The second method identifies *unusual changes* in behavior over time using a Markov model approach. Finally, we present a fusion approach that integrates evidence from both methods to improve the accuracy and robustness of the anomaly detection system. We illustrate the performance of our proposed approaches on an insider threat detection problem using a real-world work-practice data set.

Keywords: anomaly detection, insider threat detection, information fusion, machine learning

1 Introduction

Anomaly detection is a subfield of data analytics research, with theory routed in AI, data mining, and machine learning. It aims at identification of data points (e.g., iterms, events, time series, and relational data) that deviate from an expected norm state. Anomaly detection has a broad range of applications including insider threat detection [1], fault diagnostics in machine operation and fraud detection [2]. In many practical situations, anomaly detection is a challenging problem because the datasets are big and diverse. Existing anomaly detection systems however operate separately on individual homogeneous components of the data and fail to suitably exploit and fuse the heterogeneous sources of information, which results in lower precision (robustness) and lower recall (accuracy). In this paper, we propose a layered architecture that simultaneously processes the heterogeneous information sources and fuses the processed output to detect anomalies in a more robust and accurate manner.

1.1 Motivating anomaly detection problem: insider threat detection

This work presents general anomaly detection concepts that can apply in several contexts. We use insider threat detection as a motivating example application to demonstrate the problem and our proposed methods. Malicious insiders pose significant threats to information security, and yet detection of malicious insiders is still an open problem. In this paper we report our effort on detecting malicious insiders

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 5, number: 2, pp. 39-58

*Corresponding author: Intelligent Systems Lab., Palo Alto Research Center (PARC), A Xerox Company 3333 Coyote Hill Road, Palo Alto, California, 94304, Email: hoda.eldardiry@parc.com, Tel: +1-(650)-812-4790, Web: `http://www.parc.com/hodaeldardiry`

from large amounts of work practice data comprising users' IT traces on their workstations, recording activities such as logging on/off, sending and receiving emails, accessing external devices or files, and accessing web sites. Work practice data is remarkably diverse and heterogeneous. We refer to different categories of data as "domains", e.g., "logon domain" and "email domain". The dataset is described in section 5.1.

1.2 Related work

There are many novel technologies for detecting malicious insider behavior. One of the most popular approaches is based on hand-written rule-based filters [3, 4, 5] but it is difficult to create rules with good coverage and keep them up to date.

Instead of trying to exhaustively identify an open-ended set of behaviors, one could try to entrap adversarial insiders with decoys [6, 7, 8]. The strategy is particularly effective for certain information theft or espionage scenarios, but not as relevant for on-line vandalism.

Various models of adversarial insiders have been developed in an effort to identify outsiders from more general observations that go beyond online work related behaviors. These models include physical behaviors that are indicators of adversarial intent (e.g. foreign travel, signs of wealth) [9], as well as measurements related to motivation, personality, and emotion [10, 11, 12]. Supervised learning techniques that learn to identify malicious behavior from hand-labeled historical cases avoid the need for infrastructure that measures behaviors, or to write explicit rules, or to develop comprehensive theories of insider psychology. While all these models are valuable, none incorporate all of the possible situational triggers, context variables and indicators. We believe such attributes are necessary to establish a close connection between psychology and behavior.

The majority of learning-based methods assume homogeneous data is analyzed for a specific type of anomaly [13]. For example, nearest neighbor and density estimation methods were proposed for low-dimensional multivariate data. These techniques work well for structured data that can be summarized in a small number of dimensions such as the well-known UCI Breast Cancer set with 30 real-valued attributes [14], but fail when the data is unstructured and high-dimensional such as text, time-series [15, 16], social networks, images, and video [17]. Supervised methods also suffer generally from the problem of being limited to detecting existing forms of fraud for which training data has been identified.

Since malicious behaviors are relatively rare in the broader population, unsupervised statistical anomaly detection techniques which simply look for behavior outliers can be applied. For example, some works use machine learning techniques to model user's habitual information searching patterns on their own [18]. One can then recognize malicious masquerading users from the fact that they do not seem to know their way around their own hard drives. The approach generates impressive results for scenarios directly addressed by the approach, but is limited in the types of attacks it can handle. Similarly, specialized approaches can detect anomalies in analyst queries with respect to a Hidden Markov Model of document content [19], and deviations from models of user processes [20].

Social network data is an increasingly important information source as social networks capture a significant portion of people's interaction with the external world. While the social network data is rich and diverse, it introduces its own set of challenges for analysis [21]. Specialized techniques which can exploit the correlations between people enmeshed in a network have been developed to identify key individuals in organizations based on their communication patterns [22]. It has been fruitfully used by the defense and intelligence community to study covert networks [23] in an attempt to target the most important enemies and disrupt their organization [24].

Despite these tools, the number of incidents of insider attacks continues to rise in the government and commercial sectors. For example, a recent survey found that 28% of respondents would take sensitive enterprise data to negotiate a new position in the event their employer terminated their current position [25]. Indeed, insider attacks have been reported as the most frequent [26] or second most frequent [27] source of security incidents in recent years in the United States.

Clearly new approaches are still needed. We observe that behavior in real world scenarios can often be characterized by multiple heterogeneous data sources that include descriptive demographic data, time-series behaviors, unstructured text, and images. If we can analyze multiple types of behavior jointly, we could detect anomalous relationships that do not appear in any one source. Some existing work on anomaly detection has attempted to handle multiple types of data, by trying to convert heterogeneous data types into a common format that can be handled by a particular technique. However, state-of-the-art results demonstrate that different kinds of data are best modeled by techniques specifically designed for individual data types. In this paper we present an approach that can be used to integrate multiple heterogeneous forms of analysis in rich ways to detect subtle violations in relationships across data types.

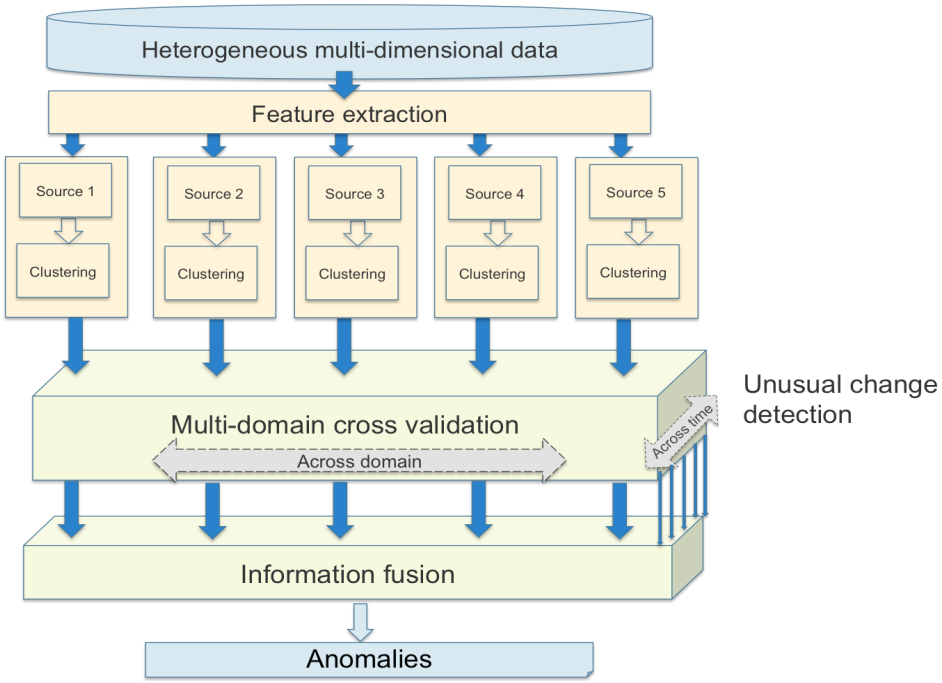

Figure 1: Anomaly Detection Framework

1.3 Problem framework

We consider the following generic setting. Our dataset is assumed to describe individual person, and we observe information concerning each individual over several information channels or sources at several instances in time. Given this data set, our goal is to accurately and robustly detect which of these entities (or users) are anomalous, which information sources they are anomalous within, and at which time instances the anomalous behavior occurred.

