Visualizing Frequent Patterns in Large Multivariate Time Series

M. Hao[1], M. Marwah[1], H. Janetzko[2], R. Sharma[1], D. A. Keim[2], U. Dayal[1], D. Patnaik[3], N. Ramakrishna[3]
[1]Hewlett Packard Labs, USA, [2]University of Konstanz, Germany [3]Virginia Tech, USA

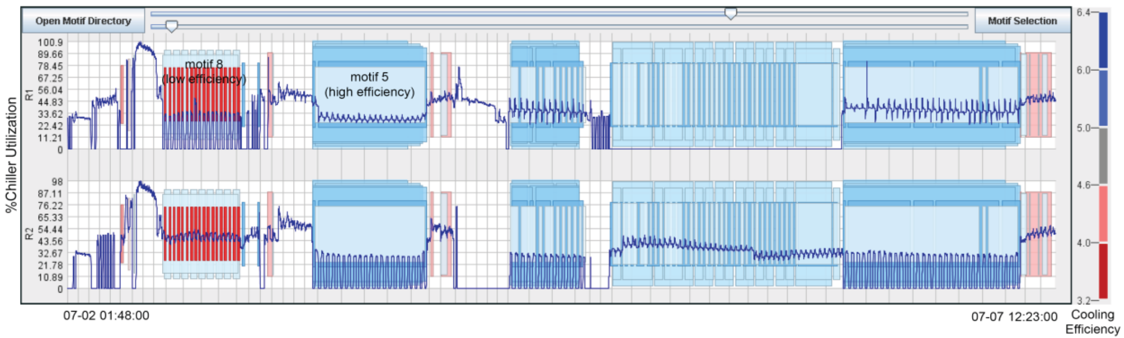


Figure 1: Frequent Patterns (Motifs) Discovered in Data Center Chiller Time Series
(x-axis: time in 1-minute intervals, y-axis: %utilization of chillers R1 and R2, color: chiller's cooling efficiency)
Motifs are represented by rectangles of a different size.
The height of a motif is proportional to the average duration time of all occurrences of the same motif.

ABSTRACT

The detection of previously unknown, frequently occurring patterns in time series, often called motifs, has been recognized as an important task. However, it is difficult to discover and visualize these motifs as their numbers increase, especially in large multivariate time series. To find frequent motifs, we use several temporal data mining and event encoding techniques to cluster and convert a multivariate time series to a sequence of events. Then we quantify the efficiency of the discovered motifs by linking them with a performance metric. To visualize frequent patterns in a large time series with potentially hundreds of nested motifs on a single display, we introduce three novel visual analytics methods: (1) **motif layout**, using colored rectangles for visualizing the occurrences and hierarchical relationships of motifs in a multivariate time series, (2) **motif distortion,** for enlarging or shrinking motifs as appropriate for easy analysis and (3) **motif merging,** to combine a number of identical adjacent motif instances without cluttering the display. Analysts can interactively optimize the degree of distortion and merging to get the best possible view. A specific motif (e.g., the most efficient or least efficient motif) can be quickly detected from a large time series for further investigation. We have applied these methods to two real-world data sets: data center cooling and oil well production. The results provide important new insights into the recurring patterns.

Keywords: recurring patterns, time series, multivariate data, motifs, distortion, merging

1. INTRODUCTION

1.1 Motivation

Many time series contain sequences of frequent patterns, often called motifs. Motif discovery is used to reveal trends, relationships, anomalies, and assist users in hypothesis evaluation and knowledge discovery. Efficient algorithms for detecting motifs in time series data [4] have been used in many applications, such as identifying words in different languages, detecting anomalies in patients' medical records over time [5], and chiller efficiency in data centers [14].

Figure 1 shows an example of the visual analysis of a pair of data center chiller time series in which different motifs were discovered. A chiller is a key component of the cooling infrastructure of a data center [15, 16, 3, 17]. The cooling efficiency of a chiller unit, also called its coefficient of performance (COP), indicates how efficiently the unit provides cooling and is defined as the ratio between the cooling provided and the power consumed. Motifs are a sequence of frequently occurring patterns (represented by rectangles). Each motif is specified in terms of its starting and ending times, marked here on the original chiller utilization time series. Motifs can be of varying lengths, with many shorter motifs nested within longer motifs, as a consequence of the level-wise motif mining algorithm [14]. Motifs are colored according to how efficiently the chiller ensemble performs within the motif. As a result, from motifs like the ones shown in Figure 1, service managers can examine historical behavior to draw inferences and gain new insights regarding the efficiency of different operational configurations. Specifically, key questions include:

- Are there repeating patterns of different types of motifs? How do they relate to a cooling efficiency metric (Coefficient of Performance [6])?
- How does the chiller ensemble transition over time from a low efficiency motif to a high efficiency one?

The service manager can quickly answer the first question using the cooling efficiency color map. The motifs in darker shades of blue are the most efficient. The second question requires further investigation as detailed in Section 4.1. The visual analysis allows data center service managers to avoid the operational configurations associated with motifs that have a low cooling efficiency. A large time series data set can contain hundreds or even thousands of motifs. Analyzing and visualizing these motifs involve several challenges:

- Displaying a large number of potentially overlapping motifs associated with multivariate time series.
- Searching and retrieving the most efficient motifs.
- Analyzing both the motifs and the context around the motifs for root-cause analysis.

