Multiple Queries with Conditional Attributes (QCATs) for Anomaly Detection and Visualization

Simon Walton, Eamonn Maguire and Min Chen

University of Oxford, UK

ABSTRACT

This paper describes a visual analytics method for visualizing the effects of multiple anomaly detection models, exploring the complex model space of a specific type of detection method, namely Query with Conditional Attributes (QCAT), and facilitating the construction of composite models using multiple QCATs. We have developed a prototype system that features a browser-based interface, and database-driven back end. We tested the system using the "Inside Threats Dataset" provided by CMU.

General Terms

Algorithm, visual analytics

Keywords

Anomaly detection, QCAT, information theory, multivariate data visualization, parallel coordinates, model visualization.

1. INTRODUCTION

Anomaly detection and visualization has been a vigorous research topic in visual analytics, offering a crucial technique to many applications, such as cybersecurity, image processing, financial management, text analysis, and so on [5]. There is a very large collection of works on anomaly detection, and for details readers may consult several surveys (e.g., [5, 6, 14]). One family of anomaly detection methods uses information-theoretic measures to determine if a data record x is anomalous in relation to a dataset $\mathbb{X}$ such that $x \in \mathbb{X}$. Because this technique is a direct implementation of the probabilistic definition of anomaly, the measured values of data records represent the ground truth of the frequency of their occurrence in the corresponding dataset. As Chandola et al. pointed out, this family of techniques have a number of advantages, including no requirement for supervised learning, no assumption about statistical distribution and no influence from human perceptual biases [5].

However, it also has some shortcomings. The most critical shortcoming is that their performance "is highly dependent on the choice of the information theoretic measure" [5]. Given a multivariate dataset, there can be numerous information-theoretic metrics, each focusing on a subset of variables. For example, the technique *Query with Conditional ATtributes* (*QCAT*), which was first deployed for detecting anomalies in the ASA flight dataset [11], features such a challenge. One can define a variety of metrics (referred to as QCATs) for detecting different types of anomalies, e.g., anomalous delay patterns among airports, anomalous patterns in weekends, anomalous patterns in relation to Taxi-in and out, and so forth. In fact, there is nothing fundamentally wrong to define different anomaly detection QCATs. The challenge is for a user to know if a specific QCAT is effective in comparison with others, or to decide when several QCATs should be deployed jointly.

In this paper, we propose the use of a parallel coordinates visualization to support users' in the creation of QCATs, selecting one or more appropriate QCATs for detecting a specific type of anomaly and observing their performance. This forms an effective visual analytics loop, where QCATs are analytical models, and parallel coordinates visualization serves as a visual interface.

In the remainder of this paper, we first give a brief overview of anomaly detection in Section 2, highlighting the related works on anomaly visualization. We then describe the mathematical concept of a QCAT in Section 3, and our implementation of a visual analytics prototype for supporting QCATs in Section 4. This is followed by a discussion of our experience in applying multiple QCATs to the CMU CERT dataset [4] in Section 5. We offer our concluding remarks in Section 6.

2. RELATED WORK

The earliest work on *anomaly detection* is often attributed to Edgeworth's paper in 1887 [8]. Since then, thousands of papers have been published, including several important surveys (e.g., [5, 6, 14]). The subject overlaps significantly with *outlier detection* and *novelty detection*, though many scholars prefer to differentiate them. Techniques for anomaly detection fall into several categories [5]:

- *Classification-based Methods* – This family of techniques rely on machine learning to train a detection model (also referred to as a classifier). Machine learning frameworks that have been used for anomaly detection

VizSec '14, November 10 2014, Paris, France.

http://dx.doi.org/10.1145/2671491.2671502

include neural networks, Bayesian networks, support vector machines, and rule-based reasoning.

- *Nearest Neighbor-based Methods* – This family of techniques rely on one or a few distance or similarity metrics that measure the proximity among data points. Such a technique typically judges whether or not a data point is anomalous based on its distance to its k^{th} nearest neighbor, or based on the relative density of its neighboring data points.
- *Clustering-based Methods* – This family of techniques rely on a clustering algorithm to group data points based on a distance or similarity metric. Data points that do not belong to any clusters of an appropriate size are considered to be anomalies.
- *Statistical Methods* – This family of techniques rely on a statistical model that encodes the underlying distribution of data points. Data points that occur in the low probability region of the model are considered to be anomalies. Machine learning is often used to train the statistical model.
- *Information-theoretic Methods* – This family of techniques rely on one or a few information-theoretic metrics to analyze the information content of a data point in relation to a data set. The more information content a data point contains, the more anomalous it is.
- *Spectral Methods* – This family of techniques rely on the discovery of a lower dimensional embedding of the original dataset. In such an embedding, normal and anomalous data points appear to be significantly different, hence easily detectable using a simple classifier. The commonly used techniques for discovering such an embedding include Principal Component Analysis (PCA), Compact Matrix Decomposition (CMD) and other dimensionality reduction techniques.

