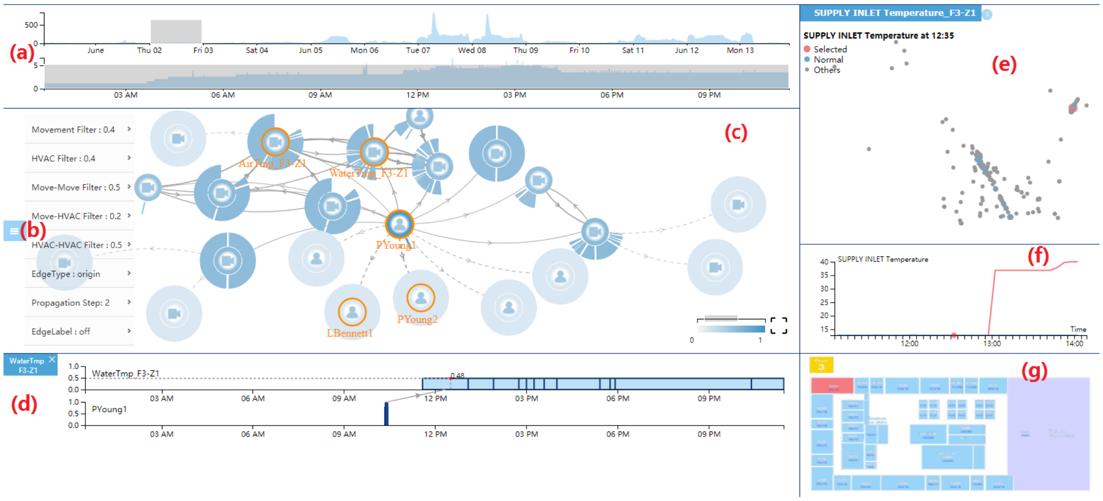
Visual Analysis of Collective Anomalies Through High-Order Correlation Graph

Jun Tao[1] Lei Shi[2] Zhou Zhuang[3] Congcong Huang[2] Rulei Yu[2] Purui Su[2]
Chaoli Wang[1] Yang Chen[3]

[1]University of Notre Dame, [2]SKLCS, Chinese Academy of Sciences and UCAS, [3]Fudan University



Movement Filter : 0.4
HVAC Filter : 0.4
Move-Move Filter : 0.5
Move-HVAC Filter : 0.2
HVAC-HVAC Filter : 0.5
EdgeType : origin
Propagation Step: 2
EdgeLabel : off

Figure 1: The visualization interface of high-order correlation graph (HOCG): (a) double overview+detail timeline selectors; (b) visualization controller; (c) correlation graph view; (d) the anomaly time series of individual nodes (objects); (e) visual interpretation of a selected point anomaly event; (f) the data value of the selected anomaly; (g) spatial detail view.

Abstract

Detecting, analyzing and reasoning collective anomalies is important for many real-life application domains such as facility monitoring, software analysis and security. The main challenges include the overwhelming number of low-risk events and their multifaceted relationships which form the collective anomaly, the diversity in various data and anomaly types, and the difficulty to incorporate domain knowledge in the anomaly analysis process. In this paper, we propose a novel concept of high-order correlation graph (HOCG). Compared with the previous correlation graph definition, HOCG achieves better user interactivity, computational scalability, and domain generality through synthesizing heterogeneous types of nodes, attributes, and multifaceted relationships in a single graph. We design elaborate visual metaphors, interaction models, and the coordinated multiple view based interface to allow users to fully unleash the visual analytics power over HOCG. We conduct case studies in two real-life application domains, i.e., facility monitoring and software analysis. The results demonstrate the effectiveness of HOCG in the overview of point anomalies, detection of collective anomalies, and reasoning process of root cause analysis.

Index Terms: correlation graph visualization, collective anomaly

1 Introduction

Anomaly detection is a critical interdisciplinary research area [6] that expands its applications in many strategic domains, such as intrusion detection, fraud analysis, and software security. If not well contained, the anomalous behavior often translates to hazardous, fatal actions, such as the compromise of machines for potential attacks, or terrorist activities in real life. In this work, we consider one of the most complicated anomaly types, namely *collective anomaly*. The collective anomaly is identified as coordinated events on a group of interrelated objects, which individually behaves normally, but their co-occurrence is seen as highly anomalous. For example, in the software analytics scenario, the stack-overflow and the call function transfer itself can be just programming tricks or low-risk software bugs. When these two events happen sequentially, the normal operation upgrades severely to a malicious attack of code injection through the exploitation of software vulnerabilities. Another example is the denial of service (DoS) attack to web servers [2]. While a single request to a server is legitimate, numerous connection requests occurring simultaneously may indicate a collective anomaly.

The detection of collective anomaly is challenging, mainly because the anomalous behavior is not only revealed by each individual event (known as point anomalies), but also depends heavily on the relationship among these events. The combination of point anomalies with their relationship leads to an explosion of potential states to examine for anomaly detection algorithms. To overcome this data proliferation, most previous approaches for the collective anomaly detection problem focus on a single type of data relationship, such as sequential [5], spatial [12], and graph relationship [22]. For each type of data, they reduce the data objects and their relationship into a finite feature space, and apply point anomaly detection algorithms to resolve. Therefore, these techniques are often limited to a single type of data and problem.

On the other hand, visualizations have been widely developed for the purpose of anomaly detection, such as the correlation graph for agnostic anomaly detection in wireless sensor networks [20, 27], spatiotemporal anomaly detection [30] and information diffusion anomaly visualization over social media [33]. These visualization approaches, either directly visualize the raw data set and do not scale to big data, or are specially designed for certain domain and do not generalize to the case of generic collective anomaly.

In this paper, we study the problem of designing a collective anomaly detection technique that meets the following three objectives simultaneously. First, to adapt to the versatility of collective anomalies, the technique should bring users in the loop to combine the power of automatic computation and human analytics to detect previously unknown collective anomalies. Second, the technique should scale to support huge data volume and a variety of data types, such as time series, sequential, and spatial data. Third, the technique should be generic to support the collective anomaly detection job in different domains and be able to incorporate prior domain knowledge in normal and abnormal data models.

Motivated by the above problem, we propose a novel concept of *high-order correlation graph* (HOCG), which is defined at the multivariate event-level, beyond its lower-order ancestor over univariate data variables [20]. Compared with existing collective anomaly detection methods, HOCG enjoys several advantages. First, *interactivity*: HOCG is fully customizable by users and provides the flexibility to analyze data objects and their relationship for unknown collective anomaly. Second, *scalability*: through the introduction of temporal and anomaly score filtering, and the object-centric abstraction, a large HOCG can be greatly reduced in the overview, while allowing the access of spatial, temporal, and anomaly details upon user interactions. Third, *generality*: the construction of HOCG follows principled analytics framework that can be generalized to different domains and data types, while incorporating the user's knowledge through domain-specific anomaly detection algorithms and configurations. Our contributions can be summarized as below.

- We formally define HOCG in a domain and data type independent way. A principled yet flexible framework is proposed to construct HOCG by integrating point anomaly detection, multifaceted correlation analysis, and anomaly propagation methods.
- We design a visual analytics system to overview the large HOCG through visual abstractions. The system supports several interaction models to validate individual point anomalies, visually detect collective anomaly, and finally conduct root cause and dynamic analysis for containment actions.
- The proposed HOCG concept and the visual analytics system are evaluated through two case studies in facility monitoring and software analysis domains. The case study result and the feedback from domain experts demonstrate the effectiveness of the system in the visual reasoning of collective anomalies.

2 Related Work

2.1 Anomaly Detection Algorithms

Anomaly detection has been extensively studied during the past decades. For a thorough understanding of the literature, we refer readers to the surveys [2,3,6,26,32]. Many types of approaches have been proposed, including classification based techniques [7], nearest neighbor based techniques [4], clustering based techniques [8], statistics based techniques [9], and information theoretic techniques [18].

Closely related works to our approach are the anomaly detection methods in sensor networks which also depend on the graph structure. These approaches can be classified into prior-knowledge based approaches [19, 24] and prior-free approaches [14, 21, 23]. The prior-knowledge based approaches require assumptions or experience to provide a normal profile. For example, Liu et al. [19] assumed that the Mahalanobis squared distance between networking attributes was subject to the chi-squared distribution. In contrast, the prior-knowledge free approaches usually produce a normal profile through a training procedure. For example, Khanna et al. [14] applied a genetic algorithm to measure the fitness of nodes.

