Visual Analysis of Collective Anomalies Using Faceted High-Order Correlation Graphs

Jia Yan, Lei Shi, Jun Tao, Xiaolong Yu, Zhou Zhuang, Congcong Huang, Rulei Yu, Purui Su, Chaoli Wang, and Yang Chen

Abstract—Successfully detecting, analyzing, and reasoning about collective anomalies is important for many real-life application domains (e.g., intrusion detection, fraud analysis, software security). The primary challenges to achieving this goal include the overwhelming number of low-risk events and their multimodal relationships, the diversity of collective anomalies by various data and anomaly types, and the difficulty in incorporating the domain knowledge of experts. In this paper, we propose the novel concept of the faceted High-Order Correlation Graph (HOCG). Compared with previous, low-order correlation graphs, HOCG achieves better user interactivity, computational scalability, and domain generality through synthesizing heterogeneous types of objects, their anomalies, and the multimodal relationships, all 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 of the HOCG. We conduct case studies for three application domains and collect feedback from domain experts who apply our method to these scenarios. The results demonstrate the effectiveness of the HOCG in the overview of point anomalies, the detection of collective anomalies, and the reasoning process of root cause analyses.

Index Terms—Correlation graph visualization, collective anomaly

1 Introduction

Anomaly detection is a critical interdisciplinary research area [1] that expands its applications to a variety of strategic domains (e.g., intrusion detection, fraud analysis, software security). If not well contained, the anomalous state often translates into hazardous fatal actions, e.g., compromise of machines for potential attacks, real-life terrorist activities. In this work, we consider one of the most complicated anomaly types: the *collective anomaly*. The collective anomaly is identified as coordinated events on a group of interrelated objects, which individually appear to be normal, or of limited suspicion; yet, their co-occurrence is highly anomalous. For example, in software analytics, the stack-overflow and the call function transfer itself can solely be programming tricks or low-risk software bugs. When these two events happen sequentially, the normal operation severely upgrades to a malicious attack of code injection through the exploitation of software vulnerabilities. Another example is the distributed denial of service (DDoS) attack on web servers [2]. While a single request to a server is legitimate, numerous connection requests occurring simultaneously with a high frequency may indicate a collective anomaly.

The detection of collective anomalies is challenging, because their anomalous states are revealed by each individual event on the objects (known as point anomalies), and heavily dependent on the relationship among the events. The combination of low-risk events with their relationships leads to an explosion of potential states to examine for anomaly detection algorithms. To overcome this data proliferation, most techniques on the collective anomaly detection focus on a single type of relationship among events, such as sequential [3], spatial [4], or graph relationship [5]. For each type of relationship, specific feature extraction algorithms are designed to reduce the event data and their relationships into a vector of features within a given feature space. The point anomaly detection algorithms are then applied to discover the collective anomalies from the extracted feature vector. Therefore, these techniques are often limited to a single type of data and application.

On the other hand, visualizations have been widely developed for the purposes of anomaly detection, e.g., the correlation graph for agnostic anomaly detection in wireless sensor networks [6], [7], or spatiotemporal [8] and information diffusion anomaly visualization [9] over social media. These approaches, either directly visualize the raw dataset and do not scale to the big data, or are specially designed for a certain domain and do not generalize to solve the common problem of collective anomaly detection.

- J. Yan is with TCA/SKLCS, Institute of Software, Chinese Academy of Sciences, Beijing 100864, China. E-mail: yanjia@iscas.ac.cn.
- P. Su is with TCA/SKLCS, Institute of Software, Chinese Academy of Sciences, Beijing 100864, China, and also with the University of Chinese Academy of Sciences, Huairou 101408, China. E-mail: purui@iscas.ac.cn.
- L. Shi is with the School of Computer Science, Beihang University, Beijing 100083, China. E-mail: leishi@buaa.edu.cn.
- J. Tao and C. Wang are with the Department of Computer Science & Engineering, University of Notre Dame, Notre Dame, IN 46556. E-mail: {Jun.Tao.12, chaoli.wang}@nd.edu.
- X. Yu, Z. Zhuang, and Y. Chen are with the School of Computer Science, Fudan University, Shanghai 200433, China. E-mail: {xlyu17, zzhuang14, chenyang}@fudan.edu.cn.
- C. Huang and R. Yu are with SKLCS, Institute of Software, Chinese Academy of Sciences, Beijing 100864, China. E-mail: {huangcc, yurl}@ios.ac.cn.

Manuscript received 5 Sept. 2018; revised 1 Dec. 2018; accepted 15 Dec. 2018. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Lei Shi.)
Recommended for acceptance by B. Lee.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2018.2889470

In this paper, we study the problem of designing a collective anomaly detection technique to achieve three key objectives. First, to adapt to the versatility of the collective anomalies, the technique should bring users into the loop to combine the power of automatic computation and human analytics. This is conducted to detect the previously unknown collective anomalies. Second, the technique should scale to analyze the dataset with a huge volume and a variety of data types, e.g., time series, sequential, and spatial data. Third, the technique should be generic enough to detect the collective anomalies in different application domains and be able to incorporate the prior domain knowledge from the normal and abnormal data models.

Motivated by this problem, we propose the novel concept of the faceted *High-Order Correlation Graph* (HOCG), in which anomalous events detected from the behavior of individual objects at multiple facets are modeled as nodes, while their high-order correlations are modeled as edges. Essentially, HOCG is defined at the multivariate-event level, in comparison to the lower-order correlation graph [6], which is defined over univariate data variables. There are several advantages to detecting the collective anomalies that fulfill the design objectives. The first is *interactivity*. The HOCG is fully customizable by users and provides the flexibility to analyze data objects and their relationships for an unknown collective anomaly. The second is *scalability*. Through graph simplification and object-centric abstraction techniques, large HOCGs can be greatly reduced in the overview visualization, while allowing access to spatial, temporal, and anomaly details upon user interactions. The third is *generality*. The construction of HOCG follows an 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.

The contributions of this work can be summarized as follows.

- We formally define HOCG in a domain and data type independent manner. A flexible framework is proposed to construct the HOCG by integrating point anomaly detection, multimodal correlation analyses, and anomaly propagation algorithms.
- We design novel metaphors to visualize the HOCG concept, and a visual analytics system to display large HOCGs through visual abstraction. The system provides several interaction models to validate the individual point anomalies, visually detect the collective anomalies, and conduct a root cause and dynamic analysis for the containment actions.
- The proposed HOCG framework and the visual analytics system are evaluated through three case studies in the facility monitoring, intrusion detection, and software analysis domains. The case study results and the feedback from the domain experts demonstrates the effectiveness of the system in the visual reasoning of the collective anomalies.

Note that this is an extended version of the conference paper published in PacificVis'18 [10]. We improve the original work by augmenting the HOCG concept with facets and proposing an enhanced metaphor design to support the scalable visualization. The other changes in the visual analytics framework, the anomaly detection algorithms, and the evaluation can be found in the main body of this paper.

2 RELATED WORK

2.1 Anomaly Detection Algorithms

Anomaly detection has been extensively studied in the past decade. We refer readers to the following surveys [1], [2], [11], [12], [13] for a thorough understanding of this area. Many types of anomaly detection algorithms have been proposed, including classification-based [14], nearest-neighbor-based [15], clustering-based [16], statistics-based [17], graph-based [18], [19], and information-theoretic techniques [20].

Among this literature, the most related works to ours are the anomaly detection techniques on sensor networks which also depend on the underlying graph structure. These techniques can be further classified into prior-knowledge based approaches [21], [22] and prior-knowledge free approaches [23], [24], [25]. The prior-knowledge based approaches require assumptions or experience to provide a normal profile for the anomaly detection. Liu et al. [22] assumed that the Mahalanobis squared distances between the attributes of a sensor network follow a chi-squared distribution. In contrast, the prior-knowledge free approaches usually construct the normal profile through the training process. Khanna et al. [24] applied a genetic algorithm to measure the fitness of network nodes.

Compared with the existing approaches, the point anomaly detection method in this work adopts a hybrid strategy. It can take a normal profile for a higher detection accuracy. It can also be prior-knowledge free when the normal profile is unavailable and the anomalies are rare. In the meanwhile, our collective anomaly detection method relies on human intervention through visual analytics, which does not fall into the algorithm-centric category.

2.2 Visual Analytics for Anomaly Detection

The visual analytics techniques for anomaly detection have gained increasing attention in the visualization community.

On cybersecurity, Fischer et al. [26] visualized attacks on a large-scale network by mapping the monitored network as a treemap and the attacking host as an isolated node. They did not provide a way to identify the anomalous events but instead relied on an external intrusion detection system. Teoh et al. [27] applied a statistical model to detect anomalies in the Border Gateway Protocol. The anomaly of each event is visualized by line graphs and a series of circles indicating the time and signature of the event.

On sensor networks, Shi et al. [7] proposed multiple designs to visualize and analyze their anomalies to allow the 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 the attributes. And the dimension projection graph maps the sensor nodes to a scatterplot. Liao et al. [28] further extended this work to consider the membership changes of the node communities, so that anomaly detection is less sensitive to the activity of each individual node.

On geospatial intelligence, Liao et al. [29] developed GPSva, a visual analytic system to study anomalies in GPS streaming traces. The anomalies are detected using the conditional random field and visualized on a map. Thom et al. [8] detected and visualized spatiotemporal anomalies based on geo-located twitter messages. A cluster analysis is used to distinguish the global and local messages. The aggregated messages are then visualized as the term clouds on a geographic map.

On social media, Zhao et al. [9] developed #FluxFlow to visually analyze anomalies in the information diffusion over social media. The anomalous retweeting threads are detected using an one-class conditional random field 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 the overview and the detail.

On finance, aka the fraud detection, the visual analytics systems such as WireVis [30] and EVA [31] were developed. They combine multiple coordinated views to illustrate the complex and time-varying behavior of large-scale transactions in financial institutions. The objective is to discover the fraudulent events such as the money laundering and the unauthorized transaction. In the VISFAN [32] and TAXNET [33] systems, the financial reports and/or records, e.g., the transactions and the shareholdings, are synthesized to build the financial activity network. The network visualization techniques are integrated with the graph clustering and pattern matching algorithms to identify the financial crimes and suspicious activities such as the tax evasion.

Among this literature, the correlation graph proposed in Ref. [7] is the closest to ours. However, the correlation graph only considers one sensor node and one type of relationship. Our approach scales to analyze the interactions among multiple types of nodes and their multimodal relationships by visually synthesizing all of the information in a single high-order correlation graph. Therefore, our method is more suitable to apply to analyze the collective anomaly in a sophisticated context.

Meanwhile, the visualization methods for the multivariate and dynamic graphs [34], [35] are also related to our work. The difference is, the attributes displayed on the nodes/links of HOCG represent the suspicious events happened on the nodes and the correlation among these events. This is designed for the task of anomaly detection. In comparison, the generic multivariate/dynamic graph visualizations display the first-order attributes and relationships of the graph nodes. The work by Wang and Mueller [36] also studied the graph-based visual analytics method to discover causalities from data. Again, their approach constructs the causality graph from the subdivided raw data, which is not used to detect the relationship of the point anomalies hidden in the raw data.

3 Problem

3.1 Definition and Requirement Analysis

We consider a group of *objects* (e.g., facilities, persons, computers), whose behaviors are captured by a set of *event* data (e.g., sensor readings of a facility, movements of a person, network traffic of a computer). The events are interconnected by *multimodal relationships* (e.g., the spatial/temporal closeness between sensors, the role similarity between persons, the network traffic between computers).