1.2 Related Work

A common method to visualize time series patterns is to use line charts. Line charts are widely used and are intuitive and easy to understand. But if the data set contains many time series with a large number of observations and many repeated patterns, the time series will have a high degree of overlap, which obscures important information. Buono [2] provided the ability to interactively search patterns in multivariate time series by pre-selecting an interesting pattern. Munzner's LiveRAC [12] supports the analysis of large system management time series with a visual comparison of devices and parameters. In work by Hao et al. [7], the problem of visualizing large time series is addressed by pixel cell-based high density displays.

Motif mining is the task of finding approximately repeated subsequences in multivariate time series, which is studied in various works, e.g., [16, 8, 10]. Mining motifs in symbolized representations of time series can be found in the rich body of literature in bioinformatics, where motifs have been used to characterize regulatory regions in the genome. As the work closest to ours, we explicitly focus on the SAX representation [9], which also provides some significant advantages for mining motifs. First, a random projection algorithm is used to hash segments of the original time series into a map. If two segments are hashed into the same bucket, they are considered as candidate motifs. In a refinement step all candidate motif subsequences are compared using a distance metric to find the set of motifs with the highest number of non-trivial matches. A contrasting framework, referred to as the frequent episode discovery, is an event-based framework that is also applicable to symbolic data which is non-uniformly sampled. This enables the introduction of junk or "don't care" states into the definition of what constitutes a frequent episode.

To visualize motifs, Lin's VizTree [11] transforms a large time series into a symbolic representation, encoding the data into a tree with branches to represent symbols and motifs. The frequency of a motif is encoded in the thickness of a group of branches. Lin employs both tree and line charts to link different pieces of information. To understand a motif, VizTree requires user domain knowledge and interactions on the tree. To simplify the motif analysis process, Ordinez [13] adds radial representations to their line charts for further analyzing the relationships among their patients' medical records over time.

All the above techniques have contributed innovative visualization solutions emphasizing the finding of motifs and transforming large volumes of data into valuable information. However, analysts want to have an overview of repeated patterns and transitions between those patterns within a single view. In addition, they want to search for a motif that is the most or least efficient one based on a performance metric, e.g., the chiller utilization metric for a data center cooling infrastructure, or an oil well production metric for oil well data.

1.3 Our Contribution

For analyzing frequent patterns in large time series, we derive three new techniques: (1) motif discovery and layout, using colored rectangles for visualizing the occurrences and hierarchical relationships of motifs in a multivariate time series, (2) motif distortion which enlarges either motif or non-motif areas to allow the analyst to focus on the content and the structure of the areas and (3) motif merging which allows analysts to combine repeated motifs into a single area for data reduction and visual un-cluttering. In order to quickly identify the most efficient motifs from a large time series, each motif is linked to its performance coefficient (COP) [6] for quick retrieval of information as needed.

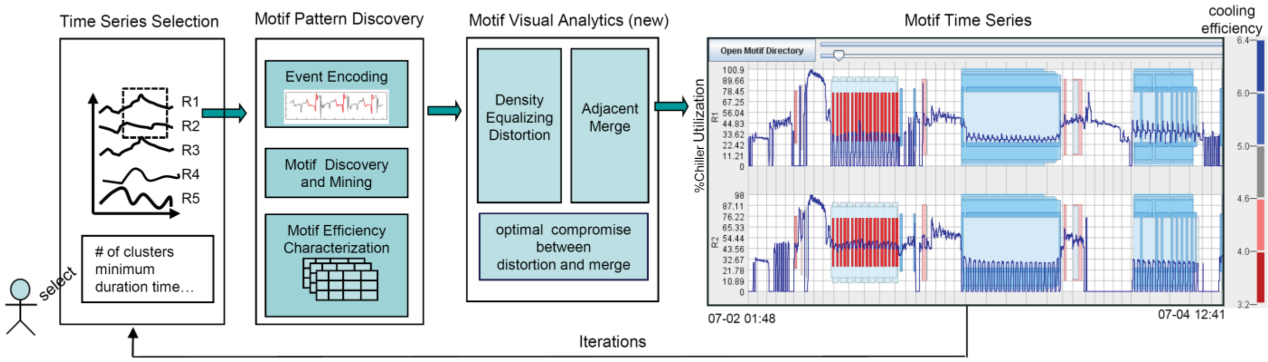


Figure 2: Visual Frequent Pattern (Motif) Analysis Pipeline

We have combined the above visual analytics techniques to provide a better understanding of the results of the motif mining algorithms, allowing the service managers to explore the big picture, namely the sequence of motifs and their behaviors, including their dependency on other attributes such as the cooling efficiency in a data center. Our motif discovery and data mining approach provides both qualitative and quantitative characterizations of the time series. Finally, we evaluate these techniques with respect to two real-world applications: data center chiller utilization and oil well flow production.

The paper is structured as follows: In section 2, we introduce a visual pattern analysis pipeline and describe the main stages used to discover motifs in a large multivariate time series. In section 3, we present the construction of visual motif layouts and our new visualization techniques. Section 4 describes two applications in which real-world data is used. An evaluation of the effectiveness of our techniques are presented in section 5. Section 6 contains the conclusions and future work.

