Redirect Your Attention: Interactive Situation Awareness Guiding by Users' Feedback

Submission ID: 3651

ABSTRACT

Situation awareness systems have been developed to help understand the environment and detect anomalous situation, and have been used in many domains like network security. However, due to the complexity and imperfection of real-world data, a reliable system is hard to establish. Human judgment is needed to better define and detect anomalous situation, and to revise the imprecise results. This paper introduces a novel algorithm framework for the analysis process of situation awareness, where human judgment is repeatedly involved to update the risk of environment. The paper also provides two implementations, respectively based on Bayes' theorem and metric learning. A controlled experiment and case studies on real-world datasets were done to verify usability of the framework and to evaluate the performance of the implementations along with a baseline. The study result shows that the framework is helpful and that the metric learning technique outperforms the other two in both time and accuracy.

INTRODUCTION

Situation awareness (SA) is the perception of environmental events with respect to time or/and space as well as the comprehension of their meanings and the projection of their ongoing situations [9]. Its goal is to help decision-makers act effectively and correctly in their work environment when anomalous situations occurred [8, 26, 15]. Due to its importance, various situation awareness systems have been developed in different application domains such as network security [44, 23, 13], military operation [28, 36], and crisis analysis [39, 42].

The aforementioned systems usually calculate to produce a visual environmental summary, through which decision makers can passively observe the ongoing situations. In this process, the correctness of the summary results heavily depend on the analysis model, whose computation results, unfortunately, are often imprecise and even misleading due to the complexity of the real world environment, the uncertainty of the data input, and the unavailability of ground truth [28, 31]. Decision makers still need to, even with the help of a situation awareness system, rely on their own experience and prior knowledge to make a judgment. Thus, a more reliable system that can always automatically and correctly redirect the users' attention to the high risky regions is highly desired.

However, to design such a reliable system is difficult. First, situation awareness system requires an efficient online estimation of the environment and a real time detection of anomalous situations, but designing such an efficient algorithm while ensuring its precision is usually difficult. Second, due to the lack of ground truth, the system should be able to leverage users' immediate feedback to amend the unreliable results in real time and guiding the future analysis. However, to design such an efficient feedback mechanism is nontrivial. Third, in real world applications, users have to rely on imperfect data and imprecise analyses results to detect real anomalies and to prevent an anomalous event from happening. This requires a repetitive situation investigation to eliminate false positives. In this process, informative clues that guide the investigation are desired but are difficult to extract.

To address the above issues, in this paper, we introduce a novel algorithm framework for supporting the analysis process of situation awareness. Our framework provides an efficient and light-weighted situation update mechanism to help revise the imprecise results from arbitrary algorithm, and also provides clues to guide the anomaly investigation process. Based on the framework, we introduced two implementations respectively based on Bayes' theorem and metric learning, whose performances are evaluated based on both quantities evaluation and controlled user study. Our evaluation verified the usefulness of the algorithm framework and the power of the implemented algorithm. In general, this paper has following contributions:

- **Algorithm Framework.** We introduce, to the best of our knowledge, the first interactive situation update algorithm framework for supporting situation awareness. The proposed framework can be used on top of most existing situation awareness algorithms, and provide informative clues to guide the investigation of the focal environment.
- **Situation Updating Algorithm.** We introduced two novel algorithms implemented based on the framework.
- **Comprehensive Evaluation.** We conducted the quantitative evaluation, a controlled user study and a case study with a domain expert to verify the usefulness of the framework and the efficiency of the proposed algorithms.

BACKGROUND

We review techniques that are most relevant to our work including (1) situation awareness systems and algorithms and (2) interactive anomaly detection.

Situation Awareness

Situation awareness, according to Endsley's concept model, is to (1) identify and (2) comprehend both the current and (3) the future situation [9]. He also believed situation awareness was an iterative procedure, in which decision makers tried to perceive and process the environmental data they needed, and the salient cues from the data activated updates in goals in models [10]. Multiple real-world cases have proved situation awareness to be important [29, 18, 10].

Many studies in this area focused on dealing with the complex real world data to improve the accuracy of the analysis. For example, Overby et al. [28] proposed temporal and spatial metrics to identify the failure patterns due to the latency of data delivery over network. Reilly and Kellihan [31] associated each input source with an uncertain measurement to reshape the decision-making model. Kalashnikov extracted spatial information [19] from uncertain, free-text reports. Applebaum et al. [1] formalized the noisy data collected from network sensors with an argumentation framework to help administrators reason about situation awareness. In comparison, our technique improves the analysis accuracy by leveraging decision makers' domain knowledge and experiences instead of relying too much on the analysis algorithm. Moreover, the proposed technique can be easily interacted with many the existing techniques as an additional plugin for enhancing the initial analysis results.

Some studies focused on human's understanding of the situation and the roles they played during the process of seeking awareness in emergency situations. Perceptually, Habibi and Mohammad [14] argued that surveillance or diagnostic based situation awareness information was best communicated in visual forms. To this end, multiple visual systems have been developed to reflect situation awareness by showing "when", "where", and "what" have happened. Livnat et al. [26] argued these were the three necessary attributes for any events, and phrased them as the w^3 premise. Hao et al. [15] visualized network security alerts as ensemble data. Senseplace2 [27] also concerned about place-time-theme information. Following these convention, our work is also designed to reflect "what" happens at "where" at a given time point.

In terms of comprehensibility, a useful situation awareness system is considered to be the one that best meets decision maker's needs. Therefore, Erbacher et al. [11] collected feedback from decision makers to understand their actual needs before building the actual visualization. Frank et al. [13, 2] tried to maximize the awareness by preprocessing the overwhelming environmental data (e.g., compiling, processing, and fusing data). Lan et al. [23] summarized multiple strategies to focus on the events of special interest in the cyberspace, including event simplification, filtering, fusion, and correlation. Despite all these efforts on shaping the initial data into what decision makers need, shaping data according to decision makers' dynamic feedback has yet to be studied.

There are also some studies that focused on the perception of situation based on human produced data. These studies tried to enhance understandings on hazard events with so-called "human censors", or with information collected from social media [42, 39, 27]. However, human produced data are not decision makers' feedback and they are used for generating the initial analysis results instead of refining the analysis, thus are different from our work.

Interactive Anomaly Detection

Anomaly detection is a well-established field aiming at finding patterns in data that deviate from normal behavior [16]. Researchers have improved SA with anomaly detection algorithms [33, 32]. Multiple automated approaches including classification-based [34, 38], clustering-based [3, 5, 12], statistical [17, 35], spectral methods [7, 22], etc., have been employed to reveal such information. Surveys [30, 6, 16] have comprehensively investigated these methods and acknowledged the difficulty of designing a universal anomaly detection algorithm [6]: The notion of "anomaly" is induced by domain-specific factors including nature of the data, availability of labeled data, the evolution of normal behaviors, etc.

In response to the indecisive "anomaly" definition problem, prior studies have also proposed interactive strategies to help practitioners personalize their detection models based on their own decision boundaries. Researchers have proposed multiple forms of feedback. For instance, Konijn and Kowalczyk [20] enabled users to iteratively label outliers until no more interesting outliers can be found. Krasuski and Wasilewski [21] improved the detection of outlying Fire Service's reports by discussing the features and decision boundaries with domain experts. The most common feedback form is to acquire new or updated labels for items. Cui et al. [45] perceived users' decision boundaries on "what are outliers" from examples provided by the users. Riveiro et al. [32] updated Self Organizing Map (SOM) as users label new instances as normal or abnormal instances. Cao et al. presented TargetVue [4], a visualization system that conveys anomalous online user behavior analysis results and collects analyst feedback. GPSvas [24] embeds an active learning process in its visual analysis, through which users can input manual labels for further training. In SODIT, Zhang [43] collected labels to determine localized thresholds for anomaly detection.

