Visual Exploration of Frequent Patterns in Multivariate Time Series

Ming C. Hao[1], Manish Marwah[1], Halldór Janetzko[2], Umeshwar Dayal[1], Daniel A. Keim[2] ,
Debprakash Patnaik[3], Naren Ramakrishnan[3] and Ratnesh K. Sharma[4]

[1] HP Labs, Palo Alto, CA, USA
[2] University of Konstanz, Konstanz, Germany
[3] Virginia Tech, USA
[4] NEC Laboratories America, Inc., USA

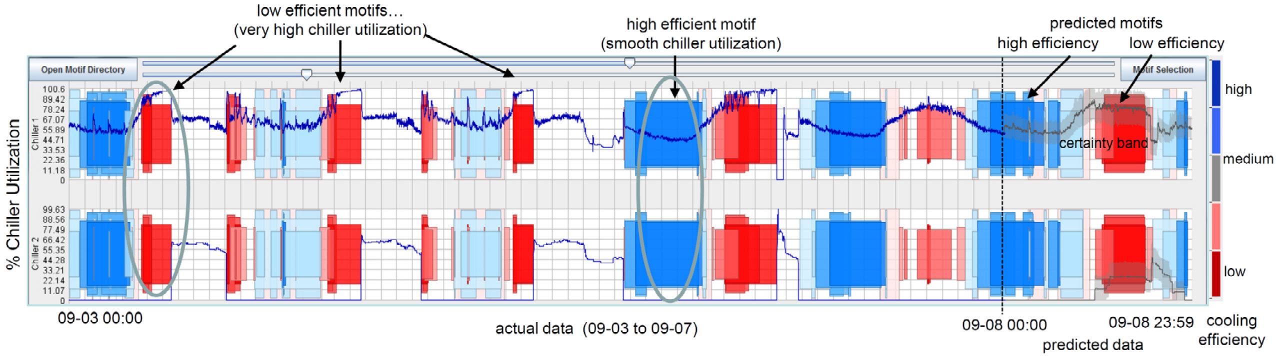


Figure 1: Exploring frequent patterns (motifs) in data center chiller utilization multivariate time series
(x-axis: time, y-axis: %utilization of two chillers, color: blue represents high cooling efficiency and red, low efficiency).

Figure 1 shows a data center chiller utilization multivariate time series line chart with actual and predicted data measured in 1-minute intervals. Frequently occurring patterns in the time series also known as motifs, which are represented by rectangles of different sizes. The height of a motif is proportional to its average duration. The color of a motif represents its cooling efficiency, which is the ratio between the cooling it provides and the power it consumes. Efficiency coded motifs allow service managers to compare chiller efficiency at different periods of time. Motifs discovered in the predicted data provide information about future chiller cooling efficiency. The certainty band shows the confidence of prediction. The key contribution is to discover and provide visual analytics of frequently occurring patterns for system monitoring and planning.

ABSTRACT

The detection of frequently occurring patterns, also called motifs, in data streams has been recognized as an important task. To find these motifs, we use an advanced event encoding and pattern discovery algorithm. Since a large time series can contain hundreds of motifs, there is a need to support interactive analysis and exploration. In addition, for certain applications, such as data center resource management, service managers want to be able to predict the next day's power consumption from the previous months' data. For this purpose, we introduce four novel visual analytics methods: (1) **motif layout**, using colored rectangles for visualizing the occurrences and hierarchical relationships of motifs; (2) **motif distortion,** enlarging or shrinking motifs for visualizing them more clearly; (3) **motif merging,** combining a number of identical adjacent motif instances to simplify the display; and (4) **pattern preserving prediction**, using a pattern preserving smoothing and prediction algorithm to provide a reliable prediction for seasonal data. We have applied these methods to three real-world data sets: data center chilling utilization, oil well production, and system resource utilization. The results enable service managers to interactively examine motifs and gain new insights into the recurring patterns to analyze system operations. Using the above methods, we have also predicted both power consumption and server utilization in data centers with a 70-80% accuracy.

Keywords: frequent patterns, multivariate time series, motifs, distortion, merging, seasonal data, and prediction.

1. Introduction

1.1 Motivation

Many time series contain sequences of frequently occurring patterns, often called motifs. Motif discovery is used to reveal trends, relationships, and 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 chillers (chiller 1 and chiller 2), a %utilization time series in which different motifs were discovered. A chiller is a key component of the cooling infrastructure of a data center [3, 16, 17]. The cooling efficiency of a chiller unit, also called its coefficient of performance (COP) [6], indicates how efficiently the unit provides cooling and is defined as the ratio between the cooling provided and the power consumed. In Figure 1, chiller 1 is the primary chiller. Chiller 2 is the secondary chiller, which is used only when the utilization of the primary chiller becomes high (close to 100%). Motifs are a sequence of frequently occurring patterns, depicted by rectangles. Each motif is specified in terms of its starting and ending times. 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.

In addition to discovering frequent patterns in the past data, users also want to predict future behavior. For example, data center service managers and system analysts want to predict the next day's chiller utilization from the past data. A retailer needs to predict the number of products to be stored in the warehouse this month using last year's sales data. In this paper, we also apply pattern-preserving methods to predict the next day's resource utilization, thus avoiding the risk of exceeding the provided resource capacities, which can lead to damage or unavailability of equipment.

Chiller operators can examine and explore the motifs discovered in the historical data (before September 7 in Figure 1). The motifs are color coded by their efficiency; the red ones are less efficient than the blue ones. Figure 1 has one day of predicted data, starting at 09-08 00:00 and ending at 23:59. The motifs in the predicted period inform the operator of the future efficiency of the system. When low efficiency motifs (red) are predicted, the service manager could make suitable configuration changes, if possible to transform the operation to more efficient motifs. Furthermore, in this specific instance, the predicted time series indicates that chiller 2 would likely turn on during the time interval 11:06-18:08 to assist chiller 1.