In particular, we assume that our data consists of measurements for N entities. For each of these N entities, we assume that we observe F different information sources (also interchangeably referred to as features, channels, or domains) at any time-instance t. Finally, we assume that we observe a total of T

time instances denoted by $t = 1, \ldots, T$. Our data can therefore be represented as a matrix of dimensions $N \times F \times T$. Given this data cube, our goal is to detect anomalous behavior w.r.t. the N entities in a robust manner by fully exploiting the heterogeneous nature of the data, both along the temporal dimension and the multiple information source dimension.

1.4 Overview of proposed approach

We propose a layered architecture for heterogeneous anomaly detection which is illustrated in Figure 1.2. First, for each information source f, we use k-means clustering to model peer groups. We do not assume peer group information is given. The N entities are clustered based on their behavior in f. We model individual activity over time by clustering activity features at each time t independent of the individual. A cluster label c_{tfn} is generated for each individual n in each information source f at each time t.

After the clustering stage, the layered architecture has two main components - one component for detecting inconsistencies across information sources (or domains), and another for detecting inconsistencies across time - which are then fused together to improve accuracy and robustness. These components are briefly described next in the context of related work.

Across-domain behavior inconsistency detection to identify blend-in anomalies - from unsupervised learning to supervised. A blend-in anomaly is a point that fits well w.r.t. each individual information source, but does not fit well when all the sources are considered jointly. For example, an engineer who logs on to multiple computers, exhibiting logon activity similar to network administrators, and not engineers, is considered a *blend-in* anomaly. Note that this user's activity w.r.t. logging into machines is consistent with network administrators, but her browsing activity will significantly differ from theirs.

To detect blend-in anomalies, we use a multi-view learning approach in which we leverage information from multiple activity domains. We assume peer-group consistency across domains. Going back to our example, a user that behaves like engineers in one activity domain (e.g., web browsing) should behave like engineers in all other activity domains (e.g., logons). The problem of detecting suspicious activity without any given ground truth has been conventionally formulated as an unsupervised learning problem. However, we formulate it as a supervised learning problem as follows. To detect across-domain inconsistency, we assume a user's behavior in one domain can be predicted from the same user's behavior in other domains and identify users with unpredictable behavior as anomalous.

Unusual change detection. Temporal anomaly detection methods focus on identifying changes in activities of a user compared to that user's past activities. The problem with this approach is that it treats any change as suspicious. Back to our running example, an employee who starts working on a new project or takes up a new role will change her activities, but this change is not suspicious.

To counter this, we define a new type of anomalous activity and refer to it as *unusual change*, which is a change that is not common to observe over the entire population. Instead of analyzing users independently, we use a Markov model approach to capture the transition probability of changing activity (or state) and declare a change as anomalous when a user exhibits changes in activities that are unusual compared to the user's peers or the rest of the population.

Multi-source information fusion. Our goal is to combine suspicion/anomaly scores that have been generated from each of the aforementioned methods to detect anomalies. We note that in a more general analytics framework, the same technique can be used to combine scores based on relative importance or surprise/risk levels.

In anomaly detection, an individual can be anomalous in one domain or at a time instance, but not in another. Also, the relative suspicion of an individual can vary from one indicator (or information source) to another. However, most anomaly detection applications require a single overall conclusion about the relative suspicion of each individual. Therefore, we developed a technique to combine multiple sources of evidence from multiple domains. In a broad sense, each source of information provides a suspicion score and the goal is to combine these scores in order to identify anomalies with greater accuracy. Going back to our insider threat detection example, we combine the predictability score of each activity domain and each time instance to compute a single, combined suspicion score for each user. These combined scores are subsequently used to identify the anomalous individuals.

Automatic threshold selection. After identifying anomalous individuals, we use an automatic entropy-based threshold selection method for outlier identification to identify the specific time instances as well as the particular anomalous events (i.e., information sources) responsible for the anomalous individuals. In particular, given a set of anomaly scores S, this method defines a corresponding threshold $T(S)$ such that all the points beyond this threshold are anomalous w.r.t. the score set S.

1.5 Outline

In section 2, we define the concept of *blend-in* anomalies, and present the technique we developed to detect this kind of anomaly. We define the second type of anomaly, *unusual change*, in section 3, and present a Markov Model approach to discover this type of anomalies. We describe our information fusion technology in detail in section 4. We next describe our proposed entropy-based staistical outlier detection technique in section 4.3.1, and use this technique to identify individual events that contributed to the anomaly scores of individuals. Experimental results are described in section 5. Finally, we present our conclusions in section 6.

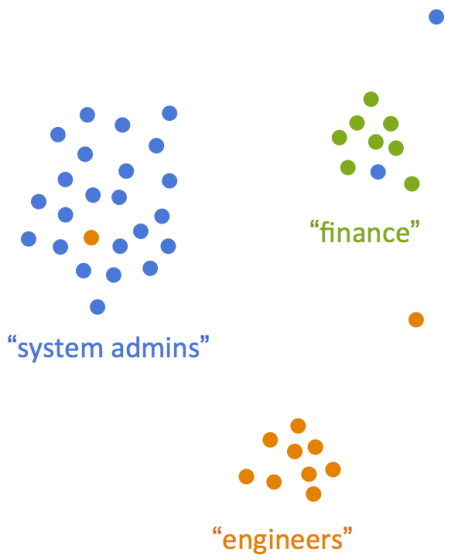

Figure 2: Multiple roles

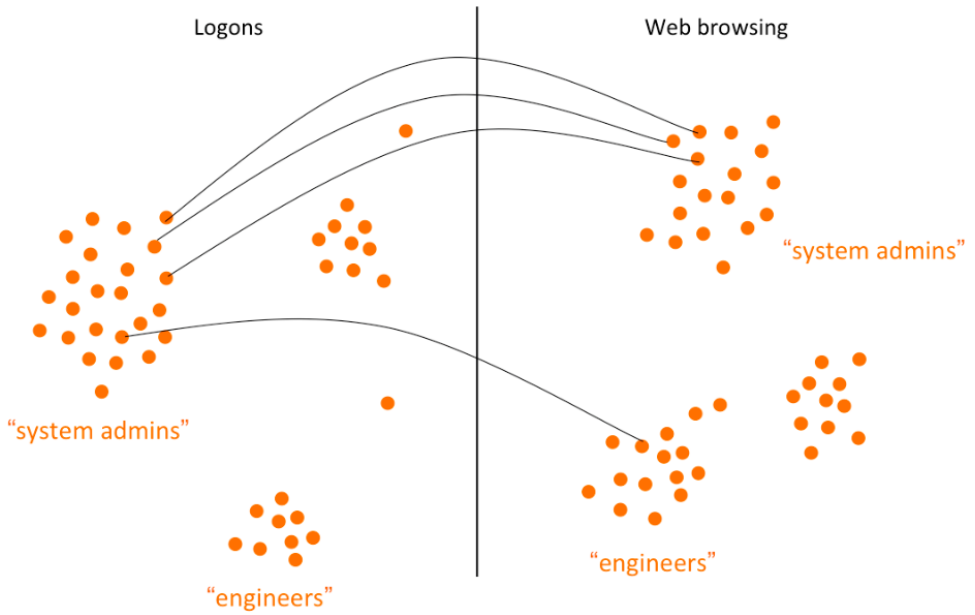

Figure 3: Detection of inconsistencies

2 Approach 1: Multi-domain cross validation (MDCV)

Given a population distribution, a blend-in anomaly is a point that blends within the distribution such that it is not a statistical outlier, but this point fits in the wrong group of points so it is not an outlier to that group. However, it can be discovered using deeper analysis of group associations.