In comparison, our point anomaly detection method adopts a hybrid strategy: it can take normal profiles for a higher accuracy, and it can also be prior-knowledge free when normal profiles are unavailable. Meanwhile, our collective anomaly method relies on the human intervention through visual analysis and does not fall into the algorithm-centric categories.

2.2 Visual Analytics for Anomaly Detection

Developing visual analytic approaches for anomaly detection has gained increasing attention in the visualization community. Many systems are developed for the anomalies in a variety of applications. Fischer et al. [10] visualized attacks on the large-scale network by mapping the monitored network as a treemap and the attacking host as an isolated node. They did not provide mechanism to identify anomalous events but relied on an additional intrusion detection system. Teoh et al. [29] applied a statistical model to detect anomalies in the Border Gateway Protocol. The anomaly score of each event is visualized by line graphs and a series of circles indicating the time and signatures of the event. Liao et al. [16] developed GPSva, a visual analytic system to study anomalies in GPS streaming traces. The anomalies are detected using conditional random field and visualized on a map. Shi et al. [27] proposed multiple designs to visualize and analyze anomalies in sensor networks to allow different aspects of data to be investigated. The temporal expansion model graph displays the network as a directed tree; the correlation graph visualizes the correlations among attributes; and the dimension projection graph maps the sensor nodes to a scatterplot. Liao et al. [15] further extended this work to consider the membership changes of node communities, so that anomaly detection is less sensitive to the activity of each individual node. Thom et al. [30] detected and visualized spatiotemporal anomalies based on geo-located twitter messages. A cluster analysis approach is used to distinguish global and local messages. The aggregated messages are then visualized as term clouds on a geographic map. Zhao et al. [33] developed #FluxFlow to visually analyze anomalies in the information diffusion over social media. The anomalous retweeting threads are detected using one-class conditional random fields model. The users involved in the anomalous threads are visualized as circles inside a streamgraph. Coordinated multiple views are designed to allow anomaly detection in both overview and details.

Among these literature, the correlation graph proposed in Ref. [27] is the closest to ours. However, the correlation graph only considers one sensor and one type of relationship, while our approach scales to analyze the interactions among multiple types of nodes and their multifaceted relationship by visually synthesizing all these information in a single high-order correlation graph. Therefore, our method is more suitable to analyze the collective anomaly.

3 Problem Description and Requirements Analysis

Our goal is to develop a visual analytic system that could help detect, analyze, and reason about collective anomalies on a group of interrelated objects from their observed behaviors. In this section, we will start from formally defining the problem to be addressed and the corresponding challenges, and then provide a detailed requirement analysis of solving these problems in a typical application domain.

3.1 Problem Description

We consider a group of *objects* (e.g., sensors, persons, computer programs, etc.), whose behaviors are captured by a set of *event* data (e.g., measured values from sensors, movement of persons, execution of programs, etc.), and the objects are interrelated by *multifaceted relationships* (e.g., sensors' spatial/temporal/behavior closeness, persons' role similarity, etc.).

Here the single event on an object is formatted as the 4-tuple: {object, space, time, measured value} (see formal notations in Section 4.1). Normally, the amount of the event data is huge as the target objects are often measured on a real-time, continuous basis. This provides the possibility to detect abnormal events, i.e., which object behaves anomalously and when and how by comparing the extracted suspicious behaviors with the large amount of normal behavior of

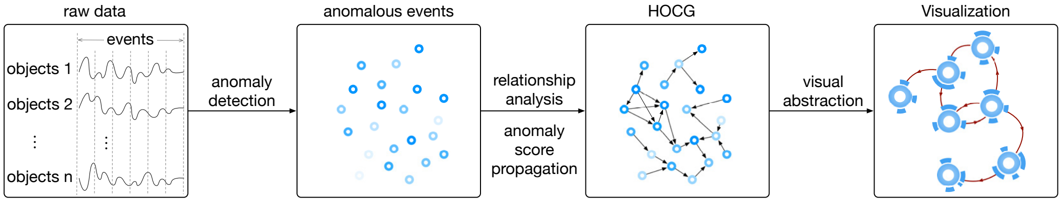


Figure 2: The workflow of our framework for analysis of collective anomalies.

this and other objects. Two levels of anomalies are considered: the traditional *point anomalies* defined by the abnormal events on a single object, and the advanced type of *collective anomalies* by synthesizing the point anomalies on multiple interrelated objects. In this work, we focus on the analysis of collective anomalies, for which the event on a single object may not be highly anomalous by itself, but several coordinated events occurring together on distributed objects can raise the anomaly level and become noteworthy.

To visually detect, analyze, and reason about collective anomalies, the following problems should be addressed.

P1. Rate individual anomalous events. Instead of classifying each event as a point anomaly or not, for our problem there should be an anomaly score calculated on each event to indicate how anomalous it is. The anomaly score serves two purposes: first, it allows to identify the moderately anomalous events as well, in order to detect the collective anomalies. Second, it provides a criterion for users to rank and filter the anomalous events independent of the data type. Users can integrate their domain knowledge to make decision on whether an event is anomalous, and finally compose and identify collective anomaly.

P2. Understand relationship among events. Given that the collective anomaly is composed of multiple interrelated events, it becomes critical to answer the question: are two events related to each other? We should consider the measured value on events as well as the underlying objects' attributes and intrinsic relationships, e.g., spatial, temporal, and categorical closeness of objects, and whether two objects demonstrate frequent interactions in history. This allows to correlate objects and events in different types.

P3. Identify and interpret collective anomalies. Knowing the anomaly scores of individual events and their relationships, the next problem will be how to identify collective anomalies and visually interpret them. In this paper, we consider two types of collective anomalies: a group of strongly interrelated events that are moderately anomalous; and a group of events that show strong connections to another highly anomalous event. The former type identifies the hidden collective anomalies that cannot be discovered by point anomaly detection alone, while the latter type enables the root cause analysis after the anomaly detection. A unified design should be proposed to represent these two anomaly types simultaneously and also resolves the scalability issue as the number of individual events is huge.

3.2 Requirement Analysis

We showcase the requirement for the visual analysis of collective anomalies in the typical scenario of facility monitoring. The facility monitoring considers two types of objects: sensors and employees. There are multiple types of sensors, e.g., to monitor the status of room heating, ventilation, and air conditioning. The behavior of each sensor is captured by their measured values. On the other hand, the behavior of employees is captured by their movements (i.e., measured locations). Detecting suspicious coordinated activities from the behavior data is one of the major tasks for facility monitoring. This can be perfectly achieved by our visual analytics system for collective anomalies. In details, facility monitoring users need to complete the following tasks with our system:

R1. Overview. Two levels of overview should be obtained: first, the overview of anomalous activities over time. For example, when do sensors/employees exhibit suspicious behaviors? With this overview, users can quickly narrow down to a specific time window for exploration. Second, the overview of all point anomalies within a selected time window. For example, which anomalies have higher anomaly scores than the others and which anomalies last longer? Users also need an overview of relationship among all point anomalies as well. For example, which anomalies have more connections to the others and which group of anomalies involve more objects?

R2. Validation of point anomalies. Once the suspicious objects and events are noticed in the overview, the system should allow users to validate them by comparing with the normal behavior data. For example, if a sensor reads an abnormal value, the system should present all the other normal values, as well as their spatial and temporal information. Users then can make the final judgment on the anomaly by incorporating their domain knowledge with the provided information.

R3. Exploration of connections among point anomalies. The system should allow to discover the relationship among point anomalies. In details, given an anomalous object, what are the associated anomalous events and all the other related objects; given an anomalous event, what are the related objects and events? For example, when a sensor reads an abnormal value, the system should help to reason the event, i.e., which equipment and/or person lead to this anomaly. Examining the related events will help users to identify the root cause and potential impact of anomalies. More importantly, the interrelated point anomalies provide a visual hint for users to identify the collective anomalies.

R4. Preserving collective anomalies during anomaly filtering. The system should allow the anomalies to be zoomed and filtered. While time and anomaly score can be used to filter point anomalies, the relationships between events should also be considered to preserve the intactness of collective anomalies. Otherwise, the events not highly anomalous may be filtered out. For example, when an employee performs a deliberate harmful action, he is likely to disguise himself and behave normally. To identify these type of events, the system should help to trace back from the detected anomalies using the relationships among the events.