In summary, visual exploration of motifs in multivariate time series has to overcome the following challenges:

- Displaying and predicting a large number of potentially overlapping motifs associated with multivariate time series,
- Searching and retrieving the most efficient motifs by efficiency, and
- Visually 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 LivePRAC [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, Ordonez [13] adds radial representations to their line charts for further analyzing the relationships among their 15 patients' medical records over time.

Holt [15] and Winters [22] both used exponentially weighted moving averages to forecast seasonal sales data. Their forecast is a function of past and current sales using exponential smoothing. Taylor [19] applied the Holt-Winters techniques to predict daily supermarket sales using exponentially weighted quantile regression for inventory control. Taylor extended the exponential smoothing based forecast to cumulative distributed function level forecast for better prediction. We apply Holt-Winters algorithms to predict the next day's chiller utilization for the data center. The results from Holt-Winters are very close to our prediction results, but peaks are missing from their prediction. Ichikawa [20] introduced a visualization environment which allows users to view a large number of stock price predictions using different types of line charts, texture, color, and 3D graphs. Masse [21] proposed a visual approach for the U.S. presidential election prediction.

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 the transitions between those patterns in a single view. In addition, they want to identify a motif as the most or least efficient one using a performance metric, e.g., the chiller cooling efficiency metric for a data center or an oil well production metric for oil well data. For data center and resource utilization seasonal data, we would like to inform the analysts how many system resources are needed for the next day.

1.3 Our Contribution

For analyzing frequent patterns in large time series, we derive four 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, (3) motif merging, which allows analysts to combine repeated motifs into a single area for data reduction and visual uncluttering, and (4) motif seasonal data prediction using pattern-preserving prediction algorithms that service managers can use for resource planning. In order to quickly identify the most efficient motifs from a large time series, each motif is linked to its performance coefficient for quick retrieval of information as needed.

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 three real-world applications: data center chiller utilization, oil well flow production, and system resource utilization prediction.

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, our new visualization techniques, and pattern preserving prediction. Section 4 describes three applications and evaluations in which real-world data is used. Section 5 contains the conclusions and future work.

2. Pattern Finding in Large Multivariate Time Series

A schematic overview of our approach is provided in Figure 2. The illustrated process can be subdivided into three phases: (1) motif pattern discovery phase, where motifs are discovered in a multivariate time series and characterized in terms of an efficiency metric; (2) the motif visual analytics phase, to layout the discovered motifs in the same multivariate time series; and (3) the visual prediction, to visualize the predictions for the next day's data with preserved patterns. With the 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.

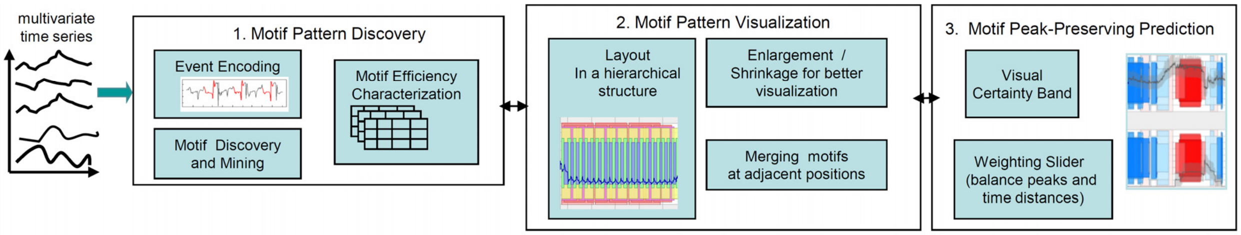

multivariate time series
1. Motif Pattern Discovery
Event Encoding
Motif Discovery and Mining
Motif Efficiency Characterization
2. Motif Pattern Visualization
Layout In a hierarchical structure
Enlargement / Shrinkage for better visualization
Merging motifs at adjacent positions
3. Motif Peak-Preserving Prediction
Visual Certainty Band
Weighting Slider (balance peaks and time distances)

Figure 2: A schematic overview of our approach

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.

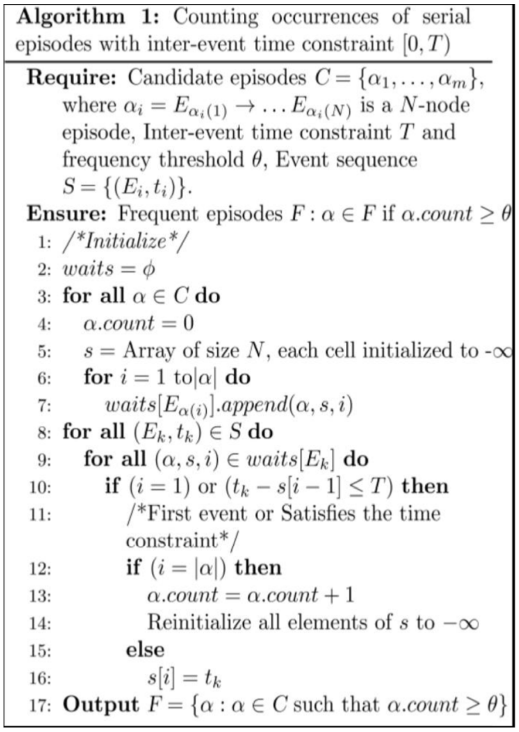
Definition (Time Series): A multivariate time series $T = \langle \vec{v}_1, \cdots, \vec{v}_m \rangle$ is an ordered set of real-valued vectors. Each real-valued vector $\vec{v}_i$ represents utilizations across all the chiller units.