Each single event on an object is represented by a 5-tuple: {object, facet, space, time, measured value} (refer to the notations in Section 4.1). Normally, the number of such events is huge as the objects are often measured on a real-time, continuous basis. This provides an opportunity to detect abnormal events, i.e., on which facet the object behaves anomalously, when, where, and how, by comparing the extracted suspicious events with a large number of normal events of this and other objects. Two levels of anomalies are considered: the traditional point anomalies and the collective anomalies. The point anomalies are defined by the abnormal events on a single object-facet pair. The collective anomalies are characterized by synthesizing the point anomalies on multiple object-facet pairs having interrelated events. In this work, we focus on the analysis of collective anomalies, for which the event on a single object-facet pair may not be highly anomalous by itself, but several interrelated low-risk events occurring together on multiple object-facet pairs can raise the anomaly level and become noteworthy.

Our work aims to meet the following requirements in visually detecting, analyzing, and reasoning about the collective anomalies.

R1. Rate individual events. Instead of classifying each event as a point anomaly or not, for the detection of the collective anomaly, there should be an anomaly score calculated on each event to indicate how anomalous the event is. The anomaly score serves two purposes: it allows us to identify the moderately anomalous events, which potentially composes the collective anomaly; it also provides a criterion for users to rank and filter the anomalous events independent of the data type.

R2. Understand relationships among events. Given that the collective anomaly is composed of multiple interrelated events, it becomes critical to answer the question of whether the two events are related to each other or not. We should analyze the correlation between these two events, e.g., their spatial/temporal/facet closeness, the underlying objects' intrinsic relationship, and the historical interaction among the objects.

R3. Detect and interpret collective anomalies. Knowing the anomaly scores of individual events and their relationships, the final and most important problem of this work becomes determining how to visually detect the collective anomalies and further interpret them. In this paper, we consider two types of collective anomalies. The first is composed of a group of strongly interrelated events that are moderately anomalous. The second is composed of a few highly anomalous events and the other less anomalous events that are tightly connected to these strong anomalies. The former type identifies the hidden collective anomalies that cannot be discovered by the point anomaly detection algorithm 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 resolve the scalability issue as the number of events is huge.

3.2 User Tasks

After fulfilling the above requirements, our visual analytics system can support several key user tasks in analyzing collective anomalies. Below we characterize these tasks in the

TABLE 1
Notations Used in This Paper

SYMBOL	DEFINITION
$\Phi = < o, c, s, t, v >$	An event defined by the 5-tuple
$\alpha(\Phi) = A_{< o,c,s,t >}(v)$	The anomaly score of an event
$\rho(\Phi_i, \Phi_j) = \rho_F(\rho_S, \rho_T, \rho_C, \rho_O)$	The high-order correlation
$\Phi(o_i, \mathbf{T})$	The events related to o_i in $\mathbf{T}$
$\mathbf{H} = (\mathbf{V}, \mathbf{E})$	The high-order correlation graph
$\mathbf{H}(\mathbf{T}) = (\mathbf{V}(\mathbf{T}), \mathbf{E}(\mathbf{T}))$	Dynamic HOCG at time $\mathbf{T}$
$\mathbf{H}^+ = (\mathbf{V}^+, \mathbf{E}^+)$	The augmented HOCG

typical scenario of facility monitoring. In this scenario, two types of objects are considered: facilities and employees. To monitor the facility, multiple types of sensors are deployed. On the other hand, the behavior of the employees is captured by their measured locations.

T1. Overview. Two overview tasks should be supported. The first level is the overview of the anomalous events over time. This helps to answer the question of when the status of the facilities or the movement of the employees exhibits suspicious behaviors? With this overview visualization, users can quickly narrow down to a specific time period for exploration. The second level is the overview of all point anomalies within a selected time period. This helps to answer the questions of which event has the highest anomaly score, which object has the longest period of an anomalous event, and what is the relationship among all point anomalies? These overview tasks depend on satisfying *R1* and *R2*.

T2. Validation of point anomalies. Once the potential anomalous events are detected in the overview, the users need to validate these anomalies by comparing them with the normal data. For example, to evaluate an abnormal reading of a sensor, the system should present all the related normal readings, as well as their spatial and temporal context. Based on the visual comparison, users can make a better judgment about the degree of the anomaly by incorporating their domain knowledge. This helps to reinforce *R1*.

T3. Visualization of relationships among point anomalies. Given all the point anomalies, users should be able to perceive their relationships. At the object level, they need to determine the associated events with the object. At the event level, they need to determine the interrelated events. For example, to reason about the abnormal reading of a sensor, it is helpful for users to understand which facility and/or employee contributes to this anomaly. The interrelated point anomalies provide a visual hint for users to further identify the collective anomaly. This task is based on meeting *R2*.

T4. Interactive root cause analysis of collective anomalies. Users should be allowed to zoom and filter point anomalies, and their relationships, to identify the related point anomalies for the composition of the collective anomalies. To reveal the less anomalous events which connect to a few highly anomalous events, the anomaly scores could be propagated among the graph of the events. For example, when an employee performs a deliberate harmful action, s/he is likely to disguise herself/himself and behaves normally. To identify these anomalies, the technique should help users to trace back to the detected significant anomalies through the event relationship. This tasks mainly fulfills *R3*.

4 High-Order Correlation Graph

In this section, we first introduce the concept of the High-Order Correlation Graph. Next, we provide an overview of the visual analytics framework over the HOCG to detect, analyze, and reason about collective anomalies. Finally, we detail each stage of the framework.

4.1 Overview

HOCG. HOCG is defined on a group of objects with multiple facets. The behavior of each object is captured by a set of event data over the studied time period. As shown in Table 1, each event is defined by a 5-tuple $\Phi = < o, c, s, t, v >$. Here o denotes the associated object of the event (e.g., a zone/floor composed of building facilities, an employee of the company, a host computer in the network), c denotes the facet of the object on which the event is captured (e.g., a sensor of the zone/floor, a listening port/application of the host), s denotes the spatial location/region of the event, t denotes the time point/interval when the event happens, and v denotes the measured value(s) on $< o, c >$ during time t. Each event is assigned an anomaly score $\alpha(\Phi) = A_{< o,c,s,t >}(v)$ by executing the point anomaly detection algorithm.

Furthermore, the interrelation between the two events Φ_i and Φ_j, denoted as $\rho(\Phi_i, \Phi_j)$, is defined by their high-order correlation. To construct the high-order correlation, we consider four classes of single-type correlations. $\rho_S(s_i, s_j)$ denotes the spatial correlation (e.g., happened on the same floor), $\rho_T(t_i, t_j)$ denotes the temporal correlation (e.g., happened in the same minute/hour), $\rho_C(c_i, c_j)$ denotes the facet correlation (e.g., belonging to the same group of sensors), and $\rho_O(o_i, o_j)$ denotes the object-level correlation (e.g., having traffic flows between the two hosts). These correlations are combined by the fusing function $\rho_F(\rho_S, \rho_T, \rho_C, \rho_O)$ to compute the high-order correlation score.

Finally, HOCG is defined as $\mathbf{H} = (\mathbf{V}, \mathbf{E})$. $\mathbf{V}$ denotes the set of nodes in which each node is an event made up of its 5-tuple. $\mathbf{E}$ denotes the set of edges in which each edge represents the high-order correlation between the events. In the real usage, HOCG is often studied within a user-specified time interval $\mathbf{T}$, which is defined by the dynamic HOCG, i.e., $\mathbf{H}(\mathbf{T}) = (\mathbf{V}(\mathbf{T}), \mathbf{E}(\mathbf{T}))$. In another setting, HOCG is extended to include the events that are closely related to the existing highly anomalous events through the anomaly score propagation. The extended HOCG is denoted as $\mathbf{H}^+ = (\mathbf{V}^+, \mathbf{E}^+)$.

Compared with the original concept of the correlation graph [7], HOCG is high-order in three aspects. First, each individual node of the HOCG is a multivariate event associated with several contextual attributes, i.e., object, facet, space, and time of the event. This is far more comprehensive than using the single measured variable as a node in the original correlation graph. Second, the edge between the events is composed of multimodal correlations detected between the multivariate events, including their spatial, temporal, facet, and object-level correlations. In comparison, the edges of the original correlation graph only focus on the temporal correlation between the measured variables. Third, and most importantly, based on the node and edge definition, the HOCG detects the point anomaly on each single event by computing an anomaly score for each of them, and then connects the dots among point anomalies

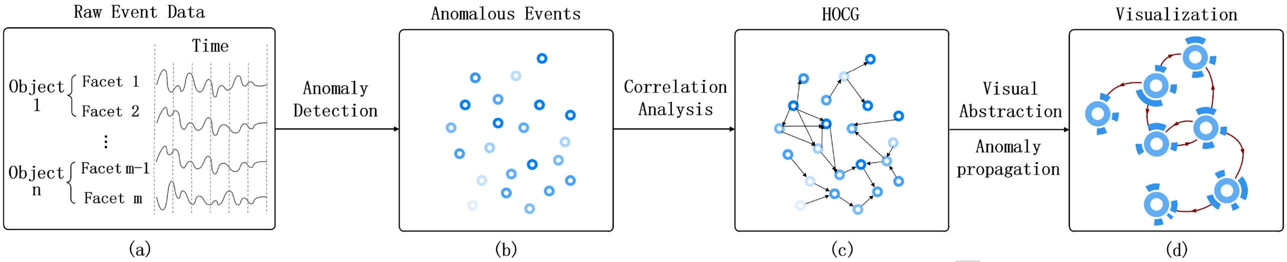

Fig. 1. The workflow of our visual analytics framework on collective anomalies.

for analyzing the collective anomaly, which often involves multiple objects. On the other hand, the original correlation graph detects anomalies from the relationship among the measured variables on a single object. Thus, they are limited to the analysis of point anomalies.

Visual Analytics Framework. As illustrated in Fig. 1, we propose a three-stage visual analytics framework to construct and visualize the HOCG for the collective anomaly detection. The raw input is the list of event data (Fig. 1a). In the first stage, we apply the point anomaly detection algorithm on the events at each facet of an object. Each event is assigned an anomaly score, which is indicated by the darkness of the node fill color in Fig. 1b. In the second stage, the correlations among events are discovered, based on which the HOCG is constructed. Finally, the raw HOCG is abstracted in an object-centric way for an efficient, compact visualization. The graph simplification, based on time and anomaly score filtering, is also supported to reduce the visual complexity. In addition, the mechanism of the anomaly propagation is employed to augment the object-level HOCG. This allows the users to identify the hidden anomalies in the studied dataset.

4.2 Point Anomaly Detection

The point anomaly can be detected by comparing a single data instance with the rest of the data. In our framework, the point anomaly is detected on each event by comparing its measured value with the other events on the same facet of an object. For example, a sensor reading on one building floor is considered anomalous if there have been few similar readings measured on the same sensor and floor previously. There are a number of established point anomaly detection algorithms [1], e.g., the statistics-based, the classification-based, and the nearest-neighbor-based algorithms. In theory, each of these algorithms can be plugged into our framework to detect the point anomalies. We will describe the two algorithms that work well with the scenarios in our case studies.

The input to each algorithm is the list of events on the same facet of an object. We assume there is a set of events known to be normal, or there is no such normal dataset, but the portion of abnormal data is known to be very small. In the latter case, we will use the entire dataset as the normal dataset. The basic idea behind this is to develop a model based on the normal data and estimate the probability for each incoming event to deviate from the normal model. We then translate this probability into a point anomaly score. Two types of events are considered and analyzed using separate models.

Events with Continuous Measures. The network traffic volume in the intrusion detection scenario and the measured temperature in the facility monitoring scenario are both measured continuously. We apply the Gaussian Mixture Model (GMM) [37] to characterize the continuous normal event data, which has a probability density function by

$$P(v|k, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{i=1}^{k} w_i \cdot \mathcal{N}(v|\mu_i, \sigma_i), \tag{1}$$

where v denotes the value of the normal event, k is the number of Gaussian components, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the means and standard deviations, and w_i is the weight of each component. The GMM model can be estimated by the Expectation Maximization (EM) algorithm [38]. The number of components can be determined by the Bayesian information criterion (BIC) [39] for model selection.

