Visual Analytics of Anomaly Detection in Large Data Streams

Ming C. Hao, Umeshwar Dayal, Daniel A. Keim*
Hewlett-Packard Palo Alto Laboratories, Palo Alto, CA
University of Konstanz, Germany*

ABSTRACT

Most data streams usually are multi-dimensional, high-speed, and contain massive volumes of continuous information. They are seen in daily applications, such as telephone calls, retail sales, data center performance, and oil production operations. Analysts want insight into the behavior of this data. They want to catch the exceptions in flight to reveal the causes of the anomalies and to take immediate action. To guide the user in finding the anomalies in the large data stream quickly, we derive a new automated neighborhood threshold marking technique, called ***AnomalyMarker***. This technique is built on cell-based data streams and user-defined thresholds. We extend the scope of the data points around the threshold to include the surrounding areas. The idea is to define a focus area (marked area) which enables users to (1) visually group the interesting data points related to the anomalies (i.e., problems that occur persistently or occasionally) for observing their behavior; (2) discover the factors related to the anomaly by visualizing the correlations between the problem attribute with the attributes of the nearby data items from the entire multi-dimensional data stream. Mining results are instantaneously presented in graphical representations (i.e., tooltip) for the user to zoom into the problem regions. Different algorithms are introduced which try to optimize the size and extent of the anomaly markers. We have applied this technique with success to detect data stream anomalies in large real-world enterprise server performance and data center energy management.

Keywords: Large Data Streams, Multi-Dimensional, Visual Correlation Queries, Anomaly Markers.

1. INTRODUCTION

A continuous data stream is a sequence of data items that is ordered by time. They are usually collected in real-time. Most data streams are high speed and multi-dimensional with massive volumes of information. Data streams contain highly valuable information and occur in many different applications, such as telephone calls, financial data (stock data or credit card transactions), oil production operational data, energy management data, server performance data, and network monitoring. Business analysts and administrators need to analyze the data streams to understand their business and service performance. They want to quickly identify patterns, trends, and exceptions in a single view instead of from many windows. The administrator also wants to catch anomalies in flight to reveal the causes and to take immediate action before the system consumes too many resources. Examples of anomalies are: an extremely busy data warehouse server, an overheated temperature sensor in a data center, and a low oil production well.

Novel methods are needed to enable the users to get an overview of large data streams and the ability to analyze patterns and trends. To guide the user to quickly identify anomalies we need to address the following issues:

- How to provide a single view for users to visualize interesting information from large complex data streams?
- How to extend the marked anomaly area to include interesting data items in the nearby time series for analyzing their behavior (i.e., does the problem occur persistently or occasionally)?
- How to find the root-cause of an anomaly? What are the related factors from the streams?

2. RELATED WORK

2.1 State of the Art

Visual analytics on large data streams is an emerging technology. Most visual analytics techniques are still limited to the presentation of a few hundred or thousand items and display screens are filled up quickly with the incoming data stream. One early approach to deal with this problem with high-density displays is Eick's SeeSoft [1]. Eick allowed users to analyze up to 50,000 lines of code simultaneously by mapping each line of code into a thin row allowing the users to find interesting patterns. Later Eick addressed high-density display issues in his Visual Scalability paper [2].

Afterwards, Ben Schneider man's Time Searchers [3, 4, and 5] developed three well-known interactive visual exploration techniques for large time-series. In TimeSearcher 1, users with the use of the Time box are able to define a pattern and search for similar patterns throughout the data. TimeSearcher 2 extends TimeSearcher 1 by allowing a visualization of long time series (>10,000 data points). TimeSearcher 2 provides the ability to view multivariate time series data by showing up to ten simultaneous plots on the same screen. TimeSearcher 3 adds the ability of forecasting. The forecast is represented graphically as a river plot showing statistical information about similar patterns. All three visualization techniques begin with an overview and allow the user to zoom into the areas of interest. The TimeSearcher applications for large time series data are financial and medical systems.

The scalability of time series displays may be maximized by resorting to the pixel level. In [9], a recursive scheme which maps sets of time series to color-coded raster pixel displays is introduced. In [6], Yost and North describe the results from their visualization scalability study. It shows that by using a combination of perceptual abilities and navigation, users were able to effectively visualize more data on a large display. In [10], Munzner sacrifices visual details for display capacity. In our own previous work [7, 11, 12], we address the problem of space-filling visualizations of many moderate length time series using cell-based multi-resolution techniques.

2.2 Our Contribution

Following the above methods, our research provides overview and detailed views of the performance of the data streams. In addition, for the stream visualization, we introduce three important visualization techniques: (1) avoiding overlapping; (2) detecting whether anomalies are persistent or occasional; and (3) performing some correlation analysis to detect root causes or related factors. To avoid overlapping, instead of using a conventional line chart (Fig. 1), we use a cell based-time data stream (Fig. 2) to show 7,701 observation values in a single view. Users can easily navigate through the cells and drilldown for the details.