4 Analysis Framework for Collective Anomalies

4.1 Overview

We propose a novel concept of high-order correlation graph (HOCG) to visually analyze collective anomalies. As shown in Figure 2, the HOCG preserves the node-link graph structure. Compared with the original correlation graph [27], HOCG is high-order in two aspects: first, each node in HOCG is an event associated with multiple attributes beyond the single measured value in the correlation graph, e.g., the space, time, object category information of the event, and most importantly, its anomaly score; second, the relationship between two nodes (events) is high-order, since there are multifaceted correlations derived between the two events, including their spatial, temporal, categorical, and historical correlations. The HOCG concept is better illustrated in the formal notation.

As shown in Figure 2, the analysis framework by HOCG consists of three stages: *anomaly detection*, *relationship analysis*, and *visual abstraction*. In the first stage, the anomaly detection assigns each event an anomaly score, as indicated by the fill color of the event circles. The relationship analysis in the second stage discovers

Table 1: Notations used in this paper.

SYMBOL	DEFINITION
$\Phi =< o, s, t, v >$	an event
$\alpha(\Phi) = A(v)$	the anomaly score of an event
$\rho(\Phi_i, \Phi_j)$	the high-order correlation between two events
$\Phi(o_i, \mathbf{T})$	events related to an object o_i in a time span $\mathbf{T}$
$\gamma(o_i, o_j, \mathbf{T})$	the historical correlation between two objects o_i and o_j in a historical period $\mathbf{T}$
$\mathbf{H} = (\mathbf{V}, \mathbf{E})$	high-order correlation graph (HOCG)
$\mathbf{H}(\mathbf{T}) = (\mathbf{V}(\mathbf{T}), \mathbf{E}(\mathbf{T}))$	dynamic HOCG in a time span $\mathbf{T}$
$\mathbf{H}^+ = (\mathbf{V}^+, \mathbf{E}^+)$	augmented HOCG

the multifaceted correlations among events and construct the raw HOCG. A historical correlation graph is also generated to describe the latent relationships among objects of the HOCG events. The latent relationships allow us to identify the hidden and collective anomalies through propagating the anomaly scores on the historical correlation graph. Finally, the raw HOCG is abstracted over time and in an object-centric way for efficient, compact visualization.

In general, HOCG provides a foundation to solve the problems in Section 3.1 and fulfill the requirements in Section 3.2. First, it synthesizes all event attributes and their multifaceted relationships in a single graph, so that the relationships among different event types can be understood (**P2**, **R3**), and the anomaly scores of individual events can be evaluated (**P1**). Second, propagating the anomaly scores on HOCG increases the anomaly scores of the hidden and collective anomalies, allowing them to be discovered (**P3**, **R3**, and **R4**). Third, the HOCG abstraction, together with the visualization interface, supports the user discovery of collective anomalies over a large number of heterogeneous events (**R1** and **R2**).

Notations. The notations used throughout this paper are listed in Table 1. Each event is a tuple $\Phi =< o, s, t, v >$, recording its four attributes: the associated object o (e.g., a sensor, a person, or a program), spatial region s, time duration t, and a series of measured values v in t. Each event is assigned an anomaly score $\alpha(\Phi) = A(v)$, determined by its behavior difference from other relevant events. The events with high anomaly scores, indicating that they behave differently from others, are identified as the point anomalies. The relevance between two events Φ_i and Φ_j is described by their *high-order correlation*, which is the fusion of three types of correlations: $\rho(\Phi_i, \Phi_j) = F(\rho_S(s_i, s_j), \rho_T(t_i, t_j), \rho_C(o_i, o_j))$, where $\rho_S(s_i, s_j)$, $\rho_T(t_i, t_j)$, $\rho_C(o_i, o_j)$ are the spatial, temporal, and categorical correlations between the two events, respectively, and F is a customizable fusing function. The fusing function allows different aspects of correlation to be emphasized in the analysis. The fused correlation reflects the relevance between the two events. In addition, to discover the latent relationships between two objects o_i and o_j in a historical period T, we introduce the historical correlation $\gamma(o_i, o_j, T)$.

The high-order correlations among all events are organized into a HOCG, defined as $\mathbf{H} = (\mathbf{V}, \mathbf{E})$, where each vertex is an event and each edge is a high-order correlation between two events. To provide a compact description of the anomalies and their relationships, a dynamic HOCG $\mathbf{H}(\mathbf{T}) = (\mathbf{V}(\mathbf{T}), \mathbf{E}(\mathbf{T}))$ will be generated during the exploration, where $\mathbf{T}$ is a users-specified time span to filter the original HOCG $\mathbf{H}$. Finally, to identify the hidden and collective anomalies, we extend $\mathbf{H}$ to include events that are closely related to the detected point anomalies. The augmented HOCG is denoted as $\mathbf{H}^+ = (\mathbf{V}^+, \mathbf{E}^+)$.

4.2 Point Anomaly Detection

The point anomaly detection discovers the object's suspicious behaviors on their own by analyzing their event data. In general, each event is compared to the other related events belonging to the same object category using a distance function, by which the anomaly score is computed on the target event. In more detail, all the events are classified into two event types according to the nature of object categories: events with normal profiles and events without normal profiles. For example, the operational sensor data on facility monitoring [1] are considered to be events with normal profiles, since the range of regularly measured value (e.g., the power consumption of air conditioners) can be identified by domain knowledge. In contrast, the employee movement data are considered to be events without normal profiles, as it is difficult to accurately predict the everyday activity of all the employees. Based on these two event types, we have designed separate anomaly detection methods.

Events with Normal Profiles. To identify anomalies in this type of events, we utilize the knowledge from users to select a set of sampled normal events $\{\Phi_{n_1}, \ldots, \Phi_{n_m}\}$. The anomaly scores of other events are then derived from their relationships with these normal events. Each sampled normal event Φ_{n_i} associates with a Gaussian distribution $\mathcal{N}(\Phi_{n_i}, \sigma^2_{n_i})$, and whether an event Φ_j is normal compared to Φ_{n_i} is given by the probability

$$p(\Phi_j|\Phi_{n_i}, \sigma^2_{n_i}) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{d(\Phi_j, \Phi_{n_i})}{2\sigma^2}}, \quad (1)$$

where $d(\Phi_j, \Phi_{n_i})$ is the distance between events Φ_j and Φ_{n_i}. Φ_{n_i} denotes the expectation of this distribution, and indicates that an event is more likely to be normal when its distance to Φ_{n_i} is smaller. The variance $\sigma^2_{n_i}$ is determined by the sparsity of normal events around Φ_{n_i}, given by the average squared distance from Φ_{n_i} to its k-nearest neighbors (kNNs). Intuitively, a higher density of normal events around Φ_{n_i} leads to smaller variance and higher probability of neighboring events being normal. On the other hand, a lower density leads to larger variance, indicating less confidence in rating neighbors as normal. In our experiments, we use a fairly large k of 50 since the normal events usually have dense neighborhoods. Finally, the anomaly score of an event Φ_j is calculated by

$$1 - \frac{1}{k} \sum_{\Phi_{n_i} \in \mathbf{n}(\Phi_j)} p(\Phi_j|\Phi_{n_i}, \sigma^2_{n_i}), \quad (2)$$

where $\mathbf{n}(\Phi_j)$ is the kNNs of Φ_j in the sampled normal events. The use of a set of sampled normal events can be considered as an approximation of Gaussian mixture model describing multiple patterns of normal events. Note that the distance definition may vary for different kinds of data. For example, each sensor event $\Phi_i =< o_i, s_i, t_i, v_i >$ is associated with a series of measured scalar values v_i from the sensor o_i. The distance between two sensor events is defined as the Euclidean distance between the two series of scalar values.

Events without Normal Profiles. For some object categories, it is difficult to identify normal events using domain knowledge. In this case, we first identify an average event for each object category, and then compute the anomaly score of an event as its distance to the average event. For example, each movement event $\Phi_i =< o_i, s_i, t_i, v_i >$ records the movement v_i of an employee o_i in a day t_i. We compute a histogram of the movement event Φ_i where each bin is the total time that the employee o_i stays in a zone. The movement event Φ_i is compared to two average events: first, an average event defined as the average histogram of all employees in the same department (category) on the same day; second, an average event defined as the average histograms of this employee o_i in all days. The difference between two histograms is measured by Jensen-Shannon divergence [17].