In an insider threat detection context, the intuition is that user activity should reflect the user's job role in any domain, and users with similar job roles exhibit similar behavior within each domain. Figure 1.5 shows an example of a blend-in anomaly where an engineer (orange circle) logs on to multiple computers, exhibiting logon activity similar to network administrators (shows up within the blue group) and not engineers (not the orange group). Assuming the job roles are unknown, straightforward outlier detection will not detect the orange circle in the midst of the blue group as an outlier.

Figure 1.5 shows them all in the same color to illustrate the hidden job role information. That is to say, we clustered them according to activity, but we do not know the job roles associated with each cluster. However, to leverage multiple activity domains providing additional sources of information. The apparent network administrator, according to logon activities, is browsing the Web similar to a different group of users (the engineers). This across-domain inconsistency in peer-groups reveals the blend-in anomaly.

Problem formulation. We define the problem as follows. An anomalous individual is one that exhibits inconsistent behavior across the information sources (or activity domains). We formulate the across-domain behavior consistency assumption as a classification task, in which clusters are used as training features. We predict an individual's cluster (or peer group) in one domain from her cluster in all other domains. The prediction accuracy for an individual's cluster in each domain reflects her behavior consistency across domains.

At each time-instance t, $c_{n,f,t}$ is the cluster for individual n in domain f. For individual n, we say individual n has domain f activity that is consistent with other domains' activities if the individual's cluster

$c_{n,f,t}$ is predictable from other domains' clusters $\{c_{n,j,t}\}_{j\neq f}$. To measure how predictable the behavior of entity n in domain f from other domains, prediction is formulated as a multi-label classification task, in which a classifier is trained using cluster information from all-but-one domains to predict the cluster information in the remaining (target) domain. In the simplest case, we may use cluster labels of other individuals $m \neq n$ to learn a mapping from $\{c_{m,j,t}\}_{j\neq f}$ to $c_{m,f,t}$, and then check whether this mapping generalizes to individual n.

Our MDCV method uses cluster labels from the observed domains as features for learning, and predicts cluster labels to evaluate user predictability. Denote the predicted user's cluster label by $\hat{c}_{nft}$. The evaluation is not based on just whether or not the true cluster is predicted, but instead on how well the true cluster is predicted. This is in essence a density estimation problem. The predictability is measured as one minus the likelihood of observing the true cluster given the cluster of its peers. In particular, define the anomaly score of individual n w.r.t. time instance t and domain f as

$$r_t(n,f) = 1 - Pr[c_{n,f,t} = \hat{c}_{n,f,t}].$$

To combine suspicion scores for a particular individual, we must distinguish between domains in which suspicious activity is commonly observed from domains in which suspicious activity is very rare. In the latter case, this score should be given a higher weight. This idea is inspired by the TF/IDF (term frequency–inverse document frequency) scheme [28], reflecting the relative importance of a word to a document in a corpus. The TF/IDF value is proportional to the frequency that a word appears in the document, but is offset by the global frequency of the word in the corpus. Words that are frequent and yet unique to the document have high TF/IDF scores. This property justifies the use of TF/IDF as a weighting factor in information retrieval and text mining and we leverage this idea in anomaly detection score computation. Denote the TF-IDF weights for each domain by by w_f. We obtain the combined score for each individual n at each time-instance t by the weighted combination

$$r_t(n) = \sum_{f=1}^{F} w_f r_t(n,f).$$

3 Approach 2: Unusual change detection (CD)

We note that while a particular activity may not be suspicious, a rare change in activity can be. In this section we propose a method that detects individuals with unusual changes in activity. In this context, changes that are common among peers or the population are considered acceptable changes, and the goal is to detect changes that are less likely to happen within the group to which a user belongs. The key strength of our proposed approach is that it avoids detecting common changes that can be mistakenly detected by typical temporal anomaly detection mechanisms.

In an insider threat detection context, the intuition is that user activity should reflect the user's job role in any domain, and users with similar job roles should exhibit similar behavior changes within each domain over time, for example, due to a change in project assignment. Peers will not be expected to exhibit similar changes in behavior at similar time episodes, but they will be expected to do so over longer time intervals.

Problem formulation. Within each domain f, we assume a cluster $c_{n,f,t}$ (or group) label for each user u at each time t. We model activity changes over time as a Markov sequence. As shown in figure 3, for each user u, within each domain f, we have a sequence of cluster labels $c_{n,f,1}$, $c_{n,f,2}$, .. , $c_{n,f,T}$. Here, clusters correspond to model states. We construct a transition probability matrix Q_f for each domain f

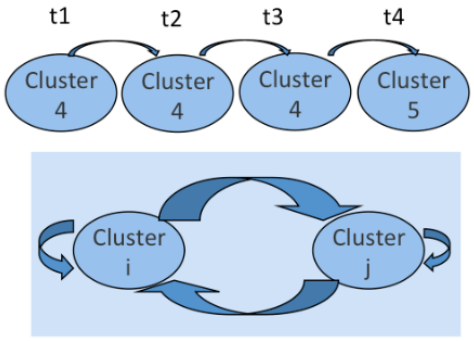

Figure 4: Change Detection illustration

by computing the transition probabililties $q_f(c_i, c_j)\ \forall_{i,j}$ (all possible clusters) by counting the number of such changes over all users over time.

Individuals are scored based on their total transition likelihood over time, and suspicious individuals with unusual transitions between temporal states are detected. Since we assume that clusters reflect peer groups, our idea of looking at cluster transitions is to compare an individual's change with that individual's peers' changes. The intuition is that peers will belong to the same set of clusters (or Markov states) and transition among that set in a similar fashion over time.

The anomaly score $r_f(n)$ for each individual n within domain f is calculated by estimating the user's transition likelihood over time. The anomaly probability score is computed as

$$r_f(n) = p_f(c_{n,1}) \prod_{t=1}^{T-1} q_f(c_{n,t}, c_{n,t+1})$$

where $p_f(c_{n,1})$ is the prior probablility of being in cluster c_1 which is the start state for entity n.

Given the scores generated by MDCV and CD, our goal is to robustly combine these scores to identify anomalous individuals. The fusion method is described in Section 4.

4 Information fusion for combining anomaly indicators

In this section, we describe the process for combining the anomaly scores generated from the Multi-domain cross-validation (MDCV) approach and the change detection (CD) approach in a robust manner. In particular, recall that the output of MDCV is given by a matrix of $N \times T$ probability scores $r_j(i)$ where $j \in \{1,\ldots,T\}$ denotes the domain, and $i \in \{1,\ldots,N\}$ is the index for individuals. Likewise, the output of CD is given by a matrix of $N \times F$ probability scores $q_j(i)$ where $j \in \{1,\ldots,F\}$ and $i \in \{1,\ldots,N\}$. We concatenate these two outputs to obtain a joint matrix of dimension $N \times (T+F)$ scores $p_j(i)$ where $j \in \{1,\ldots,F+T\}$ and $i \in \{1,\ldots,N\}$. Our goal is to combine these set of scores to robustly determine which of the N individuals is anomalous.