For each incoming event Φ_j with value v_j, we introduce the Extreme Value Theory (EVT) [40] to compute the probability for v_j to deviate from the GMM model. The theory essentially estimates the probability for v_j to be larger/smaller than the maximal/minimal value in all normal data instances. The details of the computation will be described in three steps.

In the first step, the Gaussian component closest to v_j in the GMM model is determined, which is denoted as the k^*th component. Here the Mahalanobis distance measure is applied, in which the distance between v_j and the k^*th Gaussian component is computed by

$$h_{k^*}(v_j) = \frac{|v_j - \mu_{k^*}|}{\sigma_{k^*}}. \tag{2}$$

In the second step, this distance is further normalized by the number of normal data instances belonging to the k^*th Gaussian component, denoted as m_{k^*}.

$$y_m = \frac{h_{k^*}(v_j) - \mu_m}{\sigma_m} \quad \mathit{where} \tag{3}$$

$$\mu_m = \sqrt{2\ln m_{k^*}} - \frac{\ln\ln m_{k^*} + \ln 2\pi}{2\sqrt{2\ln m_{k^*}}}, \quad \sigma_m = \sqrt{2\ln m_{k^*}}. \tag{4}$$

In the third step, the probability for the measured value to deviate from the k^*th Gaussian component is computed by

$$p(v_j \geq \max v || v_j \leq \min v) = e^{-e^{-y_m}}. \tag{5}$$

In the final step, the anomaly score of the event is translated from the probability by

$$\alpha(\Phi_j) = \min\left(\frac{-\ln(1-p)}{\Gamma}, 1\right), \tag{6}$$

where Γ is the expected highest anomaly score for normalization. Note that, the proposed method inherently extends to support the event with multivariate values.

Events with Discrete Measures. The employee's movement data in the facility monitoring scenario takes on categorical values, e.g., F3Z1,[1] F3Z2, etc. Because these categorical values are less related to each other than the continuous values, we cannot use the GMM to characterize them. Instead, we introduce a histogram based algorithm. In the facility scenario, the event value v_i denotes the location of employee o_i at time point t_i. We compute a daily movement histogram for employee o_i in which each bin of the histogram indicates the total time that the employee stays in the corresponding zone on that day. To identify the anomaly score of the employee on an incoming day, we compare the movement histogram of the employee on the incoming day with two normal histograms: 1) the average daily movement histogram of the employee on all the days belonging to the normal data; and 2) the average daily movement histogram of all the employees in the same department on the same incoming day. Each histogram can be represented by a discrete probability distribution, i.e., $P(v)$ for the distribution on an incoming day to be evaluated, $A(v)$ for the average distribution in comparison. The difference between the two histograms is measured by the Kullback-Leibler divergence $D_{KL}(P \parallel A)$ from $A(v)$ to $P(v)$ [41]. To capture the anomaly of each event, the KL divergence is decomposed. The anomaly score of each event with value v_j is then computed by

$$\alpha(\Phi_j) = \min\left(\frac{\max\left(\log\frac{p(v_j)}{a(v_j)},\ 0\right)}{\Gamma},\ 1\right), \tag{7}$$

where $p(v_j)$ and $a(v_j)$ are the probabilities of the value v_j in the two distributions $P(v)$ and $A(v)$ respectively, and Γ is the maximum anomaly score for normalization. Only the positive anomaly, i.e., $p(v_j) > a(v_j)$, is captured. The larger anomaly score computed from the two comparisons is used as the final score.

4.3 Correlation Analysis

The correlation between the 5-tuple event data is multimodal in that all the object, facet, space, and time information of the events may be related to each other. These correlations are fused to form the high-order edges in the HOCG.

Spatial Correlation. The spatial correlation indicates the closeness of the locations where the events occur. In the facility monitoring scenario, the spatial regions of a facility are defined as three hierarchies, i.e., floors, zones of a floor, rooms of a zone. The spatial correlation is calculated as the probability of two events occurring in the same region. We apply $\rho_S = 1$ for the two events occurring in the same room, $\rho_S = p_{\text{room}}/p_{\text{zone}}$ for those events in the same zone, $\rho_S = p_{\text{room}}/p_{\text{floor}}$ for those events on the same floor, and $\rho_S = 0$ for the events that do not share regions at any level. Here p_{room}, p_{zone}, and p_{floor} are the probabilities for the event being in a particular room, zone, and floor, respectively. Users can incorporate their domain knowledge to refine the spatial correlation. For example, the correlation between an event in the server room and any other facility events can be set to at least 0.5, as all the facilities can be controlled in the server room.

Temporal Correlation. The temporal correlation indicates the closeness of time in relation to when the events occur. Depending on the type of the object and its facet, we consider either the overlapping time period of the events or the difference between their starting times. For events having a causal relationship, e.g., the setpoint of an air conditioner and the room temperature, their starting time difference, denoted as ΔT, is more important. The correlation is formulated as

$$\rho_T = \begin{cases} 1, & \text{if } \Delta T \le T_{\min} \\ \left(\frac{T_{max}-T_{min}}{T_{max}-\Delta T}\right)^{-\beta_T}, & \text{if } T_{\min} < \Delta T < T_{\max}, \\ 0, & \text{if } \Delta T \ge T_{\max} \end{cases} \tag{8}$$

where $T_{\min}$ and $T_{\max}$ are the boundary parameters of ΔT, beyond which the correlation is set to 1 and 0 respectively. $\beta_T > 0$ is the exponent of the power-law decay between $T_{\min}$ and $T_{\max}$.

For parallel events, e.g., the movement of two employees, the length of the overlapping time period, denoted as T_o, is more useful to define the temporal closeness, which is formulated as

$$\rho_T = \begin{cases} 0, & \text{if } T_o \le T_{\min} \\ \left(\frac{T_{\max}-T_{\min}}{T_o-T_{\min}}\right)^{-\beta_T}, & \text{if } T_{\min} < T_o < T_{\max}, \\ 1, & \text{if } T_o \ge T_{\max} \end{cases} \tag{9}$$

where $T_{\min}, T_{\max}, \beta_T$ are the set of parameters similar to Eq. (8).

Facet Correlation. The facet correlation indicates the closeness of the source of the events. In the facility monitoring scenario, this is determined by the hierarchy of the associated object-facet category. The sensors of the facilities and the movement of the employees are the two categories at the highest hierarchy. The sensors are further divided into heating-related, air circulation-related, and power-related categories. The movements are grouped by the employee's department. The events belonging to the same category at a lower hierarchy will be assigned a larger facet correlation score because they are closer to each other. The exact correlation score can be determined by the domain knowledge.

Object Correlation. The object correlation indicates the intrinsic long-term relationship among the objects, in comparison to the opportunistic spatial and temporal correlation between the short-term events. In separate scenarios, we consider two types of object correlation. The first type integrates the event data to capture the long-term object relationship. The second type leverages the external data to model the object relationship.

In the facility monitoring scenario, we compute the object correlation between two employees, denoted by o_i and o_j, by their spatial co-occurrence in the history. Consider a time period $\mathbf{T}$, this correlation is defined as the average spatial correlation weighted by the length of the overlapping event time period.

$$\rho_O(o_i, o_j, \mathbf{T}) = \frac{1}{\mathbf{T}} \cdot \sum_{\Phi_a \in \Phi(o_i,\mathbf{T}),\Phi_b \in \Phi(o_j,\mathbf{T})} \rho_S(\Phi_a, \Phi_b) \| t_a \cap t_b \|, \tag{10}$$

1. For convenience, we denote zone i on floor j as F j Z i.

where $\Phi(o_i, \mathbf{T})$ and $\Phi(o_j, \mathbf{T})$ are the sets of movement events for o_i and o_j during $\mathbf{T}$; Φ_a and Φ_b are the events in each set; and t_a and t_b are their corresponding time periods respectively. The object correlation between the sensor readings are not used because this has been captured by the facet correlation.

In the intrusion detection scenario, we compute the object correlation of the two hosts by the average network traffic between them. In the software analysis scenario, we use the data flow between the line of codes as their object correlation, which is the external source to the event data.

Fusing of Multimodal Correlations. Multiple fusing functions are provided to allow users to focus on the different aspects of the correlation. The *uniform fusing* is as follows:

$$\rho_F = \begin{cases} \rho_S + \rho_T + \rho_C + \rho_O, & \text{if } \rho_S \neq 0 \text{ and } \rho_T \neq 0 \\ 0, & \text{otherwise} \end{cases}, \qquad (11)$$

which is the summation of the spatial, temporal, facet, and object correlations when both the spatial and temporal correlations are not zero. To emphasize the impact of time, the *time-critical fusing* is defined as multiplying the uniform fusing by the temporal correlation, i.e., $\rho_{TF} = \rho_T^{P_T} \rho_F$, where P_T is a user-defined parameter. Similarly, the *space-critical*, *object-critical*, and *facet-critical* fusings can also be defined as multiplying the uniform fusing result by the respective correlations.

4.4 Abstraction of HOCG

The raw HOCG created by the point anomaly detection and correlation analysis often suffers from an overwhelming visual complexity. This is because the number of nodes (events) and edges (correlation) could be extremely large. Consequently, we introduce two methods to alleviate this effect.

Graph Simplification. We provide a filtering scheme that allows users to specify a time period $\mathbf{T}$ to generate a dynamic HOCG ($\mathbf{H}(\mathbf{T})$) that is smaller than the full-time HOCG ($\mathbf{H}$). The filtering starts from selecting the events whose corresponding time falls into $\mathbf{T}$, i.e., $\{\Phi_i | t_i \in \mathbf{T}\}$. To allow users to focus on the anomalies, a threshold on the anomaly score is selected; it is denoted by α_0. The events with higher (equal) anomaly scores than the threshold are kept. The correlation analysis is only conducted between these selected events. Similarly, a threshold of the fused correlation score is specified, denoted by ρ_0, so that only the correlations stronger (equal) than the threshold are retained. After the filtering process is conducted, the isolated events on the HOCG will be removed.

Object-Centric Abstraction. After filtering the HOCG, the remaining graph may still be large in size and complex in structure. To provide users with a feasible HOCG overview (*T1* in Section 3.2), we propose to abstracting the graph by the associated object of each event for visualization. This involves several steps.

First, on each object-facet pair $< o_i, c_i >$, we retrieve the list of related events $\{\Phi_j\}$ after the time and anomaly filtering. These events are merged together over time to form several continuous anomaly intervals, as shown in Fig. 2a. The merging rule is conducted to combine every pair of consecutive anomaly intervals if they are back to back on the timeline.

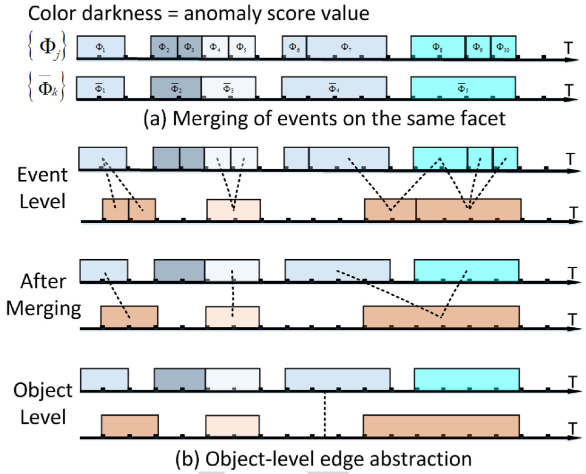

Fig. 2. Merging of events and event correlations over time.

To maintain consistency, we cut each interval at all the time points when the event's measured value changes. The final anomaly intervals are denoted as $\{\overline{\Phi}_k\}$. On each reconstructed anomaly interval, we compute its anomaly score by the function $\overline{\alpha}(\overline{\Phi}_k)$ over all the point anomaly scores of this interval. By default, we apply the $\max$ function to reveal the most notable anomaly

$$\overline{\alpha}(\overline{\Phi}_k) = \max_{\Phi_j \in \overline{\Phi}_k} (\alpha(\Phi_1), \ldots, \alpha(\Phi_j)). \qquad (12)$$

Second, the events for the same object are abstracted as a single object node. The associated events are organized by their facets on the object, sorted according to time, and visualized as the context of the node.