2. PATTERN FINDING IN LARGE MUTIVARIATE TIME SERIES

A schematic overview of our approach is given in Figure 2 which shows an example of how to monitor chiller efficiency in data centers using a pair of data center chiller time series in which different motifs were discovered. The illustrated process can be subdivided into three phases: (1) the input selection phase to select the multivariate time series and parameters, (2) the motif pattern discovery phase to map a multivariate time series to the characteristics of the frequent patterns (motifs), in the form of motif start/end times, which allows the motif's efficiency (COP) to be computed, and (3) the motif visual analytics phase to lay-out the discovered motifs into the same multivariate time series. With our new motif distortion and merging techniques, users are able to visualize the relationships and efficiencies of the motifs. As we will show, a combination of visual and motif analysis is the key to finding trends and anomalies in the time series.

Motif pattern finding techniques have previously been described in [14]. Our primary goal is to link the multivariate, numeric, time series data to high level efficiency characterizations. We decompose this goal into symbolic representation, event encoding, motif mining, and efficiency characterization, thus using motifs as a crucial intermediate representation to aid in data pattern analysis and reduction. The following are the main stages involved in discovering frequent motifs:

Event encoding. We are given a multivariate time series $T = \langle t_1,\ldots,t_m \rangle$ where each real-valued vector t_i captures the utilization values of an ensemble of chillers. We first perform k-means clustering on the multivariate time series considering each time point as a vector and use the cluster labels as symbols to encode the time series. The number of clusters can be appropriately chosen; in this particular instance we found 20 clusters works particularly well [14]. Observe that the multivariate series is now encoded as a single (one-dimensional) symbol sequence [14]. Essentially, we have stripped off the temporal information, clustered the data, and put the temporal information back, thus "redescribing" the data. The resulting sequence of cluster labels is analyzed to detect change points. Change point detection transduces the symbol stream into a sequence of events where an event is defined as a transition in the cluster label.

Motif discovery and mining. *Frequent* episode mining is conducted on the transition event stream to detect repetitive motifs. The framework of serial episodes with inter-event time constraints is used. The structure of a serial episode α is given by:

$$\alpha = \left\langle E_1 \xrightarrow{(0,d_1]} E_2 \ldots \xrightarrow{(0,d_{n-1}]} E_n \right\rangle$$

$E_1, \ldots, E_n$ are the transition events characterized by a pair of cluster IDs participating in the transitions. Each pair of event-types in α is associated with an inter-event constraint which specifies the maximum allowed time gap between them. The mining process follows a level-wise procedure similar to the *Apriori* [1] algorithm, that is, candidate generation followed by counting. As shown in Figure 3, the episode mining framework allows other “don’t care” events to occur between any pair of events in an episode. In our application this helps accommodate spurious or noise transitions.

The frequency measure for an episode is based on non-overlapped counting. Two occurrences i.e. sets of transition events corresponding to a motif are said to be non-overlapped if they do not share any portion of the time series. This seems most applicable in the context of time series data. It also simplifies the counting procedure as one does not have to track more than one occurrence of a particular motif at any given time. Figure 3 below gives an example of two non-overlapped occurrences of the motif B→A→B→A. The counting algorithm resets all its internal states after completing each occurrence of the motif.

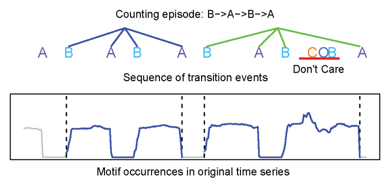


Figure 3: Example of Non-Overlapping Counting

Efficiency characterization. Finally, each motif is characterized in terms of an efficient metric. It is difficult (and subjective) to compare two motifs in terms of their efficiency by inspecting them visually. Therefore, it is necessary to quantify the efficiency of all motifs by computing a suitable metric for each of them. This enables efficiency comparisons between motifs: their categorization as 'good' or 'bad' from the efficiency metric point-of-view. Furthermore, this information helps provide guidance to an administrator or a management system regarding the most 'efficient' configurations.

In general, we use the above methods to map a multivariate time series to frequent patterns. Now the challenge is to translate these discovered patterns back to the original time series for users to continue to analyze the patterns and their behaviors. This gap requires visualization to map the discovered motifs back to the time series.

3. MOTIF PATTERN VISUALIZATION

3.1 Motif Layout

After applying the above mentioned methodology, we present the discovered motifs in a single display. For nested motifs, it is often difficult to recognize their starting and ending time; a long duration motif can contain several short duration motifs or can overlap other motifs. To overcome these difficulties, we derive a new layout algorithm (Algorithm 1) and draw rectangles to represent the occurrence of motifs. The color of a rectangle represents its numerical importance or – if no such metric exists as in most examples in this paper – different colors are used to distinguish different patterns. The nested rectangles are used for visualizing the hierarchical relationships among motifs. The rectangle’s height is linearly proportional to the statistical rank of the average duration of a motif. The statistical rank is used to distinguish motifs with nearly the same height.

Algorithm 1

```
Algorithm 1: Layout and draw motifs
Input: Array of motifs: Motif [ ] allMotifs

// Draw all occurrences of each motif
forEach Motif m in all motifs sorted by average occurrence length do
  int heightOfMotif = scaled statistical rank of motif m according to
                      the height of the line chart
  forEach TimeInterval t in occurrences of motif m do
    // the method calcXCoords determines the (possibly distorted)
    // x-coordinate of a given timestamp
    double startX = calcXCoords(t.startTime);
    double endX = calcXCoords(t.endTime);
        setColor(m.motifColor);
    setBorderColor( according to selection property );

    // draw rectangles vertically centered
    paintRectangle(startX,
      heightOfLineChart / 4 - heightOfMotif.get(m) / 2,
      endX - startX,
      heightOfLineChart / 2 + heightOfMotif.get(m));
  end
end
```

Figure 4 shows 11 consecutive occurrences of motif 5. Each motif is represented by a blue rectangle. Service managers can mouse over an embedded motif to find the detailed information. Motif 5 is on the lowest level and is nested in the other motifs. The Each motif 2 contains two occurrences of motif 3, two of motif 4, and three of motif 5. Motif 4 overlaps motifs 2, 3, and 5. To analyze this behavior, service managers can enlarge the motif areas to analyze their structures and their nesting relationships.