The work featured in this paper belongs to the family of information-theoretic methods. Chandola et al. provided a detailed comparison of this family and others [5]. In short, information-theoretic methods do not require annotated training datasets and can be used in an unsupervized situation. They make no assumption about statistical distribution, or semantically sensitive distance metrics and parameters (such as weight). In general, they are not influenced by human perceptual biases through annotation, distance metrics and weighting functions. As stated in [5], "A key challenge of such techniques is to find the optimal size of the substructure that would result in detecting anomalies." Part of this work is to address this challenge by allowing multiple QCATs, each representing a different "substructure".

Information-theoretic methods have been used for anomaly detection in a variety of datasets, including: multivariate sequence data (anonymous ftp records, shuttle-landing data, echocardiogram data) [2], time series data [10], spatial data (criminal incidents records) [12], text data [1], graph data [13]; network traffic data [19]; imagery data [16] and so on.

In recent years, more research effort has been channeled towards *anomaly visualization*. Much of the focus has been placed on depicting anomaly detection results in conjunction with one or a few visual representations. The visual representations features in the existing works on anomaly visualization include: radial tree layout [17], network visualization [17], bar charts [3], geospatial visualization [18, 15] and pixel-based visualization [9]. In addition, numerous papers in anomaly detection featured time series plots and scatter plots. In this work, we use parallel coordinates plots to visualize multivariate data records. In addition to depicting anomalies detected by each QCAT, we map the detection results of multiple QCATs to different axes, facilitating interactive exploration of different QCAT models and creation of composite QCATs.

3. QCAT MEASUREMENT

3.1 Introductory Overview

A QCAT, which stands for a *Query with Conditional ATtributes*, is a metric that measures the surprisal level of a data record in relation to other data records in the same dataset. It is based on information-theoretic measurement about information content of a data record, that is, the rarer the record occurs, the more surprise it brings when it occurs, and the more information content it carries [7].

Given an n-dimensional multivariate dataset, $\mathbb{D}$, if the probability of occurrence of every record $R \in \mathbb{D}$ is known, it is trivial to obtain the surprisal level of R. One can simply make a judgement based on the probability $p(R)$ itself. Alternatively one can use the information-theoretic measurement of self-information, $I(R) = -\log_2(p(R))$, which has a unit of bit. However, in practice, it is rare for most people, if not all, to know the probability of a record describing a complex event. For example, on a sunny day, at 12:00noon, a female bus driver, in Oxford, drove a No.2 bus, passing Magdalen Bridge, in the company of 14 passengers (3 elderly ladies, 2 elderly gentlemen, 3 female students, 1 male student, and 1 middle-age lady with 2 boys and 2 girls). It is obvious that such a record, R, is most likely to be unique among a large collection of records of this nature, i.e., $\mathbb{D}$.

There are many different questions that could be posed in this scenario. Each would lead to a different estimation of probability and a different assessment of normality. For example, how common is to have a female bus driver in Oxford? How common is it to have four students on a No.2 bus at 12:00noon? How common is it to have sunny weather at Oxford around noon? How common is for a No.2 bus to pass Magdalen Bridge with 14 passengers?

It is not difficult for one to observe that each question considers only a subset of attributes in the record. We therefore need to *query* a subset of variables in $\mathbb{D}$. One can also observe that many questions consider a specific *condition*, such as "in Oxford", "around noon". Hence, it is helpful to define a specific value, or value range for some variables, when estimating the probability distribution of other variables in the subset. This is what is meant by the term *Query with Conditional Attributes* (*QCAT*). The *conditional attributes* define the context of a query. The variables that are used to estimate the probability are *variants of normality* (*VONs*).

The QCAT measurement for anomaly detection is an information-

theoretic measurement, which will be detailed in the following subsection. Such a measurement is not a traditional detection algorithm typically derived from a machine learning process. With an information-theoretic measurement, anomalies are defined mathematically based on the probability of events captured by the historical data. Hence in relation to this definition of normality vs. anomaly, the probabilistic ranking of events derived from the measurement is always correct. On the other hand, most machine learning methods use a different definition, where an event is anomalous if it is subjectively annotated as an anomaly. The goal of a learned algorithm is thereby to mimic human perception of an anomaly. For a qualitative comparison, readers are referred to the survey by Chandola et al. [5], where a few other approaches are also considered.

3.2 Definitions

Let $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ be a set of n attributes (or *dimensions*) in a dataset. Each data record, $R = \{v_1, v_2, \ldots, v_n\}$ is an n-tuple, where v_i represents a valid value of attribute $\mathbf{a}_i$. Here the term "attribute" is interchangeable with "variable". We use the former to help differentiate $\mathbf{a}_i$ from v_i. In a practical scenario, an attribute, $\mathbf{a}_i$, may have a very large or infinite number of valid values as the value may be numerical with an unknown range, or a floating point range. Therefore, a common strategy for estimating a probability distribution is to divide a broad value range into an appropriate number of bins. For example, the value range of a timestamp attribute can be divided into bins based on every five minutes in an hour (12 bins), every hour in a day (24 bins), every day in a week (7 bins), and so on. The appropriate selection of the number of bins and the bin width not only facilitates an accurate estimation of the probability distribution, but also captures the normality vs. anomaly effectively. In the following discussion, the probability distribution of an attribute, $p(\mathbf{a}_i)$, is assumed to be estimated in conjunction with an appropriate binning scheme.