Definition (Time Series segment): A segment of the time series T is an ordered subset of consecutive vectors of T denoted by $T[i,j] = \langle \vec{v}_i, \vec{v}_{i+1}, \cdots, \vec{v}_j \rangle$, where $1 \leq i < j \leq m$.

Definition (Motif): A motif represents a set of segments of the time-series T, $\{T[i_1, j_1], T[i_2, j_2], \cdots, T[i_n, j_n]\}$, where $1 \leq i_1 < j_1 \cdots \leq i_{k+1} < j_{k+1} \cdots j_n \leq m$, such that any pair of segments in the set satisfy a similarity requirement and $n \geq \theta$, where θ is a user-defined minimum count.

The following are the main stages involved in discovering frequent motifs:

Event encoding: We are given a multivariate time series T. First we 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, k, is an input to the k-means algorithm, and depends on the application. Choosing k is not yet automated; however, there are several techniques that can be used to determine k. such as using 1) domain knowledge; 2) average silhouette coefficients; 3) ratio of sum of square within and between cluster distances; and 4) information theoretic metrics.
The number of clusters can be appropriately chosen; in this particular instance we found 20 clusters provide a good trade-off between separation of clusters and size of individual clusters [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 "re-describing" the data. The resulting sequence of cluster labels is analyzed to detect change points. Change point detection maps the symbol stream into a sequence of events where an event is defined as a transition in the cluster label.



```
Algorithm 1: Counting occurrences of serial episodes with inter-event time constraint $[0, T)$
Require: Candidate episodes $C = \{\alpha_1, \ldots, \alpha_m\}$,
    where $\alpha_i = E_{\alpha_i(1)} \rightarrow \ldots E_{\alpha_i(N)}$ is a $N$-node
    episode, Inter-event time constraint $T$ and
    frequency threshold $\theta$, Event sequence
    $S = \{(E_i, t_i)\}$.
Ensure: Frequent episodes $F : \alpha \in F$ if $\alpha.count \geq \theta$
/*Initialize*/
$waits = \phi$
for all $\alpha \in C$ do
    $\alpha.count = 0$
    $s$ = Array of size $N$, each cell initialized to -$\infty$
    for $i = 1$ to$|\alpha|$ do
        $waits[E_{\alpha(i)}].append(\alpha, s, i)$
for all $(E_k, t_k) \in S$ do
    for all $(\alpha, s, i) \in waits[E_k]$ do
        if $(i = 1)$ or $(t_k - s[i-1] \leq T)$ then
            /*First event or Satisfies the time
            constraint*/
            if $(i = |\alpha|)$ then
                $\alpha.count = \alpha.count + 1$
                Reinitialize all elements of $s$ to $-\infty$
            else
                $s[i] = t_k$
Output $F = \{\alpha : \alpha \in C$ such that $\alpha.count \geq \theta\}$
```

Motif discovery and mining: *Frequent* episode mining is conducted on the transition event stream to detect repetitive motifs. A serial episode is defined as an ordered sequence of event-types. Event types in this case represent transitions from one cluster to another in the time series data. All frequent episodes occur in the data more than θ times in the data, where θ is a user-defined threshold. The mining process follows a level-wise procedure similar to the *Apriori* [1] algorithm, that is, candidate generation followed by counting. Frequent serial episodes are discovered iteratively, in ascending order of their size and it is often referred to as a level wise procedure. The procedure alternates between counting and candidate generation. By joining the frequent *(i-1)*-size episodes found in the previous iteration, a set of candidate *i*-size episodes is created. Then the event sequence is scanned for determining the frequency or count of each candidate episode and the frequent *i*-size episode are extracted from the candidates.

In this work we use the class of serial episodes with inter-event time constraints. The structure of a serial episode α is given by:

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

where $E_1, \cdots, E_n$ 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 that specifies the maximum allowed time gap between them.

The candidate generation scheme is based on matching the *(n-1)* size suffix of one *n*-node frequent episode with the *(n-1)* size prefix of another *n*-node frequent episode at a given level to generate candidates for the next level. For example the 4-size candidate episode $\langle A \rightarrow B \rightarrow C \rightarrow D\rangle$ is generated from 3-size frequent episodes $\langle A \rightarrow B \rightarrow C\rangle$ and $\langle B \rightarrow C \rightarrow D\rangle$. Here the prefix (B->C) in (B->C->D) matches the suffix (B->C) in (A->B->C). The time complexity of the candidate generation process is $O(m^2n)$, where n is the size of each frequent episode in the given level, and m is the number of frequent episodes on that level, since all pairs of frequent episodes need to be compared for a prefix-suffix match.

The algorithm for counting the set of candidate episodes is given in Algorithm 1. The frequency measure for an episode is based on non-overlapping counting. Two occurrences, i.e. sets of transition events corresponding to a motif, are said to be non-overlapping if they do not share any portion of the time series. In Algorithm 1 each candidate episode is assigned an array s of size same as that of the episode (Lines 2-7). This array is used to track event occurrences that constitute the episode and also satisfy the inter-event expiry constraints. The $waits(.)$ map is used to access a subset of episodes in $O(1)$ when an event common to each of these episodes arrives. In one pass of the data the counts of each of the episodes is determined as shown in Line 8-16. Each time a set of events that constitute an episode occurrence are found the count of the episode is incremented and the data structure tracking the event times is reset (Line 12-14). After all episode counts have been determined the procedure returns only those episodes that meet minimum frequency threshold θ.