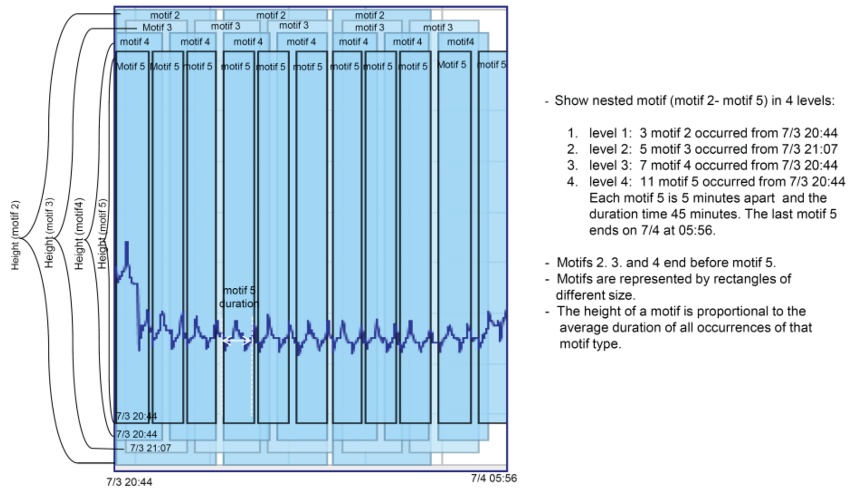


Figure 4: Motif Visual Layout
Each rectangle represents an occurrence of a motif

Visualizing the properties and behaviors of motifs in a massive multivariate time series is a complex task because of the large number of motifs (hundreds or even thousands) and the fact that they may be nested and overlapping. We introduce two new techniques, motif distortion and motif merging, to enable analysts to perform the following tasks:

- Explore motifs and their structure.
- Find the root-cause of a low efficiency motif by analyzing a sequence of transition events in a time series before the low efficiency motif occurred.

3.2 Motif Distortion

Distortion enlarges either areas with motifs or areas without motifs using a user-activated slider. In Figure 5, distorting the time series is done by applying a specific density-equalizing distortion technique. We calculate weights for each time interval and use them as the input to the distortion algorithm. These weights are based on the number of motifs occurring at that time. In a preprocessing step, we calculate the weights for both motif areas and non-motif areas within each time interval. To enlarge the motifs, we use the number of motifs. To enlarge areas without motifs, we use the inverse of the number of motifs in the time interval. If there are no motifs we use a constant weight of 1. The calculation of weights for enlarging motifs and enlarging non-motif areas is depicted in Algorithm 2. Figure 6 shows how the distortion algorithm works. Our technique tries to enlarge or shrink areas according to the weights. In Figure 5, our technique divides the time series into equal size parts and resizes each part according to the aggregated weight of the part.

Algorithm 2

```
Algorithm2: Build array with importance values
Input: Array of motifs: Motif [ ] allMotifs
Output: Arrays of importance values:
    weightsMotifs // used for enlarging motifs
    weightsNotMotifs // used for enlarging areas without motifs

weightsMotifs = new double[ number of timestamps ];
weightsNotMotifs = new double[ number of timestamps];
forEach Motif m : motifs do
  TimeInterval[] intervals = m.m_occurrences;
  forEach TimeInterval t : intervals do
    for i = t.startTime to t.endTime do
      weightsMotifs[ i ] += 1.0;
    end
  end
end

for i = 0 to number of timestamps do
  if weightsMotifs[ i ] > 0 then
    weightsNotMotifsArea[ i ]= 1 / weightsMotifs [i];
  else
    weightsNotMotifsArea[ i ] = 1.0;
  end
end
```

We first calculate a fully distorted view for each task (enlarging motifs or enlarging areas without motifs) and then calculate the zero slider position. When the user moves the slider to the left, areas without motifs are enlarged, the slider's middle position is its origin scale, and when the user moves the slider to the right, motifs are enlarged. For determining the distortion for the intermediate positions of the slider, we use a weighted interpolation between the original scale and the fully distorted view.