While the aforementioned methods have revealed valuable information in the field of intrusion detection, most of the aforementioned algorithms assume the input data is accurate enough to draw a clear decision boundary. In other words, such interactive models rarely consider cases where users fail to correctly identify normal and anomaly points as the input data is not observable, and therefore provide false feedback.

ALGORITHM FRAMEWORK AND IMPLEMENTATION

In this section, we introduce the situation update algorithm framework and two algorithm implementations respectively based on Bayes' Theorem and metric learning.

Online Situation Update Algorithm Framework

The goal of the framework is to guide the data navigation and help decision makers to efficiently locate the anomalous events inside a focal environment that is under investigation. We achieve the goal by designing an efficient situation update mechanism that interactively adopts decision makers'

Algorithm 1: interactive situation update framework

Input: $\mathbf{E} = \{e_1, ..., e_n | e_i \in R^k\}$; $\mathbf{Q} = \{q_1, ..., q_n | q_i \in [0,1]\}$

1 **P** = situation(**R**);
2 **while** *true* **do**
3 **if** *no more anomalous events can be found* **then**
4 break;
5 **end**
6 **L** = sort(**R**, **P**);
7 Users investigate high risky regions $\{r_i | r_i \in \mathbf{L}\}$ and provides feedback f_i for each of the corresponding region to indicate whether or not anomalous events have been found in it;
8 $\mathbf{P}' = \phi$
9 **for** $r_j \in L$ **do**
10 $p'_j = \text{update}(f_i, p_j, q_i)$;
11 $\mathbf{P}' = \mathbf{P}' \cup \{p'_j\}$;
12 **end**
13 $\mathbf{P} = \mathbf{P}'$
14 **end**

judgments during their investigation process. The proposed framework is light-weighted and independent to the underlying situation awareness or anomaly detection techniques. Any algorithm that can produce regional suspect rate in a given environment can be embedded into the framework.

As summarized in Alg. 1, the proposed framework first evaluates the suspect rate p_i of each region r_i in the focal environment E to provide initial clues for the investigation process (line 2). Every time the decision maker inspects a region r_j, a binary feedback f_j indicating whether or not the he/she finds an anomaly in the region is collected (line 7) to update the global situation (line 8 - 11). In this process, the environmental complexity Q is also considered to help tolerate uncertain situations of the environment causing human to make mistakes. The process iteratively performs until no more anomaly can be found in the environment.

In Alg. 1, $E = \{e_1, ..., e_n | e_i \in R^k\}$ indicates the focal environment, in which e_i is k-dimensional feature vector of the i-th region, capturing the region's situation. $Q = \{q_1, ..., q_n | q_i \in [0,1]\}$ is the environmental complexity, in which q_i indicates how difficult to find out the anomaly in the i-th region if the anomaly do exist in the region. $P = \{p_1, ..., p_n | p_i \in [0,1]\}$ is the situation estimation results where p_i indicates likelihood of an anomaly in the i-th region. In this algorithm framework, the "update$(\cdot)$" function, is the key components to be designed and implemented in a concrete algorithm.

Implementation of the Update Function

We introduce two algorithm implementations based on the above framework. The first algorithm employs Bayes' Theorem, which handles users' feedback one at a time. The second algorithm is designed based on metric learning [41], which is able to handle multiple feedback simultaneously. Both of these implementations employ One-class SVM as the situation awareness algorithm due to its unsupervised nature, computational efficiency, and good performance [37].

Implementation I. The Bayes' theorem $P(A|B) = P(B|A)P(A)/P(B)$ calculates the probability of the occurrence of an event A when the occurrence of event B is observed. In our case, let A_i represent "an anomaly exits in the i-th region" and let B_j represent "fail to find an anomaly in the j-th region even it exists", which is positively relevant to q_j, i.e., the environmental complexity of the j-th region. Thus, $P(A_i|B_j)$ indicates the likelihood of an anomaly exists in region i when the user fail to find it in another region j, i.e., p_i defined in Alg. 1. Therefore, based on Bayes' theorem, the situation update function in the framework can be defined as:

$$update(f_i = 0, p_j, q_i) = P(A_i|B_j) = \begin{cases} \frac{P(B_i|A_i)P(A_i)}{P(B_i)} = \frac{p_i q_i}{1-p_i(1-q_i)}, i = j \\ \frac{P(B_i|A_j)P(A_j)}{P(B_i)} = \frac{p_j q_i}{1-p_i(1-q_i)}, i \neq j \end{cases} \quad (1)$$

Note that the above update rules only takes negative feedback as input ($f_i = 0$), i.e., it only updates the overall situation when anomaly has not been found. In addition, it also implicitly assumes that there is only one anomaly to be detected in the investigation space; thus every failing attempt of finding one at an investigated region changes (increases) the conditional probabilities of finding one in the rest of regions. Once the anomaly has been found, i.e, $f_i = 1$, the region will be removed from the investigation space by setting $p_i \equiv 1$, so that users can start to find the next anomaly in the environment.

Implementation II. The second implementation embeds metric learning into One-class SVM [37] to help refine a distance function that determines the anomaly score of each data point via decision makers' feedback.

One-class SVM. Given a dataset, One-class SVM detects anomalies by finding a tight boundary in the feature space that encloses a majority of highly related data points, while points outside the boundary are identified as the anomalies. The distance between the points and the boundary indicates the anomaly score. In particular, the algorithm projects the data points into a latent hyperspace with a higher dimensionality, in which the points can be easily separated by a hyperplane:

$$\sum_{i=1}^{n} w_i K(\mathbf{x}, \mathbf{x_i}) - \rho = 0 \quad (2)$$

where w_i and ρ are parameters learned based on the input data; $K(\mathbf{x}, \mathbf{x_k})$ is a kernel function that estimates the distance between x (a testing point) and all the data points in the input dataset. Among various kernel functions, Gaussian kernel [40] is the most frequently used in non-linear cases. The algorithm finds the tight boundary by ensuring the distance from a point to the hyperplane in the hyperspace to be exactly the same as the distance from the same point to the boundary in the feature space. Here, the distance is defined by the above plane function:

$$dist(\mathbf{x_i}) = \sum_{j=1}^{n} w_j K(\mathbf{x_i}, \mathbf{x_j}) - \rho \quad (3)$$

Intuitively, when the testing data point x_i lies on the hyperplane, the distance is zero; otherwise the distance is a positive

or negative number respectively indicating the point lies inside (normal) or outside (abnormal) the boundary.

Situation Update. We introduced a new kernel function (K_m) based on metric learning to help implement the update function introduced in Alg. 1, which is defined as:

$$K_m(\mathbf{x}, \mathbf{x_i}) = \exp(-\frac{d_M^2(\mathbf{x}, \mathbf{x_i})}{2\sigma^2}) \tag{4}$$

where $d_M(\cdot)$ is a distance metric defined as:

$$d_M(\mathbf{x_i}, \mathbf{x_j}) = \sqrt{(\mathbf{x_i} - \mathbf{x_j})^T M (\mathbf{x_i} - \mathbf{x_j})} \tag{5}$$

The goal is to find an optimal matrix M (denoted as M^*) that best matches the users' judgments in terms of separating the normal and abnormal situations. To this end, we employ the least-square metric learning (LSML) [25], in which M^* is learned based on a set of constrains $\mathscr{C}$ in the form of

$$\mathscr{C} = \{(\mathbf{x_a}, \mathbf{x_b}, \mathbf{x_c}, \mathbf{x_d}) : d_M(\mathbf{x_a}, \mathbf{x_b}) < d_M(\mathbf{x_c}, \mathbf{x_d})\} \tag{6}$$

where $x_{\{a,b\}}$ are the data points in the same class and $x_{\{c,d\}}$ are in different classes. $\mathscr{C}$ ensures any pair of points within the same class to be closer than the ones from different classes, which can be automatically generated based on users' feedback f_i, the last situation estimation p_i , and the environmental complexity q_i. To be simple, we use the function $M(f_i, p_i, q_i)$ to denote the constraint generation and metric learning process as summarized in Alg. 2. Thus, the update function is implemented as follows:

$$\begin{aligned} &update(f_i, p_i, q_i) \\ &= N(\sum_{j=1}^{n} w_j \exp(-\frac{(\mathbf{x_i} - \mathbf{x_j})^T \mathbf{M}(f_i, p_i, q_i)(\mathbf{x_i} - \mathbf{x_j})}{2\sigma^2}) - \rho) \end{aligned} \tag{7}$$

where $N(\cdot)$ normalizes the results into $[0, 1]$.