One important question is how an anomaly develops over time. Is it persistent or occasional? For example, in a data center containing hundreds of servers, the administrator wants to know if server utilization for some of the servers becomes enormously high and whether that situation persists for some time. In so, the administrator might want to do better load balancing among the servers. One common method is to mark each individual threshold (small rectangles in Fig. 2), but this is insufficient for the analysts to detect the persistency of the anomaly. Once the users identify the anomalies, they want to find any correlations between the problem attributes and the attributes of the entire multi-dimensional data stream which might impact the anomaly. The solutions to both problems are described in Sections 3 and 4.

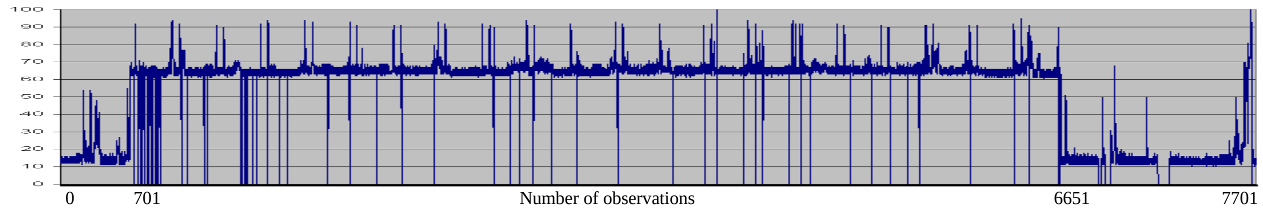


Fig. 1: A conventional time series data stream with overplotting

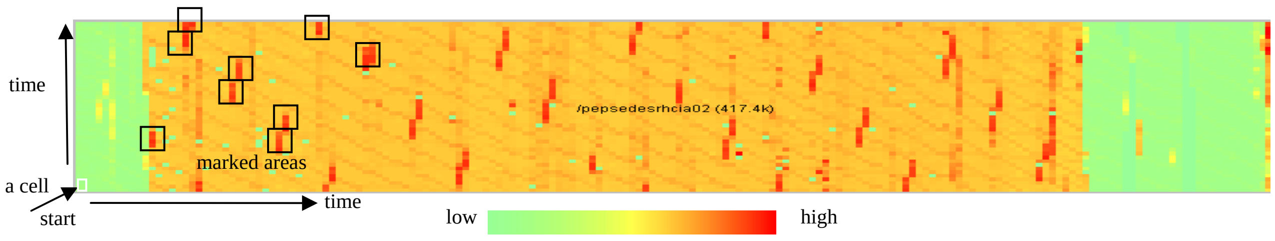


Fig. 2: A new-cell based time series data stream using individual threshold marking

Visualize 28 days worth of network monitor probes from one server resulting in 7,701 observation values shown in Fig. 1. Each observation is represented by a color-cell. The color of a cell represents the value of an observation. Cells are arranged from bottom to top (column) and left to right. This color-cell data stream visualization is able to display each observation without losing information. Each data item which exceeds the user-defined threshold is marked by a small rectangle to guide the user to detect anomalies in the streams.

3. OUR APPROACH

To detect anomalies from the real-time large multi-dimensional data streams, the approach consists of three iterative steps:

Step 1: Continuously layout the incoming data stream in a cell-based high-resolution display.

Step 2: Mark the incoming data points which exceed the user-defined threshold. Then optimize the scope of the marked area to include nearby marked areas using our AnomalyMarker algorithm.

Step 3: Enable users to perform visual query [8] to find the relationship of problem attributes with the other attributes in the stream; then present the mining results (i.e., persistency and correlations) from the marked areas in an interactive visual representation (i.e., tooltips) for finding the impact factors.

3.1 Cell-Based Visualization of Data Streams

Fig. 3A illustrates an enterprise data warehouse server performance data stream. There are 8 systems (SYS 1 to SYS 8). Each system has 16 servers (0 to 15). Each server has a cell-based time series. Each cell represents a record showing server busy% at 5 minute intervals. The new data streams will be read in and added to the end of the time series (right side of each time series). In Fig. 3A, users can visually compare 128 time series (each block) to analyze changes, patterns, and exceptions at a glance. Measurement time is from 5:00 to 14:55. The 128 parallel servers have very similar busy patterns. The workload has been evenly distributed among servers except for a few servers which show more red (e.g., Server 8 in SYS 1; Server 3 in SYS 3). In Fig. 3A, the administrator is able to instantaneously identify that Server 3 in SYS 3 has the highest busy % (red).