4.1 Score-based fusion

One set of options for fusing the anomaly scores is to directly operate on the anomaly scores $p_j(i)$ and produce an aggregate score $p(i)$. A couple of options for aggregating the score are either choosing the

maximum score across dimensions

$$p_{\max}(i) = \arg\max_j p_j(i). \tag{1}$$

or by computing the probability that a given user is anomalous w.r.t. *at least* one of the $T+F$ dimensions, which is equivalent to the complement of the probability that a user is not anomalous in any of the $T+F$ dimensions. In this case, we combine the scores using

$$p(i) = 1 - \prod_{j=1}^{T+F} (1 - I[p_j(i) > t] p_j(i)), \tag{2}$$

for some user-specified threshold t.

The idea behind the maximum score scheme is that we mark each individual by the highest score it has recorded across domains and time so as to catch any single infraction by each user. This scheme is therefore aggressive. On the other hand, the probability-based fusion scheme takes a more conservative approach by ensuring that a one-off anomalous event for an individual is penalized less severely compared to repeated transgressions. This prevents users from getting incorrectly flagged due to one-off statistically rare events that most likely happened by chance.

4.2 Rank-based fusion

The primary issue with using score based fusion is that the scores are generated from different mechanisms and as a result, provide no common ground for comparison. For instance, the scores generated from MDCV might in general be higher than the scores generated by CD, or alternatively, the scores generated during particular time-instances (for e.g., weekends) might be lower in general compared to scores over weekdays. Proceeding to combine these scores directly might therefore not result in any significant improvement subsequent to fusing the scores.

To counter this, we propose rank based fusion of the multiple anomaly scores. In particular, from the given probability scores $p_j(i)$, for each domain j, we determine the corresponding ranks $R_j(i)$, where $R_j(i)$ is simply the index at which the score $p_j(i)$ occurs in a descending-sorted array of the entries $[p_j(1), \ldots, p_j(N)]$. Next, we fuse these ranks to a single rank $R(i)$ for each individual i as

$$R(i) = \arg\min_j R_j(i). \tag{3}$$

The advantage of this scheme is that the conversion from $p_j(i)$ to $R_j(i)$ ensures that the ranks can now be compared across dimensions j. In particular, notice that the fused ranks $R(i)$, unlike in the score-based fusion methods, are robust to any monotonic transformation of the individual scores $p_j(i)$ within any particular domain. In summary, this proposed rank based fusion method clearly is more robust to the score-based alternatives.

The final algorithm based on rank based fusion is listed in Algorithm1. In the subsequent experimental section, we contrast the performance of the proposed rank based fusion with the max score and probability based fusion methods.

4.3 Exploration of anomalous individuals

Once we ascertain the anomaly ranks of the individuals, the question remains as to what was the anomalous behavior that was responsible for that ranking. To answer this question, for any given user n, we seek to identify the information sources f and time-instance t with high values of $r_t(n)$ and $q_f(n)$. In order to address what qualifies as a high value, we develop an automatic threshold selection algorithm

Algorithm 1 Heterogeneous anomaly detection algorithm

1: Input: Data matrix of dimension $N \times F \times T$
2: Output: Ranked list of anomalies
3: Compute peer group clusters c_{nft}
4: Use clusters to identify anomaly scores $r_t(n)$ across information sources using MDCV
5: Use clusters to identify anomaly scores $q_f(n)$ across time using CD
6: Concatenate scores $r_t(i)$ and $q_f(i)$ to form a single score matrix $p_j(i)$
7: Convert anomaly scores $p_j(i)$ to ranks $R_j(i)$
8: Fuse scores using $R(i) = \arg\min_j R_j(i)$.
9: **return** Anomaly rankings $R(i); i \in \{1,\ldots,N\}$

that, given a set S_N of scores $S_N = \{s_1,\ldots,s_N\}$ corresponding to N different individuals, automatically addresses (i) if any of the N individuals should be classified as anomalies w.r.t. the scores s_i, and (ii) if the answer to (i) is yes, then identifying which of the individuals are anomalous by identifying a threshold score $T(S_N)$. Given the threshold score, the anomalous individuals within S_N can simply be identifies as the elements s_i such that $s_i > T(S_N)$. Here, the score set can correspond to the set of scores within any single time-instance t_0, i.e. $S_N = r_{t_0}(1),\ldots,r_{t_0}(N)$ or any single domain f_0, i.e. $S_N = q_{f_0}(1),\ldots,q_{f_0}(N)$.

4.3.1 Automatic selection of thresholds (TS)

We assume without loss of generality that the scores S_N are anomaly scores, i.e. lower the score s_i, more normal the samples are. Stated in an other way, we assume that the scores set S_N satisfies the following monotonicity property: if an item with score s_i is an anomaly, then all items with scores $s_j > s_i$ are also anomalous.

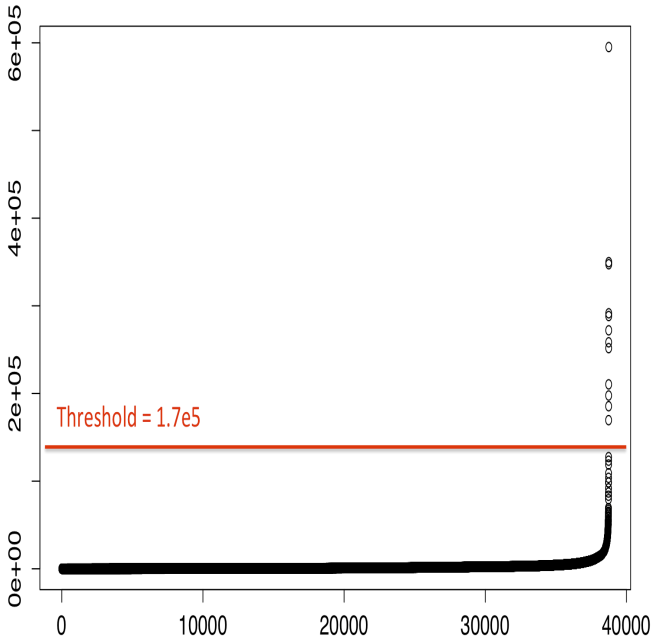

Figure 5: Outlier identification

The details of this automatic procedure, which we will henceforth denote by TS, are enumerated below:

1. The scores in S_N are sorted in ascending order. Denote these sorted entries by $\bar{S}_N = \{s_{(1)}, \ldots, s_{(N)}\}$. An illustration of these sorted scores is shown in Figure 4.3.1.

2. Collect subsets $\bar{S}_N[j]$ of $\bar{S}_N$ for j varying from 1 to N, where

$$\bar{S}_N[j] = \{s_{(1)}, \ldots, s_{(N-j+1)}\}.$$

3. Estimate the entropies $H(\bar{S}_N[j])$ of the probability density functions corresponding to the samples in $\bar{S}_N[j]$. The entropies for these 1-dimensional samples are estimated using the Vasicek spacing estimator [29].

4. Next, we identify if there is a sharp decrease in entropy $H(\bar{S}_N[j])$ as we vary j from 1 to N. If there is no sharp decrease, then we declare that no anomalies are present in the data. On the other hand, if there is a sharp decrease in the entropy as we are transitioning from $j = k$ to $j = k+1$, then we set the threshold for anomaly detection as $T(S_N) = s_{(k)}$.

5. Finally, we flag all individuals with $s_i \geq T(S_N)$ as anomalies.

The key idea behind this automated approach is that if anomalies are present in the set $\bar{S}_N[j]$, then the empirical distribution corresponding to $\bar{S}_N[j]$ will be dispersed due to the additional mode corresponding to the anomalies and as a result, the corresponding entropy $H(\bar{S}_N[j])$ will be high. On the other hand, if $\bar{S}_N[j]$ contains no anomalies, then the scores will be more concentrated and the entropy of the set will be correspondingly smaller. As a result, when we are transitioning from the set $\bar{S}[k]$ which contains anomalies to the set $\bar{S}_N[k+1]$ which contains no anomalies, there will be a sharp decrease in entropy, i.e. $H(\bar{S}_N[k]) \gg H(\bar{S}_N[k+1])$.