Algorithm 2: M(f_i,p_i,q_i)

Input: f_i, p_i, q_i , $\mathscr{C} = \varnothing$;
$\mathscr{A} = \{r_j | p_j \in \mathbf{P}, p_j > 0.5, j = 1, 2, ..., n\}$;
$\mathscr{N} = \{r_j | p_j \in \mathbf{P}, p_j \leq 0.5, j = 1, 2, ..., n\}$;
for *j=1 to NumOfConstraints* **do**
  $a, b, c \leftarrow sample(\mathscr{N})$, $d \leftarrow sample(\mathscr{A})$;
  $\mathscr{C} \leftarrow \mathscr{C} \bigcup \{(a, b, c, d)\}$
end
if f_i= *True* **then**
  $\mathscr{A} = \mathscr{A} \bigcup r_i$;
  for *j = 1 to n* **do**
    $a \leftarrow sample(\mathscr{A})$, $b \leftarrow sample(\mathscr{N})$;
    $\mathscr{C} \leftarrow \mathscr{C} \bigcup \{(r_i, a, r_i, b)\}$, with probability $1 - q_i$;
  end
else
  $\mathscr{N} = \mathscr{N} \bigcup r_i$;
  for *j = 1 to n* **do**
    $a \leftarrow sample(\mathscr{N})$, $b \leftarrow sample(\mathscr{A})$;
    $\mathscr{C} \leftarrow \mathscr{C} \bigcup \{(r_i, a, r_i, b)\}$, with probability $1 - q_i$;
  end
end
$\mathbf{M}^* \leftarrow LSML(\mathscr{C})$;

EXPERIMENT DESIGN

We conducted a within-subjects study with the goal of evaluating the effectiveness of our framework and the two algorithm implementations. We also raised a baseline method, i.e., situation awareness without any situation update (denoted as *No-Update*) for comparison. In this section, we describe the details of the experiment design.

Task and Data

This study focuses on evaluating the algorithms' capability in terms of supporting users to identify anomalies in a focal spatial environment. We design a task that simulates the real process of situation awareness, in which a small portion of anomalies need to be found among a large collection of data points distributed in a two-dimensional area. Human involved in the task are required to find the anomalies and mark the regions containing them. In our experiment, the user task is:

Locate the regions that contain anomalous events (i.e., the data points that have different feature values compared with that of other points) in a 2D spatial environment.

A multidimensional testing dataset to simulate the real-world data and to support the study task. Thousands of 6-dimensional data points are generated from the normal distribution with mean $\mu = (1, 1, 1, 1, 1, 1)^T$ and co-variance matrix $\Sigma = \text{diag}\{1, 1, 1, 1, 1, 1\}$. The data points that are 3σ far away from the mean μ were labeled as anomalies. The data points are randomly placed into a two dimensional spatial environment mixed with a fixed number of anomalies.

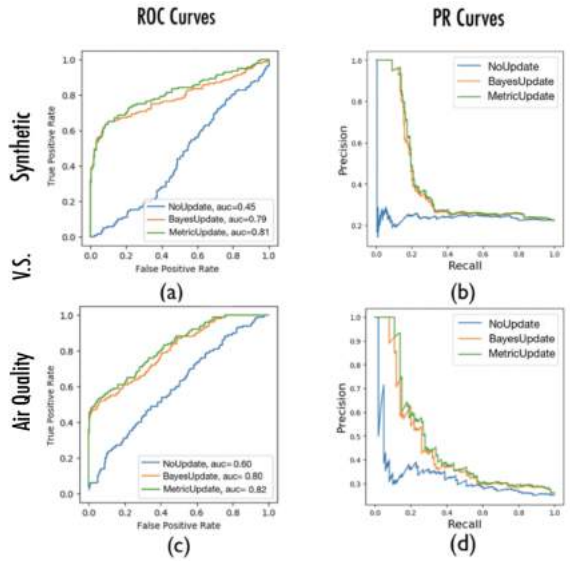

Figure 1. The evaluation curves of the three methods. Figure (a) and (b) were drawn from the synthetic test data. Figure (c) and (d) were drawn from the China air quality data on 1/15/2017. Figure (a) and (c) display the ROC curves, while Figure (b) and (d) display the PR curves. An ideal ROC curve should be close to the upper left, and an ideal PR curve should be close to the upper right.

Study Hypotheses

A preliminary quantitative evaluation on the three methods based on both the synthetic and real world data is performed to help make a reasonable hypothesis. We let the update methods always pick the most risky region and make an update of the current situation. The evaluation results in Fig. 1 suggested that *Bayes-Update* and *Metric-Update* have similar performance and are better than *No-Update*.

Therefore, we hypothesize that when operating by users, the update framework can enhance users' performance and the *Metric-Update* is the best in terms of supporting situation awareness. Specifically, we make the following hypotheses:

H1: The situation update framework is able to enhance users' performance in terms of detecting anomalous events in a focal spatial environment.

H2: With the help of *Metric-Update*, users are able to detect suspicious regions more correctly when compared to the cases of using *No/Bayes-Update*. This is due to the fact that *Metric-Update* always recalculates the situation while the other two do not.

H3: With the help of *Metric-Update*, users are able to complete the task with less time, when compared to these cases of using *No/Bayes-Update*. This is due to the fact that *Metric-Update* accepts multiple feedback but others are not.

Experiment Platform

We visualize the investigation environment via the *global view* (Fig. 2(a)) of the interface by a set of inspection grids. The initial situation awareness results and the situation update results are represented by grid colors. A grid with deeper blue indicates the region has a higher probability of containing an anomaly, i.e., a larger p value.

The *detail view* (Fig. 2(b)), reveals data points inside a focal region. The data points are shown as blue dots with the opaque ones indicating anomaly and the transparent ones indicating the ordinary points. Here, color opacity is chosen to differentiate anomalies so that the environmental complexity q can be properly represented by the overlapping rate of the dots. Intuitively, a set of highly overlapped transparent dots will be a great distraction and will result in difficulties in finding the opaque anomalies. We control the environmental complexity

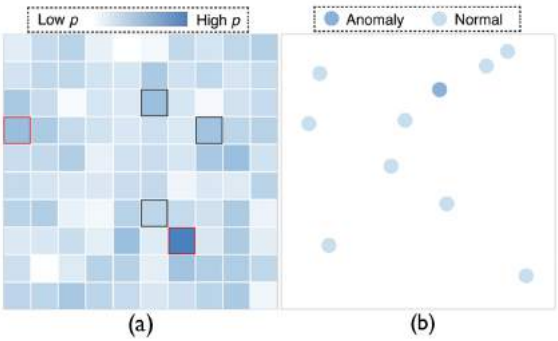

Figure 2. The interface for the user study, with (a) the global view: squared grid map simulating the spatial environment, and (b) the detail view: data dots simulating events in a region.