The attribute set, $\mathbf{A}$, is divided into three mutually-exclusive subsets, $\mathbf{A}_{cnd}$ for all Conditional Attributes, $\mathbf{A}_{von}$ for all Variants of Normality (VONs), and $\mathbf{A}_{ins}$ for the rest of attributes in $\mathbf{A}$, i.e., $\mathbf{A}_{ins} = \mathbf{A} - (\mathbf{A}_{cnd} \cup \mathbf{A}_{von})$. For a specific query defined by a configuration $(\mathbf{A}_{cnd}, \mathbf{A}_{von})$, $\mathbf{A}_{ins}$ are attributes considered to be "insignificant" for this query.

As anomalies are context-sensitive, $\mathbf{A}_{cnd}$ defines the context of a type of anomaly as a particular condition, such that all attributes in $\mathbf{A}_{cnd}$ are associated with specific values, with respect to their defined bins. For example, if an attribute $\mathbf{a}_k$ represents the dates of a user logging into a system, we may divide a 24 month period into 7 bins, each representing a day in a week. When $\mathbf{a}_k$ is chosen as a conditional attribute, we can estimate the probability distribution of attributes in $\mathbf{A}_{von}$ for a specific condition, such as Mondays, or Weekends (Saturdays and Sundays).

The attributes in $\mathbf{A}_{von}$ play the primary role in determining an anomaly score for each record that has met the condition defined by $\mathbf{A}_{cnd}$. Since attributes in $\mathbf{A}_{ins}$ are considered to have an "insignificant" influence on a query, they are excluded from the computation. Such a decision is usually made based on some known factors or logical reasoning by the user. It is obvious that decisions of this nature can be unreliable. This leads to the need for selecting most appropriate queries, or QCATs, which is a long standing challenge in the family of information-theoretic methods for anomaly detection [5]. The objective of this work is to use a visual analytics approach to address this challenge.

A combined configuration of $\mathbf{A}_{cnd}$ and $\mathbf{A}_{von}$ in relation to the overall attribute set $\mathbf{A}$ subsequently determines how anomaly scores are estimated for each record. Given a record R, we first retrieve all records that have the same conditional attribute values as R. Let this collection of records be $R_1, R_2, \ldots, R_W$, where W is usually a very large number. We now consider only the variants of normality defined by $\mathbf{A}_{von} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_s\}$. In conjunction with a binning scheme, each attribute, $\mathbf{x}_j$, may take valid values that are mapped to a set of t_j bins $B_j = \{b_{j,1}, b_{j,2}, \ldots, b_{j,t_j}\}$. For the s attributes in $\mathbf{A}_{von}$, there are a total of: $t_1 \times t_2 \times \ldots \times t_s$ different combinations of bins across different attributes. These combinations collectively define an alphabet $\mathcal{Z}$, and each unique combination is a letter $z \in \mathcal{Z}$.

The selection of an appropriate binning scheme for each attribute $\mathbf{x}_j$ is essential for ensuring that the total number of letters $|\mathcal{Z}|$ is smaller than the total number of records W. Ideally, we have $|\mathcal{Z}| << W$. We can, then, estimate the probability of each letter $z \in \mathcal{Z}$ based on the collection of records $R_1, R_2, \ldots, R_W$, resulting in a probability distribution function $p(z)$. For the given record R, we obtain its probability $p(R)$ by mapping it to its corresponding letter in $\mathcal{Z}$. The level of self-information is $I(R) = -\log_2(p(R))$, which is also called *surprisal*. We use this surprisal value as the anomaly score for the given record R. The entropy of the alphabet $\mathcal{Z}$, i.e., $H(\mathcal{Z})$, indicates the uncertainty of the data space. When $H(\mathcal{Z})$ is close to the maximum entropy $\log_2(|\mathcal{Z}|)$, the difference between two surprisal values is likely to be less meaningful. When it is much lower than the maximum entropy, $\log_2(|\mathcal{Z}|)$, the difference is usually more meaningful.

3.3 Multiple QCATs

It is necessary to emphasize that the anomaly score obtained for R reflects only the type of anomalies encoded by the specific query configuration $(\mathbf{A}_{cnd}, \mathbf{A}_{von})$. Conveniently and metaphorically, we call each configuration a QCAT (a qu cat). As mentioned above, a user may experiment with different QCATs in order to find one or a few effective query configurations for anomaly detection. An immediate need for conducting such an experimental process is the ability to observe the performance of each individual QCAT, as well as to compare different QCATs. Since the information-theoretic measurements discussed above, namely the anomaly scores and their uncertainty, all use "bits" as their units, they are not particularly intuitive for ordinary users.