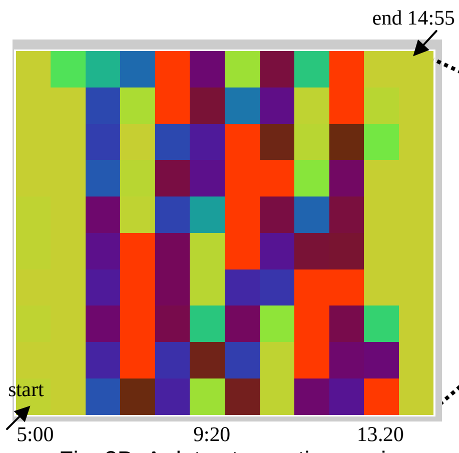
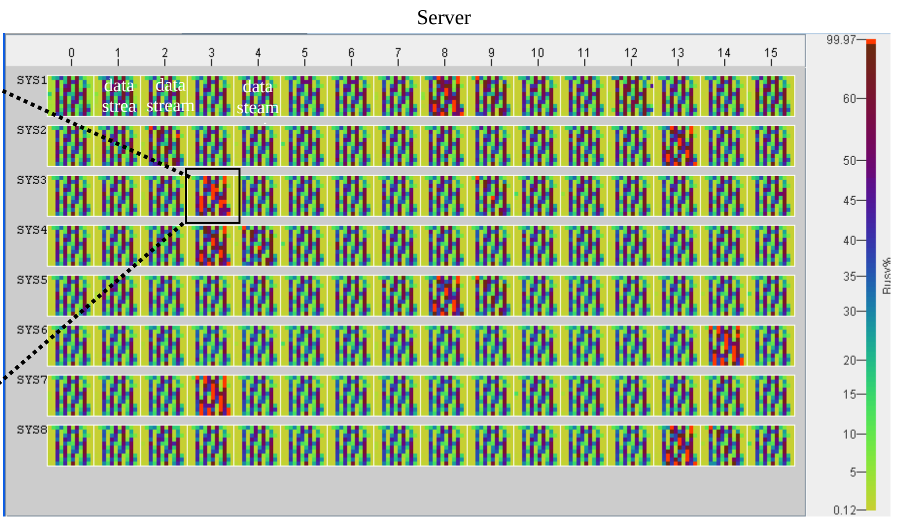


Fig. 3B: A data stream time series at SYS 3, Server 3



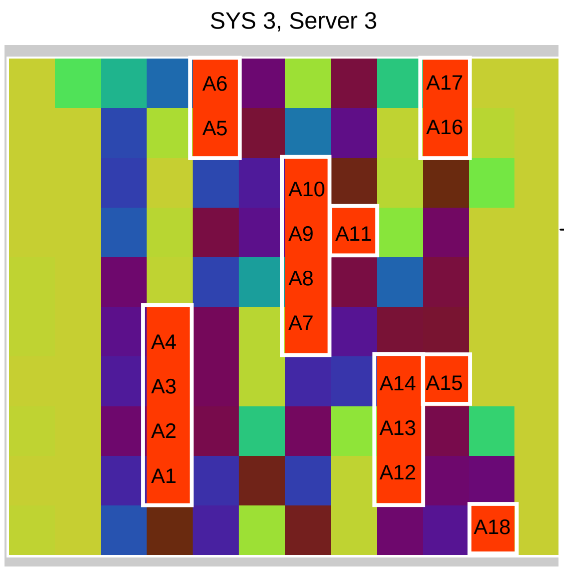
- Each data stream contains all the cells (data points) from 5:00 to 14:55.
- Each cell represents a data point showing Server Busy% at 5-minute intervals.
- Cells are arranged from bottom to top and left to right according to the time values.
- The color of a cell is the value of server busy%.

Fig. 3A: A cell-based time series visual analytics showing a large enterprise data warehouse server performance (8 SYS x 16 Server, total 128 data streams)

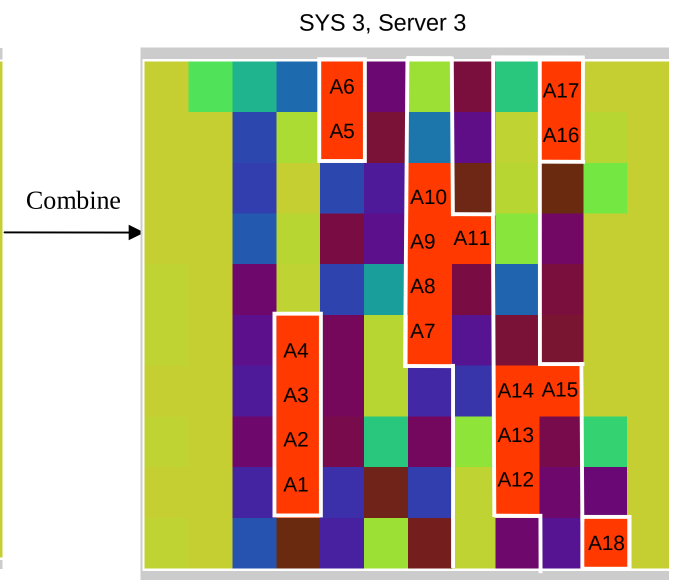
3.2 Enlarged Anomaly Marker

To detect anomalies in data streams, this paper introduces a new technique, called AnomalyMarker. This technique first marks the data points exceeding a user-provided threshold and then combines adjacent data points and previously marked areas into larger marked areas. This is different from just marking each data point which exceeds the threshold as shown in Fig. 2 in that it enlarges the scope of data points for (1) visualizing the anomaly persistency, and (2) visualizing the anomaly correlations with the other attributes for root-cause discovery (described in section 3.3).