Efficiency characterization: Finally, each motif is characterized in terms of efficiency 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 them. This enables efficiency comparisons between motifs: their categorization as 'good' or 'bad' from the efficiency metric point-of-view. Furthermore, this information helps to provide guidance to a service manager 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 which is based on the length of the occurrence of motifs. In order to be independent of small variations of their length we use the statistical rank of the average occurrence length and use this information to represent the occurrences of motifs with rectangles. The width of the rectangle is defined by the duration of the occurrence and the height is calculated using the following formula:

$$height(motif) = \frac{\text{nrOfMotifs} - \ rank(avg(motif.length))}{\text{nrOfMotifs} - 1}$$

As stated in the formula above the height of the rectangles of the same motif have the same height to ensure the visual identity of rectangles of the same motif. Additionally, we scale the result of the equation above to the interval from 0.5 to 1.0 making sure that all occurrences – even the ones with low ranks – are visible.

The color of a rectangle represents the efficiency of a motif – different colors are used to distinguish between different efficiency levels. The definition of efficiency is application-specific and is usually defined by the service manager. 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. Figure 2 shows 11 consecutive occurrences of motifs (blue rectangles) nested in two other types of motifs (yellow and pink rectangles). Visualizing the properties and behaviors of motifs in a massive multivariate time series is a complex task because there may be a 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 that contain motifs or areas that do not contain motifs using a user-activated slider. Distorting the time series is done by applying a specific density-equalizing distortion technique. We calculate weights for each time interval between two consecutive data points and use them as the input to the distortion algorithm.

These weights are based on the number of motifs occurring during that time interval. 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 in the time interval, we use a constant weight of 1. The calculation of weights for enlarging motifs and enlarging non-motif areas is depicted in Algorithm 3. Figure 3 shows how the distortion algorithm works. Our technique tries to enlarge or shrink areas according to the weights.

Algorithm 3

```
Algorithm 3: Build array with efficiency values
Input: Array of motifs: Motif [ ] all motifs
output: Arrays of efficiency values:
weightsMotifs // used for enlarge 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
```

In Figure 4A, our technique divides the time series into equal size parts and resizes each part according to the aggregated weight of the part. We first calculate a fully distorted view for each task (enlarging motifs or enlarging areas without motifs) and then calculate the zero slider position.

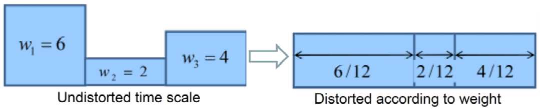


Figure 3: Distorting the Time Scale According to Given Weights

When the user moves the slider to the left, areas without motifs are enlarged. The slider's middle position is its origin scale. 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.

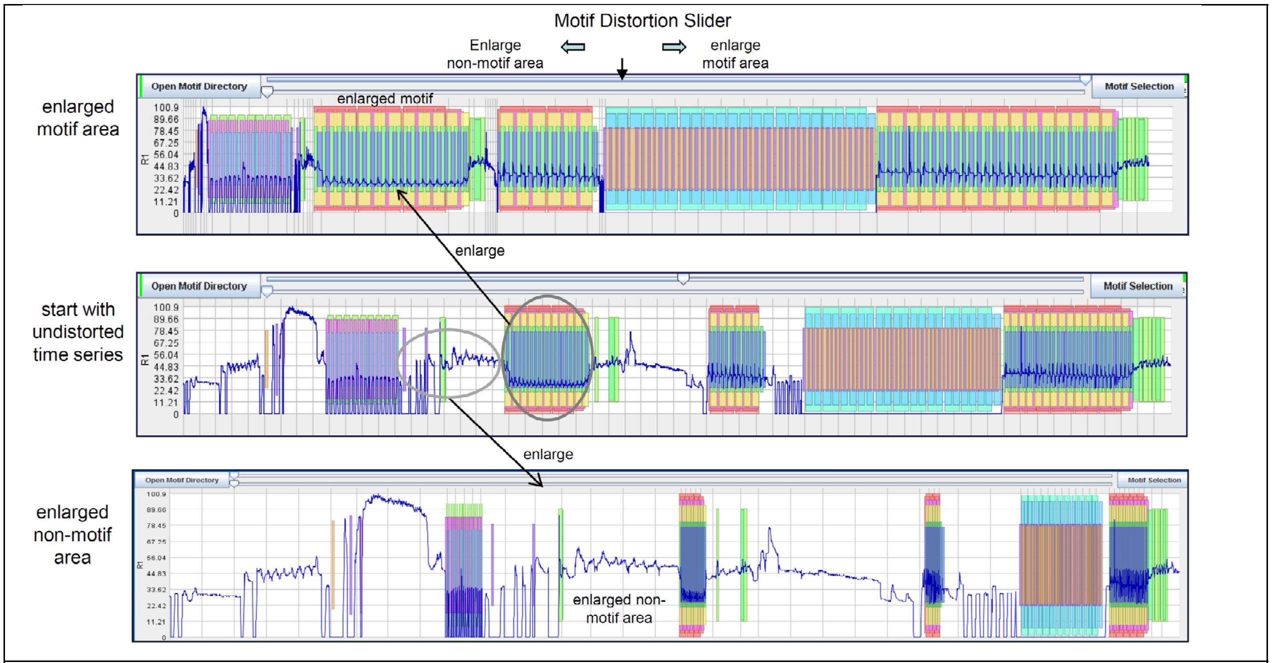


Figure 4A: Motif Visual Distortion
(x-axis: time, y-axis: %utilization of chiller R1, rectangles: motifs, color: cooling efficiency)
Move the distortion slider to the right to enlarge motifs. Move the distortion slider to the left to enlarge the non-motif area.