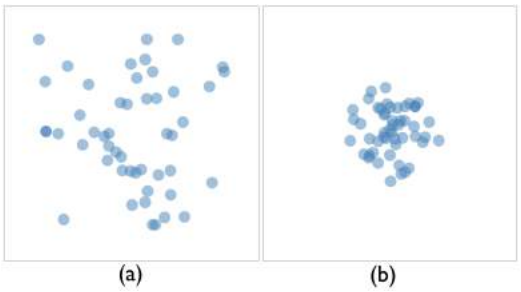

Figure 3. Two detail views under different levels of environmental complexity: (a) $q = 0.1$, and (b) $q = 0.8$.

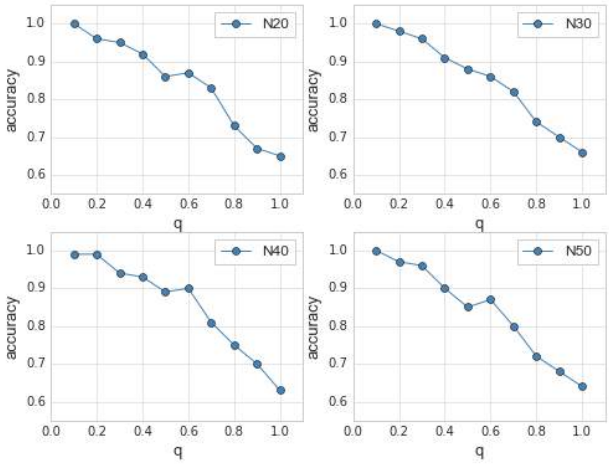

Figure 4. The accuracy-q line charts under different numbers of data points per grid. The charts revealed the correlation between users' performance and environmental complexity under different numbers of events, among which $N30$ displayed the strongest linearity.

via the distribution of the data $N(0, 1-q)$, where $N(\cdot)$ is the normal distribution. As shown in Fig. 3, a larger q value results in a more concentrated layout of dots. In the formal study, the environmental complexity of each grid in each testing trial is randomly determined.

A pilot study involving 20 participants was performed to check if the environmental complexity determined by $N(0, 1-q)$ matches with the users' intuition. To this end, we generated different collections of testing data based on $N(0, 1-q)$ by varying $q \in \{0.1, 0.2, \cdots, 1\}$ and test users' performance (i.e., accuracy). A linear regression model is fitted to the study results, which suggests a significant linearity between the accuracy and q ($t = 21.03, p < 0.01$), i.e., our design of environmental complexity fitted well to the users' intuition. We also investigated the "accuracy-q" correlation in views containing different number of data points. The results, as shown in Fig. 4, suggest the best linearity exists, when there are 30 points in the view. Thus, we choose 30 as a constant number of data points in each grid in the formal study. During the study, a user can hover on a grid to check its containing data points in the *detail view* and can click to mark the grid if he/she finds an anomaly. The marked grid will be highlighted by a red border, which can be canceled by a double click. The grids that have been explored are also marked with thick black borders to avoid duplicated investigation.

Task Conditions

The goal of the study is to evaluate the algorithms' capability in supporting situation awareness under different conditions, which is determined by the scale of the investigation space (determined by the number of grids g^2) and the total number of anomalies to be found (denoted as n_a).

Another pilot study with 20 university students was conducted to determine the best setting of the above two condition parameters so that users' performances can be clearly differentiated. The participants were required to finish the aforementioned study task based on *Metric-Update*.

We tested a range of possible g and n_a and found the accuracy dropped and the task completion time increased significantly (both $p < .05$ according to t-test) when increasing g from 10 to 15 or increasing n_a from 3 to 7, thus have been adopted in the formal study. In this way, each participant was asked to use 3 algorithms to complete the task under four different conditions by enumerating the number of grids and number of anomalies. Each condition was repeated for 3 times to reduce random noise, thus resulting in 36 testing trials per participant as summarized in Table 1.

Table 1. The design of the user study task

	3	Algorithms
×	2	Number of grids (*small (10^2), large (15^2)*)
×	2	Number of anomalies (*small (3), large (7)*)
×	3	Repetitions
	36	Trials

Task Performance Measures

To evaluate users' performance of detecting anomalous events via different algorithms, we quantify the effectiveness by accuracy and completion time of each task.

There are two common metrics of accuracy: the precision and recall rate. The precision indicates the proportion of anomalous grids among those checked by a user, and the recall rate measures the proportion of successfully found anomalies among all the anomalous grids. The pilot study shows that the precision rate is not a good choice to reflect the accuracy as users rarely make mistakes in identifying an anomalous grid. In comparison, the recall rate are more differentiable, thus has been chosen as a performance measure in the formal study.

The completion time measures the duration starting at the time when the dataset is loaded, and ending at the time when users click the "next" button to begin the next trial. It is automatically recorded by our system, and the duration includes both the data inspection time and response time. Users can click the "pause" button when a break is necessary, during which the time recorder is held up.

USER STUDY

We recruited 18 users (9 females) to participate in our study to test our framework and compare three update methods: *No-Update*, *Bayes-Update*, and *Metric-Update*. The users were university students majoring in computer science, math, and design, who are aging from 19 to 26 (mean: 22.06, sd: 1.95). In this section, we describe the user study procedure and results, then discuss the findings and potential challenges.

Procedure

Before the formal study, we conducted a 20-minute tutorial, during which we introduced the concept of situation awareness and its application in real-world scenarios. We also introduced the proposed situation updating algorithms and their distinct characteristics, followed by a detailed description of the experiment platform and the operation methods. In the follow-up practice session, the participants were required to test the study system with all three update methods by their own based on a sample dataset containing all the testing conditions. They were encouraged to ask questions and all their questions were carefully answered to make sure they all fully understand the task and the corresponding procedures.

After fully practiced, each participant was required to finish all the testing trials. The completion time and the answers were recorded automatically for later analysis. We used random data with the same distribution under the same condition across different algorithms to guarantee a fair comparison and minimize the learning effects. We also counterbalanced the order of comparing techniques. Finally, a post-study questionnaire session was conducted.

The study was performed on a 13.3-inch laptop computer with a display resolution of 2560×1600 pixels. The experiment was conducted within a 2000×1200 pixel window with a white background. The size of each grid was adjusted automatically according to g, i.e., the number of grids. The whole study took approximately one hour.

Results

The Effect of Study Variables

We investigated how the two study variables (number of the grids and number of the anomalous events) affecting the task performance through a series of analysis. To this end, we divided the study results into four parts based on different task conditions (i.e., small/large numbers of grids, small/large numbers of anomalous events). In each part, RM-ANOVA was introduced to make the comparison between the two levels of a variable. The normality and homogeneity of the data were respectively tested by the Shapiro-Wilk test and the Lenvene test. The inverse degree of freedom was used to transform the data into a normal distribution.

When the number of grids was small (G10), RM-ANOVA showed that the number of anomalous events significantly affected users' performance ($F(1,17) = 15.81, p < 0.05$) in terms of using all three methods: a larger anomaly number decreased the recall rate (Fig. 5(a)). The analysis results showed no significant change in task completion time when the number of anomalies changed. Compared to *No/Bayes-Update*, *Metric-Update* was the least sensitive to the change of the number of anomalous events in both time and accuracy.

When the number of grids was large (G15), users' recall rate were significantly lower in datasets with more anomalous events ($F(1,17) = 12.65, p < 0.05$) across all methods. The completion time was also significantly influenced by anomaly

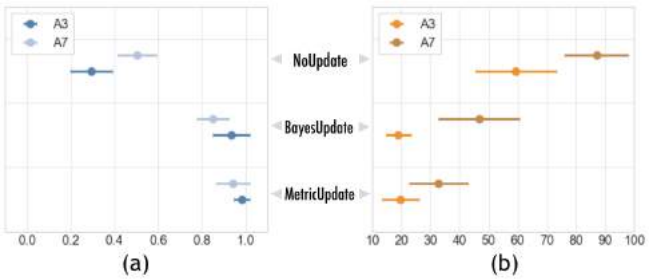

Figure 5. The effect of number of anomalous events when number of grids is 10×10 (G10): (a) recall rate and (b) average completion time.