4.3 Correlation Analysis

Correlation analysis determines the relevance between individual events, which is crucial to identify the collective and hidden anomalies from point anomalies on interrelated objects. Specially, the correlation between the 4-tuple event data is multifaceted in that both the attributes of space, time, object category and the object in history can be related with each other. We describe each of these correlations below. Finally, these multifaceted correlations are fused together to form the high-order relationship in HOCG.

Spatial Correlation between Events. Spatial correlation evaluates the location closeness of two events. We rely on domain knowledge to build a hierarchy of spatial regions and determine the spatial correlation based on the probability of two events occurring in the same region. For example, consider the facility monitoring data in a three-floor building, where each floor is partitioned into multiple zones and each zone contains multiple rooms. We use $\rho_S = 1$ for two events occurring in the same room, $\rho_S = p_{\text{room}}/p_{\text{zone}}$ for two in the same zone, $\rho_S = p_{\text{room}}/p_{\text{floor}}$ for two on the same floor, and $\rho_S = 0$ for events that do not share regions at any level, where p_{room}, p_{zone}, and p_{floor} are the probabilities of two events being in the same rooms, the same zones, and the same floors, respectively. An exception is that the spatial correlation between an event in the server room and any other event is at least 0.5, since the air conditioning equipment in the entire building can be controlled in the server room.

Temporal Correlation between Events. Temporal correlation evaluates the closeness of their time durations using Pareto distribution with zero tail, which gradually approaches zero when the value of the random variable increases. Depending on the object category, we may consider the overlapping duration of two events or the starting time difference. For example, for two sensor events with causal relationships, the resulting event is not possible to occur much later than the cause. Therefore, the difference of starting times is more important. In this case, temporal correlation is formulated as

$$\rho_T = \begin{cases} 1, & \text{if } \Delta T \le T_{\min} \\ (T_{min}/\Delta T)^{\beta_T}, & \text{if } T_{\min} < \Delta T < T_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where ΔT is the difference of starting times between two events, $T_{\min}$, $T_{\max}$, and β_T are three user-specified parameters to determine two events are fully related or completely irrelevant, respectively. For two movement events, we consider the overlapping duration to be more important and formulate temporal correlation as

$$\rho_T = \begin{cases} 1, & \text{if } T_o \ge T_{\max} \\ \left(\frac{T_o - T_{\min}}{T_{\max} - T_{\min}}\right)^{\beta_T}, & \text{if } T_{\min} < T_o < T_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where T_o is the length of overlapping duration, and $T_{\min}$, $T_{\max}$, and β_T are user-specified parameters.

Categorical Correlation between Events. Categorical correlation evaluates whether the objects of two events are similar given by the user-specified domain knowledge. Similar to spatial correlation, we assign different weights to different levels of category. For example, sensor objects and movement objects are the two categories at the highest level. Sensor objects can be further be partitioned into heating-related, air circulation-related, and power-related, and movement objects (employees) can further be grouped by their departments.

Historical Correlation between Events. The historical correlation can be considered as a supplement to the categorical correlations, capturing the latent relationships between objects. The historical correlation of two objects aggregates the correlation between all related events in the historical records. The exact definition relies on the specific data being processed.

For the facility monitoring, we compute the historical correlation between two sensor objects and two movement objects differently. For two sensor objects, it is critical to reveal the causal relationship between their corresponding events. If the events of one object frequently result in the events of another object, we consider that the two objects are closely related and have high historical correlation. The causal relationship between two events is measured by the cross correlation of their corresponding value series

$$\rho_{cc}(\Phi_i, \Phi_j) = \max_{\tau' \in \mathbf{T_{cc}}} \| \frac{1}{|\mathbf{t_i}|} \sum_{\tau \in \mathbf{t_i}} \frac{(v_i(\tau) - \bar{v_i})(v_j(\tau + \tau') - \bar{v_j})}{\sigma_{v_i}\sigma_{v_j}} \|, \tag{5}$$

where τ is an offset applied to v_i in the duration t_i of event Φ_i, τ' is an offset applied to v_j in a user-specified range T_{cc}, and $\bar{v_i}$, $\bar{v_j}$, σ_{v_i} and σ_{v_j} are the averages and standard deviations of v_i and v_j , respectively. In our implementation, we use [-1 hour, 1 hour] for T_{cc}. The historical correlation of two objects o_i and o_j over a historical period $\mathbf{T}$ is given by the maximum cross correlation between their corresponding events

$$\gamma(o_i, o_j, T) = \max_{\Phi_a \in \Phi(o_i, \mathbf{T}), \Phi_b \in \Phi(o_j, \mathbf{T})} \rho_{cc}(\Phi_a, \Phi_b), \tag{6}$$

where $\Phi(o_i, \mathbf{T})$ and $\Phi(o_j, \mathbf{T})$ are all events related to objects o_i and o_j, respectively, in the period $\mathbf{T}$.

For two movement objects, their coincidence in the same region is a more important factor. In this case, the historical correlation of two movement objects o_i and o_j over a historical period $\mathbf{T}$ is given by the summation of overlapping durations of their corresponding events weighed by their spatial correlation

$$\gamma(o_i, o_j, \mathbf{T}) = \sum_{\Phi_a \in \Phi(o_i, \mathbf{T}), \Phi_b \in \Phi(o_j, \mathbf{T})} \rho_S(\Phi_a, \Phi_b) \| t_a \cap t_b \|. \tag{7}$$

The spatial correlation is involved to emphasize the periods that two objects stay close to each other.

Fusing of multifaceted correlations to HOCG. Multiple fusing functions are provided to allow users to focus on different aspects of correlation. A *uniform fusing*

$$\rho_F = \begin{cases} \rho_S + \rho_T + \rho_C, & \text{if } \rho_S \ne 0 \text{ and } \rho_T \ne 0 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

is the summation of spatial, temporal, and categorical correlations when both the spatial and temporal correlations are not zero. To emphasize the impact of time, a *time-critical fusing* is provided by multiplying the resulting correlation of general fusing by temporal correlation to some degree, i.e., $\rho_{TF} = \rho_T^{P_T} \rho_F$, where P_T is a user-defined parameter. Similarly, *space-critical*, *object-category critical*, and *space-time critical* fusing can be achieved through multiplying the uniform fusing result by the respective correlations.

4.4 Anomaly Score Propagation

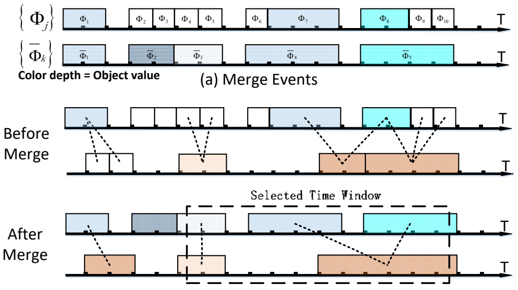
To tackle the problem **P3** in Section 3.1 and fulfill the requirement **R3** in Section 3.2, we need to raise the anomaly scores for: 1) events that are closely related to high anomalous ones for root cause analysis; and, 2) multiple strongly interrelated anomalous events for collective anomaly detection. To this end, we leverage the random walk with restart on the historical correlation graph to propagate the anomaly scores from the point anomalies to other events.

Historical correlation graph construction. This graph is directed, where each node is an object, and each edge is associated with a relative anomaly score $A(o_i|o_j)$ indicating the probability of object o_i being anomalous given that object o_j is abnormal. We formulate this relative anomaly score as the historical correlation between o_j and o_i divided by the total historical correlation between o_i and any object in the historical period $\mathbf{T}$

$$A(o_j|o_i) = \frac{\gamma(o_j, o_i, \mathbf{T})}{\sum_{o_k \in \mathbf{O}} \gamma(o_k, o_i, \mathbf{T})}. \tag{9}$$

where $\mathbf{O}$ denotes all objects in $\mathbf{T}$. In this construction, the network is not symmetric, i.e., $A(o_i|o_j) \ne A(o_j|o_i)$. For example, if o_i only relates to o_j in the historical period T, we consider o_j is likely to be the cause of o_i. Therefore, the value of $A(o_j|o_i)$ should be large. On the other hand, if o_j relates to many objects other than o_i, the value of $A(o_i|o_j)$ will be small, so that o_i will not become an anomalous even if o_j is detected as a point anomaly.