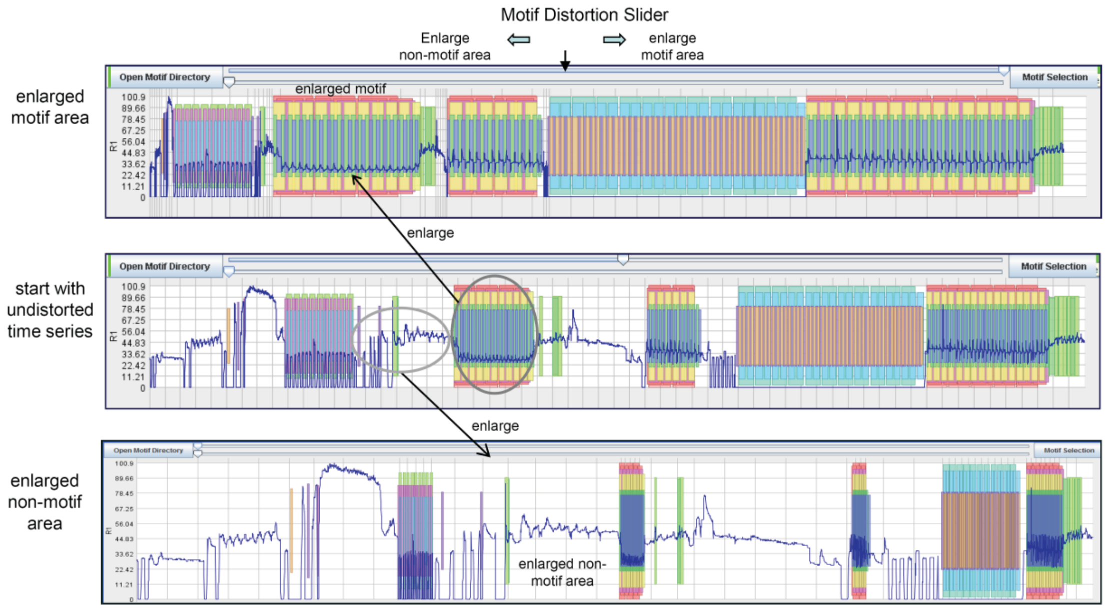


Figure 5: Motif Visual Distortion
(x-axis: time intervals, y-axis: %utilization of chiller R1, rectangles: motifs, color: motif types)
Move the distortion slider to the right to enlarge motifs. Move the distortion slider to the left to enlarge the non-motif area.

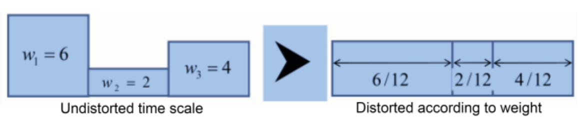


Figure 6: Distorting the Time Scale According to Given Weights

3.3 Motif Merging

In Figure 7, we provide a second slider to merge multiple occurrences of motifs to a single rectangle to reduce the data and the visual clutter. If the slider is moved, motifs of the same type that begin or end at adjacent positions are combined. We define two occurrences of the same motif as adjacent if the time duration between those occurrences does not exceed a given threshold. The threshold is set by the user via a slider. The value is measured in minutes and ranges from zero minutes to a calculated upper bound. For each motif, we compute the minimum gap length between its occurrences and average values over all instances of the motif. Note that only the same types of motifs are merged. Users can mouse over the time series in a merged motif to display the current time interval and the efficiency measure value.

After applying various degrees of distortion and merging, the motif time series greatly simplified for further visual analysis.

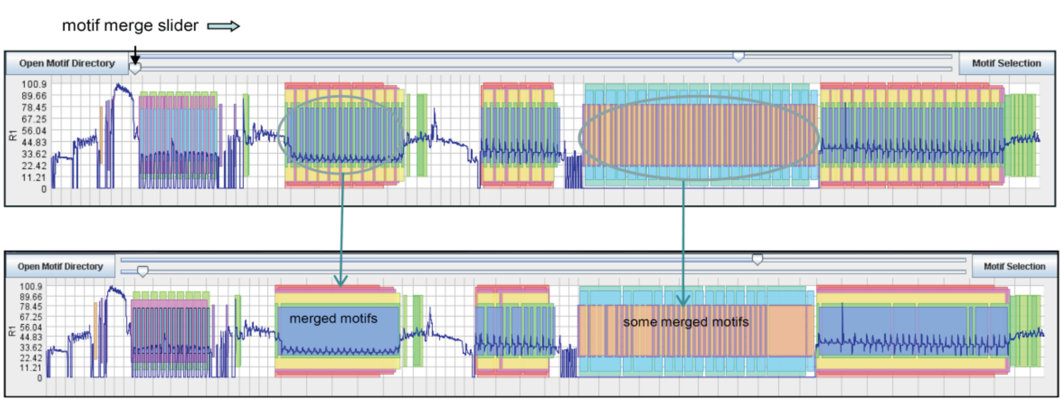


Figure 7: Motif Visual Merging
(x-axis: time intervals, y-axis: %utilization of chiller R1. rectangles: motifs, color: motif type)
Move the slider to the right to merge adjacent motifs of the same type.

4. APLICATIONS

Motif visual analysis has a large number of applications, including anomaly detection, prediction, and clustering. We will demonstrate the above techniques with data center chiller sensor time series and oil well production sensor data (e.g., oil flow, pressure). The identified motifs help the users to visualize cooling/oil production efficiency quickly. Most importantly, service managers are enabled to avoid the inefficient patterns and guide the operations towards more efficient ones.

4.1 Data Center Cooling Monitoring

Finding low efficiency motifs

The motif time series in Figure 8 show the utilization of four chillers (R1-R4) with 13,578 records at 1-minute intervals. The color shows the motif efficiency computed from the cooling efficiency metric. Service managers can quickly identify that motif 5 is more efficient than the other motifs. Furthermore, service managers are able to interact with the other motifs to analyze the characteristics of these motifs. For example, in motif 5, chiller R2 runs at medium utilization, while chiller R4 runs at high utilization. In motif 8, chiller R1 operates in a low utilization. By visual observation, we can see that the utilization of chiller R4 in motif 5 is the highest.