We note that an alternative but completely equivalent interpretation of our algorithm is as follows: For a set S_N of scores, compute the entropy $H(S_N) = \sum_{i \in S_N} P(s_i) * (-\log P(s_i))$ and for each element i, compute the surprise ratio $r_i = \frac{-\log p(s_{(i)})}{H(S_N)}$. The surprise ratio is a measure of how consistent or random a given sample point s_i is w.r.t. the rest of the data in S_N. Finally, we identify samples $s_{(i)}$ as anomalies if the corresponding surprise ratio r_i is large. This automated approach has clear advantages to currently used practices for identifying anomalies given a score set, such as reporting the k individuals $S_k \subset S_N$ with the k smallest scores or reporting all individuals $S_\delta \subset S_N$ with scores $s_i < \delta$ for some user-specified, but arbitrarily chosen δ. In particular, it is possible that the actual set of anomalies is a superset or subset of S_k (or S_δ), including the extreme case where there are no anomalies. In this case, reporting the top k individuals (or all the individuals with score less than δ) would be incorrect. Our automated approach ensures that we accurately identify if anomalies are present or not, and if they are present, correctly estimate the threshold for identifying the anomalies.

Threshold selection for streaming data. In the insider threat monitoring setting, the data is constantly being updated as new activities are being observed. It follows that the threshold for detecting anomalies should be updated as new data is being added. The procedure detailed in the previous section assumes a static setting where are all the scores $S_N := \{s_1, \ldots, s_N\}$ are available. We now extend the procedure to the online case where the scores $\bar{S} := \{s_1, s_2, \ldots\}$ are observed in a streaming fashion. Assume that we have access to the first N scores S_N from S. The threshold $T(S_N)$ for these scores S_N can be determined using the procedure in the previous section. When the next sample score s_{N+1} arrives, the threshold $T(S_{N+1})$ can be determine via a brute force approach by operating on the set S_{N+1}. However, this brute

force approach is computationally expensive as the procedure has to be repeated each time a new sample score s_{N+1} is computed.

Instead, we describe a faster alternative by making the following observation. If the new sample $s_{N+1} < T(S_N)$, then the threshold should remain unchanged, *i.e.* $T(S_{N+1} = T(S_N)$. This is because the score s_{N+1}, by virtue of satisfying $s_{N+1} < T(S_N)$, classifies as a normal score and this in turn implies that the threshold $T(S_N)$ need not be adjusted in order to avoid incorrectly mis-classifying s_{N+1} as anomalous. Equivalently, this implies that the threshold needs to be updated only when $s_{N+1} \geq T(S_N)$. Also, by the very nature of anomalies being sparse, the event $s_{N+1} \geq T(S_N)$ will occur rarely and as a result, the procedure for determining automatic thresholds scales easily for streaming data.

Illustrative Example. As an illustrative example, consider a set S_N of $N = 4000$ anomaly scores plotted in ascending order in the illustration in Figure 4.3.1. From the figure, it is intuitively clear that (i) anomalies are present in this data, and (ii) the anomalous entries are the ones with scores greater than $1.5e5$. On application of our algorithm to this data set, we identify a threshold $T(S) = 1.7e5$, which is illustrated via the red line in the figure.

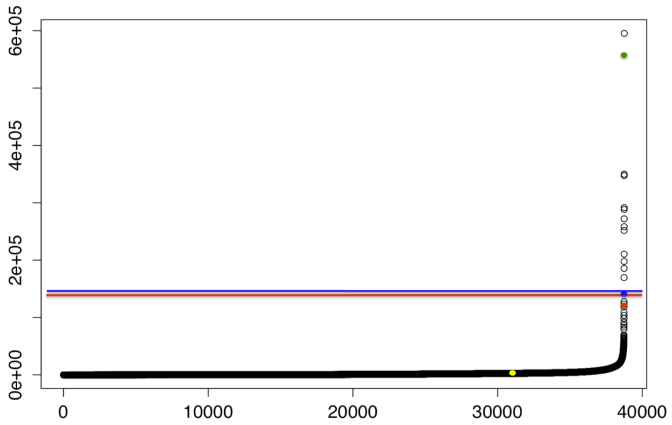

Figure 6: Outlier identification for online setting. Threshold remains unchanged (original red line) when the new data points are below the original threshold (red and yellow dots). For new data points above threshold (blue and green dots), threshold should be reevaluated. The threshold changes to the new line marked in blue if the new data point corresponds to the blue dot. On the other hand, the threshold does not change for the green dot even after reevaluation because of the extreme nature of the green dot.

We extend this illustrative example to the online case next. The online situation is illustrated via Figure 4.3.1. The new sample s_{N+1} could be any one of the four differently colored -green, red, yellow and blue - dots. It is clear from this figure that the original threshold (marked by the red line) should remain unchanged when the new data point s_{N+1} is below the threshold, as is the case with the yellow and red dots. If s_{N+1} corresponds to the blue dot or the green dot, then as per our algorithm, the threshold should be reevaluated. We note that the threshold changes after reevaluation (to the new blue line) only in the case of the blue dot, and not in the case of the green dot because of the extreme value of the green dot.

4.3.2 Application to entity exploration

The automated threshold selection algorithm is run on the set of scores $p_j(i)$ for each j in $1,\ldots,T+F$ and $T+F$ threshold scores T_j are identified. Subsequently, all the tuples $E_i = \{(i,j); p_j(i) > T_j\}$ are

recorded and maintained for each entity i. Finally, when a particular user n is being investigated due to high anomaly rank (generated as output from Algorithm 1), the anomalous events E_n are reported. The investigation of specific users is illustrated in Figure 5.3 in the experimental section.

5 Experimental Evaluation

We apply our proposed framework (illustrated in figure 1.2) on an indsider threat detection problem, starting with a multi-dimensional dataset as input until we finally generate a list of anomalies.

5.1 Dataset

We present results using a real dataset provided by a large defense contractor. The dataset contains workpractice information about 4334 users collected over 30 days. The data volume is approximately 89 million records per day. To simplify processing, we bin events into user day records. For each (user, day) pair, we compute aggregated statstics as shown below. The data also contains synthetically injected anomalies based on real world malicious behavior scenarios. Note that the scenario labeling was not made available to the learning, modeling or detection algorithms.

5.2 Feature Extraction

Work pratice data falls into the domains listed below. Each event is tagged with auxilary information such as user id, host PC id, activity code (whether it is a logon / logoff, file upload / download, etc.), and a timestamp. We consider six different activity domains: “device”, “email sent”, “email received”, “file”, “http” and “logon”.

- Logon and logoff events.
- Use of removable device such as USB thumb drives or removable hard disks. Device name and type are logged with each usage event.
- File access events: e.g., file created, copied, moved, written, renamed, or deleted. For each file access the record, file name, path, type, and content are logged.
- Http access events, tagged with URL and domain information, activity codes (upload or download), browser information (Internet Explorer, Firefox, or Chrome), and whether the website is encrypted.
- Email sent and viewed are tagged with from address, to / cc / bcc addresses, subject line, sent date, text, attachment info, and whether the email is encrypted.

Furthermore, our system associates a set of tags to raw events. For instance, we label (1) whether the event happens after normal working hours and (2) whether the event happens on a user’s own designated PC, someone else’s designated PC, or a shared PC. As malicious insiders often need to steal information from their colleagues, labeling the host PC is semantically important. In addition, events concerning activities external to the organization (e.g., email sent to or received from external addresses, file upload/download from external URLs) are labeled.