Propagation on HOCG. The propagation starts from the detected anomalies $\mathbf{O_a}$ on the historical correlation graph. At each iteration, the random walk updates the anomaly score of each object based on the anomaly scores of its neighbors and their relative



(a) Merge Events

(b) Object-level Correlation

Figure 3: Merging of events and event correlations over time.

anomaly score. This procedure can be formulated as

$$A^{\tau}(o_i) = \begin{cases} (1-\alpha)NA^{\tau-1}(o_i), & o_i \notin \mathbf{O_a} \\ (1-\alpha)NA^{\tau-1}(o_i) + \alpha RS^{\tau-1}(o_i), & o_i \in \mathbf{O_a} \end{cases} \tag{10}$$

$$\text{where } NA^{\tau-1}(o_i) = \sum_{o_j \in \mathbf{N}(o_i)} A(o_i|o_j)A^{\tau-1}(o_j),$$

$$RS^{\tau-1}(o_i) = \sum_{o_k \in \mathbf{O_a}} A^{\tau-1}(o_i|o_j),$$

where τ is the current iteration number, $\mathbf{N}(o_i)$ is the neighbors of o_i, and α is damping factor. We can consider $NA^{\tau-1}(o_i)$ to be the weighted anomaly scores from neighbors and $RS^{\tau-1}(o_i)$ to be the restart values. The initial anomaly scores $A^0(o_i)$ are the original anomaly scores computed in the point anomaly detection stage.

4.5 Visual Abstraction on HOCG

The original HOCG may contains tens of thousands of nodes (events), which is impractical to visualize and analyze.

Filtering of HOCG. We provide a filtering scheme that allows users to specify a time period $\mathbf{T}$ to generate a dynamic HOCG $\mathbf{H}(\mathbf{T})$ of the original HOCG $\mathbf{H}$. The filtering starts from selecting the events whose corresponding time falls into the period $\mathbf{T}$, i.e., $\{\Phi_i | t_i \in \mathbf{T}\}$, and the edges between these events. In addition, to allow users to focus on anomaly analysis, a threshold of anomaly score is provided to filter the events according to their anomaly scores. The isolated events will be removed as well. However, showing only the anomalies in period $\mathbf{T}$ may not lead to the root cause of these anomalies. Therefore, we augment the dynamic HOCG $\mathbf{H}(\mathbf{T})$ to further include the events closely related to at least one of the selected events. An event Φ_i is considered to be closely related to another event Φ_j, if one of the following two criteria is fulfilled: first, the fused correlation $\rho(\Phi_i, \Phi_j)$ is larger than a user-specified correlation threshold; second, the historical correlation between their corresponding objects $\gamma(o_i, o_j, \mathbf{T}')$ in the historical period $\mathbf{T}'$ before $\mathbf{T}$ is large. The former criterion is used to discover the explicitly connected events and form collective anomalies, and the latter one is used to identify the hidden anomalies whose relationships to the detected ones are not directly available from spatial and temporal closeness or domain knowledge.

Object-centric abstraction. After filtering by anomaly score and time period, a relevant HOCG can be obtained for visualization. Yet, in some cases, the remaining graph is still large in size and complex in structure. To provide users a feasible overview, we propose to visually abstract HOCG according to the host object of each event.

Specifically, on each object o_i, we have retrieved a list of events $\{\Phi_j\}$ that matches the time and anomaly filtering criteria. These events are first merged together over time to form several continuous anomaly intervals, as shown in Figure 3. The merging rule is to combine every pair of consecutive anomalies if they are back to back in the timeline. To maintain consistency, we further cut each interval at time points when the object's value changes. The final anomaly intervals are denoted as $\{\overline{\Phi}_k\}$, which are represented as nodes in the HOCG visualization. On each reconstructed anomaly interval, we compute its anomaly score by a function $\overline{\alpha}(\overline{\Phi}_k)$ over all point anomaly scores in this interval. By default, we apply the max function to reveal the most notable anomaly

$$\overline{\alpha}(\overline{\Phi}_k) = \max_{\Phi_j \in \overline{\Phi}_k} (\alpha(\Phi_1), \cdots, \alpha(\Phi_j)). \tag{11}$$

Among these abstracted nodes, i.e., object-centric anomaly intervals, we form object-level links by aggregating the event-level correlations. As shown in Figure 3, the correlation between two events will be merged into the object-level correlation with two endpoint intervals covering each event in the low-level correlation. By default, the max function is also used to compute the object-level correlation score from their low-level components.

Over the visual abstraction of HOCG, we also support multiple methods to drill-down to its low-level events and correlations, which will be described in Section 5.

5 Visualization

We implement a web-based visualization interface of HOCG, as shown in Figure 1. For more detail, please refer to the video demonstration at `http://lcs.ios.ac.cn/~shil/video/HOCG_PacificVis.mp4`. It is composed of four views: the correlation graph view (Figure 1(c)) that displays the HOCG structure for static anomaly analysis within a certain time window; the double overview+detail timeline selectors (Figure 1(a)) that filter HOCG by sliding the two time windows and empower dynamic analysis on collective anomalies; the event view (Figure 1(d)) that shows the anomaly score time series of one node and helps to examine the root cause of anomalies on that node; and the anomaly detail view (Figure 1(e)(f)(g)) that visually explains the source of each point anomaly and their causal relationships.

5.1 Design Principle

We follow three principles in designing the interface, for the same goal to optimize the visual analysis process on collective anomalies:

- *From macro to micro*: The central idea of this work is to detect, analyze and reason collective anomaly from large amount of low-risk point anomalies. Therefore, it is important to present an overview map of point anomalies first so that users can zoom (on the time axis) and filter (by anomaly and correlation levels) to access the details. Essentially this resembles Shneiderman's visual information seeking mantra [28].
- *From static to dynamic*: On analyzing collective anomalies, both static and dynamic patterns are critical. The static pattern reveals relationship among point anomalies, and the dynamic pattern illustrates their formation and evolution over time. In fact, there is an inherent paradigm in users' analysis process: we observe the static relationship first, and then proceed to discover how it forms, and reason why it develops. Based on this paradigm, the dynamic visualization is built over static views in fixed time windows.
- *Building the reasoning path*: The ultimate goal of our technique is to analyze the root cause of a certain fatal anomaly or failure. This means detecting a primary anomaly path from the fatal anomaly back to the potential root cause. The visualization is therefore designed to help completing this task. We have introduced the interaction to manually inspect point anomalies and the path-based correlation to connect the dots among verified point anomalies.

5.2 Timeline Selectors View

Both point and collective anomalies evolve over time. Therefore, it is important to visualize the dynamics of HOCG to understand the development of anomalies. In our work, we propose an overview+detail design to filter the HOCG according to the selected time window. As shown in the top row of Figure 1(a), a first overview chart is displayed to represent the time series of the number

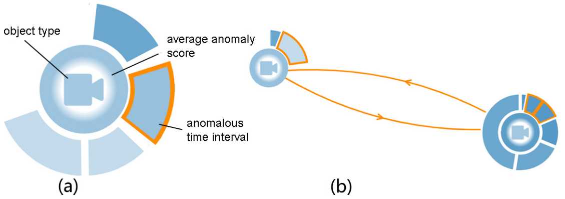


Figure 4: Wedge-based metaphor design: (a) The node composed of multiple anomaly wedges, each wedge corresponds to a time interval having the same anomaly score on this node ; (b) When users hover one wedge on an object, the wedges having correlations with it on the other objects will be highlighted.

of anomalous events above the current anomaly threshold. Users can get a full picture of what is happening on the entire timeline. On the first overview chart, a selection window can be adjusted to specify the detailed time window to examine.

In the bottom row of Figure 1(a), the detailed time window selected in the top row is expanded. To conduct a finer-grained time series analysis, users can choose to select a subset of the currently selected time window, and the HOCG in Figure 1(c) will be filtered to nodes and edges on this subset of time. This double filtering design allows to drill-down to very small time window when some critical anomalies fill in.

5.3 Correlation Graph View

At the center of the interface, as shown in Figure 1(c), the correlation graph view visualizes HOCG as a node-link graph. Each node in the graph represents an object (sensor variable, person, execution of program) that is anomalous in the selected time window, and each edge between two nodes represents their directed relationship from the multifaceted correlation. We employ the stress majorization algorithm implemented in the GraphViz package [11] to compute the layout of HOCG.