Finally, we form the object-level edges by merging the event-level correlations. As depicted in Fig. 2b, the correlation between two events will be merged into the correlation between the anomaly intervals covering these events, then to the correlation between the associated objects. The $\max$ function is used to compute the object-level correlation from the low-level components.

4.5 Anomaly Propagation

To fulfill the requirement *R3* in Section 3.1 and support the task *T4* in Section 3.2, other anomalies that are not currently in the HOCG should also be considered: 1) the event with a low anomaly score, but closely related to many highly anomalous events, which is critical for the root cause analysis; and 2) multiple mildly anomalous events strongly correlated to each other, which could potentially be a collective anomaly. We introduce an anomaly propagation based method that can detect these hidden anomaly patterns.

The basic idea is to propagate and re-distribute the anomaly score over the HOCG so that the anomaly score of the events in the above cases could be raised higher than the threshold, and be displayed in the visualization. The key challenge is that by default the unabstracted HOCG should be used as the input of the propagation, which can be extremely large at the event level. Moreover, computing the correlations among all these events leads to quadratic complexity. To tackle the challenges, we apply the anomaly propagation on the object-level HOCG after the abstraction. This object-level

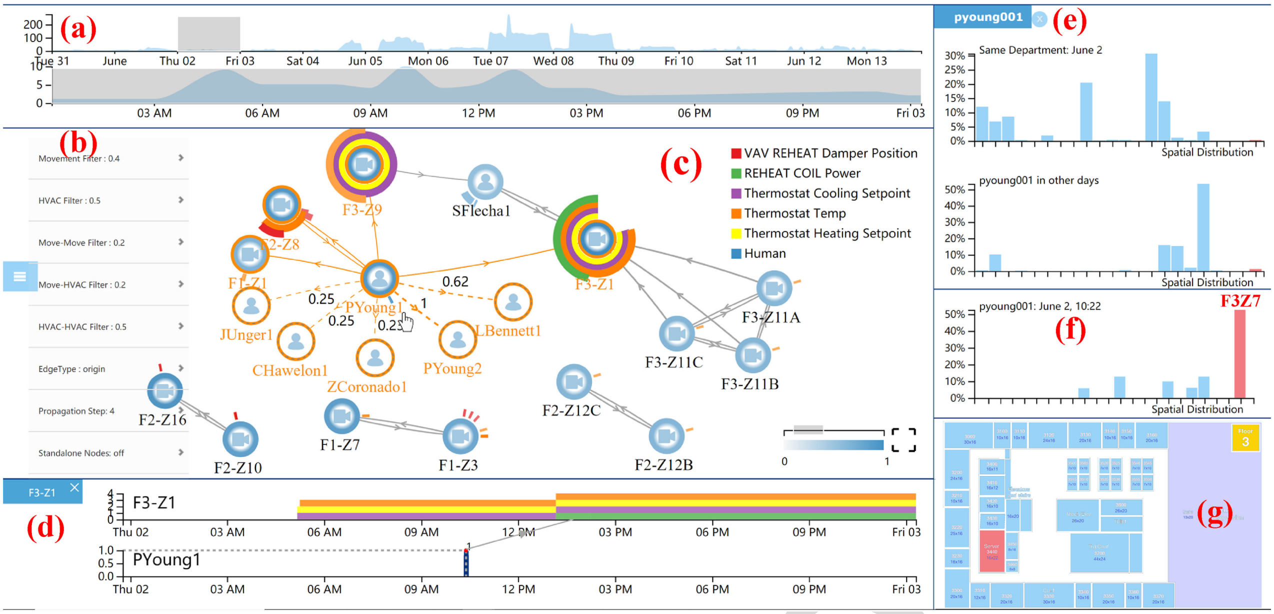

Fig. 3. The visualization interface of high-order correlation graph (HOCG): (a) double overview+detail timeline selectors; (b) visualization controller; (c) correlation graph view; (d) the anomaly time series of individual nodes (objects); (e) visual interpretation of a selected point anomaly; (f) the data value of the selected anomaly; (g) spatial detail view.

HOCG is then augmented by adding the other objects without any anomalies higher than the threshold. To avoid the full-scale correlation analysis among the events, we use the object correlation as the edge of the object-level HOCG.

The propagation starts from all the objects having their anomaly scores above (equal) the threshold α_0. They are denoted as the anomalous node set $\mathbf{O_a} = \{o|\alpha(o) \geq \alpha_0\}$. The algorithm of random walk with restart [42] is applied, which computes a similarity between any two nodes in the graph, denoted as $w(o_i, o_j)$ between o_i and o_j. After the propagation, each object o_i having an anomaly score lower than the threshold ($\alpha(o_i) < \alpha_0$) will be updated to a new anomaly score.

$$\alpha^*(o_i) = \alpha(o_i) + \sum_{o_j \in \mathbf{O_a}} (w(o_i, o_j) \cdot \alpha(o_j)) \qquad \forall o_i \notin \mathbf{O_a}. \tag{13}$$

In the augmented object-level HOCG, the objects with the new anomaly score lower than the threshold will again be removed.

5 Visualization

We designed and implemented a web-based visualization interface of the HOCG (Fig. 3). The interface is composed of four coordinated views: 1) the correlation graph view (Fig. 3c) that displays the HOCG structure for the static anomaly analysis within a certain time window; 2) the overview+detail timeline selectors (Fig. 3a) that filter the HOCG by the selected time window and enable the dynamic analysis; 3) the event view (Fig. 3d) that shows the event time series on interrelated object-facet pairs and helps to examine the root cause of certain anomalies; and 4) the anomaly detail view (Fig. 3e, 3f, 3g) that visually explains the source of each point anomaly and its static/dynamic context.

5.1 Design Principles

We follow three principles in designing the interface, 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 about the collective anomaly from a large amount of low-risk point anomalies. Therefore, it is important to present an overview map of the point anomalies first, so that users can zoom (on the time axis) and filter (by the anomaly and correlation scores) to access the details. Essentially this resembles Shneiderman's visual information seeking mantra [43].
- *From static to dynamic*: On analyzing the collective anomalies, both the static and dynamic patterns are critical. The static pattern reveals the relationship among the point anomalies. The dynamic pattern illustrates their formation and evolution over time. In fact, there is an inherent paradigm in the users' analysis process: we observe the static relationship first and then proceed to discover how it forms. Finally, we reason about 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 work is to discover the root cause of a certain fatal anomaly or failure. This requires detecting a primary anomaly path from the fatal anomaly back to the potential root cause. The visualization is therefore designed to help complete this task. We introduce the interactions to manually inspect the point anomalies and the path-based correlation to connect the dots among the verified point anomalies.

5.2 Timeline Selector View

Both point and collective anomalies evolve over time. In our interface, we propose an overview+detail design to filter the HOCG according to the selected time window. As illustrated in the top row of Fig. 3a, a first overview chart is displayed to represent the number of anomalous events over time. Users can obtain a full picture of what is happening

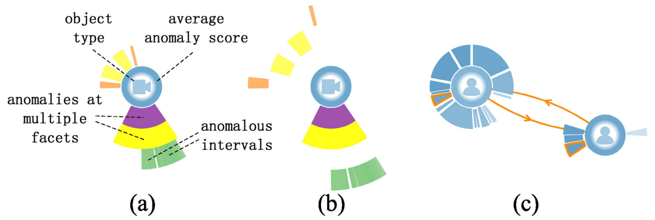

Fig. 4. The multi-layered wedge-based visual metaphor: (a) the node with stacked wedges, where each colored layer corresponds to a facet of the object, and each wedge in a layer corresponds to a time interval having the same anomaly score on this facet; (b) the design without folding; (c) hovering one wedge of an object, the correlated wedges on the other objects will be highlighted.

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 Fig. 3a, the detailed time window selected in the top row is expanded. To conduct a finer-grained time series analysis, users can choose a subset of the currently selected time window. The HOCG in Fig. 3c will be filtered to the nodes and edges on this subset of time. This double filtering design allows for drilling-down to very small time windows when some critical anomalies occur intensively.

5.3 Correlation Graph View

The correlation graph view in the center (Fig. 3c) visualizes HOCG as a node-link graph. Each node in the graph represents an object (a room/zone/floor of a facility, an employee of a company, a line of code) on which at least one anomalous event happens during the selected time window. Each edge between the two nodes represents their relationship by the multimodal correlation. We apply GraphViz [44] to compute the layout of HOCG, which provides multiple algorithm options, e.g., stress majorization, hierarchical layout.

For each node, a multi-layered wedge-based metaphor is designed to visualize the anomaly time series on this object. As shown in Fig. 4a, 4b, the visual metaphor is composed of an icon in the center, a filled ring surrounding the icon, and multiple layered rings in the outermost section. Each layered ring is further composed of several wedges arranged in a circular layout. The icon in the center of the node represents the object type. For example, the facility measured by sensors is drawn as a camera icon, the employee is drawn as a people icon, and the host is drawn as a computer icon. On the surrounding ring, the darkness of the fill color indicates the average anomaly score of the object in the selected time window. A larger anomaly score will be displayed in a darker color. In the outermost layered rings, each ring is colored with a different hue and represents a separate facet of the object, e.g., the cooling/heating setpoint, the air temperature (also shown in the legend of Fig. 3c). Each wedge of a layered ring indicates a time interval having the same anomaly score on the corresponding facet. The starting position of the wedge indicates the beginning time of the interval within the selected time window. The angle of the wedge indicates the length of this anomalous time interval. Each layered ring corresponds to the entire time window selected in Fig. 3a. In this way, we can interpret the node as a clock with the earliest time mapped to 12 AM. The wedges are displayed on the clock to visualize the temporal distribution of the anomalies on each facet. The fill color darkness of each wedge indicates the anomaly score of the corresponding time interval, using the same color mapping as the inner ring.

The default multi-layered metaphor design in Fig. 4b suffers from two drawbacks: 1) the node size will grow quadratically as the number of facets increases; and 2) it is difficult to perceive the dynamics of all the anomalies on the same object. To alleviate these drawbacks, we improve the design by folding the layered rings. As shown in Fig. 4a, starting from the second layer (yellow), each wedge of the ring will be collapsed towards the center of the node if it does not overlap with any wedge in the inner rings. By conducting this folding operation, each node will be displayed in a more compact manner, and the overall anomaly time series can be easily perceived. A side effect of this design lies in the inappropriate visualization of the per-facet anomaly time series except for the first facet. We further introduce an interaction method, as the user clicks on one outer ring, this ring will be switched to the first inner layer so that its anomaly time series can be revealed.

In our design process, we once considered the GrowthRingMap [45] as the node metaphor of HOCG. Each anomalous event is represented by a filled ring and is stacked on the central icon of the node in a radial order according to the event time. The color hue and darkness of the ring represent the time and anomaly score of the event respectively. This ring-based design is later discarded due to three limitations: 1) both the event time and the anomaly score are at least ordinal variables, which can not be simultaneously displayed in the visual channel of color; 2) the design can not visualize the facet information of HOCG; 3) the size of the node grows indefinitely with the number of anomalies, leading to an unbalanced view with large variations on the node size. The multi-layered wedge-based metaphor in our final design applies the clockwise order to encode the time and stacks multiple facets in the radial order. The node size is bound by the limited number of facets and further reduced by the folding design.

Meanwhile, the edges drawn in the solid line style indicate the high-order relationship computed in Section 4.3. The dashed edge indicates the extended relationship by the anomaly propagation in Section 4.5. The edge thickness indicates the fused correlation score. The edge direction is determined according to the anomalous time intervals of the two connecting nodes. By the visual abstraction in Section 4.4, the node with an earlier time interval will point to the other nodes with later time intervals, except for object correlations, where we use their inherent directions. As there are cases where two nodes have a bidirectional relationship, we draw curved edges to distinguish the edge directions.