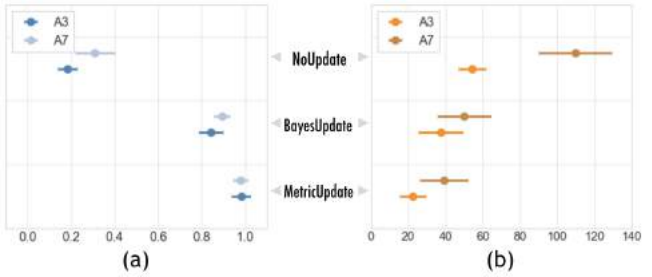

Figure 6. The effect of number of anomalous events when number of grids is 15×15 (G15): (a) recall rate and (b) average completion time.

number ($F(1,17) = 68.23, p < 0.01$). Fig. 6 showed that *Metric-Update* and *Bayes-Update* were less influenced in both recall rate and time.

When the number of anomalies was small (A3), *Metric-Update* was less sensitive to the number of the grids in terms of recall rate (Fig. 7(a)). The recall rate of both *No-Update* and *Bayes-Update* were significantly decreased respectively with ($F(1,17) = 19.95, p < 0.05$) and ($F(1,17) = 10.17, p < 0.05$) when the grid number increased. However, *Metric-Update* is not significantly influenced. In terms of task completion time (Fig. 7(b)), only *Bayes-Update* is significantly influenced ($F(1,17) = 63.79, p < 0.01$).

When the number of anomalies was large (A7), the change of grid number did not have conspicuous impact on *Metric/Bayes-Update* in terms of both recall rate and task completion time, whereas *No-Update* was significantly influenced in terms of both recall ($F(1,17) = 23.28, p < 0.01$) and completion time ($F(1,17) = 46.36, p < 0.01$), as shown in Fig. 8.

Comparison of Algorithms

We compared *No/Bayes/Metric-Update* under different conditions to evaluate their performances in supporting situation awareness and finding anomalous events in a focal environment. The mean and stand error of recall rate and completion time is presented in Fig. 9 and Fig. 10. RM-ANOVA was used again to quantify the differences, and Bonferroni correction was used to conduct the pairwise comparisons. The null hypothesis is that there is no difference in means between these situation updating algorithms in terms of both recall rate and task completion time. We verified normality and homogeneity and transformed unsatisfying data or degree of freedom if assumptions were violated.

T1 (G10-A3)*: finding 3 anomalies in 10×10 grids.* The tests of within-subjects effect showed that these three methods are significantly different in terms of both task completion time ($F(2,34) = 79.16, p < 0.01$) and recall rate ($F(2,34) = 6.05, p < 0.05$). Compared to *No-Update*, *Metric-Update*'s performance is significantly better with respect to time($F(2,34) = 51.90, p < 0.01$) and recall rate($F(2,34) = 13.31$, p<0.01). Compared to *Bayes-Update*, *Metric-Update* is significantly better in terms of completion time ($F(2,34) = 457.49, p < 0.01$).

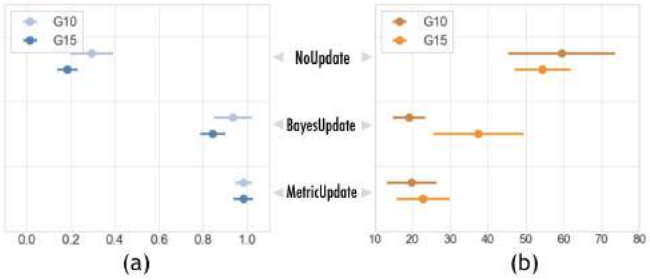

Figure 7. The effect of number of grids when number of anomalous events is 3 (A3): (a) recall rate and (b) average completion time.

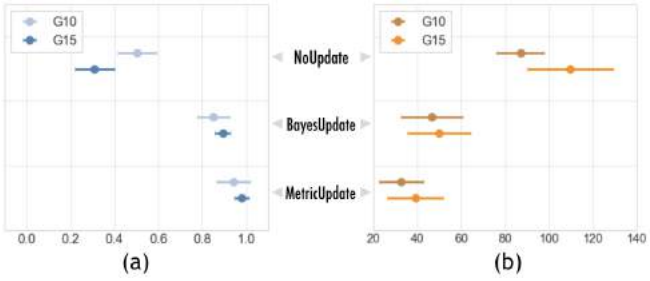

Figure 8. The effect of number of grids when number of anomalous events is 7 (A7): (a) recall rate and (b) average completion time.

T2 (G10-A7)*: finding 7 anomalous events in 10×10 grids.* The tests of within-subjects effect showed that these three methods are significantly different in terms of both task completion time ($F(2,34) = 206.83, p < 0.01$) and recall rate ($F(2,34) = 34.11, p < 0.01$). Compared to *No-Update*, *Metric-Update*'s performance is significantly better with respect to time ($F(2,34) = 94.85, p < 0.01$) and recall rate ($F(2,34) = 22.53, p < 0.01$). Compared to *Bayes-Update*, *Metric-Update* is also significantly better in terms of both completion time ($F(2,34) = 308.36, p < 0.01$) and recall rate ($F(2,34) = 27.86, p < 0.01$).

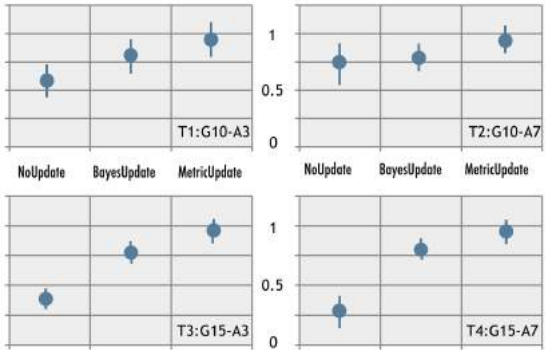

Figure 9. Comparing the recall rate of the three algorithm implementations under different conditions.

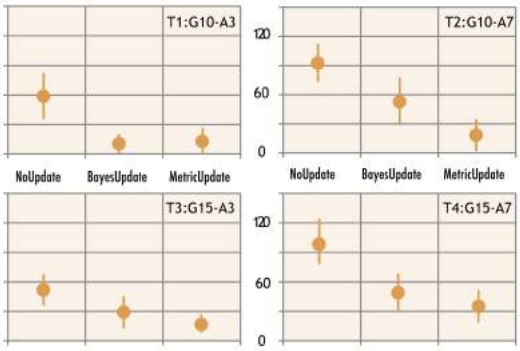

Figure 10. Comparing the average task-completion time of the three algorithm implementations under different conditions.

T3 (G15-A3)*: finding 3 anomalous events in 15×15 grids.* The tests of within-subjects effect showed that these three methods are significantly different in terms of both task completion time ($F(2,34) = 98.32, p < 0.01$) and recall rate ($F(2,34) = 15.09, p < 0.01$). Compared to *No-Update*, *Metric-Update*'s performance is significantly better with respect to time ($F(2,34) = 103.55, p < 0.01$) and recall rate ($F(2,34) = 15.88, p < 0.01$). However, there is no significant difference between *Metric/Bayes-Update*.