For each node, we design a wedge-based metaphor to visualize the anomaly score time series on this object. As shown in Figure 4(a), the visual metaphor is composed of an icon in the center, a filled ring surrounding the icon, and multiple wedges in the outermost ring. The icon in the center represents the node type. For example, the sensor variable is represented by a camera icon, and the person is represented by a people icon. On the surrounding ring, the luminosity of the filled color in the HSL color space indicates the average anomaly score of the object in the selected time window. A larger anomaly score will be displayed in a darker color, thus more noticeable in the visualization. In the outermost ring, each wedge corresponds to an time interval having the same anomaly score. The starting position of each wedge indicates the beginning time of the interval within the selected time window. The radian of each wedge indicates the length of this anomalous time interval. The entire outermost ring corresponds to the whole time window selected in Figure 1(a). In this way, we can interpret the node as a clock with the earliest time mapped to the 12AM position. The wedges are spatially placed on the clock to visualize the temporal distribution of anomalies. The fill color luminosity of each wedge indicates the anomaly score of the time interval, using the same color mapping as the inner ring.

For each edge, the solid edge style indicates the regular high-order relationship computed in Section 4.3; while the dashed edge style indicates extended relationship from the anomaly score propagation (Section 4.4). The edge thickness indicates the fused correlation score. Each edge is directed by comparing the anomalous time interval of the two connecting nodes. By the visual abstraction in Section 4.5, the node with an earlier time interval will point to the other node with a latter time interval. There is also cases that two nodes have bidirectional relationship. To visually represent these edges, we draw curved edges to distinguish the edge direction.

5.4 Event View

On the correlation graph view (Figure 1(c)), users can drill down to each node by a single-click. The anomaly score time series of the clicked node will then be displayed as bar charts in the event view, as shown by the top row in Figure 1(d). To reason about the root cause of anomalies, users can click on another related node that contributes to the anomaly of the previous node. Then another row is added on the bottom with its anomaly bars linked back to the previous anomalies, thus forming a reasoning path. When users click on a new node unrelated to the existing reasoning path, another tab will be opened to show the new path for the root cause analysis.

5.5 Detail View

On the event view (Figure 1(d)), users can further drill down to examine each point anomaly event. It starts with selecting a time point on the anomaly time series of the event view. The corresponding event is then displayed in the detail view on the right part of the interface (Figure 1(e)-(g)). Figure 1(e) shows a scatterplot of all events related to the selected one. The distance between two dots preserves the similarity between the corresponding events. The selected event will be drawn in red. The events known to be normal will be drawn in blue. Other events are drawn in grey. This scatterplot visually explains why the selected event is anomalous by illustrating how it behaves as an outlier in the distribution. In other words, this is a visual interpretation of our point anomaly detection algorithm. Below the projection view, the raw data value of the selected event is displayed. For the sensor data in facility monitoring scenario, we show a time series of 36 data measurements surrounding the selected event (Figure 1(e)), which also compose the vector used in the projection. In addition, the location of the selected event is displayed in Figure 1(g). Note that for different data types, the design of detail view can be customized. For the movement data, we turn to depict the histogram of the selected persons' spatial distributions, as well as the other distributions under comparison.

5.6 Interaction

In terms of interaction, HOCG supports most basic interactions, including zoom&pan, node drag&drop, neighborhood highlight, etc. Specially, when users select one wedge by a mouse hover action in Figure 1(c), this wedge and all the other wedges having direct correlation will be highlighted, as shown in Figure 4(b). In addition, we also introduce three advanced interactions for the visual analysis of collective anomalies. The first is the network-based HOCG filtering. The original HOCG can have a huge amount of nodes/edges, whose visual complexity hampers the analysis. As shown in Figure 1(b), we build node and edge filters that allow users to access point anomalies and relationships over a certain anomaly threshold and correlation score. Note that the filters are arranged by node type (e.g., movement, sensor) and edge type (e.g., mhFilter, according to the node type of two endpoints). The other two network interactions are time-based filtering for dynamic anomaly analysis (Section 5.2) and node/edge detail accessing for root cause analysis (Section 5.4).

6 Case Studies

6.1 Facility Monitoring

We first present our analysis of the facility monitoring scenario released by IEEE VAST Challenge 2016 (VC16) [1]. VC16 data set contains two weeks of operation data in a building with three floors. Each floor is divided into multiple zones, where two types of data are collected: the movement data of employees and the heating, ventilation, and air conditioning (HVAC) data. The HVAC data was generated by sensors every five minutes, recording the environmental conditions, such as temperature, concentration level of carbon dioxide and other chemicals, and the heating and cooling system status, such temperature set points and damper positions. The movement data recorded the locations of the employees. The

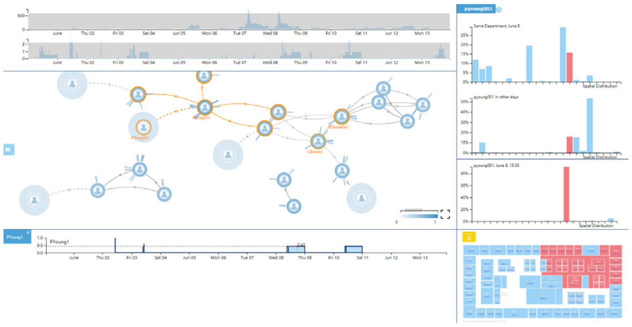
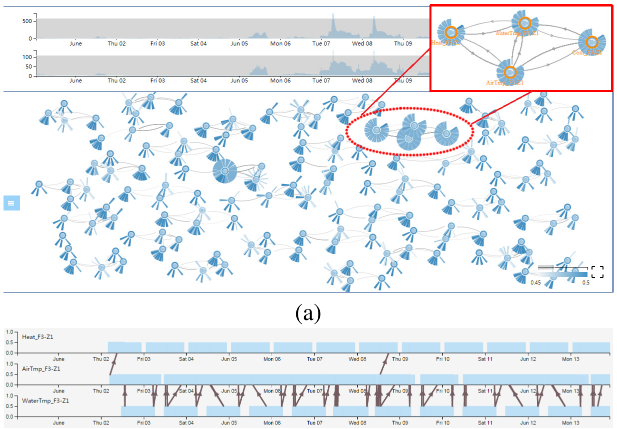


Figure 5: The dynamic HOCG of movement anomalies in two weeks.



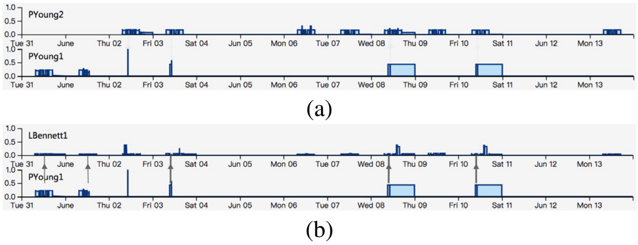
(a)

(b)

Figure 6: The dynamic HOCG of HVAC anomalies in two weeks. (a) all anomalies after filtering with a zoom-in view. (b) the anomaly time series of three sensors in F3Z1.

employees were required to carry a proximity card. The proximity card readers in each zone would generate a record with the proximity card ID, time, and the zone being entered when a proximity card moves from one zone to another. A mail delivery robot moving in the building would also generate records of the nearby proximity cards. During the time of the data set, suspicious activities were conducted in the building. Detecting, analyzing, and reasoning these activities is the major task of the challenge. For convenience, we denote zone i on floor j as FjZi.

We first investigate the suspicious employees over the entire two weeks. We filter the HOCG to remove all the HVAC anomalies from display and only show the employees with moderately high anomaly scores. We also enable the propagation of anomaly scores on the graph to identify the hidden anomalies of employees. The resulting correlation graph is shown in Figure 5. It is obvious that three employees (RMieshaber1, MBramar1, and PYoung1) have more connections to others. By investigating their anomaly details, we discover that PYoung1 is especially suspicious for two reasons. First, his anomaly score time series show a significantly larger spike on June 2, which is not found for the other two employees. Second, two anomalous event related to him last for almost the entire day of June 8 and 10. By selecting June 8 for detail exploration, the histogram of PYoung1's movement is compared to the histogram of all other employees from the same department and the histogram of his own movement on other days. The behavior of PYoung1 is different from others as he mostly stays in one zone for the entire day. In addition, we find that the normal employee PYoung2 is identified by his connection to PYoung1. This indicates that two active cards of PYoung exist at the same time, which is also suspicious.