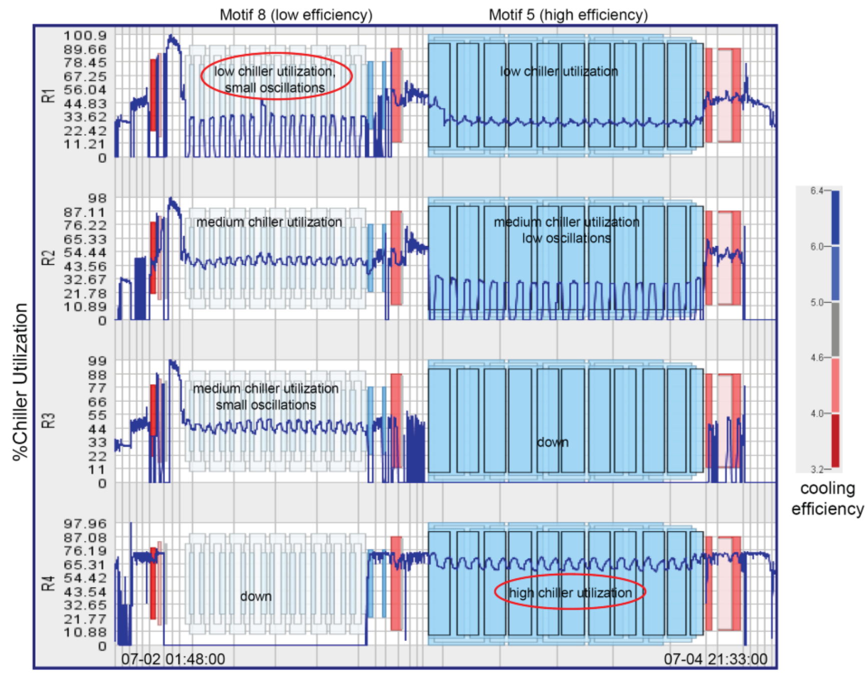


Figure 8: Motifs 5 and 8 are Enlarged to Compare their Chiller Utilization
Motif 5 is more efficient than motif 8. Motif 8's chillers R1 and R3 have some oscillatory behavior.
(x-axis: 07-02 01:48 to 07-04 21:33, y-axis: %utilization of chillers R1-R4, color: cooling efficiency).

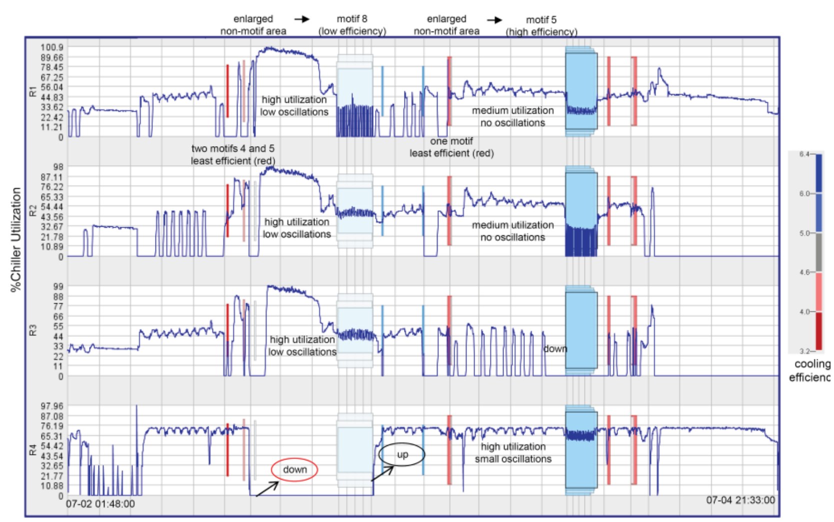
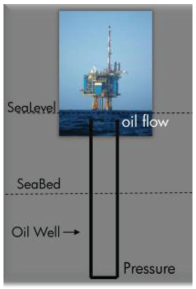


Figure 9: Enlarge the non-motif area to observe the transitions
from low cooling efficiency motif 8 to high cooling efficiency motif 5.
(x-axis: time line from 07-02 01:48 to 07-04 21:33, y-axis: %utilization, color: cooling efficiency).

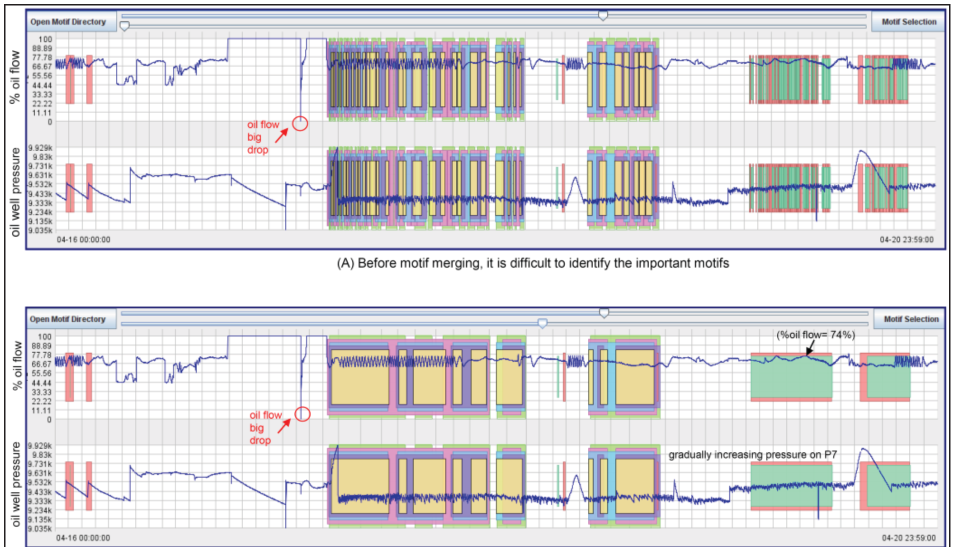
Transition from low efficiency motif to a high efficiency motif

Data center service managers want to know how low efficiency motifs transition to high efficiency motifs in a given time period. In order to examine the changes over a period of time, the service manager can move the distortion slider (shown in Figure 5) to the left to shrink the motif areas and enlarge the non-motif time intervals as shown in Figure 9.