T4 (G15-A7)*: finding 7 anomalous events in 15×15 grids.* The tests of within-subjects effect showed that these three methods are significantly different in terms of both task completion time ($F(2,34) = 147.21, p < 0.01$) and recall rate ($F(2,34) = 21.36, p < 0.01$). Compared to *No-Update*, *Metric-Update* is significantly better by both completion time ($F(2,34) = 73.31, p < 0.01$) and recall rate ($F(2,34) = 19.36, p < 0.01$). When compared to *Bayes-Update*, *Metric-Update* is also significantly better by both completion time($F(2,34) = 365.26, p < 0.01$) and recall rate($F(2,34) = 14.02, p < 0.01$).

From the statistics above, we rejected the null hypotheses. We verified that (H1): the update algorithms could better enhance users' performance than *No-Update*; (H2): *Metric-Update* gave the most precise estimation of the situation and (H3): *Metric-Update* took the least time to complete the task.

Post-Study Questionnaire

Users were asked to complete a questionnaire with 12 questions designed to estimate the ease of use (Fig. 11(a)) and usefulness (Fig. 11(b)) of the proposed techniques. The results suggested *Metric-Update* was favored by most participants in any testing conditions.

Discussion

In this section, we discuss findings about the two algorithm implementations from our user study.

Why did Metric-Update outperform Bayes-Update method?

According to the statistics, *Metric-Update* outperformed *Bayes-Update* on both the recall rate and the average completion time. The reason is that *Metric-Update* involves both human factors and the original data for updating. The distance measurement changes with the iterations to recompute the decision boundary and the anomaly score for each grid. That is, if a user determines that a grid contains an anomaly, the anomaly scores of other grids similar to the checked grid increase. By contrast, *Bayes-Update* takes human judgment as an observation that changes the prior anomaly scores, but it does not combine human factors with data. Equation 1 also shows that *Bayes-Update* changes the anomaly scores of unchecked grids at the same rate, which makes little change to human perception. As some users reported, *Bayes-Update* always set the colors of all grids lighter together, which did not guide them to find latent anomalous events.

Why was Metric-Update easy to use?

Metric-Update was rated as the easiest to use among the three methods (Fig. 11(b)). This is mostly due to the simultaneous update strategy *Metric-Update* adopts. It was commonly stated among users that they felt tired in updating after each check when using *Bayes-Update* or randomly guessing when using *No-Update*, and that they liked to update whenever they want, with several decisions at a time when using *Metric-Update*.

When should Bayes-Update be used?

According to the study results, *Metric-Update* was the most preferred method across all conditions (Fig. 11). However, *Bayes-Update* gained some votes in small-scale environment and small number of anomalous events (G10 and A3). Those who voted for *Bayes-Update* said that *Bayes-Update* responded to the clicks immediately, which made the operation very convenient and efficient. When there were only 3 anomalies to be found, the iteration times of the two algorithms were about the same. The reason for the quick response of *Bayes-Update* is that it has no need to recompute the data for the new anomaly score.

The RM-ANOVA results in T1 and T3 also show that *Bayes-Update* can be comparable to *Metric-Update* in recall rate. Therefore, *Bayes-Update* is recommended in some simple cases. That is, when the spatial environment is small-scaled, or the probability for an anomalous event to break is low.

When should Metric-Update be used?

In our study, *Metric-Update* performed better on the recall rate and completion time across all the conditions than *Bayes-Update*. In terms of robustness, it was also less sensitive to the variation of the scale of the investigation space and the number of anomalies. Most participants preferred to use

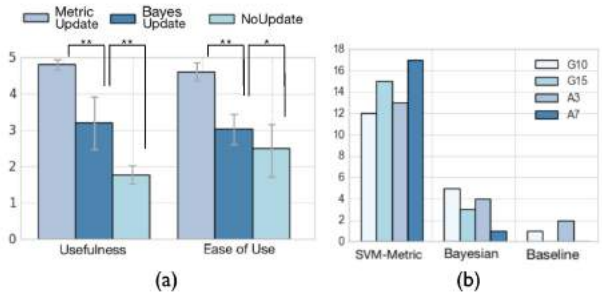

Figure 11. Questionnaire statistics: (a)The usefulness and ease of use among algorithms rated by participants. The significance was indicated by **: p<0.01 and *: p<0.05. (b) Popularity among the participants for each method, under different conditions (i.e., small/large numbers of anomalous grids, small/large numbers of anomalous events).

Metric-Update under any circumstances. Therefore, *Metric-Update* can be applied in most situation awareness scenarios, especially in the complex and large-scale cases.

What is the challenge in practical use of the system?
Our framework provides an indicator informing users of the total number of anomalous events. However, in real-world situations, the number of anomalous events is unknown. It is difficult for users to be aware of the progress of the task, that is, whether all the anomalous events are found.

Another possible challenge is that when the system is used in real-world scenarios, the environmental complexity q is not easy to measure. Also, the visual encoding of the environmental complexity varies according to the scenarios, and needs to be carefully designed.

DOMAIN EXPERT INTERVIEW

We conducted an interview with an expert, a professor of environmental engineering, to further evaluate the proposed algorithm framework and the corresponding implementations.

Procedure

The interview was conducted in the form of a short-term case study, during which the expert was asked to identify anomalies based on the hints provided by situation awareness and update algorithms. The interview started with a tutorial with the fundamental concepts about situation awareness and update. Once the expert was proficient with the prototype system, he was asked to identify regions in which air is polluted. During this procedure, we conducted a semi-structured interview that included questions about different algorithms, overall usefulness, ease of use, and general pros and cons of the approach taken. The interview lasted about one hour and was recorded.

Prototype System

To support the interview, a prototype situation awareness system (Fig. 12) was implemented based on *Metric-Update* to monitor air quality based on the data collected from the "National Urban Air Quality Real-time Release Platform"[1] in China. This data source provides the pollutant concentration data collected from more than 1,400 urban air quality monitoring stations located in 367 cities in China. Six common pollutants were measured by each station, including: nitrogen dioxide (NO_2), sulfur dioxide (SO_2), ozone (O_3), carbon monoxide (CO), and particulate matter (PM2.5 and PM10). The value of each pollutant is standardized based on *IAQI* (the individual air quality index)[2] to facilitate a comparison.

The primary view (Fig. 12(a)) of the prototype system illustrates a map of China, i.e., the environment to be inspected. The map is further segmented into grids with each grid indicates a region whose color represents the current situation, i.e., the likelihood of the air in the region been polluted. Empty grids indicate no data have been collected from those regions.

The air quality of a region is captured by a six-dimensional feature vector that indicates the averaged *IAQI* values of the six pollutants observed by the air quality monitoring stations inside the region. Based on this feature vector, One-Class SVM calculates the current situation p, i.e., the likelihood of a region been polluted by comparing the features of the focal region to the features of other regions and their features in history. The environmental complexity q of each region is negatively proportional to the number of air quality monitoring stations inside the region. Thus, based on p and q, *Metric-Update* is applied.

When hovering on a region in the primary view, its containing air quality monitoring stations will be shown as star glyphs in the detail view (Fig. 12(b)) with each axis in the star indicates a pollutant. The standard baseline (the background yellow circle) and the observed data (the foreground red shape) are visualized simultaneously in each glyph to facilitate a fast comparison. Thus, users can quickly find an anomaly when the red region exceeds the yellow boundary.

Expert Interview

We used the air quality data collected on Jan 15th, 2017 for the case study and expert interview. The expert performed two case studies by exploring and checking the air quality situations at two different areas in China to help estimate the functions of the proposed algorithm.

Case I: The first glance at the map as shown in Fig. 12(a), the expert immediately found the area of Henan Province is potentially polluted as all the grids in this area were in dark red. He clicked to check one of the region with the highest risk. From the star glyphs of the air monitoring station in the region shown in the detail view (Fig. 12(b)), he found although the values of pollutants are high but they are still normal. He believed this was not an anomalous situation, thus marked the region as normal and committed to the system to make a situation update. After the updating, some of the surrounding regions also turned into blue. After checking these regions, he found these regions had a similar situation, i.e., nothing else is abnormal but the relatively value are a little big high. He said, "it is convenient ... and smart ... as it can update regions with a similar situation based on my choice". The situation update also made some of surrounding regions into an even darker red. These regions were suspicious and thus checked by the expert. From the star glyph, he immediately found in these regions, the level of both PM2.5 and PM10 are extremely high, which is indeed abnormal compared with historical situations. These regions were thus marked as abnormal and the situation is updated accordingly.