(a)

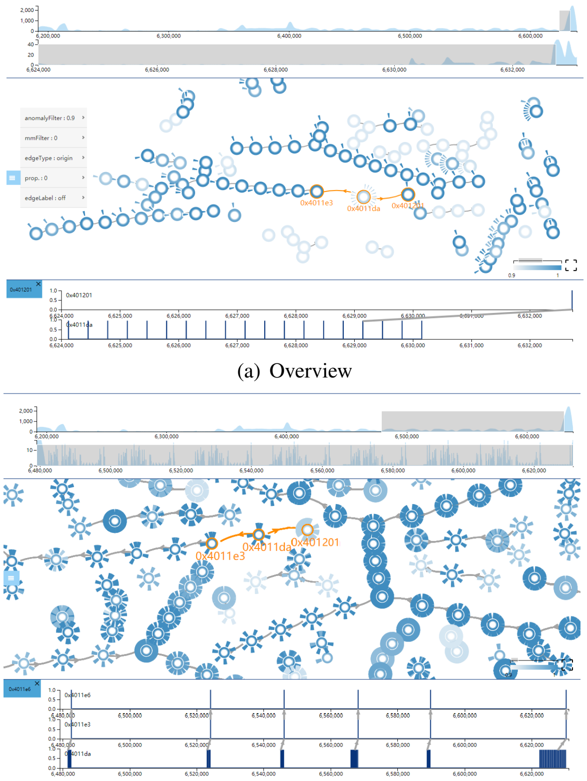
(b)

Figure 7: The anomaly score time series (a) between PYoung1 and PYoung2 and (b) between PYoung1 and LBennett1.

We then study the connectivity pattern of anomalous HVAC events. Due to the dense connectivity among HVAC events, we only show the anomalies with high correlation scores. The resulting HOCG is shown in Figure 6(a). Four closely related anomalies (highlighted in the red circle) are noticeable as these nodes have more and wider wedges, indicating significantly longer duration of anomaly. By zooming into the specific region of the graph, we can observe that the four anomalies correspond to four sensors in F3Z1: namely, the heating set point, cooling set point, water temperature, and air temperature, as highlighted in the red rectangle of Figure 6(a). The four sensors are interconnected, with edges pointing in both directions. The only exception is that no connection is found between the heating set point and cooling set point. We select the heating set point, water temperature, and air temperature for detail exploration. In the detail panel, the anomaly score time series show that the air temperature and water temperature connect to each other more frequently than the heating set point. Similar patterns can be observed for other connected components in the graph, with each connected component corresponding to temperature sensors in the same room. This indicates that the suspicious activities are likely to relate to the temperature control system.

After identifying suspicious employees and sensors, we start to investigate each individual event. We first pick the day of June 2 for exploration, when the largest spike of the anomaly score of PYoung1 is found. We display both the employees and sensors to reveal their connections. The resulting dynamic HOCG is shown in Figure 1 (c). It is obvious that PYoung1 is at the center of the graph leading to most of the HVAC anomalies and his anomaly score propagates to five employee anomalies. The two highly suspicious sensors (the air temperature and water temperature) in F3Z1 are shown in this graph and connected to PYoung1. We specify the water temperature to study its relationship to PYoung1. In the detail panel, the anomaly score time series show that after the short appearance of PYoung1's anomalous activity, the anomaly of the water temperature in F3Z1 starts. By selecting this anomaly, we find a steep rise of the water temperature in F3Z1 (the red curve in the line graph), which is different from the same sensors in other zones (the blue curves). Exploring the other HVAC anomalies shows similar relationship between them and PYoung1, indicating PYoung1 is likely to be the cause of all HVAC anomalies on June 2.

In Figure 1(c), we also find that five employees with normal movement patterns are identified through PYoung1. The largest correlation is between PYoung1 and PYoung2, indicated by the thickness of the edge between them. This is simply due to the large categorical correlation as they belong to the same employee. The second largest is found between PYoung1 and LBennett1, which is also much higher than the correlation between PYoung1 and the others. In Figure 7, we find that PYoung1 and PYoung2 do not exhibit any strong spatial-temporal correlations during the entire two weeks, but almost every single appearance of PYoung1 is accompanied by LBennett1, except for June 2. PYoung1's record on June 2 is only found for a short period resulting into small temporal correlation. In addition, in Figure 5, the longest bin of the histogram shows that PYoung1 spends almost the entire day of June 8 in F2Z7, where LBennett1's office locates. This suggests that PYoung1 is closely related to LBennett1. The second longest bin of this histogram indicates that PYoung1 visits F3Z7, where the HVAC control room locates, on the same day. The anomaly score time series of PYoung1



(a) Overview

(b) Detail

Figure 8: Software analysis case study: (a) the initial HOCG view with a smaller time window close to the crash point selected; (b) zooming out to a large time window for the root cause analysis.

show that he visits the control room each time before the HVAC anomalies. Inspecting the anomaly time series of PYoung1 and LBennett1, we find that LBennett1 never moves during the activities of PYoung1. Therefore, we suspect that LBennett1 may use the card of PYoung1 for suspicious activities in controlling the temperature of the building while leaving her own card in the office.

We invite a domain expert who had been analyzing this data for several months to evaluate our system. Generally, the expert stated that our system provided an effective and efficient way to explore the VC16 data. He found that the interface was intuitive as he only need the minimum amount of guidance to learn the tool. He commented that both the histograms displayed on the time selectors and the anomaly score time series in the detail panel are helpful for users to quickly narrow down to a specific time for exploration. He further stated that the ability of discovering hidden anomalies from the detected one was especially effective. Together with the connections on the anomaly score time series, users can easily distinguish the frequently interacting objects from the occasionally connected ones. In addition, the expert commented that the histograms and floor maps in the detail panel provide useful information to verify the findings in the correlation graph. The expert pointed out some possible improvements as well. He stated that it would be beneficial to filter the objects in the dynamic HOCG according to a user-specified object. In that way, users can focus on one anomaly and analyze its relationships to others, so that its impact and cause can be identified more easily. He also suggested that we might allow users to specify a zone to explore so that the dynamic HOCG could be further filtered to show anomalies related to that zone.

6.2 Software Analysis

In another case, we deploy the HOCG to detect collective anomalies in software runtime executions. We consider the desktop software that is known to have certain security vulnerabilities. Such software vulnerabilities can lead to a fatal crash at runtime, and if compromised by malicious attacks, can even be hijacked to execute any code on the host machine. The traditional software analysis is based on the source code inspection [13, 25, 31] because the runtime analytics of software can evolve billions of executions per second and generate a huge volume of monitoring data. Analyzing such big data and detecting collective anomaly is analogous to finding a needle in the haystack, which poses great challenges to the community.

In this scenario, the raw data are the runtime monitoring data of software executions. Each line of data corresponds to an execution of one line of code in assembly language with the following attributes: "id" is the execution sequence; "eip_addr" is the address of this line of code; "op_vals" are operator values; "src_ids" and "dst_ids" are related executions that affect or are affected by this execution.

For this data set, we construct HOCG by treating each line of code as a node, each execution of the code as an event, and the data flow between executions as the correlation link. The point anomaly on events is detected by the algorithm in Section 4.2. The same software is executed twice. In the first time, no compromise of the security vulnerability is conducted, and the execution data are used as the normal profile; in the second time, the software vulnerability is triggered and the execution data are used to construct HOCG.

The initial overview of HOCG is shown as Figure 8(a). The entire data set contains 6 million lines of executions and we load the last 400,000 lines close to the crash point of the software. We first examine the timeline overview panel in the top row of Figure 8(a). It is clear that there is a surge in the number of point anomalies close to the final crash point. We then select a small time window (about 8000 cycles) to examine the context at the crash point. The HOCG at this window is visualized in the correlation graph view of Figure 8(a). In this graph, most anomalies are shown to happen very recently, as indicated by the last wedges on these nodes. Only the node representing the line of code at 0x4011da (eip) behaves anomalously in a continuous manner, as indicated by much more wedges on the node than others. To drill-down to details, we click on this node (eip: 0x4011da) to expand its anomaly stack over time. The bottom row in Figure 8(a) shows regular anomaly pattern with a fixed cycle. We proceed to check the other nodes connected to it. There are two such nodes: eip: 0x401201 and eip: 0x4011e3. When clicking to expand the anomaly stack, we find that the node of 0x401201, as shown by the row on top of 0x4011da, contains only one anomalous event at the end of the timeline. We can conclude that 0x401201 is the line of code leading to the fatal crash, and 0x4011da behaves as the direct source of this crash.