In order to support the user interface to be detailed in Section 4, we introduced two more intuitive measurements that can be used to compare different QCATs. Since each QCAT has a different alphabet $\mathcal{Z}$, a specific surprisal value (e.g., 5.24 bits) for a record R can mean a normality in one alphabet, but an anomaly in another. We thus introduces a normalized value as an alternative, namely *percentile*. Given a QCAT that obtains W records, and computes a surprisal

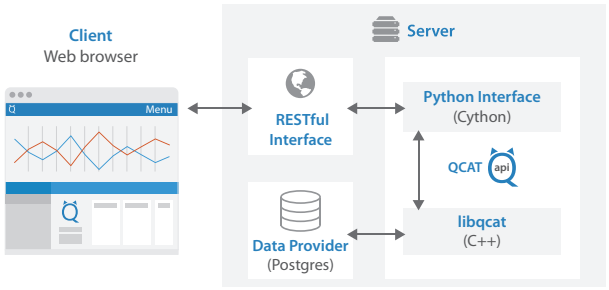

Figure 1: The QCAT Analytics Environment is comprised of a web client and a QCAT server. Interaction between the two occurs via a RESTful API.

value $I(R)$ for a record R, we map $I(R)$ to its percentile as:

$$\psi(R) = 100L/W$$

where L is the number of records that have the same surprisal value as, or a lower surprisal value than, $I(R)$. We also normalize the uncertainty measurement as $H(\mathcal{Z})/\log_2(|\mathcal{Z}|)$, where $H(\mathcal{Z})$ is the entropy of the alphabet $\mathcal{Z}$ and $\log_2(|\mathcal{Z}|)$ is its maximum entropy.

4. THE QCAT ANALYTICS SYSTEM

To support QCAT creation, result interrogation and analysis, we have developed what we call the QCAT analytics environment. This environment, illustrated in Figure 1, is comprised of a web-based visual analytics system built on top of a QCAT server. In this section we describe both of these components and how they interact.

4.1 Visual Analytics User Interface

The purpose of the visual analytics system is two-fold: 1) to support the design of QCATs; and 2) to enable the interrogation of results obtained from the QCAT server. Both purposes are interlinked since when creating a QCAT the user would ideally like to see the effect of changing conditional variables (with their ranges and bin sizes) and VONs. This refinement loop constitutes the visual analytics component of our system.

Figure 2 shows a number of screenshots of our user interface and the functionalities available to support the purposes outlined above. The main interface (central to the figure) is divided in to three panels: A) the result view panel featuring a parallel coordinates view of the data; B) a QCAT library pane to access all available QCATs and create new QCATs in the database; and C) a QCAT builder interface that provides the QCAT editing environment.

There are also four additional panels to support: composition of results from multiple QCATs (D); selection of data sets to perform the analyses on (E); editing QCAT glyphs; and viewing detailed table views of records selected in the parallel coordinate plot. In the remainder of this section we describe in more detail the features available for *designing a QCAT* and *exploring QCAT results*.

Designing a QCAT. As defined above, a QCAT consists of two sets of attributes: conditional attributes; and variants of normality. The QCAT design interface as shown in Figure 2 C presents a simple way to define a QCAT with conditional attributes and their range, and the VONs. The user can give the QCAT a name and a description, and can customize a QCAT glyph to enable visual identification of a QCAT. To support the iterative process present in analyzing multiple QCATs, the interface allows users to observe in real time how a defined QCAT performs against their data via the results panel described below. The underlying database also supports versioning of the QCATs, so users can retrieve previously defined QCATs and view how they perform against newer data.

Exploring the QCAT Results. QCATs results are displayed via the parallel coordinates plot (Figure 2 A) with more detailed numerical viewed by a supplementary table view of the data. Data can also be filtered by any of the dimensions sent from the QCAT server described in Section 4.2. This provides the means to increase responsiveness of the application when the full result set is hundred of thousands of records in size. The interactive parallel coordinates plot provides a single view of all the data with support for brushing (selecting ranges of one or more dimensions) and axis rearrangement. The default color scheme highlights in orange those records where one or more QCATs have detected an anomaly. Rules determining whether or not a record should be highlighted are configurable via the interface shown in Figure 2 D where users can: compose a rule from one or more QCATs; set the entropy score percentile at which records will be highlighted; and use logical AND/OR operations to perform intersections or joins of results from multiple QCATs.

4.2 QCAT Server

The client can query a data source via HTTP GET requests to the server. The client can also execute QCATs on the dataset to obtain the average entropy/surprisal discovered, and results for each record in the form of a list of surprisal values to insert into the parallel coordinates plot.

Our architecture is shown in Figure 1. Web-based clients (such as the one we have developed shown in Figure 2) connect to the QCAT server via asynchronous GET requests. The requests are served via a Python Flask instance connected via a Cython layer to our QCAT processing codebase *libqcat*, developed in C++ for efficiency to ensure that QCAT evaluation does not result in unworkable delays on the front end. The API is designed to be stateless, with parameters fed via a query string in the GET requests. Examples of API requests include:

- **/datatables**: provides a list of data sources (tables/views) available in the system;
- **/datadimensions**: given a data source name, provides information on the data source, such as the number, datatypes and statistics (such as range, mean, etc) of the dimensions, as well as request a number or all of the data rows;
- **/datarows**: given a data table, provides the associated data records. Results may be restricted by date and limited to a subset of rows;