5.4 Event View

On the correlation graph view (Fig. 3c), users can drill down to each node with a single click. The anomaly score time series of the corresponding object will be displayed as a row in the event view (Fig. 3d). Each row visualizes the anomalies that occurred on the object as stacked bar charts, where each stack corresponds to a facet of the object. To reason about the root cause of the anomalies, users can click on

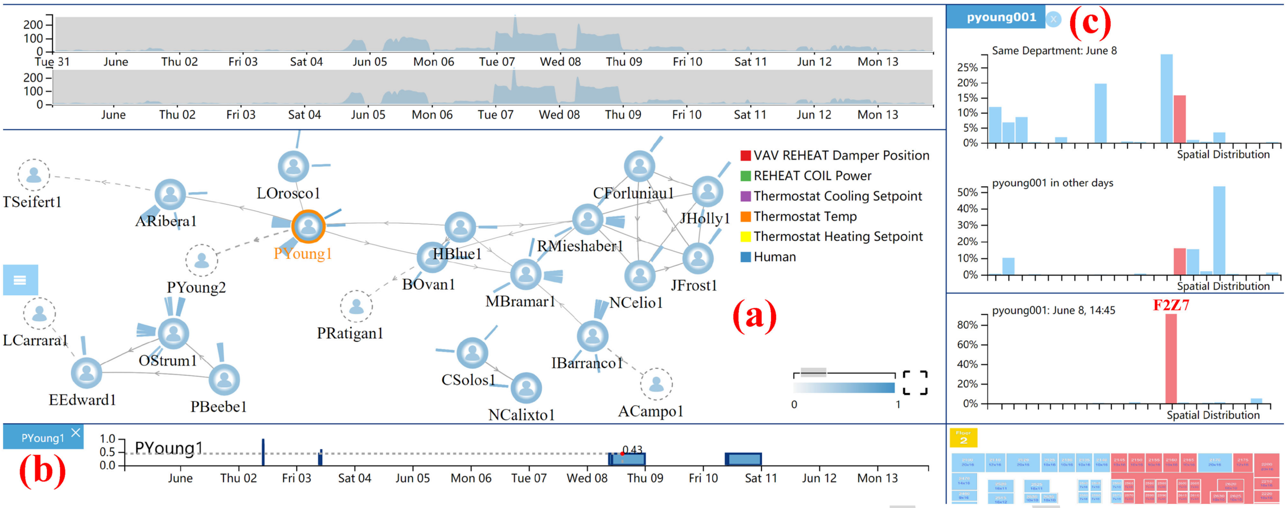

Fig. 5. The HOCG containing suspicious company employees and their anomalous events during the entire two weeks: (a) the correlation graph view; (b) the event timeline of PYoung1; (c) the detailed explanation of PYoung1's anomaly on June 8.

another node that correlates with the anomaly of the previous node. Additional rows are added to the bottom of the view. Links are drawn between the two rows to indicate their relationship, thus forming a reasoning path. When users click on a new node unrelated to the existing reasoning path, another tab will be opened to illustrate a new path for the root cause analysis.

5.5 Detail View

In the event view (Fig. 3d), users can drill down to examine each point anomaly by selecting a time point on the anomaly time series. The corresponding event is visualized in the detail view on the right part of the interface (Fig. 3, 3e, 3f, 3g). Note that for different data types, the detail view will have customized designs. For example, on the movement data, we depict the histogram of the selected employee's spatial distribution in Fig. 3f, which is compared with the average employee's distributions in Fig. 3e for the model explanation. The location of the selected event is displayed in Fig. 3g.

On the sensor data analyzed in the first case study (Section 6.1), the detail view will illustrate all the events on the selected time point. On each event, a line chart in blue is drawn to represent the GMM model of the normal profile (Fig. 6c, 6d, 6e). The measured value of the selected event will be drawn in red on the line chart. This design visually interprets our point anomaly detection algorithm by showing how the event deviates from the normal profile, i.e., as an outlier of the model. The measured values surrounding all the selected events are displayed below the chart views as time series (Fig. 6f), which enables the user to drill-down to the level of the raw data.

5.6 Interaction

In terms of interaction, HOCG supports basic network visualization interactions, including zoom&pan, node drag&drop, and neighborhood highlights, etc. When users select one wedge with a mouse hover action in Fig. 3c, this wedge and all the other wedges having a direct correlation in the event level will be highlighted, as shown in Fig. 4c.

In addition, we 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 Fig. 3b, we build node and edge filters that allow users to access point anomalies and correlations above certain anomaly and correlation thresholds. Note that the filters are arranged by the node type (e.g., employee, facility) and edge type (e.g., mhFilter indicates the edges between employees and facilities). The other two interactions are the time-based filtering for the dynamic anomaly analysis and the node/edge detail accessing for the root cause analysis, which have been introduced in Sections 5.2 and 5.5 respectively.

6 Case Studies

6.1 Facility Monitoring

We first consider the facility monitoring scenario released by IEEE VAST Challenge 2016 (VC16) [46]. The VC16 dataset contains two weeks of operation data for a company's three-floor building. Each floor is divided into multiple zones. Two types of monitoring data are collected: the heating, ventilation, and air conditioning (HVAC) data for each zone; and the movement data for each employee in the company. The HVAC data was generated every five minutes by fixed sensors, which record the environmental conditions, such as the temperature, the concentration level of the carbon dioxide and other chemicals, and the heating and cooling system statuses, such as temperature set points and damper positions. The movement data records the locations of the employees who were required to carry a proximity card. The proximity card readers in each zone would record the proximity card ID, time, and the zone being entered, when a card moved from one zone to another. During the time of the provided dataset, suspicious activities were conducted in the building. Detecting, analyzing, and reasoning about these activities is the major task of the challenge.

We apply HOCG to tackle the VC16 challenge, where the mapping from data to HOCG has previously been introduced. In the analysis, we first investigate the suspicious employees over the entire two weeks. We filter the HOCG to remove all the HVAC anomalies and only show the

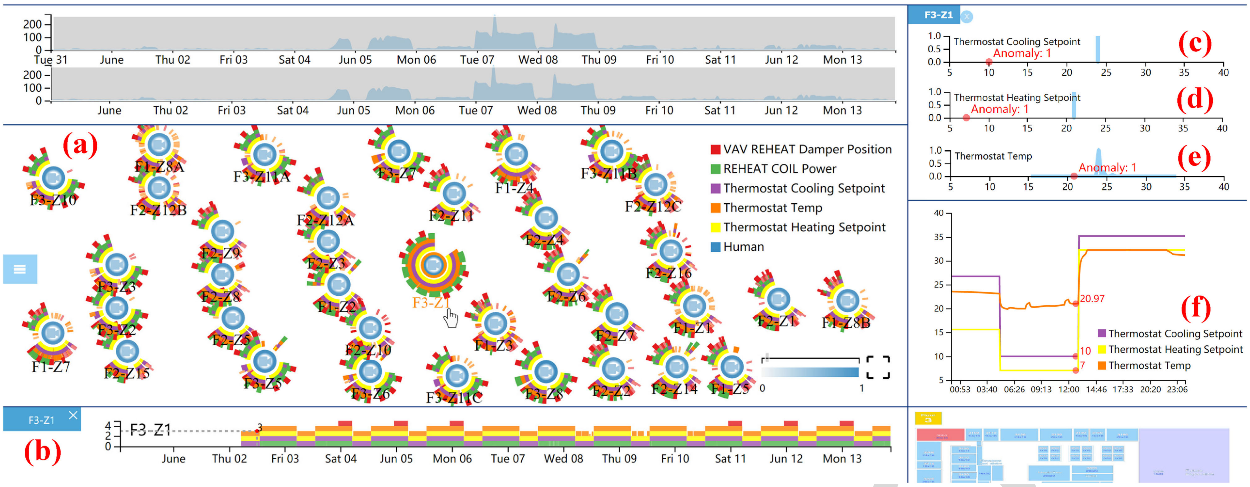

Fig. 6. The HOCG containing HVAC anomalies during the entire two weeks: (a) the correlation graph view; (b) the event time series at F3Z1; (c)(d)(e) the detailed explanation of selected anomalies at F3Z1; (f) the raw sensor readings of the selected anomalies.

employees with moderately high anomaly scores (≥ 0.4). 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 Fig. 5a. The graph illustrates that three employees (i.e., RMieshaber1, MBramar1, and PYoung1) have more connections than the others. By investigating the anomaly details for the three employees, we discover that PYoung1 is especially suspicious for three primary reasons. First, his anomaly score time series presents a significantly higher spike on June 2 (Fig. 5b), which is not found for the other two employees. Second, his anomalous events on June 8 and 10 last for almost the entire day (Fig. 5b). Third, there is another employee PYoung2 connected to PYoung1 by propagation (Fig. 5a), due to their high facet correlation. This indicates that two active cards for the employee "PYoung" exist at the same time, which is highly suspicious. By selecting June 8 for a detailed exploration, the histogram of PYoung1's movement on June 8 is compared to the histogram of all the other employees from the same department and the histogram of his own movement on other days (Fig. 5c). The behavior of PYoung1 is suspicious as he mostly stayed in one zone (F2Z7) for the entire day. This is a zone that he only visited a few times during the other days.

We then study the anomalous HVAC events. Due to the large number of HVAC anomalies, we apply an anomaly score threshold of 0.8 so that only the highly suspicious HVAC anomalies are shown. The corresponding HOCG visualization is given in Fig. 6a for the entire two weeks. Multiple types of HVAC anomalies are present. The most frequent HAVC anomalies are temperature-related, i.e., cooling/heating set points and thermostat temperature. Among the building zones, F3Z1, which is the CEO's office, has the highest number of anomalies (the center of Fig. 6a). To better understand the details of these anomalies, we click on the node of F3Z1 to retrieve its event timeline (Fig. 6b). Then we select a typical time of 12:55 PM, June 2 on F3Z1 to access the explanation for the anomaly. The detail views in Fig. 6, 6c, 6d, 6e show that all the three temperature-related anomalies have their sensor readings largely deviated from the GMM model of the normal profile. By looking at the raw sensor readings (Fig. 6f), it is revealed that both cooling/heating set points were turned up, from 10/7°C to 35/32°C at 13:00 PM. The zone temperature followed accordingly. By a similar analysis on F3Z1, we conclude that someone was altering the HVAC setting of the CEO's office repeatedly, which poses a big security threat to the company.

After identifying the suspicious employees and HVAC events, it is hypothesized that these two types of anomalies are potentially interlinked. We start to validate this hypothesis by investigating each individual event. We first pick the day of June 2 for exploration, when the highest anomaly score is found for PYoung1. We display both the employee's movement events and the building sensor's HVAC events to reveal their correlations. The resulting HOCG visualization is shown in Fig. 3c. It is observed that PYoung1 is at the center of the graph leading to most of the HVAC anomalies including those at F3Z1, and his anomaly score also propagates to five highly related employees. We then form the reasoning path from PYoung1 to F3Z1. In Fig. 3d, the event timeline view shows that after a short appearance of PYoung1's anomalous activity, a new series of anomalies happened at F3Z1 on both the cooling/heating set points, temperature, and coil power. Fig. 3g also indicates that PYoung1's anomalous activity happened at F3Z7, the HVAC control room, where the HVAC setting of all zones can be configured. A further investigation on the entire anomaly timeline of PYoung1 (Fig. 7a) reveals that all the highly anomalous events of PYoung1 occurred at F3Z7, where he potentially overwrote the HVAC setting of the building zones.