To further detect the root cause of this crash, we select a larger time window of 200,000 cycles. The corresponding HOCG is depicted in Figure 8(b). The relationship among 0x4011da, 0x401201, and 0x4011e3 is unchanged. By expanding their anomaly stack again, it is found that the line of code 0x4011da has triggered regular anomalies on 0x4011e3 for a long time, before leading to the crash by the code at 0x401201. We bring the findings to work with a source code analysis expert together. Based on our result, we are able to restore the scene of this software crash. A brief description is illustrated in Figure 9. Initially, the code at 0x401201 and 0x4011e3 (both "mov" instructions) are not related, though their read/write memory address is close to each other. After an abnormal I/O operation, in fact an invalid external user input, the line of code at 0x4011da starts to move an overlong string to its destination memory address. Then the operator of the code at 0x4011e3 gets overflown and it begins to run anomalously. The code line at 0x4011da continues to overflow at its destination memory address to write the overlong input string, until the function address of the "call" instruction at 0x401201 gets overflown. This leads to the irreversible, fatal software crash.

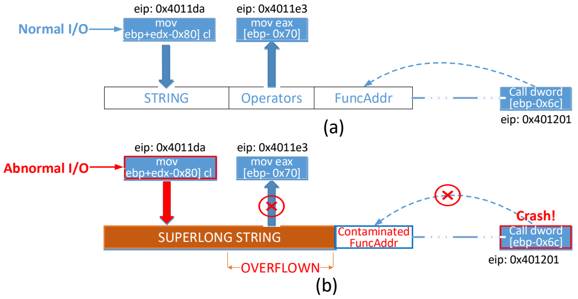


Figure 9: The illustration of compromised software vulnerabilities: (a) normal case; (b) under malicious external input.

7 Conclusion

In this paper, we describe a visual analytic framework based on the high-order correlation graph to detect, analyze, and reason collective anomalies. HOCG captures the multifaceted relationships in heterogeneous types of objects and events. It can generalize to various kinds of applications by providing domain-specific anomaly detection methods. By leveraging the random walk method, the anomaly scores of events can be propagated from the detected ones to the others, in order to identify the collective anomalies. In addition, we design an interactive interface that allows the flexible exploration of detected anomalies and their multifaceted relationships. Users can drill down to the raw data in the detail view to validate their discoveries. We demonstrate the effectiveness of the HOCG concept and the visualization system with two real-world applications.

In the future, we plan to extend our current system in the following ways. First, we will develop a node aggregation scheme to reduce visual clutter and provide high-level information. Second, we will leverage belief propagation to incorporate our point anomaly detection and anomaly score propagation in a unified framework. Messages will be passed between nodes in the HOCG to identify point anomalies and collective anomalies simultaneously. Third, as suggested by the domain expert (refer to Section 6.1), we will provide an egocentric exploration scheme that focuses on the relationships between a user-specified object and others.

Acknowledgments

This work was supported in part by China National 973 Project 2014CB340301, NSFC Grants 61379088 and 61772504, and the U.S. NSF Grant IIS-1456763. Jun Tao was partially supported by the International Visiting Scholar Program of Institute of Software, Chinese Academy of Sciences. Lei Shi is the corresponding author.

References

[1] IEEE VAST Challenge 2016. `http://vacommunity.org/VAST+Challenge+2016`.

[2] M. Ahmed, A. N. Mahmood, and J. Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016.

[3] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, 29(3):626–688, 2015.

[4] S. Boriah, V. Chandola, and V. Kumar. Similarity measures for categorical data: A comparative evaluation. In *SDM'08*, pp. 243–254, 2008.

[5] P. K. Chan and M. V. Mahoney. Modeling multiple time series for anomaly detection. In *ICDM'05*, pp. 1–8, 2005.

[6] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing surveys*, 41(3):15:1–15:58, 2009.

[7] C. De Stefano, C. Sansone, and M. Vento. To reject or not to reject: That is the question-an answer in case of neural classifiers. *IEEE TSMC, Part C (Applications and Reviews)*, 30(1):84–94, 2000.

[8] L. Ertoz, E. Eilertson, A. Lazarevic, P.-N. Tan, V. Kumar, J. Srivastava, and P. Dokas. MINDS - Minnesota intrusion detection system. In *Next Generation Data Mining*, chap. 3, pp. 199–218. MIT Press, 2004.

[9] E. Eskin. Anomaly detection over noisy data using learned probability distributions. In *ICML'00*, pp. 255–262, 2000.

[10] F. Fischer, F. Mansmann, D. A. Keim, S. Pietzko, and M. Waldvogel. Large-scale network monitoring for visual analysis of attacks. In *VizSec'08*, pp. 111–118, 2008.

[11] E. R. Gansner and S. North. An open graph visualization system and its applications to software engineering. *Software - Practice & Experience*, 30:1203–1233, 2000.

[12] G. G. Hazel. Multivariate Gaussian MRF for multispectral scene segmentation and anomaly detection. *IEEE TGRS*, 38(3):1199–1211, 2000.

[13] B. Johnson, Y. Song, E. Murphy-Hill, and R. Bowdidge. Why don't software developers use static analysis tools to find bugs? In *ICSE'13*, pp. 672–681, 2013.

[14] R. Khanna, H. Liu, and H.-H. Chen. Reduced complexity intrusion detection in sensor networks using genetic algorithm. In *ICC'09*, pp. 1–5, 2009.

[15] Q. Liao, L. Shi, and C. Wang. Visual analysis of large-scale network anomalies. *IBM Journal of R&D*, 57(3/4):13–1, 2013.

[16] Z. Liao, Y. Yu, and B. Chen. Anomaly detection in GPS data based on visual analytics. In *VAST'10*, pp. 51–58, 2010.

[17] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory*, 37(1):145–151, 1991.

[18] S. Lin and D. E. Brown. An outlier-based data association method for linking criminal incidents. *Decision Support Systems*, 41(3):604–615, 2006.

[19] F. Liu, X. Cheng, and D. Chen. Insider attacker detection in wireless sensor networks. In *ICC'07*, pp. 1937–1945, 2007.

[20] X. Miao, K. Liu, Y. He, D. Papadias, Q. Ma, and Y. Liu. Agnostic diagnosis: Discovering silent failures in wireless sensor networks. *IEEE TWC*, 12(12):6067–6075, 2013.

[21] E. C. Ngai, J. Liu, and M. R. Lyu. An efficient intruder detection algorithm against sinkhole attacks in wireless sensor networks. *Computer Communications*, 30(11):2353–2364, 2007.

[22] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *KDD'03*, pp. 631–636, 2003.

[23] I. Onat and A. Miri. A real-time node-based traffic anomaly detection algorithm for wireless sensor networks. In *ICSC'05*, pp. 422–427, 2005.

[24] W. R. Pires, T. H. de Paula Figueiredo, H. C. Wong, and A. A. F. Loureiro. Malicious node detection in wireless sensor networks. In *IPDPS'04*, p. 24, 2004.

[25] M. Pistoia, S. Chandra, S. J. Fink, and E. Yahav. A survey of static analysis methods for identifying security vulnerabilities in software systems. *IBM Systems Journal*, 46(2):265–288, 2007.

[26] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova. Anomaly detection in dynamic networks: a survey. *Interdisciplinary Reviews: Computational Statistics abbreviation*, 7(3):223–247, 2015.

[27] L. Shi, Q. Liao, Y. He, R. Li, A. Striegel, and Z. Su. SAVE: Sensor anomaly visualization engine. In *VAST'11*, pp. 201–210, 2011.

[28] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *VL'96*, pp. 336–343, 1996.

[29] S.-T. Teoh, K. Zhang, S.-M. Tseng, K.-L. Ma, and S. F. Wu. Combining visual and automated data mining for near-real-time anomaly detection and analysis in BGP. In *VizSEC/DMSEC'04*, pp. 35–44, 2004.

[30] D. Thom, H. Bosch, S. Koch, M. Wörner, and T. Ertl. Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages. In *PacificVis'12*, pp. 41–48, 2012.

[31] D. Wagner, J. S. Foster, E. A. Brewer, and A. Aiken. A first step towards automated detection of buffer overrun vulnerabilities. In *NDSS'00*, pp. 3–17, 2000.

[32] M. Xie, S. Han, B. Tian, and S. Parvin. Anomaly detection in wireless sensor networks: A survey. *Journal of Network and Computer Applications*, 34(4):1302–1325, 2011.

[33] J. Zhao, N. Cao, Z. Wen, Y. Song, Y. R. Lin, and C. Collins. #FluxFlow: Visual analysis of anomalous information spreading on social media. *IEEE TVCG*, 20(12):1773–1782, 2014.