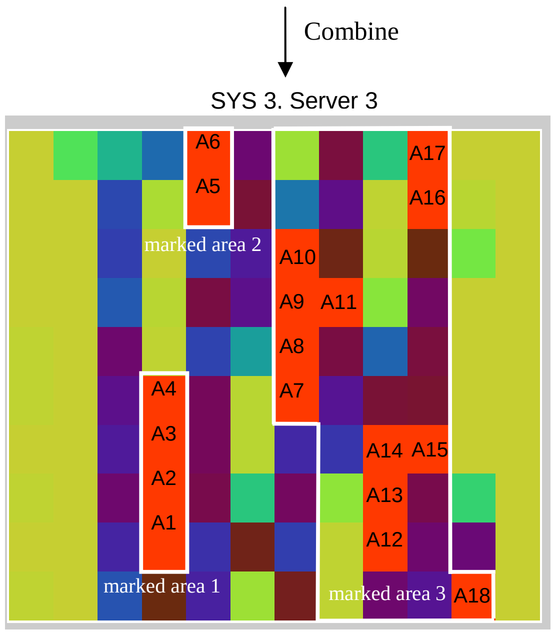
To enlarge the individual threshold marked areas, we use the above SYS 3, Server 3 (Fig. 3B) real-world data stream to illustrate the following three-step process in Fig. 4:



Step 1: Combine *vertically adjacent marked cells* such as cells A1-A4, A5-A6, A7-A10, A12-A14, and A16-A17 to enlarge marked areas. A11, A15, and A18 are single cell marked areas.



Step 2: Combine *horizontally adjacent cells* including the time continuous cells in between markers, such as: combine A11 to the marked area A7-A10 including cells between A10 and A11. Combine A15 to the marked area A12-A14 including cells between A14 and A15.



Step 3: Combine all *horizontally adjacent* marked areas, such as A7-A11, A12-A15, A16-A17, and the single marked cell A18.

The resulting graph shows the final three marked areas:

marked area 1:	A1-A4	(occasional anomalies)
marked area 2:	A5-A6	(occasional anomalies)
marked area 3:	A7-A18	(persistent anomalies)

Note:
Persistent anomalies: multiple anomalies happened in a short period of time.
Occasional anomalies: anomalies happened once a while.

Fig. 4: Three-Step Combining Process to Enlarge Marked Anomaly Areas

As a result of combining the anomaly marked areas, administrators are able to identify the enlarged area (e.g., marked area 3) has the most anomalies and happened persistently in a short period of time. The cause of these persistent anomalies could seriously impact operations in the data warehouse. However, the anomalies in the marked area 2 (two consecutive occurrences) are occasional and could be ignored.

3.3 Automated Visual Correlation Query

Streaming data is multi-dimensional with many different factors and analysts often want to know how these factors impact each other. An enlarged marked area provides a focus area for users to analyze the correlation between the problem attribute and other attributes from the entire data stream. Intelligent visual analytic queries have been introduced in our previous work [8]. Instead of using a rubber-banding technique to select the focus area, we use a marked area for correlation analysis. When the user moves the mouse over the marked area, the anomaly attribute (e.g., Server Busy%) will be compared automatically with the other attributes (i.e., Disk Usage, Memory Usage, etc.) in the same time period, as shown in Fig. 5A. The mining result will be instantaneously presented in an interactive tooltip as shown in Fig. 5B with a computed correlation coefficient [8].

Fig. 5B illustrates that server busy% is highly correlated with the attribute DIS (Disk Usage) during the time 10:00 to 14:05. The Busy% data points (black) and the corresponding disk usage data points (blue) change at the same pace. Users are able to interact with the data points in the line chart (Fig. 5B) to analyze the details. With the correlation visual analysis tooltip, analysts are able to interactively spot and identify the source of a problem. The calculation time in the example takes about 361 milliseconds. Developing these insights by a manual search would have taken much longer time, since the administrator would have had to search through many pages of tables or charts.

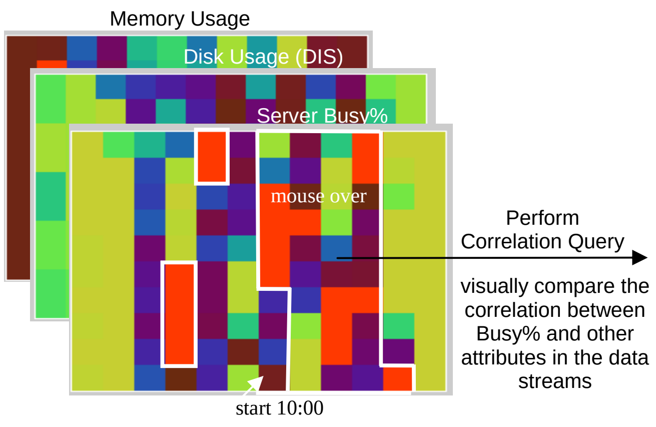

Memory Usage
Disk Usage (DIS)
Server Busy%
mouse over
start 10:00
Perform Correlation Query
visually compare the correlation between Busy% and other attributes in the data streams

Fig. 5A: A data stream time series at Sys 3, Server 3 with different attributes (e.g., *Server Busy%, Disk Usage, and Memory Usage)* at the same time period in the streams.

- mouse over a server busy marked area from 10:00 to 14:05
- get the corresponding data points in the other attributes (e.g., disk usage, memory usage, etc.)