Case II: The expert also noticed some regions in Xinjiang Province was also initially highlighted in dark red by our situation awareness system (Fig. 12(c)), thus worthed an inspection. After reading the pollutant values shown in the star glyph, he believed these are indeed abnormal events as these areas are seldom been polluted. He marked the regions as abnormal to update the current situation. This time, he found the situations in the surrounding regions were no longer dramatically changed. This is because his decision was made by only reviewing the situations from two air quality monitoring stations, which had a lower confidence. After knowing this reason, the expert was impressed by our technique and said "this is a smart

[1] http://pm25.in.
[2] https://en.wikipedia.org/wiki/Air_quality_index

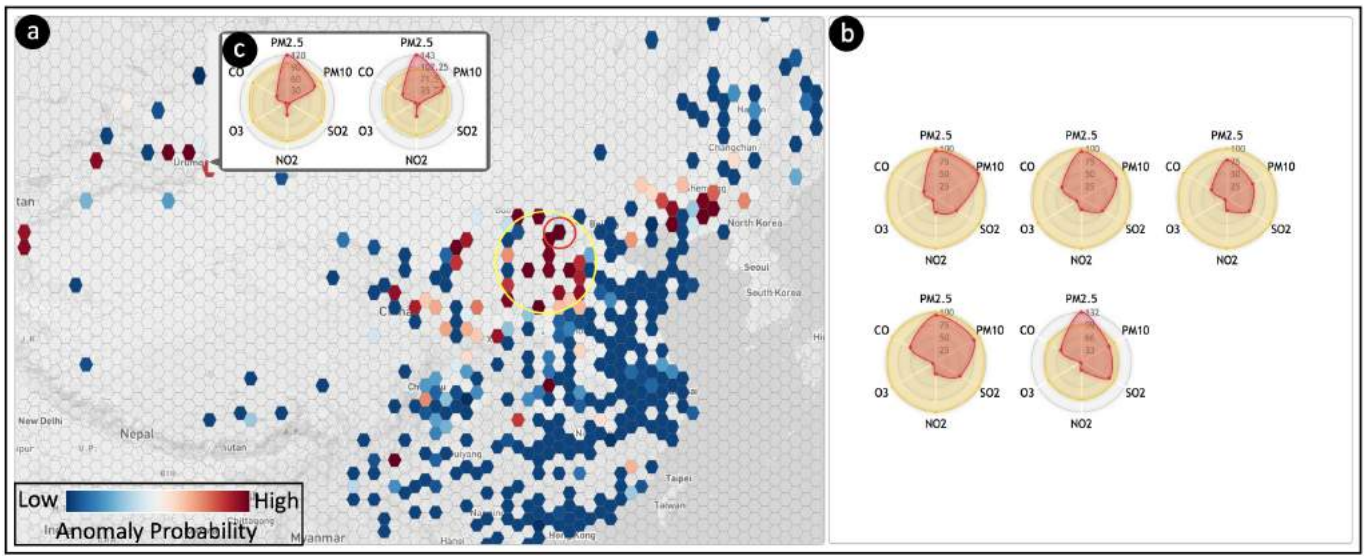

Figure 12. The prototype system developed for case study, with (a) hexagonal grids composing a map of China, (b) the detail view of the focal hexagonal grid with high air pollution risk. The red area on each radar chart displays the concentration of each substance, while the yellow area displays the standard concentration, and (c) the detail view of a remote area with high air pollution risk, where the number of stations is small.

design" and "I can imagine this [situation update] mechanism to be useful in many other applications".

Many comments from the expert were collected during the case study. In general the expert acknowledged the usefulness of our situation update techniques. During the case study, the expert said "this tool is very useful" several times during the exploration of the polluted regions. The expert also felt the situation update was a "smart idea" and "much better than the static heatmap" frequently used in traditional situation awareness systems. He also believed the proposed situation update mechanism provided "useful context information" and "can correctly guide the data exploration process". The expert especially impressed by the real-time situation calculating and updating, said "... even it is only a prototype, I can see the potential of using the system in real-world cases".

DISCUSSIONS

In this section, we discuss the limitations of our work revealed in the expert interview. We also make reflection on what we did in this study and the value of our work in terms of guiding the design of future situation awareness systems.

Limitations. Although the expert generally praised the usefulness of the proposed technique, he also raised several limitations of it. First, the system did not utilize the geographical feature of the grids, which could provide information for situation awareness. Second, *Metric-Update* methods sometimes produce sudden changes which could be difficult for a user to follow ("sometimes I can't follow the situation after a sudden change of the colors"). A more smooth updating method is desired. Third, the current system didn't take the temporal information into consideration. In particular, the second issue is key challenge of *Metric-Update*. A sudden change is majorly caused by the re-computation of the decision boundary for One-Class SVM. Smoothing the update result requires a more precise study on the metrics used for calculating SVM, which will be our future work.

Implications of Future Design. We proposed an online situation update framework for situation awareness that used human judgment to refine the estimation of an spatial environment. Our algorithm framework and the corresponding implementations accomplished three achievements:

- **A1** *Dynamic-Feedback*: The framework accepted feedback from users to rectify the situation awareness model in real time in a dynamic procedure.
- **A2** *Parallel-Updating*: Human can feedback with several judgments simultaneously, which makes the update mechanism more more efficient and precise.
- **A3** *Fault-Tolerant*: People tend to make mistakes in complex environment. By taking the environment complexity into consideration, the proposed technique tolerates fault decision made in complex situations by reducing its effect during the update process.

We believe these achievements fit better to the complicated real-world scenarios and provide a new standard for designing the future situation awareness systems.

CONCLUSION

In this paper, we introduced a novel situation update algorithm framework for the analysis process of situation awareness and two updating algorithm implementations based on Bayes and metric learning, respectively. We conducted the quantitative evaluation and a controlled user study to examine their performance in interactively detecting anomalies under various conditions, compared with the baseline technique. The study results verified that the proposed interactive situation update framework was useful and the *Metric-Update* algorithm significantly outperformed the compared algorithms. A case study with a domain expert further verified the effectiveness and usefulness of the proposed technique. Future work include refining the algorithm design to provide a smoother update results and apply our technique to more real world applications.