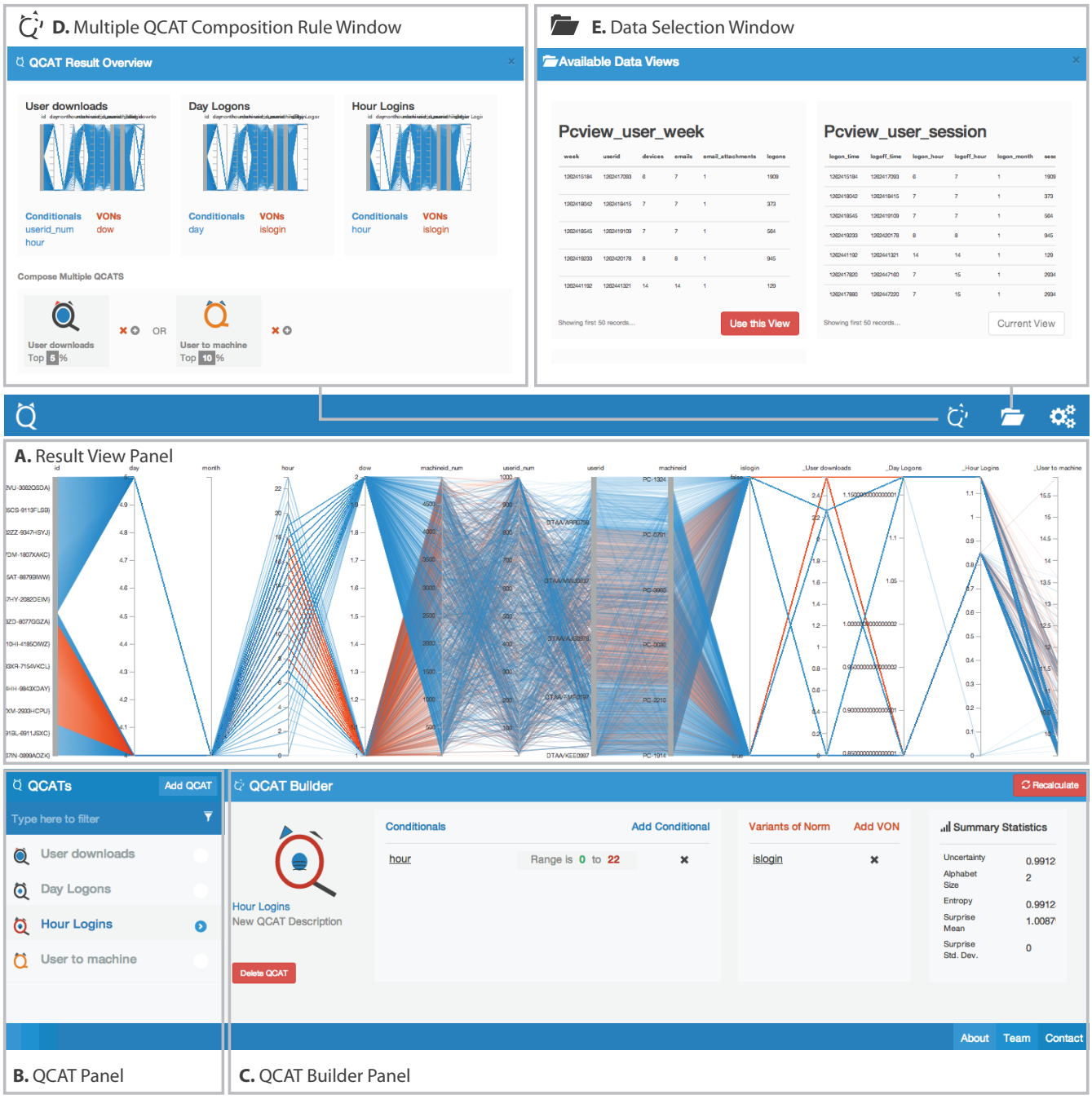

Figure 2: The User Interface for QCAT Builder consists of three primary panes: A) - the result view panel; B) - QCAT library panel; C) - the QCAT Builder panel; D) interface to support the composition of multiple QCATs enabling highlight of records only when all query conditions have been met; and E) a data selection window for selection of various data views.

- **/qcatsummary**: given the name of a data source and a QCAT definition, executes the QCAT and provides a summary of the QCAT execution such as the mean entropy/surprisal, number of rows matching the conditional, and the size of the alphabet.
- **/qcatsurprisals**: as above, but also provides a list of tuples (id, surprisal) matching the data provided by `/datarows` to surprisal values of this QCAT execution.

The QCAT server is standalone, and therefore runs independently of the aforementioned visual analytics front end. This design decision enables execution of QCATs via other agents that can provide notifications in the form of emails for example when anomalies have been detected by the QCAT processing codebase.