From the behavior of these non-motif time intervals, the data center service managers are able to make the following observations:



Oil Well Production Operation



(A) Before motif merging, it is difficult to identify the important motifs

(B) After motif merging, users are able to focus on important motifs to analyze their behavior.

Figure 10: Oil Well Production Time Series with 7 Different Frequent Patterns with Distinct Colors
(x-axis: time, y-axis: % oil flow and pressure, color: each motif has a distinct color)

- Water chiller R4 is turned off before motif 8.
- Water chiller R4 starts up before the start of Motif 5, contributing to its improved cooling efficiency.

In addition, our techniques allow data center service managers to investigate other motif characteristics and make comparisons with other attributes, such as power and temperatures in order to gauge their impact on the cooling efficiency.

4.2 Oil Well Production Motif Observations

The picture on the left of Figure10 is a typical oil well which is used to produce daily oil supplies. Figure 10 shows a real-world oil well production time series (85,035 records) with different frequent patterns (motifs) identified by seven distinct colors. The most critical problem in the oil industry is to reduce the non-productive time. The common questions are:

- Which oil well flow pattern is the most productive?
- What transitions occur after a big drop in oil well flow? How can this be recovered from?

Figure 10 illustrates two different motif visualization methods: one uses only motif distortion as shown in 10(A); the other uses a combination of both distortion and merge as shown in Figure 10(B). As an evaluation result, the use of a combination of distortion and merge makes the motif visual analytics most effective.
From Figure 10(B), the production manager can see that the green motif is the most productive with an oil flow of up to 74%. Also, the production manager can determine that after a big drop in oil flow it is best to gradually increase the pressure as shown in the green motifs.

5. EVALUTION

5.1 Evaluation from Data Center Cooling Monitoring

Our new motif finding, distortion, and merging visualization techniques have been successfully used on several data centers of different sizes, ranging from 3,000 to 14,000 sq. ft. and containing hundreds of racks. Several billion records from data centers have been analyzed.

Figure 11 is a regular time series to which data center operators have access. Operators can analyze the time series and observe the variation of utilization over time. Such interfaces are available today in building management systems. However, operators do not know which set of patterns represents an efficient mode of operation, nor do they understand whether such a pattern has occurred in the past or not. Usually, such operational patterns are characteristic of a mismatch between chiller operation and demand variation. Not all chillers can scale uniformly in capacity with a rise in demand. Also demand does not uniformly change over time. But, this kind of monitoring is essential in building efficient management systems.

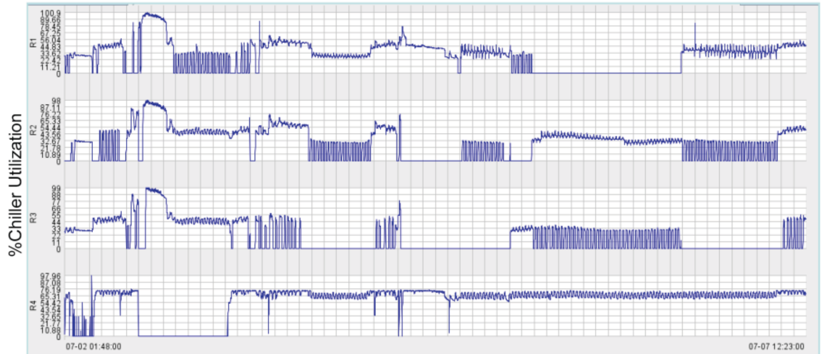


Figure 11: Data Center Chiller (R1-R4) % utilization regular time series without motif

The motif time series as shown in Figure 8 helps service managers identify motifs and their cooling efficiencies and provides guidance on how the current performance compares with the past. Our new techniques can assist service managers to move the chiller system to a more efficient state.

Analyzing transitions between different motifs of different cooling efficiencies is an important capability for administrators and helps them improve chiller performance. Figure 9 shows how service managers can look at such transitions between states from the past using the distortion and merging sliders. Using existing regular time series (Figure 11) it can take days, while using motifs (Figures 8, 9, and 10), it can be done in minutes.

