Temporal Data Mining Approaches for Sustainable Chiller Management in Data Centers

DEBPRAKASH PATNAIK, Virginia Tech
MANISH MARWAH, HP Labs
RATNESH K. SHARMA, NEC Labs
NAREN RAMAKRISHNAN, Virginia Tech

Practically every large IT organization hosts data centers—a mix of computing elements, storage systems, networking, power, and cooling infrastructure—operated either in-house or outsourced to major vendors. A significant element of modern data centers is their cooling infrastructure, whose efficient and sustainable operation is a key ingredient to the "always-on" capability of data centers. We describe the design and implementation of CAMAS (Chiller Advisory and MAnagement System), a temporal data mining solution to mine and manage chiller installations. CAMAS embodies a set of algorithms for processing multivariate time-series data and characterizes sustainability measures of the patterns mined. We demonstrate three key ingredients of CAMAS—motif mining, association analysis, and dynamic Bayesian network inference—that help bridge the gap between low-level, raw, sensor streams, and the high-level operating regions and features needed for an operator to efficiently manage the data center. The effectiveness of CAMAS is demonstrated by its application to a real-life production data center managed by HP.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining*; K.6.2 [**Management of Computing and Information Systems**]: Installation Management—*Computing equipment management*

General Terms: Algorithms, Experimentation, Measurement, Management, Reliability

Additional Key Words and Phrases: Data centers, chillers, clustering, motifs, frequent episodes, sustainability

ACM Reference Format:
Patnaik, D., Marwah, M., Sharma, R. K., and Ramakrishnan, N. 2011. Temporal data mining approaches for sustainable chiller management in data centers. ACM Trans. Intell. Syst. Technol. 2, 4, Article 34 (July 2011), 29 pages.
DOI = 10.1145/1989734.1989738 http://doi.acm.org/10.1145/1989734.1989738

1. INTRODUCTION

Modern IT infrastructure is ubiquitous, especially in the services sector that requires "always-on" capability. Practically every large IT organization hosts data centers, operated either in-house or outsourced to major vendors. Over the last decade, data centers have grown from housing a few hundred multiprocessor systems to tens of thousands of individual servers today. This growth has been accompanied with steep increases in power density, resulting in higher heat dissipation, and thus increasing both power

Authors' addresses: D. Patnaik, Department of Computer Science and Application, Virginia Tech, 1251 Progress Street NW, Apt 4900G, Blacksburg, VA 24060; email: patnaik@vt.edu; M. Marwah, HP Labs, 3000 Hanover Street, Palo Alto, CA 94304-1185; R. K. Sharma, Department of Energy Management, NEC Labs, 4 Independence Way, Suite 200, Princeton, NJ 08540; N. Ramakrishnan, Department of Computer Science and Application, Virginia Tech, Blacksburg, VA 24060.

DOI 10.1145/1989734.1989738 http://doi.acm.org/10.1145/1989734.1989738

and cooling costs. According to the EPA, U.S. data centers have become energy hogs and their continued growth is expected to demand the construction of 10 new power plants by 2011 [Koomey 2008a, 2008b; Kaplan et al. 2008]. One news report [Leake and Woods 2009], perhaps alarmist, claims that a single Web search query can use up to half the equivalent energy of boiling a kettle of water! Globally, data centers currently consume 1–2% of the world's electricity [Vanderbilt 2009] and are already responsible for more CO_2 emissions than entire countries such as Argentina or The Netherlands. If these trends hold, data center emissions are expected to quadruple by 2020 [Kaplan et al. 2008] and some estimates expect the carbon footprint of cloud computing to surpass aviation [Koomey 2008a].

Data centers constitute a mix of computing elements, networking infrastructure, storage systems along with power management and cooling capabilities all of which contribute to energy inefficiency. A plethora of approaches are hence available to curtail energy usage in each of the different data center subsystems and achieve sustainable data centers. For instance, huge inefficiencies abound in average server usage (believed to be in the single digits -to- at most 10–15%), and thus one approach to achieve greener IT is to use virtualization and migration to automatically provision new systems as demand spikes and consolidate applications when demand falls. A lot of prior work has focused on a data center's cooling infrastructure, which consumes anywhere between 30% to 50% of the total power consumption of a data center. Within the cooling infrastructure, most of the energy is spent on chillers, which refrigerate the coolant, typically water, used to extract heat from the equipment in the data center. Dynamic management of an ensemble of chiller units [Patnaik et al. 2009b] in response to varying load characteristics is an effective strategy to make a data center more energy efficient. There are even end-to-end methodologies proposed [Sharma et al. 2008] that track inefficiencies at all levels of the IT infrastructure "stack" and derive overall measures of the efficiency of energy flow during data center operation.

A key problem is the unavailability, inadequacy, or infeasibility of theoretical models or "first principles" methodologies to optimize design and usage of data centers. Admittedly, some components of data centers can be readily modeled (e.g., an operating curve for an individual chiller unit, a CFD prediction of airflows through rows of racks for static conditions) but the applicability of these models is severely limited due to simplifying assumptions and excessive computational time. Consequently, data-driven approaches to data center management have become more attractive. By mining sensor streams from an installation, we can obtain a real-time perspective into system behavior and identify strategies to improve efficiency metrics.