5. CASE STUDIES

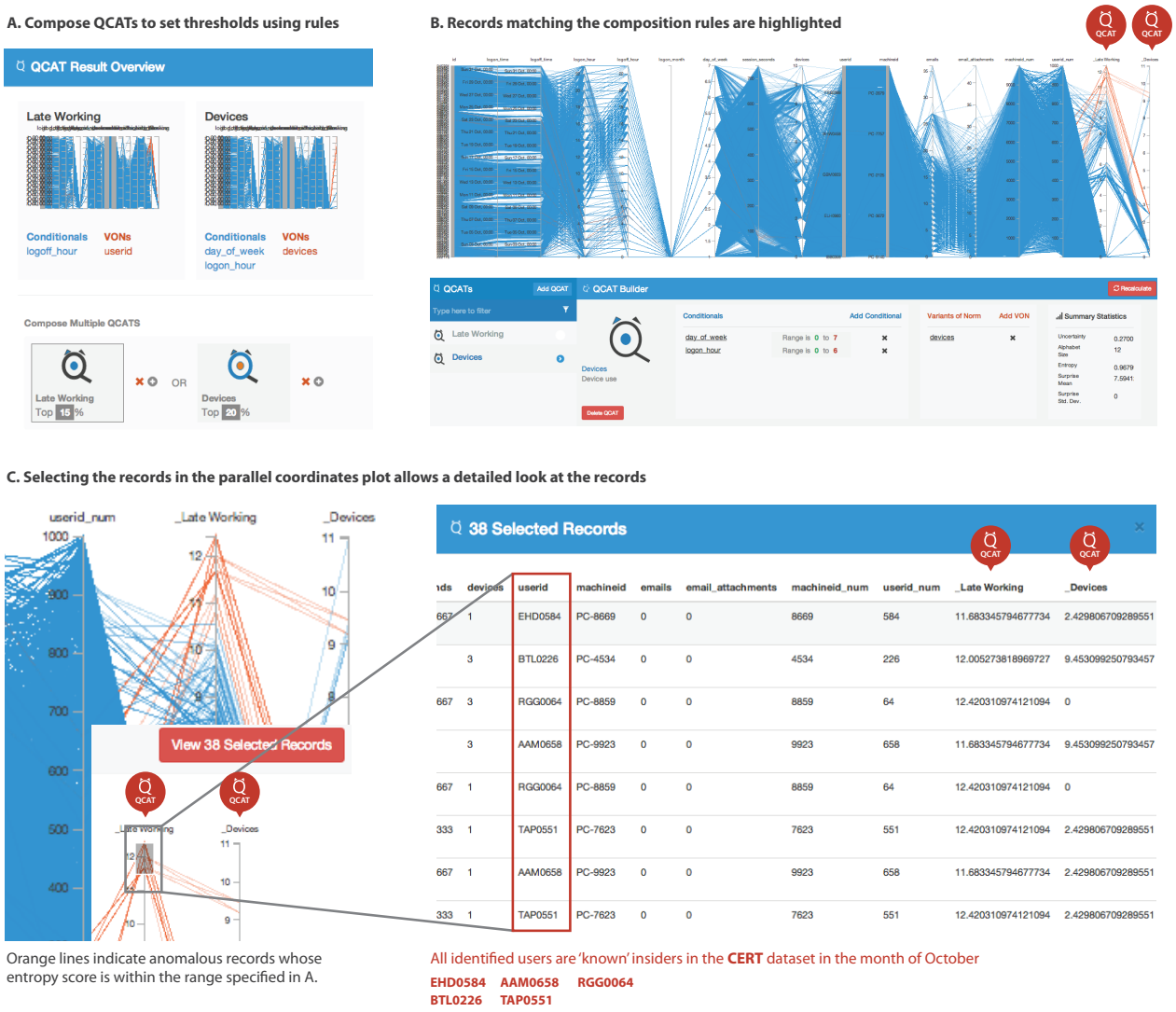

Figure 3: ***Scenario 1*** **- includes scenarios where a user begins to work late having never done so before and also inserts devices. We use two QCATs to capture this scenario, one for** ***late night working*****, the other to detect** ***device*** **insertion outside of working hours.**

To validate both the effectiveness and scalability of our approach in the security domain, we tested our system using the synthetic datasets from CMU CERT [4], which were created as a resource to test the efficacy of detection algorithms. These datasets contain insider threat events, and have documented ground truth. Using such a public dataset also allows for reproducibility of the results presented in this section.

Our test data, imported from the CMU CERT datasets (r1, r4.2 and r5.2), contain information about *users* (name, user id, email address and job role); *login/logoff events* (time, username, machine id and event type (login or logoff)); *email activity* (time, user, machine id, to (including cc, bcc), from, size, number of attachments, and content); *web activity* (time, user id, machine id, and url); and *device insertion/removal* (time, user id, machine id, and event type (insert or remove)). This provides a total of 2.6 million login/logout events, 1.24 million device insertion/removal events, 3.5 million web events, and 20 million emails.

Amongst this data are events related to five scenarios representing common forms of insider threats within an organization. Each scenario has between 10 and 30 synthetic "insiders". Here we detail the QCATs being used to find insiders in a selection of those scenarios.

5.1 Scenario 1

A user who had no previous history of using removable drives and who rarely worked after hours begins to do so, taking the data and sending it to media sites with malicious intent.