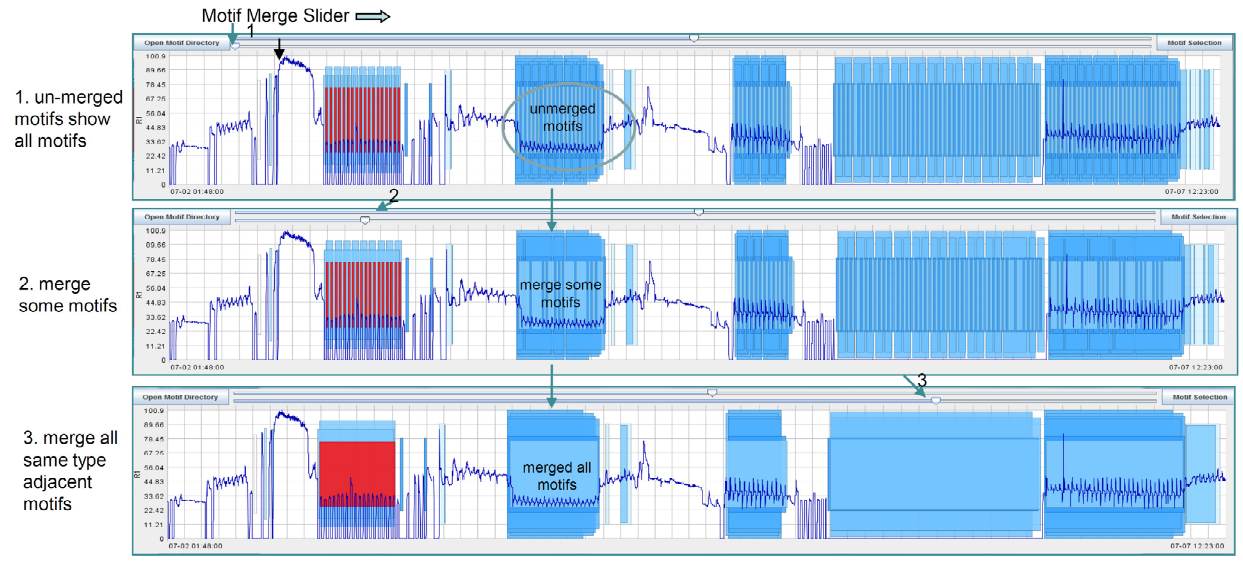


Figure 4B: Motif Visual Merging
(x-axis: time, y-axis: %utilization of chiller R1. rectangles: motifs, color: cooling efficiency)
Move the slider to the right to merge adjacent motifs of the same type.

3.3 Motif Merging

In order to merge multiple occurrences of motifs to a single rectangle and to reduce the number of motifs and the visual clutter we provide a second slider (see Figure 4B). We use a threshold using three steps for merging motif. If the slider is moved to the right, 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.

3.4 Pattern-Preserving Prediction

For our application, in data centers, in addition to motif detection, it is important to predict the resource consumption for the immediate future. With standard methods such as the well-known Holt Winters prediction model [23]; however, the prediction often does not preserve peaks as shown in Figure 5. The red box highlights that the proposed technique performs better in terms of peak preservation.

Prior to prediction, we apply smoothing. Smoothing based on moving averages using a varying time interval can help to reduce the negative effects of noise on the prediction. In our data center application, however, this is not enough. In the prediction it is especially important to retain peaks since peaks are essential in planning resource consumption. To obtain a pattern-preserving prediction, we derive a variant of the well-known Douglas Peucker [18] algorithm, which reduces a graph to its most significant data points. The algorithm starts with creating a connecting line, which connects the first and the last value. Then, it searches for the highest or lowest data point in between these values

Algorithm 3

Algorithm 3: Prediction based on daily patterns

```
double[ ] doPrediction(double[ ] pastValues,
                       Date[ ] dateOfPastValues,
                       double[ ] importancePeakWeights) {
// create temporary storage:
double valueForEachMinuteOfTheDay[ ]
   = new double[60 * 24];
int counterForEachMinuteOfTheDay[ ] = new int[60 * 24];

double c = calculateConstant(numberOfDays);
for (int i = 0; i < pastValues.length; i++) {
   Date d = dateColum[i];
   int minuteOfTheDay = d.getHours() * 60 + d.getMinutes();
   counterForEachMinuteOfTheDay [minuteOfTheDay]++;
   // Add the current value multiplied with a computed weight to
   // the right slot, as we are calculating a weighted average
   valueForEachMinuteOfTheDay[minuteOfTheDay] +=
      values[i] * combinedWeights(
      counterForEachMinuteOfTheDay [minuteOfTheDay] * c,
      importancePeakWeights[i], userSetValue);
}

return valueForEachMinuteOfTheDay;
```

with respect to the connecting line. If the absolute height of the data point exceeds a certain threshold, this data point is tagged as a peak. The algorithm performs these steps recursively again until no more peaks can be found, and then the process terminates.

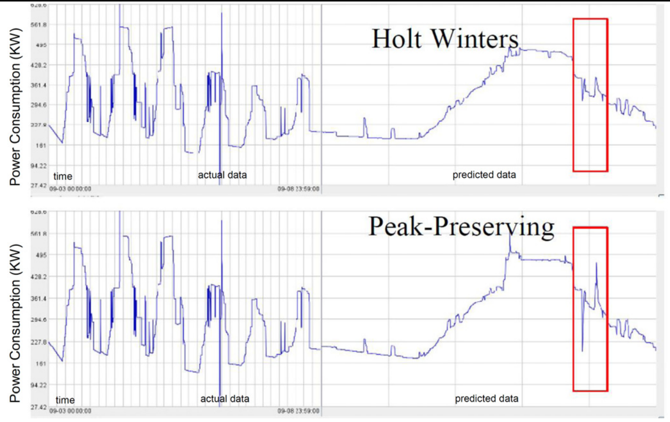


Figure 5: Comparison of Holt Winters method with Pattern-Preserving prediction which including the peak. Peak removed in Holt Winters. Our prediction handles the peak better than Holt Winters method.