We then analyze the relationship of PYoung1 with the other five employees detected through propagation. The largest correlation happens between PYoung1 and PYoung2, as indicated by the thickness/label of the edge between them (Fig. 3c). This is simply because the two cards belong to the

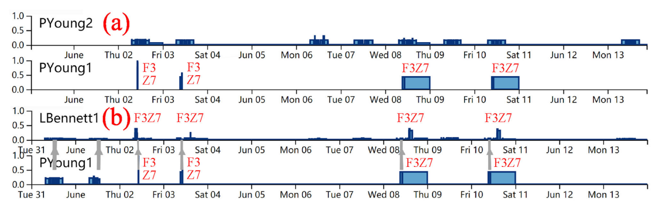

Fig. 7. The anomalous event time series over the entire two weeks: (a) PYoung1 and PYoung2; (b) PYoung1 and LBennett1.

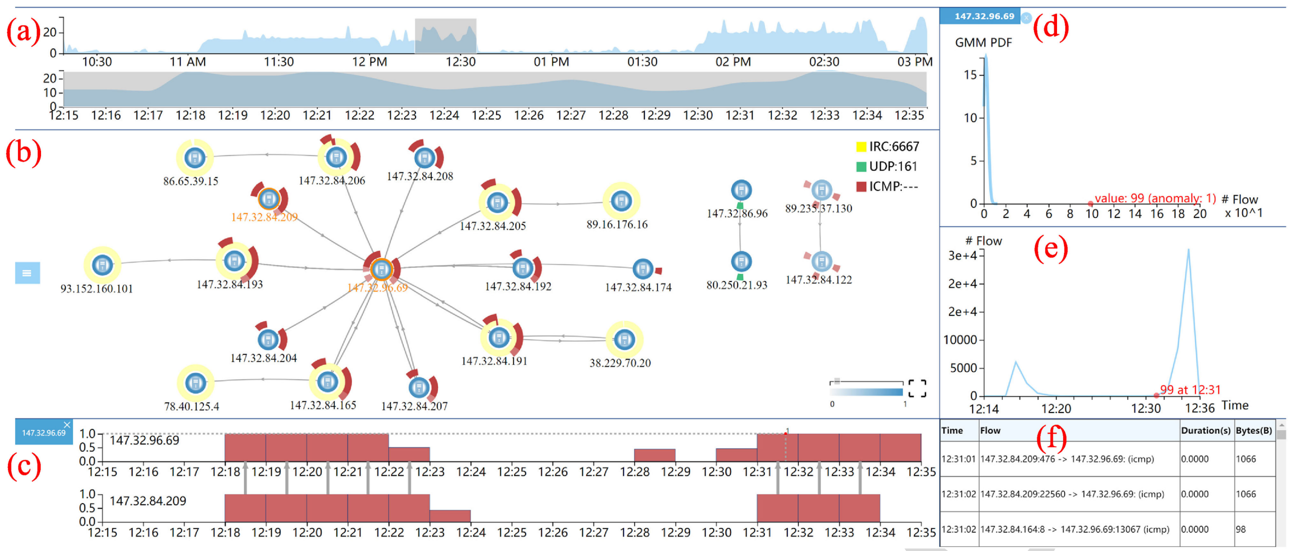

Fig. 8. The HOCG visualization of the CTU-13 dataset. The two largest anomaly spikes from 12:15 PM to 12:35 PM are selected.

same employee. The second largest correlation is found between PYoung1 and LBennett1, with a correlation much higher than the other employees. In Fig. 7a, we find that PYoung1 and PYoung2 do not exhibit any spatiotemporal correlation during the entire two weeks. Nevertheless, in Fig. 7b, we discover that almost every appearance of PYoung1 at F3Z7 with a high anomaly score is accompanied by LBennett1. In addition, Fig. 5c shows that PYoung1 spent almost the entire day of June 8 and 10 in F2Z7, where LBennett1's office is located. These findings suggest that PYoung1 is closely related to LBennett1. According to the challenge dataset, PYoung (Patrick Young) and LBennett (Loretta Bennett) both work in the facility department of the company. PYoung is LBennett's manager and has the privilege of visiting the HVAC control room (F3Z7). By summarizing the discoveries, we conclude that the major security threat to the company lies in the frequently overwritten HVAC settings, especially for the CEO's office. The direct suspect is identified as PYoung whose visitation to the control room highly correlates with the HVAC anomalies. It is possible that he may use two proximity cards to disguise his suspicious behavior. In the meanwhile, PYoung has one team member namely LBennett; they may plan all their activities together.

6.2 Intrusion Detection

We apply HOCG on a typical network intrusion detection dataset: CTU-13 [47]. The dataset is composed of large-scale botnet traffic mixed with normal traffic and background traffic. The botnet traffic is generated by executing real-world malware on the selected hosts of the network (i.e., bots). These hosts use several protocols to perform malicious actions (e.g., port scan, click fraud, email spamming). The dataset considered here contains 90 M packets out of 1.3 M flows from 20 k hosts, with a duration time of 5 hours. The original packet data has been translated into the list of directional flows between the hosts as the raw data of our system.

The primary objective of the CTU-13 scenario is to better understand the malware-based intrusion detection in typical networking environments. The design goal of HOCG fits this objective well in relation to analyzing malware anomalies. In the application, each host computer with a standalone IP address is modeled as an object (i.e., node) in HOCG. The protocol that transfers network traffic on this host at a particular (set of) port(s) is considered as a facet of the object, e.g., TCP:21, UDP:161, IRC:6667. The network traffic to/from each host using a particular protocol:port is considered as events that occurred on this object-facet pair. To reduce the number of events for a scalable analysis, we aggregate all the events into fixed time bins (one minute each in this study), so that each object-facet pair will have only one event in each time bin. For each event, several statistics in the corresponding time bin are computed as the values of the event. These statistics include the number of active flows, the number of connected hosts, the average number of active flows with each host, the size of the transmitted traffic in bytes, and the average duration of the active flows. The point anomaly detection algorithm in Section 4.2 is applied to each statistic of the event. The highest anomaly score is used as the anomaly score of the event. The dataset in an early time period, when the malware is not executed, is used as the normal data to build the model. Among the events, we treat the directional traffic flows between the hosts using the corresponding protocol:port as their correlations (edges). In other words, only the object correlation is used. The spatial/temporal/facet correlations are not considered because the network flows already represent the spatial/temporal/facet affinity between the hosts.

The initial HOCG visualization on the whole CTU-13 dataset illustrates a large network consisting of 2976 anomalies detected during the 5-hour time period. This indicates the complex behavior of the studied malware. The anomaly time series in Fig. 8a can be divided into two bursty periods. To examine the first period, we switch to an anomaly threshold of 0.5 to analyze the most significant anomalies and select the two largest spikes from 12:15 PM to 12:35 PM. The correlation graph view then displays a star-like topology in its largest connected component, as shown in Fig. 8b. The node in the center represents the host of 147.32.96.69 (96.69 in short if the IP prefix is repeated). The 10 surrounding nodes represent the hosts of 84.165, 84.191 $\sim$ 193, and 84.204 $\sim$ 209. These 10 hosts share similarly shaped wedges on the ICMP

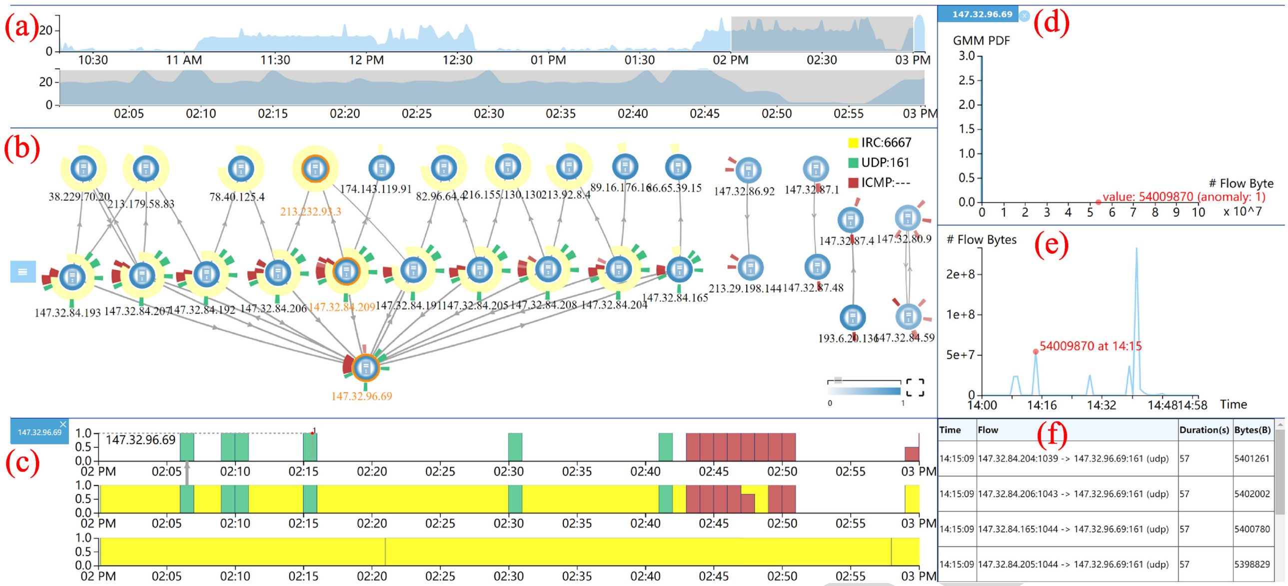

Fig. 9. The HOCG visualization of the CTU-13 dataset. The last hour (14 $\sim$ 15 PM) is selected for analysis.

protocol (red wedges), mostly composed of two continuous anomalous time periods. These two time periods also correspond to the anomaly pattern in the central host. Most of the network traffic is sent to the central host (96.69). Therefore, it is highly suspected to be a coordinated attack from the 10 internal hosts (bots) to the central host (server).

We validate this hypothesis by drilling down to the details of each host. As shown in Fig. 8c, the ICMP anomalies on the central host and one of the internal host are aligned in the timeline. There are network flows between them in most of the anomalous time periods. Furthermore, we click on one time point of the central host, i.e., the minute of 12:31 PM, to retrieve the visual explanation of the corresponding anomaly. Fig. 8d reveals that the number of flows (NF) on the central host during this minute (99, the red dot) deviates largely from the GMM model built from the normal data. In the timeline of Fig. 8e, there is also a spike on the NF measure starting from this minute. By clicking on one of the internal hosts in the following minute (Fig. 8c), we discover a similar deviation and spike on the average number of flows per host (ANF), which accounts for the root cause of the anomaly in the central host. All the following three minutes share the same pattern, i.e., a high NF in the central host and a high ANF in the internal hosts. Finally, the directional flows, as the raw data in the selected minute, are displayed in Fig. 8f, which lists a large number of flows of a small size, initiated at 12:31 PM (e.g., 1 KB). This finding confirms our hypothesis on the DDOS attack from the internal hosts to the central host using short-lived ICMP pings.

In another trial, we analyze the second anomalous time period by selecting 2 $\sim$ 3 PM on the interface (Fig. 9a). The HOCG view, as shown in Fig. 9b, reveals a three-layered structure after applying the hierarchical layout algorithm. In the central layer, the 10 internal hosts (bots) again exhibit similar anomaly patterns. Different from the first analysis trail, the anomalous events now come from three different facets (protocols:ports) of the objects: ICMP, UDP:161, and IRC:6667. These internal hosts connect to the same host of 96.69 in the bottom layer, which behaves anomalously in the ICMP and UDP protocol during the similar time periods. Drilling down to the detailed anomaly timeline in Fig. 9c, the ICMP anomalies are found to be the same type of DDOS attack as in the first analysis trail. To better understand the UDP anomalies, we select the minute of 14:15 PM. The visual explanation in Fig. 9, 9d, 9e reveals that the UDP anomalies co-occur with the spikes on the size of the transmitted traffic (NB). These spikes align well with the UDP anomaly time series on the host of 96.69 (the first row of Fig. 9c). This pattern suggests a UDP-based DDOS attack from internal hosts to 96.69. Different from the ICMP DDOS, the UDP attackers send a much larger volume of traffic to the victim. This can be found in the list of flows in Fig. 9f, where a UDP flow as large as 5.4 MB in size is initiated.