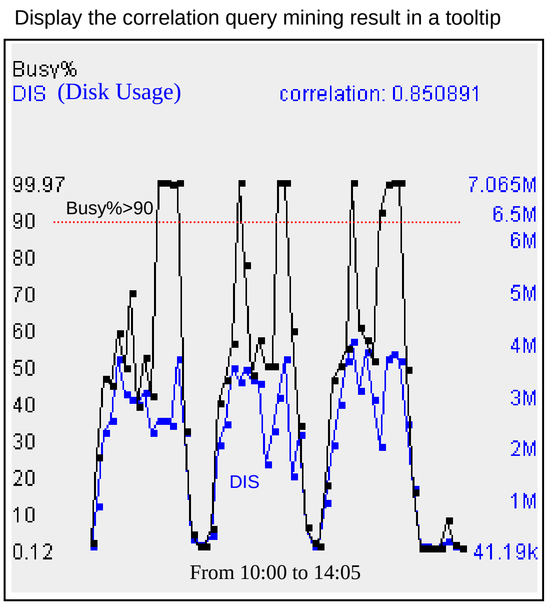


Fig. 5B: An anomaly marker visual analysis showing some correlations between Server Busy% and Disk Usage (High Disk Usage could result in Server Busy)

- Busy% line chart in black to show 12 data points which exceed the Busy% > 90 (persistent anomalies).
- A blue line chart with the corresponding 11 high DIS data points.
- Show high correlations between Busy% and DIS (Disk Usage) with the same change pace.

4. ANOMALY MARKER ALGORITHMS

In this section, we introduce three variants of our anomaly marker algorithms. Fig. 4 illustrates the process of enlarging the anomaly marked areas using a server performance data stream example. The algorithms work as follows: In the first step, we mark all cells that are above the MarkingThreshhold. If there are any such cells (checked by the function MoreMarkersAvailable), in the second step we get the first two cells (in the time sequence) above the threshold and combine it with its direct neighbors. Direct neighbors are cells in the time sequence that are directly neighboring cells in the time-sorted order. In the third step, indirectly neighboring cells are combined. This can be done in three different ways leading to three variants of the algorithm.

The first variant of our algorithms uses a visual criterion to combine indirect neighbors: Marked areas are combined if the marked areas intersect. The new marker width spans the total size of the two markers including the sequence in between. A problem of the first variant is that even two isolated marked cells which happen to be in the same row of neighboring columns may cause the two columns to be united into one large marked area (see Fig. 6B in the top left of variant 1 for an example). The second variant of the algorithm therefore only combines the marked cells if there is a minimum percentage (%MarkedCells) of cells above the MarkingThreshold within the combined marker (CombinedMarker(MA1,MA2)). This leads to more natural results (see Fig. 6B variant 2) but the cells above the threshold may still be multiple isolated events which is different from the situation where many cells within the marker are close to the threshold (see Fig. 6B variant 2). Our third variant therefore combines the marked areas if the average value of the cells with the combined marker (AvgValueOfCells) is close enough (%Value) to the MarkingThreshold (see Fig. 6B variant 3). Note that MarkingThreshold, %MarkedCells and %Value are variables that the user can control interactively depending on the data and application. As shown in Figures 6A and 6B, the third variant of our Anomaly Marker Algorithm provides nice results, even for problematic cases. Please note that depending on %Value, different results are obtained. In Fig. 6A variant 3, a smaller value for %Value will lead to two markers instead of one. A problem of the two more elaborate variants of our algorithm, however, is that depending on the parameter settings two neighboring markers may touch (such as the two markers in the upper left corner in Fig. 6B variant 2 and 3) . This however is rarely visually confusing since the marked areas are clearly separated by non relevant cells. Confusing results may only occur for inappropriate parameter settings (such as too low values for %MarkedCells or %Value).

Anomaly Marker Algorithm (Variant 1):

```
FOR ALL Cells C DO               // Mark Cells
      IF  Value(C)> MarkingThreshold  THEN  Mark(C)  ENDFOR;
If MoreMarkersAvailable          // Only continue if Markers are available
   THEN   GetFirst(MA);
          If MoreMarkersAvailable THEN GetNext(NextMA);
          REPEAT                 // Combine direct neighbors
                 WHILE DirectNeighborInSequence(MA, NextMA) DO
                        tmpMarker = CombinedMarker(MA,NextMA);
                        Add(tmpMarker); Remove(MA); Remove(NextMA);
                        MA = tmpMarker;
                        If MoreMarkersAvailable THEN GetNext(NextMA);
                 ENDWHILE;
                 MA = NextMA;
                 If MoreMarkersAvailable THEN GetNext(NextMA);
          UNTIL  NOT(MoreMarkersAvailable);
          REPEAT                 // Combine indirect neighbors
                 FOR ALL Marked Areas MA1 and MA2 (neighbors in the time sequence) DO
                        IF INTERSECT(MA1,MA2) THEN
                               Add(CombinedMarker(MA1,MA2); Remove(MA1); Remove(MA2);
                        ENDIF;
                 ENDFOR;
           UNTIL  no markers are changed;
ENDIF
```