Figure 4 illustrates this scenario where two QCATs have been used to capture: (a) anomalous working hours (*Late Working*) using the logoff_hour as the conditional variable

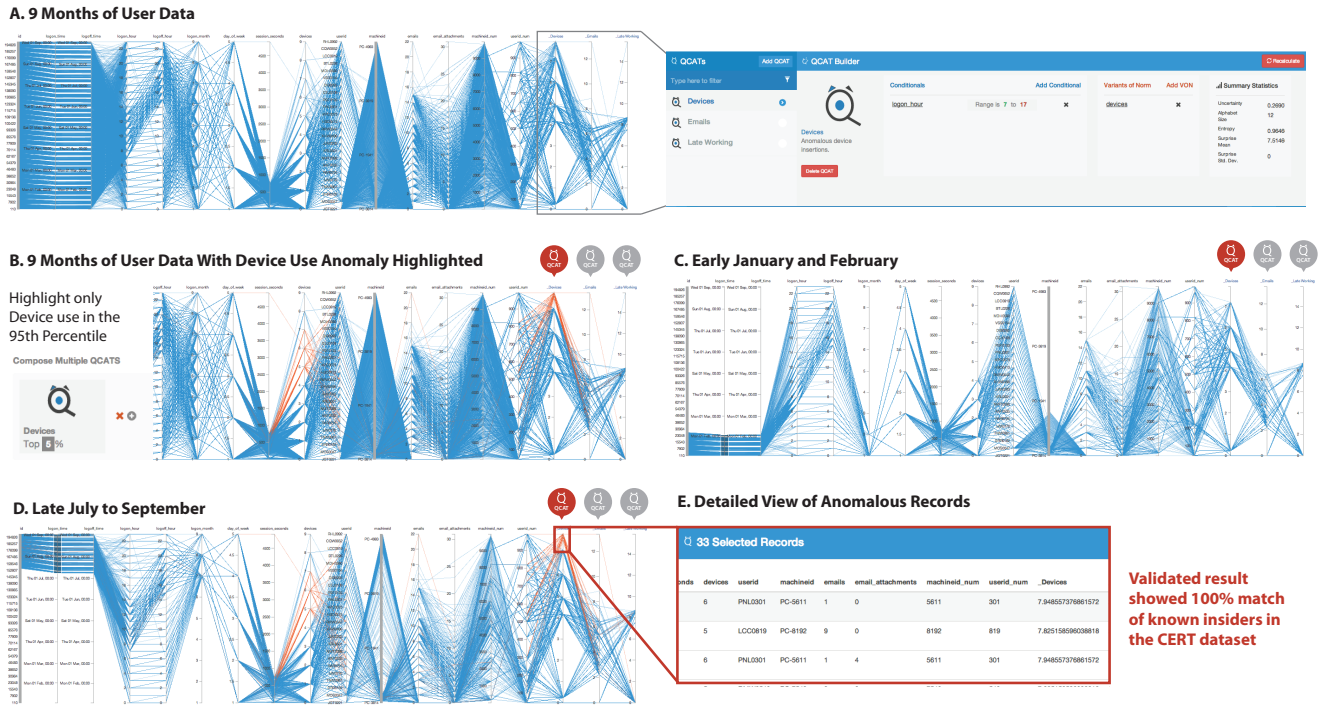

Figure 4: *Scenario 2* **- the insiders begin to use their thumb drive more often than usual with the intention of taking files with them to their new job with a competitor.**

and the userid as the VON; and (b) anomalous device insertion outside of working hours (*Devices*) with logon_hour and day_of_week as conditional variables and devices as the VON. We compose the results of both QCATs to highlight records with anomalous working hours (as shown in A), and device insertions (as shown in B). Selection of the top results in late night working gives rise to the output in Figure 4 C, where we can see a detailed view of the anomalous records. Through correlating the results displayed here with the ground truth available in the CMU CERT dataset, we show that our approach found all five "insiders" during the month of interest (October 2010). Across the entirety of the dataset, we were able to identify 100% of insiders.

5.2 Scenario 2

An employee began searching for a new job, and eventually found a position with a competitor. Before leaving the current organization, the employee started using a thumb drive much more than previously to steal data.

This particular case study tests the capability of the QCAT method to detect gradual changes of behaviour. Figure 4 A shows an example with three QCATs, *devices*, *emails*, and *late working*. Because the three QCATs fail to detect any anomalies collectively at the set threshold (95th percentile), we interactively explore each QCAT. In Figure 4 B we choose to focus on records that the *device* QCAT finds anomalous. These records are then highlighted as orange lines. To examine how the anomalies change over time, Figure 4 C and D show how the presence of anomalous records has changed over time by brushing areas of the time axis on the left of the plot. Through correlating the anomalous records highlighted in orange (as shown in E) with the ground truth in the CERT dataset, each of these records is shown to correspond to a known insider. Again, together with interactive visualization, this QCAT was able to detect 100% of anomalies for this scenario.

5.3 Scenario 3

A system administrator became disgruntled. He downloaded key logging software and installed this on his supervisor's machine via a USB drive. The following day, he used the information collected to log in to his supervisors machine followed by sending an alarming email to many in the organization, causing panic.

In the CMU CERT dataset, there are ten insiders spread throughout the dataset to represent this scenario. One such example is given by an insider with user id *BBS0099*, who acts on machine *PC-5866* which belongs to her boss *FAW0032*. *BBS0099* proceeds to install key logging software through the insertion of a USB thumb drive. Figure 5 shows how our solution was able to identify anomalous activity on the part of logging in to someone else's machine using just one QCAT. (Note that earlier examples showed that one QCAT is often not sufficient.)