After smoothing, the pattern-preserving prediction algorithm (Algorithm 3) generates the predicted data points based on the time period of the historical data as following:

1. In order to compute the predicted data points we iterate over the entire dataset of the past values.
2. Use daily grouping (which level of time to group is highly application dependent) for the weighted averaging. For example, we want to predict the data point for the time 0:00 o'clock, we look for all measurements made at 0:00.
3. A single predicted value can be therefore calculated as follows, where all value v are measured during the same time interval (minute) of the day

$$pred(\text{min Of Day}) = \sum \text{combinedWeights} \cdot v$$

4. The *combinedWeights* result from the combination of two interestingness measures namely time distance and peak importance. For the first one we use linearly decreasing weights to ensure recent measurements with higher influence. As we will use the weights for a weighted average, the weights should look like 1 * c, 2 * c, 3 * c … where c is a constant, normalizing the weights of the result.

$$\sum_{i=1}^{n} i * c = c * \sum_{i=1}^{n} i = c * \frac{n*(n+1)}{2} = 1$$

$$\Rightarrow c = \frac{2}{n*(n+1)}$$

The peak importance is measured using the recursion depth of the recursive smoothing algorithm.

4. Applications and Evaluation

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, oil well production sensor data (e.g., oil flow, pressure), and resource utilization prediction. 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

The motif time series in Figure 6 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. The cooling efficiency metric, called coefficient of performance (COP), is calculated by dividing the heat extracted by the power consumed. Service managers can quickly identify that motif 5 is more efficient than the other motifs (blue color of motif 5). 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.

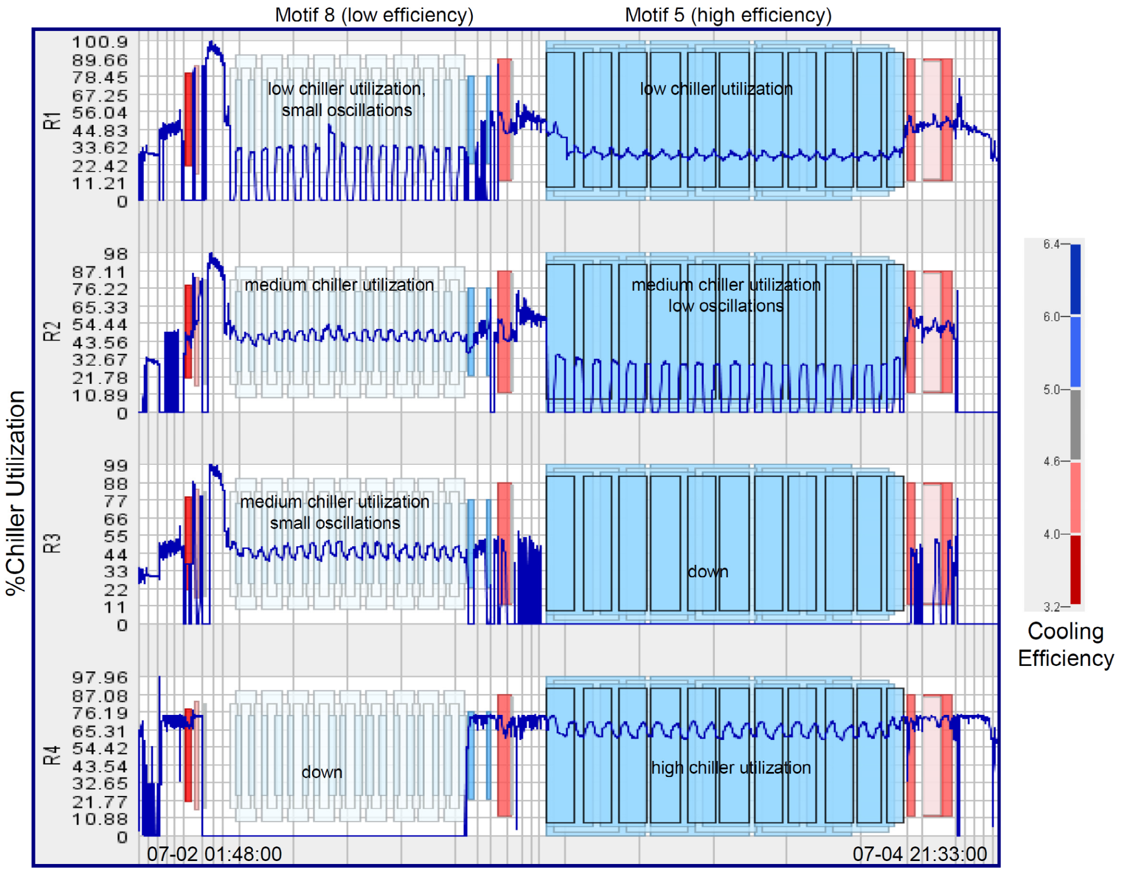


Figure 6: Visual Analytics of Data Center Cooling Management
Motifs 5 and 8 are enlarged to compare their chiller utilization
Motif 5 is more efficient than motif 8. Motif 8's chillers R1 and R3 have some oscillatory behavior.
(x-axis: 07-02 01:48 to 07-04 21:33, y-axis: %utilization of chillers R1-R4, color: cooling efficiency).

In motif 8, chiller R1 operates in a low utilization with many small oscillations. Since motifs are overlaid on the time series, it is easy to observe that the utilization of chiller R4 is the highest in motif 5.

Evaluation

The new motif finding, distortion, and merging visualization techniques have been successfully used on two production data centers of different sizes, about 3,000 and 14,000 sq. ft., respectively, and containing up to hundreds of racks. Several million records from data centers have been analyzed.