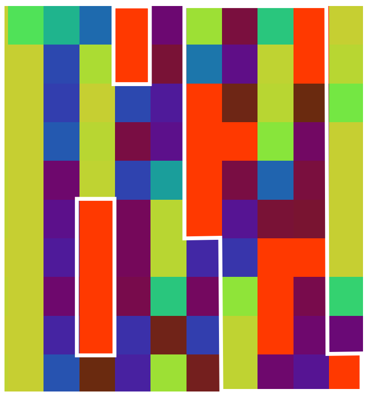
Anomaly Marker Algorithm (Variant 2):

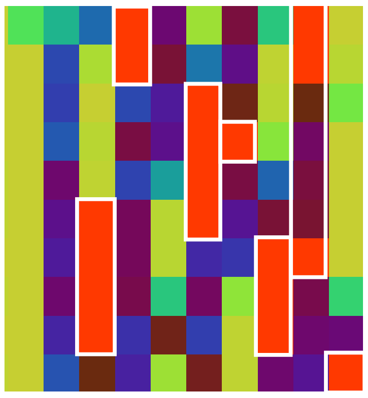
```
. . .   // Mark Cells (as in algorithm 1)
. . .   // Combine direct neighbors (as in algorithm 1)
        REPEAT                // Combine indirect neighbors
               FOR ALL Marked Areas MA1 and MA2 (neighbors in the time sequence) DO
                      tmpMarker = (CombinedMarker(MA1, MA2);
                      If #MarkedCells(tmpMarker)/#Cells(tmpMarker) > %MarkedCells
                             THEN
                                    Add(tmpMarker); Remove(MA1); Remove(MA2);
                      ENDIF;
               ENDFOR;
         UNTIL  no markers are changed;
```

Anomaly Marker Algorithm (Variant 3):

```
. . .   // Mark Cells (as in algorithm 1)
. . .   // Combine direct neighbors (as in algorithm 1)

        REPEAT
               FOR ALL Marked Areas MA1, MA2 (neighbors in the time sequence) DO
                      tmpMarker = (CombinedMarker(MA1, MA2);
                      IF AvgValueOfCells(tmpMarker) > %Value*MarkingThreshold
                             THEN
                                    Add(tmpMarker); Remove(MA1); Remove(MA2);
                      ENDIF;
               ENDFOR;
        UNTIL no markers are changed;
```


Variant 1


Variant 2

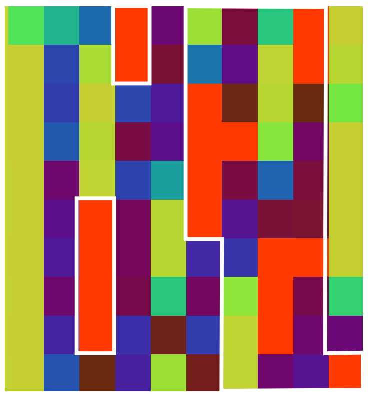
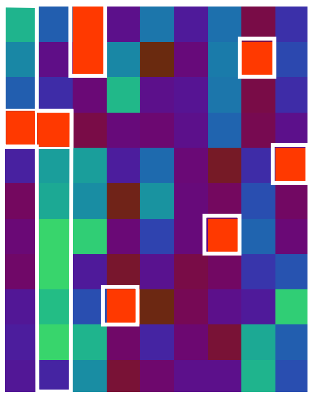
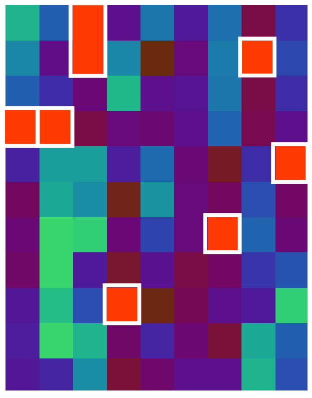

Variant 3

Fig. 6A: Anomaly Marker Algorithms – Example 1


Variant 1


Variant 2

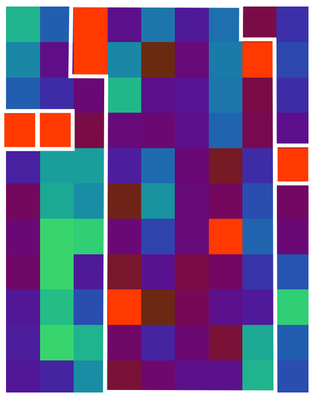

Variant 3

Fig. 6B: Anomaly Marker Algorithms – Example 2

5. A DATA CENTER TEMPERATURE MONITORING APPLICATION

We have used the AnomalyMarker to detect anomalies in streaming data with success in many different real-world applications, such as enterprise data warehouse server performance analysis (as shown in Figures 3, 4, and 5) and data center sensor temperature monitoring.

Today's data centers are becoming the computational hub of the next generation of IT services. A typical smart data center usually has more than 100 racks. Each rack has five sensors (T1, T2, T3, T4, and T5). The cooling air flows from T1 to T2, T3, T4, and T5. Within a rack, the temperature sensors (T1-T5) should be always kept in an ascending sequence, such as T1<T2<T3... The normal temperature should be below 38^{o}C. Any temperature above 38^{o}C will be considered an exception. With the advent of dynamic smart cooling and rack level sensing in today's data center, the need for visual data exploration is growing. The common questions which data center administrators want to answer are:

- What significant thermal events happened in the last few hours?
- Which racks are overheated? What is the root-cause?