In the meanwhile, there are 9 external hosts (not in the subnet of 147.32) in the top layer of Fig. 9b. Each external host communicates with 1 $\sim$ 3 internal bots and has the same anomaly timeline on the IRC protocol as the connected bots. The IRC protocol is notorious as the communication channel between the command-and-control server (C&C) and the bots. Hence, these external hosts are highly susceptible to be the C&C servers. To validate our hypothesis, we drill down to the detail view and find that the anomaly is caused by an extraordinarily long connection time on the IRC protocol, when compared with the normal behavior. The C&C server would take this long time to issue the next batch of commands to the connecting bots. Therefore, the detected collective anomaly can be concluded as the ICMP/UDP DDOS attack on a single server from multiple internal bots which are coordinated by external C&C servers.

6.3 Software Analysis

In another case, we deploy the HOCG to detect the collective anomalies in a runtime execution of software which is known to have certain security vulnerabilities. The raw data are from the monitoring of such runtime 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;

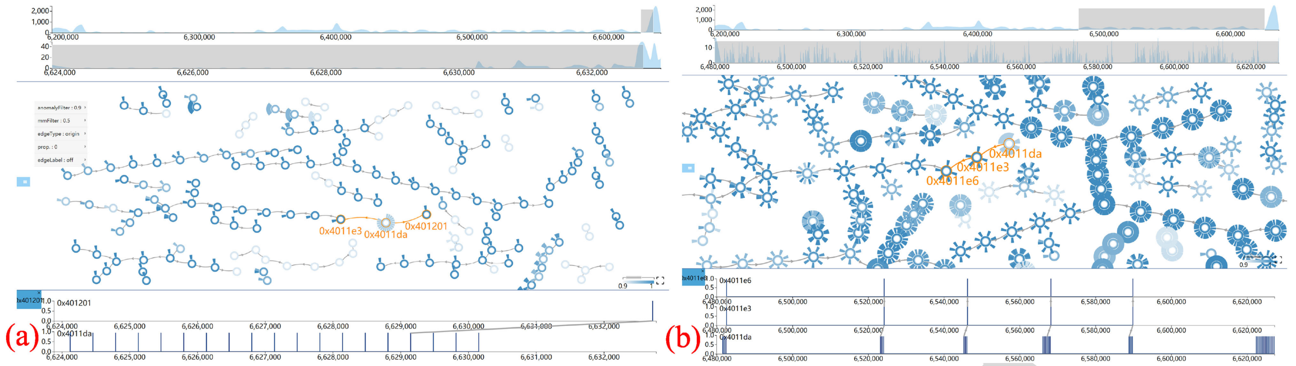

Fig. 10. Software analysis case study: (a) the initial HOCG view selecting a smaller time window close to the crash point; (b) zooming out to a large time window for the root cause analysis.

"op_vals" are the operator values; and "src_ids" and "dst_ids" are the executions affecting, or affected by, this execution.

For this dataset, 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 the events is detected by the algorithm in Section 4.2. The same software is executed twice. During the first execution, no compromise of the security vulnerability is conducted and the execution data are used as the normal profile. During the second execution, the software vulnerability is triggered and the execution data are used to construct the HOCG.

The initial overview of HOCG is shown as Fig. 10a. The entire dataset contains 6 million lines of executions. We load the last 400,000 lines, which are close to the crash point of the software. We first examine the overview panel in the top row of Fig. 10a. 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 Fig. 10a. In this graph, most anomalies are shown to have occurred 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 a greater number of wedges on the node than that of the others (the highlighted node at the center of Fig. 10a). To drill-down to the details, we click on this node to expand its anomaly events over time. The bottom row in Fig. 10a shows a 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 reasoning path, we find that the node of 0x401201, as shown by the row on top of 0x4011da in Fig. 10a, contains only one anomalous event at the end of the timeline. We conclude that 0x401201 is the line of code leading to the fatal crash, and that 0x4011da behaves as the direct cause of this crash.

To find out the root cause of this crash, we select a larger time window of 200,000 cycles before the crash. The corresponding HOCG is depicted in Fig. 10b. The relationship between 0x4011da and 0x4011e3 is unchanged. By expanding their anomaly timeline again, it is found that the line of code at 0x4011da has triggered regular anomalies on 0x4011e3 for a long time, before leading to the final crash by the code at 0x401201. We bring our findings to work with a source code analysis expert. Based on our visual analysis result, we are able to restore the situation of this software crash. 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, i.e., an invalid 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 becomes overflown and it begins to run anomalously. The code at 0x4011da continues to overflow at its destination address in writing the overlong input string until the function address of the "call" instruction at 0x401201 becomes overflown. This leads to the irreversible software crash.

7 Expert Feedback

On applying HOCG to the intrusion detection scenario, we invited three network security experts to a trial study of the CTU-13 dataset using our visualization tool. The study is composed of two sessions: the training session and the test session. In the training session, the experts were provided with a user manual to become familiar with the visualization tool, including the visual design, data mapping, and interactions. Then they were asked to conduct some simple analyses on the sample data to practice their skills with the tool. We answered all their questions during the training session to ensure an appropriate level of understanding of the visualization tool. During the test session, each expert was provided with a full CTU-13 dataset (5 hours), and was asked to complete three tasks with the visualization tool: (1) identify at least 5 anomalies in the data, and provide details on each anomaly (e.g., time, host, behavior); (2) discover the relationship among these anomalies; and (3) infer the possible root cause of these anomalies. After finishing the tasks, the experts were asked to provide detailed feedback on the pros and cons of the tool, their previous experience in working with a similar scenario, and the potential extensions of the tool on the functionality and application domains.

The first expert is the IT manager and network administrator of a large department (~ 200 employees), who is responsible for the monitoring and troubleshooting of the department's Intranet. Initially, it was not easy for him to apprehend the HOCG visualization because most commodity tools display the actual network traffic, both normal and abnormal, while ours only displays the anomalous part of

TABLE 2
The Computation Time of HOCG Analytics and Visualization in Section 6.1

Stage / Measure	Offline (all computations)		Online (the computation for Fig. 3c)		
	Point anomaly detection ($\alpha \geq 0.2$)	Correlation analysis ($\rho \geq 0.2$)	HOCG generation	Anomaly propagation	Layout
#Node (#Anomaly)/#Edge	7072/—	—/13253	44/38	15/23	20/28
Time (second)	2.55	2882	0.17	0.33	< 0.01

the traffic. Nevertheless, he was able to get used to our tool after the 30-minute training session. During the test session, the first expert quickly identified the victim of most attacks (i.e., 96.69), and several true attackers (i.e., bots) in accordance with the ground truth, as we only asked him to locate five anomalies. He also concluded with the correct root cause of these anomalies: the ICMP DDOS attack. The UDP and IRC anomalies were noticed, but the three-layered anomaly structure at the end of the dataset was not found. During the analysis, his most praised feature of the tool was the ability to generate alerts for the administrators and display them on the network topology. He thought it will be straightforward to illustrate these alerts in real time. The suggestions he provided focused on the integration of our design with the mainstream network monitoring tools (i.e., nagios, zabbix, cacti) by adding the classical network traffic visualization (e.g., time series charts). He also suggested distributing the node anomalies into the edges, which fits better with the administrator's expectations.

The second expert is a researcher in computer security, who is also the adjunct network administrator of his lab. This expert has extensive experience managing networking devices (e.g., routers, firewalls). He quickly understood the correlation graph view and the event view. Though not required as a user, he was also interested in understanding the GMM model behind our anomaly detection algorithm. During the real test, and similar to the first expert, the second expert was able to locate the central victim, a few bots, and the type of DDOS attack using ICMP and UDP. Compared with the firewall log analysis tools he was using as his role of the network administrator, he thought our tool provided a unique global view of the network anomalies. The correlation analysis was also valuable in linking these anomalies together. For future extensions, he suggests analyzing the content of the network traffic. The content data was not available in the currently studied dataset.

Our third expert is a senior engineer on network security products, who is knowledgeable with the mainstream software features on the analysis of network anomalies. He could also quickly locate the timeline of the anomalies, from which he found the victim and some of the bots in the attack. He called the ICMP/UDP scanning a "flood attack". He did not notice the three-layered structure. During the analysis, the third expert found that the interaction design of the tool was convenient, compared with the existing network administration tools. The commodity software, e.g., the security gateway, relies on the previously defined models of a network anomaly, including the known incidences, firewall rules, and security knowledgebase. Our tool has the potential to work with unknown anomalies by incorporating the flexibility of human intelligence. This is critical in the networking scenario because the network traffic is in general bursty and complex, making it difficult to be governed by a few models. In the suggestions, the third expert recommended extending the analysis to include more security information (e.g., the state of the hosts, the packet content, the firewall logs), which are intensively analyzed by the existing security products. He would like us to develop our tool as the decision-making software, beyond the general "data presentation" software in the market.

In summary, all the experts could use the tool successfully after the training. All of them could correctly detect the ICMP or UDP DDOS attack through the linked view of the anomalous hosts. No one seemed to notice the IRC C&C channel, as they seldom select a large time window for analysis. On the positive side, the experts mentioned a few features of our visualization that accelerate their analysis tasks, including the flexible visual analysis without known models, the interactive global anomaly view, and the (real-time) alert visualization together with the topology network. On the other hand, all of them mentioned the importance of customizing the HOCG visualization in the network administration domain, including adding the network traffic charts, analyzing detailed network information (e.g., packet content), and incorporating a networking and security knowledgebase.

8 Discussion

The evaluation of our visualization framework reveals several limitations of the HOCG and suggests interesting future directions.

First, our framework can scale to analyze a huge amount of raw data. In the case of facility monitoring (Section 6.1), there are 40 types of sensor readings collected on 38 zones in more than 4,000 time periods, summing up to 6M+ data entries. As shown in Table 2, all the data processing carried out offline takes 48.1 minutes on a cloud server with four virtual CPUs and 16 GB of memory. The online computations for a typical graph of Fig. 3c take less than one second, which applies the object-centric abstraction to simplify the HOCG.

Despite the scalability in the data analytics, the HOCG visualization can still suffer from overwhelming visual complexity when the number of objects is extremely large. The introduction of the facet field in the event modeling helps to reduce the visual complexity. A higher-level object hierarchy can be selected as the node of the HOCG to reduce the number of nodes/edges in the HOCG. For example, in the facility monitoring case study, we use the zones containing multiple sensors as nodes of the HOCG, rather than using the individual sensors as nodes in the conference-version design. The direct sub-hierarchies of the object can be defined as the facets to illustrate the extended information on the object, i.e., the sensors installed on the zones. In the future work, allowing the users to set and navigate the object hierarchy will be a valuable extension for the HOCG design. The visualization can then be configured by the users to manage the visual complexity through setting the appropriate object hierarchy as nodes of the HOCG (e.g., the floors containing multiple zones). On the other hand, when there are only a few objects in the HOCG, the point anomalies detected on each pair of objects could be re-distributed into the links between the objects for a finer-grained analysis. For example, the overly high traffic flows between the hosts could be visualized as the anomalies on the link between the HOCG nodes.

Second, while the HOCG visualization focuses on the anomalies extracted from the everyday data, in many scenarios, the normal data pattern plays an equally important role in analyzing the collective anomaly. For example, the average traffic chart over time helps to identify the core of a computer network (i.e., routers/servers), which are vulnerable to the distributed attacks identified as collective anomalies. It is a nontrivial problem to effectively abstract the normal data pattern and integrate this pattern with the existing anomaly visualization.