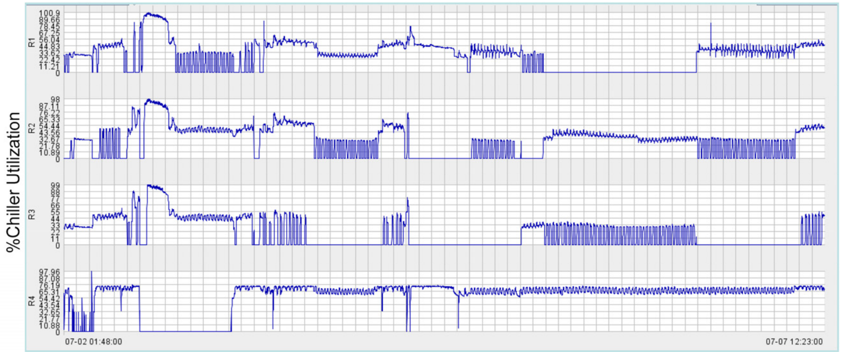


Figure 7: Data Center Chiller (R1-R4) % utilization regular time series without motif

Using existing regular time series plots, as shown in Figure 7, it may take hours for data center service managers to analyze and observe the variation of utilization over time. However, service managers cannot easily determine which set of patterns represent an efficient mode of operation, nor can they determine whether a pattern had occurred previously. Usually, such operational patterns are characteristic of a delay in matching chiller cooling supply with data center cooling demand. Not all chillers can scale uniformly in capacity with a rise in demand. Also demand does not change uniformly over time. However, this kind of monitoring is essential in building efficient management systems.

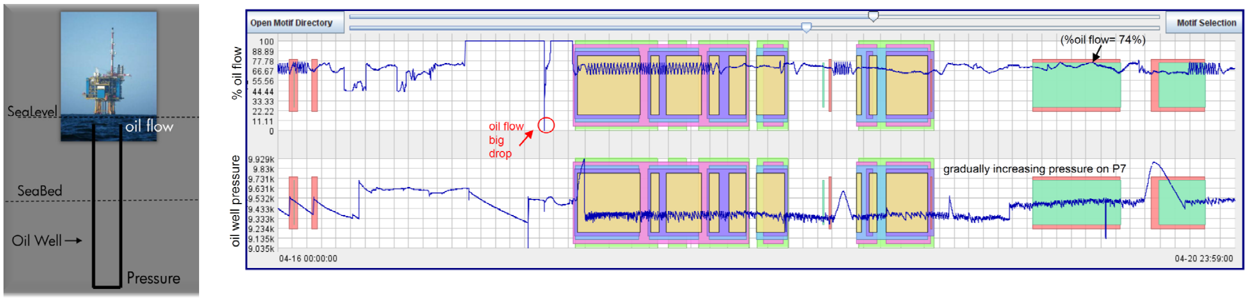
The motif time series, as shown in Figure 6, helps service managers identify motifs and their cooling efficiencies and provides guidance on how current performance compares with past performance. Our new techniques can assist service managers to move the chiller system to a more efficient state.

4.2 Oil Well Production Motif Observations

The picture on the left of Figure 8 is a typical oil well. The figure on the right shows oil flow and pressure time series (85,035 records) with different frequent patterns (motifs) identified by the efficiency of the oil well production volumes. A critical problem in the oil industry is to reduce production losses. 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 8 illustrates the use of a combination of distortion and merging to make the motif visual analytics most effective. 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.



Oil Well Production Operation

Figure 8: Oil Well Production Time Series with 7 Different Frequent Patterns with Distinct Colors
(x-axis: time, y-axis: % oil flow and pressure,
color: green represents high oil well production and yellow, low production).

Evaluation:

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 oil flow can help in the classification of such issues. Finding the motifs that are able to maximize oil flow at the normal pressure is the goal of the well production manager. Without our motif layout, it is almost impossible for production manager to find these frequent patterns, as shown in Figure 9. Using motifs, as illustrated in Figure 8, the production manager can quickly find the most efficient motifs (green). Furthermore, production managers can reduce the motifs (yellow) that cause fluctuations in pressure (or flow). The motifs with high oscillations can be detrimental to well operation and lead to reliability issues.

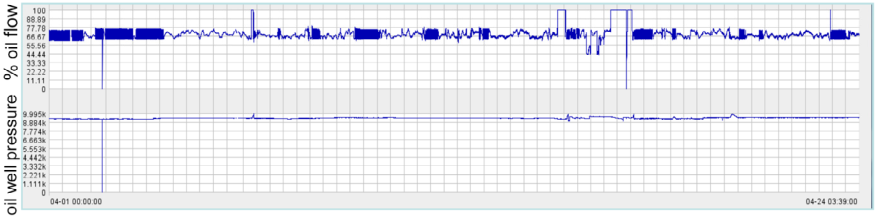


Figure 9: Oil well production regular time series without motif

4.3 Resource Utilization Prediction

Optimizing the utilization of servers has a major impact on costs in IT-Services centers. The basic power consumption of an idle server is significant–approximately 50% of peak power usage. This leads to the conclusion that a server is utilizing power best when it is fully loaded and idle servers should be turned off. To reduce the risk of performance degradation, service managers have to analyze the server utilization patterns and relocate applications away from under-utilized servers. To get a reasonably high utilization, service managers are required to consolidate applications into fewer servers.

Figure 10 shows a server's daily utilization based on two attributes (server utilization and number of users) of 36,338 measurements. The time series on the left of Figure 10 shows the actual data. The time series on the right shows the predicted data on 10/15. The colored motifs are used to show resource utilization efficiency which is the ratio between the server utilization and the number of users. The color of the motifs is used to show resource utilization efficiency (red: low, blue: high). The narrow certainty band indicates that it is safe to move the applications to another server and power down this server from hour 22 to 9 am the next day, as indicated by the red motifs. The peak time for running applications is during the day between hours 9 to 16 and at night between hours 20 to 21 for system work, as shown in blue motifs with high efficiencies. From our experiments, a power savings of up to 30% seems realistic. Interestingly, the motif occurrences are highly correlated to the number of users as all motifs contain only those areas where the number of users is high. Combining motifs and prediction is, in this case, very important to enlarging the influence of motifs in the prediction process which leads to an overall better prediction result. Using motifs, service managers can quickly recognize which time intervals have a low utilization and which servers can be shut down to save energy.