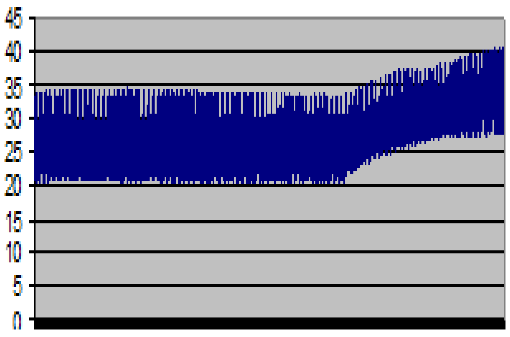


Fig. 7 An overplotted time series line chart (100 racks)

Fig. 7 shows an overplotted time series line chart using a conventional method. Administrators have difficulties in identifying which racks are overheated and what are the impact factors. To catch the overheated racks from a large multi-dimensional (e.g., sensors T1, T2, T3…) data stream, we need to employ a cell-based visual method to layout the entire data center racks, sensors, and time series in a single view as shown in Fig. 8A. The x-axis contains 100 rack names. The y-axis is the measurement time (from hours 11 to 13 on 5/10) at 1-minute intervals. Each hour contains a 60 time series (0-59 minute). The color represents the sensor temperature. From Fig. 8A, an administrator is able to recognize quickly that the data center was fairly hot in the hour 12 (most burgundy and some red), especially on the Rack EB1307. To identify the root cause of the overheating, we apply the AnomalyMarker algorithm to facilitate the finding of the temperature anomaly areas (illustrated in Fig. 8B to Fig. 8E). Fig. 8B shows the zoomed EB1307 data stream at the hour 12. Fig. 8C shows the 90^{o} rotation of the data stream with marked anomalies. Fig. 8D shows the enlarged marked areas after combining the nearby anomalies. Fig. 8E shows the mining correlation results after the administrator moves the pointer to the marked area to perform visual queries. The administrator can interact with the data points in the line chart (Fig. 8E) and find that T5 has a persistent high temperature above the threshold ($>38^{o}$C) over a long period of time from 12:30 to 12:44. Also, the T5 temperatures are highly correlated with those of T2, a coefficient (0.89). But T5 has lower temperatures than T2, which indicates that the sensor temperatures are out of sequence. Another interesting observation is that the temperatures of T3 (green line chart) have the opposite correlation to those of T5. From these factors, the administrator is able to take immediate action to reduce the heat, such as adjusting the sensor T2 and T3 temperatures on rack EB1307.

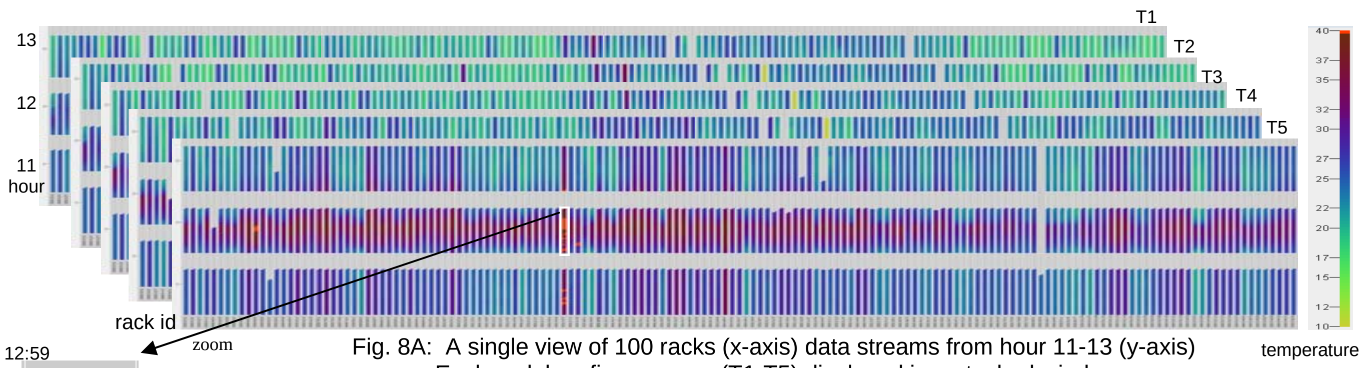


Fig. 8A: A single view of 100 racks (x-axis) data streams from hour 11-13 (y-axis)
Each rack has five sensors (T1-T5) displayed in a stacked window
(Find: Rack EB1307 with Sensor T5 has the highest temperatures at hour 12)