In real world settings users often produce a multitude of events. For each (user, day) pair, we compute aggregated statstics as shown below. Domain features are treated separately, and daily feature vectors for each domain is extracted, as summarized below.

- Logon: #logons, #PCs with logons, #after hour logons, #logons on dedicated PC, #of logons on other people's dedicated PC
- Device: #device access, #PCs with device access, #after hour device access, #device usage on dedicated PC, #device usage on other people's dedicated PC
- File: #file access, #PCs with file access, #distinct files, #after hour file access, #file access on dedicated PC, #file access on other people's dedicated PC
- HTTP: #web access, #PCs with web access, #URLs visited, #after hour web access, #URLs visited from other people's dedicated PC
- Email Sent: #emails, #distinct recipients, #internal emails, #internal recipients, #emails sent after hour, #emails with attachment(s), #emails sent from non dedicated PC
- Email Received: Similar to email sent

Finally, we apply K-means clustering to generate a 3-dimensional matrix: N x F x T.

5.3 Results

Multi-domain cross validation. We apply our multi-domain cross validation method as follows. For each of the six activity domains, the goal is to predict the user's cluster, at any given instant, from the same user's clusters in all other domains at that same instant. The prediction accuracy reflects the peer group consistency across domains. A high accuracy leads to a low anomaly score. Finally, a combined score for each (user, time) pair is generated by applying TF/IDF to the scores from all the domains.

Figure 5.3 plots the prediction probability distribution for each domain as well as the anomaly scores. The figure shows the general domain predictability over the entire population based on unadjusted scores for each domain. The varying level of predictability explains why the different domains used have different levels of importance in identifying an anomaly. Device and File domains are the most predictable (with the highest predictive accuracy), while Logon and HTTP domains are harder to predict. It appears that users show great variation in their logon and http behavior, but are more uniform in device usage and file access. Finally, it is hardest to predict activity in Email-sent and Email-received domains. This emphasizes the necessity of using TF/IDF as a fusion approach, which considers this domain variation when computing the final anomaly score.

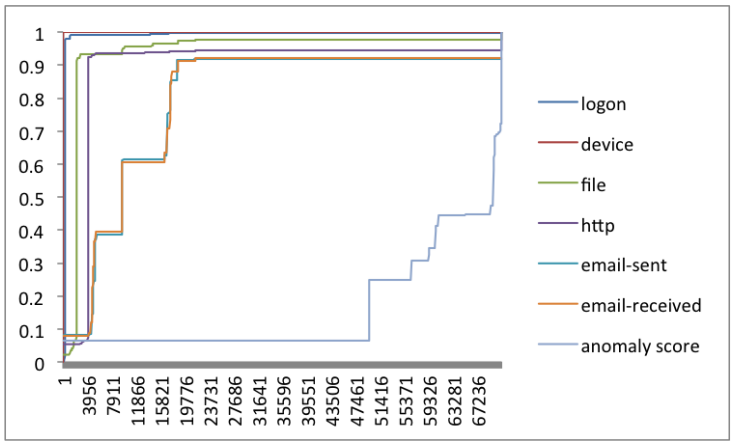

Figure 7: Multi-domain cross validation: prediction probability for each domain and final anomaly scores

Unusual change detection. We apply our unusual change detection method as follows. Within each of the six activity domains, the goal is to compute the overall user's cluster transition likelihood over time. The likelihood reflects the activity change consistency of the population within each domain. A high likelihood leads to a low anomaly score. Finally, a combined score for each user is generated using the minimum likelihood across domains. Figure 5.3 plots the transistion likelihood distribution for each domain as well as the anomaly scores.

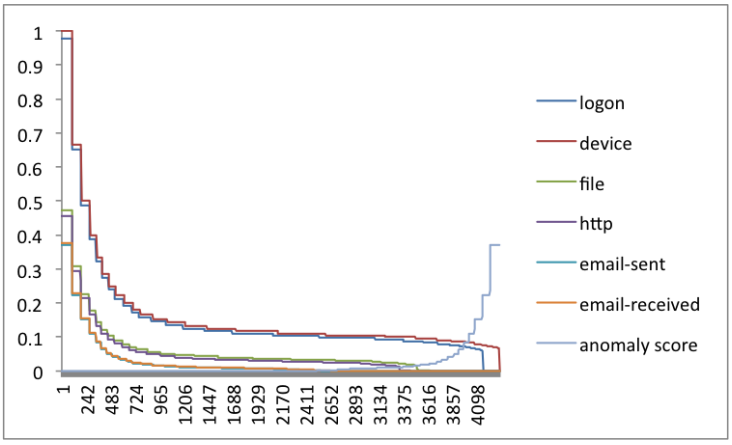

Figure 8: Markov Unusual Change Detection: transition likelihood for each domain and final anomaly scores

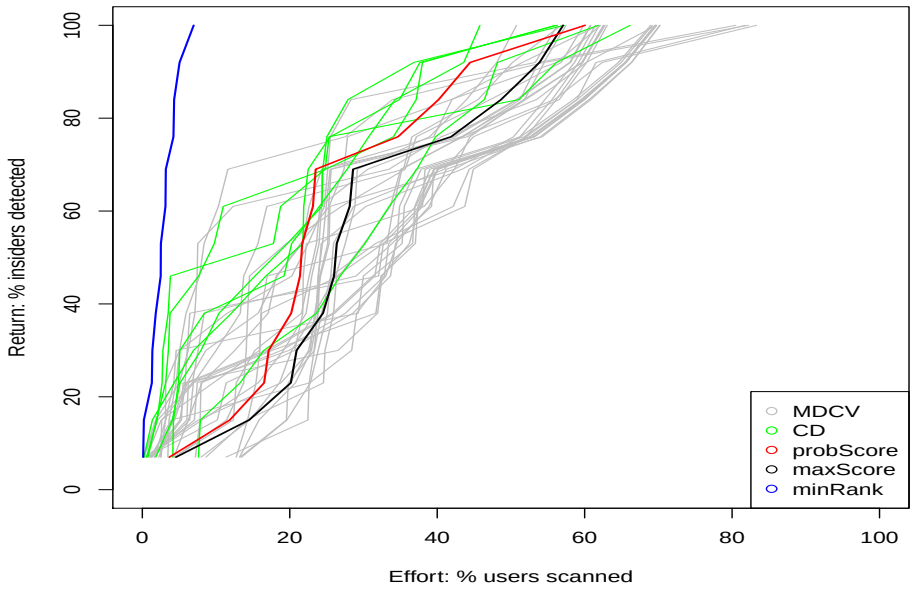

Figure 9: Insider detection

Ranking and scanning. We apply Algorithm 1 to generate the fused ranking $R(i)$ for this data set. Given these rankings, Figure 5.3 shows the investigation effort on the x-axis and the return on the y-axis

Table 1: Rank percentile corresponding to different insiders using different fusion methods. The proposed rank based fusion consistently has a higher anomaly rank across the different insiders and therefore outperforms other methods.