5.2 Evaluation from oil well production observation

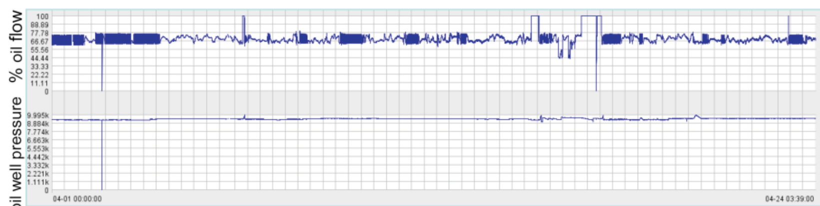


Figure 12: Oil well production regular time series without motif

Oil well pressure and flow are normally strongly correlated. However, variations do occur due to well-head problems or geological issues. The variations can be complicated and depend on the geology of the oil well and its composition. Identification of motifs in oil well pressure and flow can help in classification of such issues. Finding the motifs which are able to maximize throughput (or oil flow) at the lowest pressure is the goal of the well operator. Without our motif layout, it is almost impossible for administrators to find these frequent patterns as shown in Figure 12. Using motifs, as illustrated in Figure 10, the service managers can quickly find the most efficient motifs. Furthermore, service manager can reduce the motifs which cause fluctuations in pressure (or flow). The motifs with high oscillations can be detrimental to well operation and lead to reliability issues.

5.3 An informal user study

From an informal user study (12 users) using both regular time series (Figures 11 and 12) with and without motifs, we are getting at least an 87% times savings by applying motif layout, distortion, and merging techniques. No user can identify all the repeated patterns (motifs). Most of the users can only find about a couple of frequently repeating patterns.

6. CONCLUSION

Finding frequently occurring patterns and analyzing them allows data center and oil well service managers to determine which configurations are more efficient and which ones result in poor efficiency so the latter can be avoided. In this paper we address the whole visual analysis pipeline for motifs. First, we briefly describe a novel motif discovery algorithm, which is based on cluster analysis, event encoding, frequent motif mining, and sustainability characterization of those motifs. Second, we introduce three new visualization and interaction techniques (motif layout, distortion and merge) for the analysis of motifs discovered from mining. We allow service managers to adjust the degree of distortion and merge to generate the best view on a single display. In addition, we link the motifs to the associated efficiency metrics for service managers to query the least/most efficient motifs for root-cause analysis. Our results from both the real-world data center and oil/gas production time series sensor data show that our techniques successfully enable users to identify both efficient and inefficient patterns. This demonstrates the wide applicability and usefulness of our techniques. In the future, we want to use discovered motifs to detect the current operational motif in real-time to predict near term behavior.

Acknowledgment: The authors wish to thank C. Patel and M. Hsu for their suggestions and encouragement, and Cullen Bash for providing comments and suggestions.

REFERENCES

[1] Agrawal, R, Srikant, R: Fast Algorithms for Mining Association Rules in Large Databases. VLDB 1994: 487-499

[2] Buono, P., Aris, A., Plaisant, C., Khella, A., Shneiderman, B., Hochheiser, H. Schneiderman, B. Interactive Pattern Search in Time Series. Proceedings of Conference on Visualizaion and Data Analysis, VDA 2005, SPIE. CA.

[3] C. Bash, C. Patel, R. Sharma, “Dynamic Thermal Management of Air Cooled Data Centers”, IEEE Itherm06.

[4] Bill Chiu, Eamonn Keogh, and Stefano Lonardi. Probabilistic discovery of time series motifs. In KDD '03: proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 493-498, New York, USA, 2003. ACM.

[5] Chen, L., Ozsu, T. Oria, V. Symbolic Representation and Retrieval of Moving Object Trajectories. University of Waterloo. 2003.

[6] Michael J. Moran and Howard N. Shapiro, Fundamentals of Engineering Thermodynamics, Wiley; 6 edition, 2007.

[7] Hao, M. Dayal, U. Keim, D. A., Schreck, T., Multi-Resolution Techniques for Visual Exploration of Large Time-Series Data. Proceedings: IEEE VGTC Symposium on Visualization, EuroVis 2007.

[8] Lin, J., Keogh, E., Lonardi, E., and Patel, P., Finding motifs in time series. In Proceedings of the Second Workshop on Temporal Data Mining, 2002.

[9] Lin, J., Keogh, E., Wei, L., and Lonardi, S. (2007) Experiencing SAX: a Novel Symbolic Representation of Time Series. DMKD Journal.

[10] Patel, P., Koeogh, E, Lin, J., and Lonardi, S. Mining motifs in massive time series database. In ICDM’02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDN’02), page 370, Washington, DC. 2002.

[11] J. Lin, E. Keogh, S. Lonardi, J. Lankford, D. Mystrom. VizTree: a Tool for Visually Mining and Monitoring Massive Time Series Database. Proceedings of the 30th VLDB Conference, Canada, p#.1260, 2004.

[12] McLachlan, P., Munnzner, T., Koutsofios, E. North S., LivePRAC: Interactive Visual Exploration of System Management Time-Series Data, CHI2008, 4/08 Florence, Italy.

[13] Ordonez, P., des Jardins, M., Feltes, C. et. al. Visualizing Multivariate Time Series data to Detect Specific Medical Conditions. AMIA Annual Symposium Proceeding. 2008. MD.

[14] Patnaik, D., Marwah, M. Sharma, R. Ramakrishnan, N. Sustainable Operation and Management of Data Center Chillers using Temporal Data Mining. In the proceedings of KDD’09, 6/09, France.

[15] R. Sharma, C. Bash, C. Patel, "Dimensionless parameters for evaluation of thermal design and performance of large-scale data centers", 8th ASME/AIAA Joint Thermophysics and Heat Transfer Conf., St. Louis, MO, June 2002.

[16] Sharma, R. et al. On building next generation data centers: Energy flow in the information technology stack, in Proceedings of Compute 2008.

[17] R. Sharma et al., “Energy Flow in the Information Technology Stack”, ACM Compute 2008.