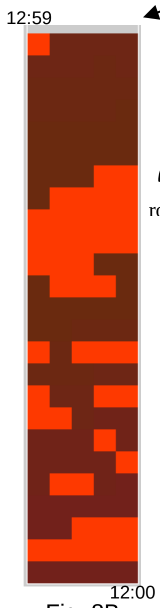


Fig. 8B
Rack EB1307

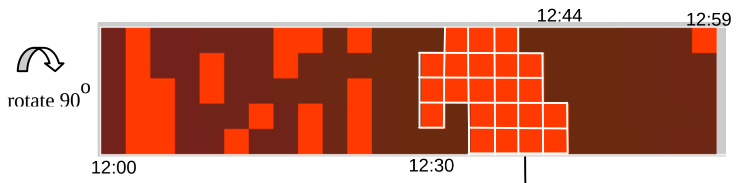


Fig. 8C: Combine the small marked areas

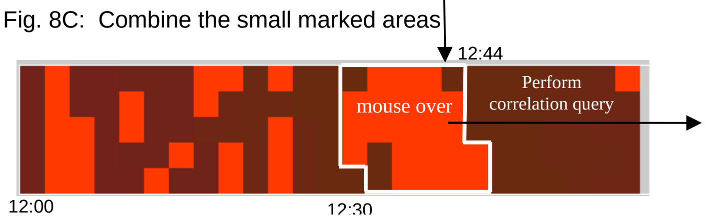


Fig. 8D: Mouse over to perform visual correlation

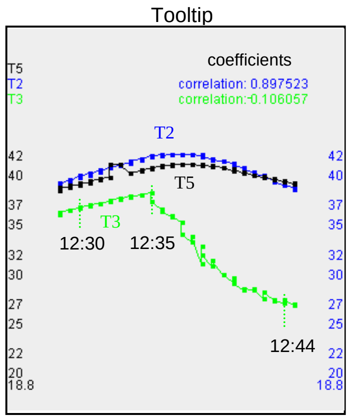


Fig. 8E: Correlation Mining Results
Showing:

- Sensor temperatures T5 and T2 in the marked area change at the same pace.
- T2>T5 (temperature out of sequence)
- T5 has a negative correlation with T3.
- T5's high temperature could be impacted by T2.

Fig. 8: Visual analytics of rack temperature anomalies in a data center

6. CONCLUSION

In this paper, we present an automated data analytics method, AnomalyMarker, to detect outliers or user-defined thresholds in large multi-dimensional data streams. This is a new concept that provides automated data detection methods and built-in data mining mechanisms to intelligently combine interactive visualizations with correlation analytic techniques. Depending on the applications and the size of the data stream, the anomaly combining algorithms optimize the size of the marked areas and return the most relevant markers to the analyst. This framework can easily be extended by additional correlation and similarity metrics. The data points in the marked area are interactive for users to zoom-in on information to reveal the relationship across different attributes. We apply this technique to mine data relationships and root-causes of enterprise server performance and data center sensor temperatures. From the recent feedback of system administrators, this technique can be used to monitor and detect the anomalies of incoming data streams and to observe the initial and ending conditions, correlations, and persistency. Future research will include the automated selection of the best suited threshold and anomaly persistency for a given data stream.

REFERENCES

[1] Eick, S., Steffen J. L., Summer Jr. E. E., Seesoft-A Tool for Visualizing Line Oriented Software Statistics. IEEE Transactions on Software Engineering, 18(11): 957 - 968, 1992.

[2] Eick, S., Karr, A., Visual Scalability. IEEE Transactions on Visualization and Computer Graphics, 6(1):44-58, 2000.

[3] Hocheiser, H., Schneiderman, B., Dynamic Query Tools for Time Series Data Sets, Timebox Widgets for Interactive Exploration, Information Visualization 2004.

[4] Buono, P., Aleks, A., Shneiderman B., Plaisant C., Khella A., Interactive Pattern Search in Time Series. Proc. of Visual Data Analysis Conference VDA05. San Jose, CA, 2005.

[5] Buono, P., Plaisant, C., Simeone, A., Schneiderman, B., Shmueli, G., Jank, W., Similarity-Based Forecasting with Simultaneous Previews: A River Plot Interface for Time Series Forecasting. Proc. 11th International Conference on Information Visualization. Zurich, Switzerland, 2007.

[6] Yost, B., North, C. The Perceptual Scalability of Visualization. IEEE Transactions on Visualization and Computer Graphics, 12(5):837-844, 2006.

[7] Hao, M., Dayal, U., Keim, D. A., Schreck T. S., Multi-Resolution Techniques for Visual Exploration of Large Time-Series Data. *Proc. of Eurographics/IEEE-VGTC Symposium on Visualization,* pp. 27-34, 2007.

[8] Hao, M., Dayal, U., Keim, D. A., Morent, D., Intelligent Visual Analytics Queries. IEEE Symposium on Visual Analytics Science and Technology, pp. 91-98, 2007.

[9] Keim, D. A., Kriegel, H. P., Ankerst, M. A.. Recursive pattern: A Technique for Visualizing Very Large Amounts of Data. Proc. IEEE Visualization, pp. 279-286, 1995.

[10] Lam, H., Munzner, T., Kincaid, R. Overview Use in Multiple Visual Information Resolution Interfaces. IEEE Transactions on Visualization and Computer Graphics. Vol. 13, No. 6, November/December, 2007.

[11] Hao, M., Keim D. A., Dayal, U., Oelke, D., Tremblay, C. Density Displays for Data Stream Monitoring. *Proc. of Eurographics/IEEE-VGTC Symposium on Visualization,* pp. 27-34, 2008.

[12] Hao, M., Keim D. A., Dayal. Large Multi-Attribute Time Series Data Visual Analytics Techniques for Large Multi-Attribute Time Series Data. Visualization and Data Analysis conference, 2008.