Insider	**meanMDCV**	**meanCD**	**probScore**	**maxScore**	**minRank**
67145F	21.4	14.9	14.5	11.9	1.3
2C7F84	25.6	12.5	26.4	21.7	3.2
D5943A	23.8	21.7	20.1	16.5	2.5
D1A30D	17.3	12.1	24.5	20.2	1.4
A6C30B	44.7	36.9	41.9	34.7	5.1
95B2BC	32.5	27.1	28.6	23.5	4.3
4EE3A0	61.8	40.9	57.1	60.1	1.8
D96B46	46.4	45.1	48.7	40.2	4.2
6044DA	24.6	13.1	20.9	17.1	2.5
1A1686	52.0	31.8	53.9	44.5	7.0
7F5DFF	24.1	14.7	26.0	21.4	3.1
663B16	13.4	4.4	4.6	3.7	0.1
EE07B6	15.5	7.4	28.1	23.1	0.2

for each approach. The effort is represented by the percentage population that needs to be investigated and the return is the percentage insiders detected. For comparison purposes, we also plot the performance curves for rankings based on the individual MDCV and CD results, and also the rankings corresponding to the maximum-score fusion approach and the probability-based fusion approach. Each gray curve corresponds to the MDCV output at time t. Each green curve corresponds to the CD approach from domain f. The red curve shows the result for the probability-based fusion approach. The black curve shows the result for the maximum-score fusion approach. The blue curve corresponds to the best performing minRank approach using which all the insiders are detected after scanning the top 7% of the population.

Table 1 summarizes the % of the population that needs to be scanned to detect each insider by each scheme. For MDCV and CD, we present the average results due to space constraints. From these results, it is clear that the rank based fusion approach comfortably outperforms the other methods used in this comparison. Indeed, it is remarkable that a suitable combination of the individual MDCV and CD scores in an appropriate fashion leads to significant improvement in performance relative to any of the individual MDCV or CD sources.

Localization and visualization of anomalous activity for insiders. Once we identify the highly ranked insiders, our automatic thresholding algorithm TS can be applied to breakdown the specific time-instances and domains that are associated with anomalous activity. Figure 5.3 shows anomalous activities by each insider (rows) within each domain (colored stack of points) at each time-instance (columns) for this insider. These anomalous activity incidents were detected using our outlier identification scheme, applied separately on one vector of one activity domain at one time instance for all the users.

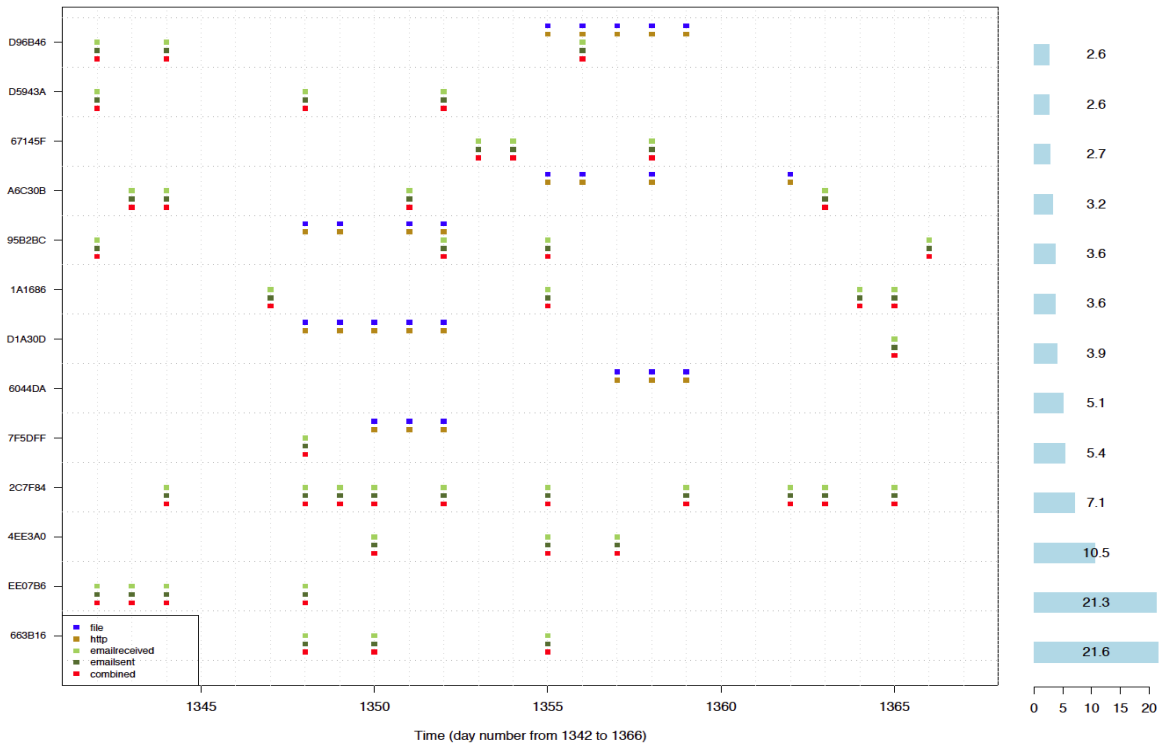

Figure 10: Visualization of anomalous incidents of insiders

6 Conclusion

A novel layered approach for robust detection of anomalies is discussed. The two main components of our framework - MDCV and CD - improve anomaly detection prediction accuracy by combining information from multiple domains and time-instances. As a result, these methods are able to determine anomalies that are not apparent in any single domain or time-instance, but only manifest in discrepancies across domains. In addition, we have proposed a robust ranking based fusion scheme to fuse the results generated from MDCV and CD. Our fusion scheme offers the advantage of being robust to variance in the individual outputs of MDCV and CD while combining their information in order to improve the quality of anomaly detection. Finally, we propose a novel outlier threshold selection algorithm that aids in analyzing the specific domain and time-instance related events that were responsible for a particular to be flagged with a high anomaly rank. We verify the improved robustness and accuracy of our proposed algorithm via experimental results on detecting insiders from a large, real-world work-practice data set.

Acknowledgments

The authors gratefully acknowledge support for this work from DARPA through the ADAMS (Anomaly Detection At Multiple Scales) program funded project GLAD-PC (Graph Learning for Anomaly Detection using Psychological Context). Any opinions, findings, and conclusions or recommendations in this material are those of the authors and do not necessarily reflect the views of the government funding agencies.

References

[1] H. Eldardiry, E. Bart, J. Liu, J. Hanley, B. Price, and O. Brdiczka, "Multi-domain information fusion for insider threat detection," in *Proc. of the 2013 IEEE Security and Privacy Workshops (SPW'13), San Francisco, California, USA*. IEEE, May 2013, pp. 45–51.

[2] H. Eldardiry, J. Liu, Y. Zhang, and M. Fromherz, "Fraud detection for healthcare." ACM, August 2013.

[3] "Raytheon oakley systems sureview," http://www.raytheon.com/capabilities/products/cybersecurity/insiderthreat/products/surview/, retrieved February 14, 2012.

[4] "Lanxoma. intelligent desktop surveillance." http://www.lanxoma.com/, retrieved February 14, 2012.

[5] "Packetmotion," http://www.packetmotion.com/, retrieved February 14, 2012.

[6] L. Spitzner, "Honeypots: Catching the insider threat," in *Proc. of the 19th Annual Computer Security Applications Conference (ACSAC'03), Las Vegas, Nevada, USA*. IEEE, December 2003, pp. 170–179.

[7] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, *Baiting inside attackers using decoy documents*. Springer, 2009.

[8] S. Stolfo, S. M. Bellovin, A. D. Keromytis, S. Sinclair, S. W. Smith, and S. Hershkop, *Insider Attack and Cyber Security: Beyond the Hacker (Advances in Information Security)*. Springer-Verlag TELOS, 2008.

[9] M. Maybury, P. Chase, B. Cheikes, D. Brackney, S. Matzner, T. Hetherington, B. Wood, C. Sibley, J. Marin, and T. Longstaff, "Analysis and detection of malicious insiders," DTIC Document, Tech. Rep., 2005.