6. CONCLUSION

In this paper, we have demonstrated that a visual analytics approach can aid an anomaly detection method (i.e. QCATs) that would exhibit some serious shortcomings if it were to operate on its own. As the design space for QCATs in relation to a high-dimensional multivariate dataset is huge, the visual analytics approach enables a user to create differ-

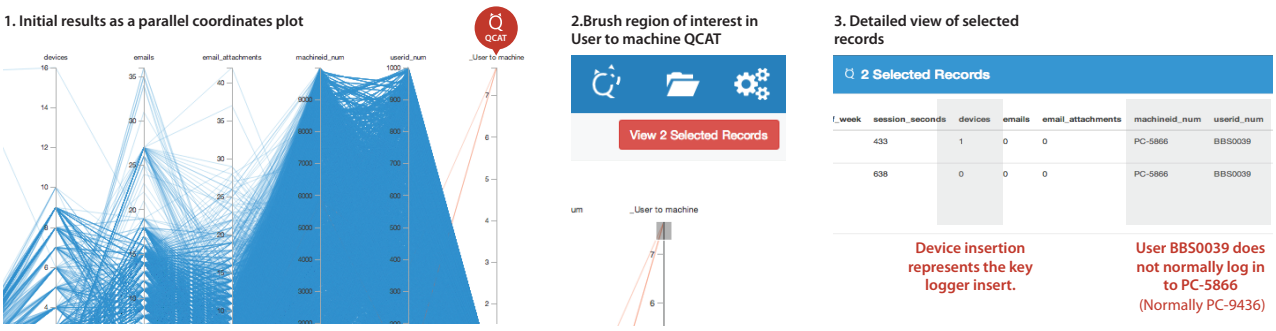

Figure 5: ***Scenario 3*** **a user, BBS0039 has become disgruntled, downloads a key logger and installs it on their bosses machine. The** ***User to machine*** **QCAT is built with the** ***userid*** **column as the conditional variable and the** ***machineid*** **as the VON.**

ent QCATs, and observe and compare their performance. When individual or composite QCATs produce false positives or false negatives, users can interactively explore the data space, by following the hints provided by QCATs, as well as utilizing their own background knowledge, visualization experience and sense-making skills. Though we have made a step forward to address the challenge of the "choice of the information-theoretic measure" [5], this challenge is not yet diminishing. We will continue this work to make further advances in this direction.

7. REFERENCES

[1] S. Ando. Clustering needles in a haystack: An information theoretic analysis of minority and outlier detection. *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 13–22, 2007.

[2] A. Arning, R. Agrawal, and P. Raghavan. A linear method for deviation detection in large databases. *Proc. ACM SIGKDD International Conference of Knowledge Discovery and Data Mining.*, pages 164–169, 1996.

[3] M. Celenk, T. Conley, J. Willis, and J. Graham. Predictive network anomaly detection and visualization. *Information Forensics and Security, IEEE Transactions on*, 5(2):288–299, 2010.

[4] CERT. Insider threat tools, http://www.cert.org/insider-threat/tools/index.cfm.

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

[6] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):823–839, 2012.

[7] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 1991.

[8] F. Edgeworth. Xli. on discordant observations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 23(143):364–375, 1887.

[9] A. Frei and M. Rennhard. Histogram matrix: Log file visualization for anomaly detection. In *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, pages 610–617. IEEE, 2008.

[10] E. Keogh, S. Lonardi, and B.-c. Chiu. Finding surprising patterns in a time series database in linear time and space. *Proc. ACM SIGKDD International Conference of Knowledge Discovery and Data Mining.*, pages 550–556, 2002.

[11] S. Ko, S. Afzal, S. Walton, Y. Yang, J. Chae, A. Malik, Y. Jang, M. Chen, and D. Ebert. Analyzing high-dimensional multivariate network links with integrated anomaly detection, highlighting and exploration. In *Proc. IEEE VAST*, 2014.

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

[13] C. C. Noble and D. J. Cook. Graph-based anomaly detection. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636, 2003.

[14] M. A. Rassam, A. Zainal, and M. A. Maarof. Advancements of data anomaly detection research in wireless sensor networks: A survey and open issues. *Sensors*, 13:10087–10122, 2013.

[15] M. Riveiro. Evaluation of normal model visualization for anomaly detection in maritime traffic. 2014.

[16] S. M. Schweizer and J. M. Moura. Hyperspectral imagery: Clutter adaptation in anomaly detection. *Information Theory, IEEE Transactions on*, 46(5):1855–1871, 2000.

[17] L. Shi, Q. Liao, Y. He, R. Li, A. Striegel, and Z. Su. Save: Sensor anomaly visualization engine. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 201–210. IEEE, 2011.

[18] D. Thom, H. Bosch, S. Koch, M. Worner, and T. Ertl. Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages. In *Pacific Visualization Symposium (PacificVis), 2012 IEEE*, pages 41–48. IEEE, 2012.

[19] F. Xie and L. Xie. Using information theory to measure call site information of system call in anomaly detection. *Communication Technology (ICCT), 2013 15th IEEE International Conference on*, pages 6–10, 2013.