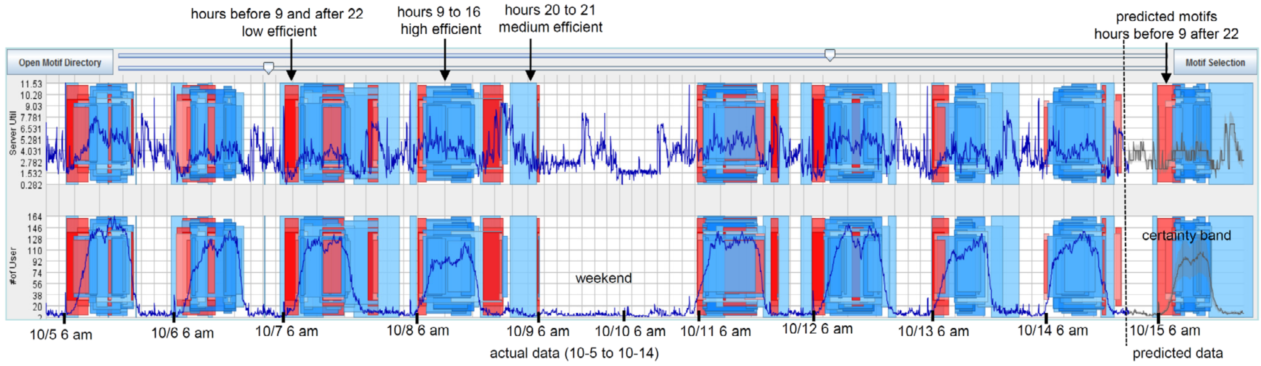


Figure 10: Resource utilization predictions using pattern-preserving visual analytics technique
An interesting observation shows a low efficient motif occurred in every morning from 6 am to 9 am due to no enough users.
(x-axis: time, y-axis: server utilization and number of users, color: blue represents high server efficiency and red low efficiency).

Evaluation on Prediction Accuracy

The server utilization from 10/06 to 10/15 has been used to measure the accuracy of the pattern-preserving prediction techniques. The values of each single day are predicted and compared to the observed actual data. We compute the difference between each real-data and predicted data over time. Then we take the average them to derive the accuracy. The result of the comparison between actual and predicted data shows an accuracy of 70% - 80% with an average accuracy of 75%. In general, the 75% prediction accuracy is for data of similar temporal repeatability. As temporal repeatability of the data decreases, the accuracy of prediction will decrease regardless. Figure 11 shows the predictions for one day, 10/14.

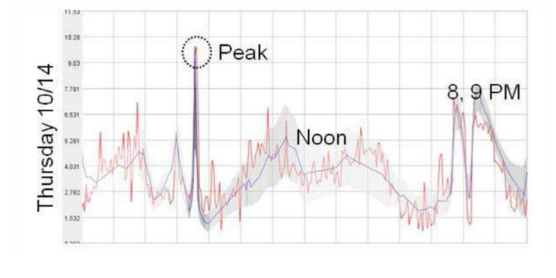


Figure 11: Prediction accuracy comparison between actual and predicted data
(blue: predicted values / red: actual values)

5. Conclusion

Finding frequently occurring patterns and analyzing them allows data center service managers and oil well production managers to determine which configurations are more efficient and which ones result in poor efficiency so that 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 the efficiency 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 users to adjust the degree of distortion and merge to generate the best view on a single display. To enable the users to find the most efficient motifs, our techniques link the motifs to the associated efficiency metrics for root-cause analysis. Furthermore, our techniques provide a visual analytics approach for the pattern-preserving prediction of large seasonal multivariate data. Our results from both the real-world data center and oil production time series sensor data show that our techniques successfully enable users to identify both efficient and inefficient patterns. Furthermore, our techniques also provide reliable predictions. This demonstrates the wide applicability and usefulness of our techniques. In the future, we want to detect motifs in real-time to perform immediate intervention.

Acknowledgment: The authors wish to thank M. Hsu, C. Patel, Laura Hill, and Cullen Bash for their suggestions and encouragement, and Walter Hill and Sebastian Mittelstädt for providing prediction 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., Munzner, 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] Holt, C. C., “Forecasting seasonal and trends by exponentially weighted moving averages. ONR Research Memorandum, Carnegie Institute 52.

[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.

[18] Douglas, D. & Peucker, T. "Algorithms for the reduction of the number of data points required to represent a digitized line or its caricature", The Canadian Cartographer 10(2), 112–122 (1973).

[19] Taylor, J. W. “Forecasting Daily Supermarket Sales using Exponentially Weighted Quantile Regression” European Operational Research, 2007, Volume 207.

[20] Ichikawa, Y., Tsunawaki, T. A Visualization environment for multiple Daytime Stock Price Predictions, 2002. IEIC Technical Report.
[21] Masse, C. F. “2008 US Presidential Election Prediction – A Visual Approach of InTrade’s Prediction Markets. InTrade, 2008.
[22] Winters, P. R. Forecasting sales by exponentially weighted moving averages, Management Science 6, 1960.
[23] ARIMA and Holt Winters prediction models are described in the Chatfield, C. The analysis of time series: An Introduction, 6th ed., CRC Press, Boca Raton, USA. 2004.