[10] K. L. Herbig, *Changes in espionage by Americans: 1947-2007*. DIANE Publishing, 2009.

[11] K. L. Herbig and M. F. Wiskoff, "Espionage against the united states by american citizens 1947-2001," DTIC Document, Tech. Rep., 2002.

[12] S. R. Band, D. M. Cappelli, L. F. Fischer, A. P. Moore, E. D. Shaw, and R. F. Trzeciak, "Comparing insider it sabotage and espionage: A model-based analysis," DTIC Document, Tech. Rep., 2006.

[13] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[14] M. Goldstein and A. Dengel, "Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm," in *Proc. of the 35th German Conference on Artificial Intelligence (KI'12), Saarbruecken, Germany*, S. Wolfl, Ed., September 2012, pp. 59–63.

[15] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2013. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TKDE.2013.184

[16] Y. Xie and S.-Z. Yu, "A large-scale hidden semi-markov model for anomaly detection on user browsing behaviors," *IEEE/ACM Transactions on Networking*, vol. 17, no. 1, pp. 54–65, February 2009.

[17] C. Brax, L. Niklasson, and M. Smedberg, "Finding behavioural anomalies in public areas using video surveillance data," in *Proc. of the 11th IEEE International Conference on Information Fusion (FUSION'08), Cologne, Germany*. Cologne, Germany: IEEE, June 2008, pp. 1655–1662.

[18] M. B. Salem and S. Stolfo, "Masquerade attack detection using a search-behavior modeling approach," Department of Computer Science, Columbia University, Tech. Rep. CUCS-027-09, 2009.

[19] P. Thompson, "Weak models for insider threat detection," in *SPIE Proc. of Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III*, vol. 5403, September 2004, pp. 40–48.

[20] P. G. Bradford and N. Hu, "A layered approach to insider threat detection and proactive forensics," in *Proc. of the 21st Annual Computer Security Applications Conference (Technology Blitz), Tucson, Arizona, USA*. IEEE, December 2005. [Online]. Available: http://www.acsa-admin.org/2005/techblitz/hu.pdf

[21] S. Wasserman, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

[22] R. S. Burt, *Structural holes: The social structure of competition*. Harvard university press, 2009.

[23] M. K. Sparrow, "The application of network analysis to criminal intelligence: An assessment of the prospects," *Social networks*, vol. 13, no. 3, pp. 251–274, 1991.

[24] K. M. Carley, J. Reminga, and N. Kamneva, "Destabilizing terrorist networks," *Institute for Software Re-*

search, p. 45, 1998.

[25] "Cyber-Ark press release, November 23, 2009." http://www.cyber-ark.com/news-events/pr_20091123.asp, retrieved February 14, 2012.

[26] "Computer Security Institute (CSI) computer crime and security survey, 2007." http://i.cmpnet.com/v2.gocsi.com/pdf/CSISurvey2007.pdf, retrieved February 14, 2012.

[27] "Computer Security Institute (CSI) computer crime and security survey, 2008." http://www.docstoc.com/docs/9484795/CSI-Computer-Crime-and-Security-Survey-2008, retrieved February 14, 2012.

[28] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, no. 1, 1972.

[29] J. Beirlant, E. J. D. L. Györfi, and E. C. V. der Meulen, "Nonparametric entropy estimation: An overview," *International Journal of Mathematical and Statistical Sciences*, vol. 6, pp. 17–40, 1997.

[30] R. S. Burt, J. E. Jannotta, and J. T. Mahoney, "Personality correlates of structural holes," *Social Networks*, vol. 20, no. 1, pp. 63–87, 1998.

[31] K. M. Carley, "Dynamic network analysis," in *Dynamic social network modeling and analysis: Workshop summary and papers*. The National Academies Press, 2003, pp. 133–145.

[32] M. Keeney, *Insider threat study: Computer system sabotage in critical infrastructure sectors*. US Secret Service and CERT Coordination Center, 2005.

Author Biography

Hoda Eldardiry is a research scientist at Palo Alto Research Center (PARC). She is a member of the Model-Based Reasoning group at PARC's Intelligent Systems Laboratory. Eldardiry's research is in the areas of machine learning, data mining, knowledge discovery and human-computer team intelligence, with a focus on statistical relational learning, information fusion and anomaly detection. She has developed techniques for fraud detection on medical insurance claims, models for insider threat detection, and algorithms for activity prediction. She has also developed techniques for multi-source information fusion and collective classification on social networks. Hoda earned her Ph.D. and M.S. in Computer Science from Purdue University, and her Bachelor's degree in Computer and Systems Engineering from Alexandria University in Egypt.

Kumar Sricharan currently focuses on statistical machine learning and data mining methods at the Palo Alto Research Center [PARC], with applications to anomaly detection and pattern recognition in multivariate, temporal, and relational data. His research interests include statistics, machine learning, data mining, and signal processing with specific focus on ensemble methods, nonparametric statistics and large sample estimation theory. Prior to PARC, Kumar was a research engineer at NASA Ames, where he conducted research on mining aviation data for anomalies with respect to fuel consumption efficiency and aviation safety. Kumar earned his Ph.D. in Electrical Engineering, Systems in 2012, M.A. in Statistics in 2011, and M.S. in Electrical Engineering: Systems in 2009, all from the University of Michigan, Ann Arbor. His doctoral work on efficient estimation of probability density functionals using neighborhood graphs has resulted in publications in esteemed peer-reviewed conferences and journals, and was nominated for the best dissertation award by the University of Michigan. He received his B.Tech degree in Electrical Engineering from IIT Madras in 2006.

Juan Liu is a Senior Research Scientist and manages the Learning, Inference, and Data Solutions (LIDS) Area at PARC. She leads a portfolio of machine learning and data mining projects. Her recent projects include DARPA Anomaly Detection at Multiple Scale (ADAMS) (4-yr, $6.4M) and Detection of Fraud, Waste and Abuse from transactional data (leading a team of 12 researchers at PARC). Her research interest includes machine learning, big data, statistical modeling and inference. She received her Ph.D. degree in Electrical Engineering from the University of Illinois at Urbana-Champaign in 2001. She is the recipient of IEEE Signal Processing Society Best Young Author Paper Award of 2002. She has published more than 50 papers in the field of statistical modeling and inference, and is the holder of more than 20 patents.

John Hanley works on large software systems deployed in situations spanning the breadth of PARC's customers, from decision support to aerospace applications. Prior to his current work in Embedded Reasoning, John contributed to a PARC and Xerox research effort aimed at retrieving documents from a personal library exactly when they become relevant. John holds a Masters in Software Engineering from Carnegie Mellon University.

Bob Price works on inference, tracking, learning, and planning applications for government and industry clients. He has developed techniques for fraud detection on social services payments, algorithms for learning cell phone user preferences from GPS traces, and models for predicting the life of machine tools. He has also developed techniques for tracking in military domains. He earned his MSc in Computer Science at the University of Saskatchewan and Ph.D. in Computer Science from the University of British Columbia.

Oliver Brdiczka manages PARC's Contextual Intelligence research area. He focuses on constructing models for human activity from various sensors – ranging from PC desktop events to physical activity sensors – by employing machine learning methods. He aims to enable context-aware applications and services that understand human activity and anticipate human needs by leveraging derived intent and goals. Oliver received his Ph.D. in Computer Science and M.S. in Imagery, Vision, and Robotics from INP Grenoble.

Eugene Bart focuses on high-level understanding and interpretation of images and has led to novel theories of human vision. He developed a system that can recognize a novel face from just a single photograph across significant variations in viewpoint, and also introduced a method of learning a novel category from a single example. Eugene earned his M.S. and Ph.D. in computer science from the Weizmann Institute.