Third, the experts in our study mentioned domain-specific requirements. To apply HOCG to a real-world scenario, it is critical to construct the HOCG visualization template for different domains (e.g., our design in Section 6.2 for analyzing the anomaly of computer networks). For the applications in the same domain, the final adaptation can be achieved by further designating a different set of parameter values, e.g., a low point anomaly threshold for more steady data center networks and a high threshold for the campus network due to its traffic randomness.

The video demonstration of this work can be found at http://lcs.ios.ac.cn/~shil/share/HOCG-TVCG.mp4, and the code repository is hosted at https://github.com/visdata/HOCG/tree/TVCG/.

9 Conclusion

In this paper, we describe a visual analytics framework based on the concept of the faceted High-Order Correlation Graph to detect, analyze, and reason about collective anomalies. The HOCG captures the multimodal relationships among the heterogeneous types of objects and events. It can be generalized 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 to identify the collective anomalies. In addition, we design an interactive visualization interface that allows the flexible and scalable exploration of detected point anomalies, their multimodal relationships, and the potential root cause of the overall collective anomaly. Users can drill down to the raw data in the detail view to validate their discoveries. We demonstrate the effectiveness of the HOCG concept, the analysis framework, and the visualization system with three real-world applications. Expert feedbacks were also reported, which confirm the usefulness of our technique and recommend several future research directions.

Acknowledgments

This work was supported by NSFC Grants 61772504, U1836117, U1736209, 61572483, 61602122, 71731004, U.S. NSF Grant IIS-1455886, DUE-1833129, and NSF Shanghai No. 16ZR1402200.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput Surveys*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[2] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J Netw. Comput. Appl.*, vol. 60, pp. 19–31, 2016.

[3] P. K. Chan and M. V. Mahoney, "Modeling multiple time series for anomaly detection," in *Proc. 5th IEEE Int. Conf. Data Mining*, 2005, pp. 1–8.

[4] G. G. Hazel, "Multivariate Gaussian MRF for multispectral scene segmentation and anomaly detection," *IEEE Trans. Geoscience Remote Sens.*, vol. 38, no. 3, pp. 1199–1211, May 2000.

[5] C. C. Noble and D. J. Cook, "Graph-based anomaly detection," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 631–636.

[6] X. Miao, K. Liu, Y. He, D. Papadias, Q. Ma, and Y. Liu, "Agnostic diagnosis: Discovering silent failures in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 12, pp. 6067–6075, Dec. 2013.

[7] L. Shi, Q. Liao, Y. He, R. Li, A. Striegel, and Z. Su, "SAVE: Sensor anomaly visualization engine," in *Proc. IEEE Conf. Visual Analytics Sci. Technol.*, 2011, pp. 201–210.

[8] D. Thom, H. Bosch, S. Koch, M. Wörner, and T. Ertl, "Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages," in *Proc. IEEE Pacific Vis. Symp.*, 2012, pp. 41–48.

[9] 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 Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 1773–1782, Dec. 2014.

[10] J. Tao, L. Shi, Z. Zhuang, C. Huang, R. Yu, P. Su, C. Wang, and Y. Chen, "Visual analysis of collective anomalies through high-order correlation graph," in *Proc. IEEE Pacific Vis. Symp.*, 2018, pp. 150–159.

[11] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: A survey," *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1302–1325, 2011.

[12] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Mining Knowl. Discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[13] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, "Anomaly detection in dynamic networks: A survey," *WIREs Comput. Statist.*, vol. 7, no. 3, pp. 223–247, 2015.

[14] 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 Trans. Syst. Man Cybern. Part C (Appl. Rev.)*, vol. 30, no. 1, pp. 84–94, Mar. 2000.

[15] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," in *Proc. SIAM Int. Conf. Data Mining*, 2008, pp. 243–254.

[16] 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*. Cambridge, MA, USA: MIT Press, 2004, ch. 3, pp. 199–218.

[17] E. Eskin, "Anomaly detection over noisy data using learned probability distributions," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 255–262.

[18] L. Akoglu, M. McGlohon, and C. Faloutsos, "Oddball: Spotting anomalies in weighted graphs," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2010, pp. 410–421.

[19] L. Akoglu, R. Chandy, and C. Faloutsos, "Opinion fraud detection in online reviews by network effects," in *Proc. Int. AAAI Conf. Web Soc. Media*, 2013, vol. 13, pp. 2–11.

[20] S. Lin and D. E. Brown, "An outlier-based data association method for linking criminal incidents," *Decision Support Syst.*, vol. 41, no. 3, pp. 604–615, 2006.

[21] W. R. Pires, T. H. de Paula Figueiredo, H. C. Wong, and A. A. F. Loureiro, "Malicious node detection in wireless sensor networks," in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, 2004, Art. no. 24.

[22] F. Liu, X. Cheng, and D. Chen, "Insider attacker detection in wireless sensor networks," in *Proc. 26th IEEE Int. Conf. Comput. Commun.*, 2007, pp. 1937–1945.

[23] I. Onat and A. Miri, "A real-time node-based traffic anomaly detection algorithm for wireless sensor networks," in *Proc. Syst. Commun.*, 2005, pp. 422–427.

[24] R. Khanna, H. Liu, and H.-H. Chen, "Reduced complexity intrusion detection in sensor networks using genetic algorithm," in *Proc. IEEE Int. Conf. Commun.*, 2009, pp. 1–5.

[25] E. C. Ngai, J. Liu, and M. R. Lyu, "An efficient intruder detection algorithm against sinkhole attacks in wireless sensor networks," *Comput. Commun.*, vol. 30, no. 11, pp. 2353–2364, 2007.

[26] F. Fischer, F. Mansmann, D. A. Keim, S. Pietzko, and M. Waldvogel, "Large-scale network monitoring for visual analysis of attacks," in *Proc. Int. Workshop Vis. Comput. Secur.*, 2008, pp. 111–118.

[27] 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 *Proc. ACM Workshop Vis. Data Mining Comput. Secur.*, 2004, pp. 35–44.

[28] Q. Liao, L. Shi, and C. Wang, "Visual analysis of large-scale network anomalies," *IBM J. Res. Develop.*, vol. 57, no. 3/4, pp. 13–1, 2013.

[29] Z. Liao, Y. Yu, and B. Chen, "Anomaly detection in GPS data based on visual analytics," in *Proc. IEEE Symp. Visual Analytics Sci. Technol.*, 2010, pp. 51–58.

[30] R. Chang, M. Ghoniem, R. Kosara, W. Ribarsky, J. Yang, E. Suma, C. Ziemkiewicz, D. Kern, and A. Sudjianto, "Wirevis: Visualization of categorical, time-varying data from financial transactions," in *Proc. IEEE Symp. Visual Analytics Sci. Technol.*, 2007, pp. 155–162.

[31] R. A. Leite, T. Gschwandtner, S. Miksch, S. Kriglstein, M. Pohl, E. Gstrein, and J. Kuntner, "Eva: Visual analytics to identify fraudulent events," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 330–339, Jan. 2018.

[32] W. Didimo, G. Liotta, F. Montecchiani, and P. Palladino, "An advanced network visualization system for financial crime detection," in *Proc. IEEE Pacific Vis. Symp.*, 2011, pp. 203–210.

[33] W. Didimo, L. Giamminonni, G. Liotta, F. Montecchiani, and D. Pagliuca, "A visual analytics system to support tax evasion discovery," *Decision Support Syst.*, vol. 110, pp. 71–83, 2018.

[34] S. Hadlak, H. Schumann, and H.-J. Schulz, "A survey of multi-faceted graph visualization," in *Proc. EuroVis STAR*, 2015, pp. 1–20.

[35] F. Beck, M. Burch, S. Diehl, and D. Weiskopf, "A taxonomy and survey of dynamic graph visualization," *Comput. Graph. Forum*, vol. 36, no. 1, 2017, pp. 133–159.

[36] J. Wang and K. Mueller, "Visual causality analysis made practical," presented at the *IEEE Visual Analytics Sci. Technol.*, Phoenix, AZ, USA, 2017.

[37] D. Reynolds, "Gaussian mixture models," *Encyclopedia Biometrics*, pp. 827–832, 2015.

[38] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Statistical Soc. Series B (methodological)*, vol. 39, pp. 1–38, 1977.

[39] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.

[40] S. J. Roberts, "Novelty detection using extreme value statistics," *IEE Proc.-Vis. Image Signal Process.*, vol. 146, no. 3, pp. 124–129, 1999.

[41] S. Kullback, *Information Theory and Statistics*. North Chelmsford. MA, USA: Courier Corporation, 1997.

[42] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *Proc. 6th Int. Conf. Data Mining*, 2006, pp. 613–622.

[43] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proc. IEEE Symp. Visual Lang.*, 1996, pp. 336–343.

[44] E. R. Gansner and S. North, "An open graph visualization system and its applications to software engineering," *Softw. - Practice Experience*, vol. 30, pp. 1203–1233, 2000.

[45] P. Bak, F. Mansmann, H. Janetzko, and D. Keim, "Spatiotemporal analysis of sensor logs using growth ring maps," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 913–920, Nov./Dec. 2009.

[46] IEEE VAST Challenge 2016. (2016). [Online]. Available: http://vacommunity.org/VAST+Challenge+2016

[47] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comput. Secur.*, vol. 45, pp. 100–123, 2014.

Jia Yan received the PhD degree from the University of Chinese Academy of Sciences, in 2015. He is an assistant professor with TCA/SKLCS, Institute of Software, Chinese Academy of Sciences. His research interests include visualization for security, malware analysis, and software security.

Lei Shi received the BS, MS, and PhD degrees from the Department of Computer Science and Technology, Tsinghua University, in 2003, 2006, and 2008, respectively. He is a professor with the School of Computer Science, Beihang University. Previously, he was a professor with SKLCS, Institute of Software, Chinese Academy of Sciences. His research interests include information visualization, visual analytics, and data mining.

Jun Tao received the PhD degree in computer science from Michigan Technological University, in 2015. He is a postdoctoral researcher with the University of Notre Dame. His major research interests include scientific visualization, especially on applying information theory, optimization techniques, and topological analysis to flow visualization and multivariate data exploration.

Xiaolong Yu received the BS degree in computer science from Fudan University, in 2017. He is now working toward the master's degree in computer science at Fudan University, and a visiting student at the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. His research interests include social computing and data mining.

Zhou Zhuang received the BS degree from the School of Computer Science, Fudan University, in 2018. He is working toward the master's degree in CS Department, Columbia University. He has been a research assistant with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences since 2016. His research interests include information visualization and machine learning.

Congcong Huang received the BS degree from the Department of Computer Science, Sichuan University. She is working toward the graduate degree in the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. Her research interests include data mining, data visualization and visual analytics.

Rulei Yu received the BS degree from the College of Microelectronics, Xidian University, in 2017. He is currently a student in the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. His research interests include information visualization and visual analytics for deep learning.

Purui Su received the PhD degrees from the University of Chinese Academy of Sciences. He is a professor with TCA/SKLCS, Institute of Software, Chinese Academy of Sciences. His research interests include malware detection, program analysis, and software security.

Chaoli Wang received the PhD degree in computer and information science from The Ohio State University, in 2006. He is an associate professor of computer science and engineering with University of Notre Dame. Prior to joining Notre Dame, he was an assistant professor of computer science with Michigan Technological University. His main research interests include scientific visualization, in particular time-varying multivariate data visualization, flow visualization, and information-theoretic algorithms and graph-based techniques for big data analytics.

Yang Chen received the BS and PhD degrees from the Department of Electronic Engineering, Tsinghua University in 2004 and 2009, respectively. He is an associate professor with the School of Computer Science, Fudan University, China. Before joining Fudan, he was a postdoctoral associate with the Department of Computer Science, Duke University and a research associate with the Institute of Computer Science, University of Goettingen, Germany. His research interests include online social networks, internet architecture, and mobile computing.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.