REFERENCES

1. Andy Applebaum, Karl Levitt, Zimi Li, Simon Parsons, Jeff Rowe, and Elizabeth Sklar. 2015. Cyber reasoning with argumentation: Abstracting from incomplete and contradictory evidence. In *IEEE Military Communications Conference*. IEEE, 623–628.
2. Tim Bass. 2000. Intrusion detection systems and multisensor data fusion. *Commun. ACM* 43, 4 (2000), 99–105.
3. Suratna Budalakoti, Ashok N Srivastava, Ram Akella, and Eugene Turkov. 2006. Anomaly detection in large sets of high-dimensional symbol sequences. (2006).
4. Nan Cao, Conglei Shi, Sabrina Lin, Jie Lu, Yu-Ru Lin, and Ching-Yung Lin. 2016. Targetvue: Visual analysis of anomalous user behaviors in online communication systems. *IEEE transactions on visualization and computer graphics* 22, 1 (2016), 280–289.
5. Philip K Chan, Matthew V Mahoney, and Muhammad H Arshad. 2003. *A machine learning approach to anomaly detection*. Technical Report.
6. Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys* 41, 3 (2009), 15.
7. Vasilis Chatzigiannakis, Symeon Papavassiliou, Mary Grammatikou, and B Maglaris. 2006. Hierarchical anomaly detection in distributed large-scale sensor networks. In *IEEE Symposium on Computers and Communications*. IEEE, 761–767.
8. Anita D'Amico and Michael Kocka. 2005. Information assurance visualizations for specific stages of situational awareness and intended uses: lessons learned. In *IEEE Workshop on Visualization for Computer Security*. IEEE, 107–112.
9. Mica R Endsley. 1995. Toward a theory of situation awareness in dynamic systems. *Human factors* 37, 1 (1995), 32–64.
10. Mica R Endsley. 2015. Situation awareness misconceptions and misunderstandings. *Journal of Cognitive Engineering and Decision Making* 9, 1 (2015), 4–32.
11. Robert F Erbacher, Deborah A Frincke, Pak Chung Wong, Sarah J Moody, and Glenn A Fink. 2010. Cognitive task analysis of network analysts and managers for network situational awareness. *VDA* 7530 (2010).
12. Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. 2002. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. *Applications of data mining in computer security* 6 (2002), 77–102.
13. Ulrik Franke and Joel Brynielsson. 2014. Cyber situational awareness–a systematic review of the literature. *Computers & Security* 46 (2014), 18–31.
14. Mohammad S Habibi. 2011. Adaptive Situation Awareness Using Visual Analytics. In *First International Symposium on Network Cloud Computing and Applications*. IEEE, 79–82.
15. Lihua Hao, Christopher G Healey, and Steve E Hutchinson. 2015. Ensemble visualization for cyber situation awareness of network security data. In *IEEE Symposium on Visualization for Cyber Security*. IEEE, 1–8.
16. Victoria Hodge and Jim Austin. 2004. A survey of outlier detection methodologies. *Artificial intelligence review* 22, 2 (2004), 85–126.
17. Zhaojie Ju and Honghai Liu. 2012. Fuzzy gaussian mixture models. *Pattern Recognition* 45, 3 (2012), 1146–1158.
18. David B Kaber and Mica R Endsley. 2004. The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. *Theoretical Issues in Ergonomics Science* 5, 2 (2004), 113–153.
19. Dmitri V Kalashnikov, Yiming Ma, Sharad Mehrotra, Ramaswamy Hariharan, and Carter Butts. 2006. Modeling and querying uncertain spatial information for situational awareness applications. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*. ACM, 131–138.
20. Rob M Konijn and Wojtek Kowalczyk. 2010. An interactive approach to outlier detection. In *International Conference on Rough Sets and Knowledge Technology*. Springer, 379–385.
21. Adam Krasuski and Piotr Wasilewski. 2013. Outlier detection by interaction with domain experts. *Fundamenta Informaticae* 127, 1-4 (2013), 529–544.
22. Anukool Lakhina, Mark Crovella, and Christophe Diot. 2005. Mining anomalies using traffic feature distributions. In *ACM SIGCOMM Computer Communication Review*, Vol. 35. ACM, 217–228.
23. Fang Lan, Wang Chunlei, and Ma Guoqing. 2010. A framework for network security situation awareness based on knowledge discovery. In *International conference on Computer Engineering and Technology*, Vol. 1. IEEE, 1–226.
24. Zicheng Liao, Yizhou Yu, and Baoquan Chen. 2010. Anomaly detection in GPS data based on visual analytics. In *IEEE Symposium on Visual Analytics Science and Technology*. IEEE, 51–58.
25. Eric Yi Liu, Zhishan Guo, Xiang Zhang, Vladimir Jojic, and Wei Wang. 2012. Metric learning from relative comparisons by minimizing squared residual. In *IEEE 12th International Conference on Data Mining*. IEEE, 978–983.
26. Yarden Livnat, James Agutter, Shaun Moon, and Stefano Foresti. 2005. Visual correlation for situational awareness. In *IEEE Symposium on Information Visualization*. IEEE, 95–102.

27. Alan M MacEachren, Anuj Jaiswal, Anthony C Robinson, Scott Pezanowski, Alexander Savelyev, Prasenjit Mitra, Xiao Zhang, and Justine Blanford. 2011. Senseplace2: Geotwitter analytics support for situational awareness. In *IEEE Conference on Visual Analytics Science and Technology*. IEEE, 181–190.
28. Derek Overby, Jim Wall, and John Keyser. 2012. Interactive analysis of situational awareness metrics. In *Visualization and Data Analysis*, Vol. 8294.
29. Raja Parasuraman and Victor Riley. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human factors* 39, 2 (1997), 230–253.
30. Poonam Rana, Deepika Pahuja, and Ritu Gautam. 2014. A critical review on outlier detection techniques. *International Journal of Science and Research* 3 (2014).
31. W Scott Neal Reilly, Sean L Guarino, and Bret Kellihan. 2007. Model-based measurement of situation awareness. In *Simulation Conference, 2007 Winter*. IEEE, 1353–1360.
32. Maria Riveiro, Goran Falkman, and Tom Ziemke. 2008a. Improving maritime anomaly detection and situation awareness through interactive visualization. In *11th International Conference on Information Fusion*. IEEE, 1–8.
33. Maria Riveiro, Fredrik Johansson, Göran Falkman, and Tom Ziemke. 2008b. Supporting maritime situation awareness using self organizing maps and gaussian mixture models. *Frontiers in Artificial Intelligence and Applications* 173 (2008), 84.
34. Volker Roth. 2006. Kernel fisher discriminants for outlier detection. *Neural computation* 18, 4 (2006), 942–960.
35. Peter J Rousseeuw and Annick M Leroy. 2005. *Robust regression and outlier detection*. Vol. 589. John wiley & sons.
36. P Salmon, M Neville, A Stanton, D Ladva, DP Jenkins, GH Walker, and L Rafferty. 2007. *Measuring Situation Awareness during Command and Control Activity: A Comparison of Brunel University Measures Study Human Factors Integration Defence Technology Centre 2007*. Technical Report. HFIDTC/2/1.2. 5/3, Version 2/25 September.
37. Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural computation* 13, 7 (2001), 1443–1471.
38. Sameer Singh and Markos Markou. 2004. An approach to novelty detection applied to the classification of image regions. *IEEE Transactions on Knowledge and Data Engineering* 16, 4 (2004), 396–407.
39. Sarah Vieweg, Amanda L Hughes, Kate Starbird, and Leysia Palen. 2010. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 1079–1088.
40. Wikipedia. 2017. Gaussian Kernel. (2017). https://en.wikipedia.org/wiki/Radial_basis_function_kernel [Online; accessed 10-Sep-2017].
41. Liu Yang and Rong Jin. 2006. Distance metric learning: A comprehensive survey. *Michigan State Universiy* 2, 2 (2006).
42. Weiwei Yuan, Donghai Guan, Eui-Nam Huh, and Sungyoung Lee. 2013. Harness human sensor networks for situational awareness in disaster reliefs: a survey. *IETE Technical Review* 30, 3 (2013), 240–247.
43. Ji Zhang, Hua Wang, Xiaohui Tao, and Lili Sun. 2013. SODIT: An innovative system for outlier detection using multiple localized thresholding and interactive feedback. In *International Conference on Data Engineering*. IEEE, 1364–1367.
44. Fangfang Zhou, Ronghua Shi, Ying Zhao, Yezi Huang, and Xing Liang. 2013. Netsecradar: A visualization system for network security situational awareness. In *Cyberspace Safety and Security*. Springer, 403–416.
45. Cui Zhu, Hiroyuki Kitagawa, Spiros Papadimitriou, and Christos Faloutsos. 2004. Example-based outlier detection with relevance feedback. *DBSJ Letters* 3, 2 (2004), 1–4.