In this article, we focus primarily on the cooling infrastructure of a data center, especially chillers. Chillers are a key ingredient to keeping data centers functioning; as a case in point, recently, the music service Last.fm had to be shut down due to overheating in its data center. We present CAMAS (Chiller Advisory and MAnagement System), a temporal data mining solution to mine and manage chiller installations. CAMAS embodies a set of algorithms for processing multivariate time-series data and characterizes sustainability measures of the patterns mined. Thus, we show how temporal data mining can bridge the gap between low-level, raw, sensor streams and the high-level operating regions and features needed for an operator to efficiently manage the data center.

The design and implementation of CAMAS makes the following contributions.

(1) We demonstrate an efficient approach to mine motifs in multivariate time-series data and which can be used for sustainability characterization. Our algorithm can accommodate "don't care" states in its definition of motifs and this enables us to uncover expressive patterns in multivariate data.

(2) We describe how simple association analysis can be utilized to identify inefficient regions of chiller ensemble operation and how protocols for improving the overall efficiency of the system can be readily derived from the results of such associations.
(3) We describe how we can construct a complete Dynamic Bayesian Network (DBN) that both captures the operation of the chiller ensemble and also enables reasoning in what-if and diagnostic scenarios.

This article builds upon two preliminary conference publications [Patnaik et al. 2009b, 2010] by providing an integrated framework for mining and reasoning about chiller data. In particular, we provide a greater coverage of chiller management issues throughout the article, demonstrate how our prior work on mining sensor streams is generalized by the CAMAS framework, and describe how external conditions can be monitored and mined to identify regions of inefficiency in the data center.

2. BACKGROUND

We present next some background about data center chillers and their chiller installations with a view toward motivating the underlying operational problems that can be solved using data mining techniques.

2.1. Architecture of a Data Center

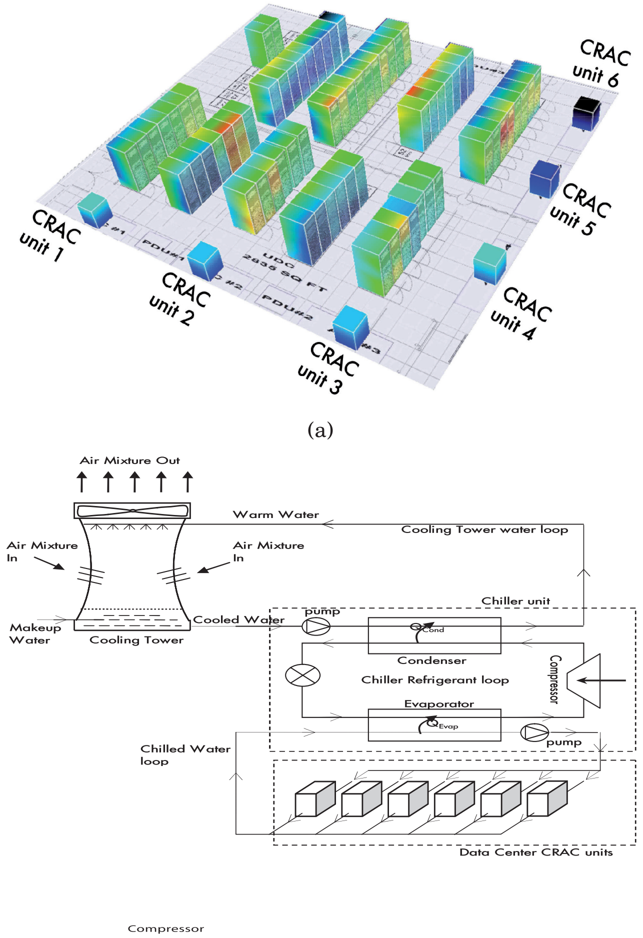
Figure 1(a) shows a data center consisting of IT equipment (servers, storage, networking) fitted in racks arranged as rows. A large data center could contain thousands of racks occupying several tens of thousands of square feet of space. Also shown in the figure are Computer Room Air Conditioning (CRAC) units that cool the exhaust hot air from the IT racks. Energy consumption in data center cooling comprises work done to distribute the cool air and to extract heat from the hot exhaust air. A refrigerated or chilled water cooling coil in a CRAC unit extracts the heat from the air and cools it within a range of 10°C to 18°C. The cooling infrastructure of a data center is shown in Figure 1(b).

Key elements of this infrastructure include CRAC units, plumbing and pumps for chilled water distribution, chiller units, and cooling towers. Heat dissipated from IT equipment is extracted by CRAC units and transferred to the chilled water distribution system. Chillers extract heat from the chilled water system and reject it to the environment through cooling towers or heat exchangers. In addition to the IT equipment, the data center cooling infrastructure can account for up to 50% of the total power demand [Belady 2007]. The CRAC units provide two actuators that can be controlled. The Variable Frequency Drive (VFD) controls the blower speed and the chilled water value regulates the amount of chilled water flowing into a unit (between 0% and 100%). These built-in flexibilities allow the units to be adjusted according to the workload demand in the data center. The demand is detected via temperature sensors installed on the racks throughout a data center.

2.2. Data Center Chillers

As stated earlier, the focus of this article is on chiller units that receive warm water (at temperature, T_{in}) from the CRAC units, extract heat from it, and recirculate the chilled water (at temperature, T_{out}) back to the CRAC units.

Each chiller is composed of four basic components, namely, evaporator, multistage centrifugal compressor, economizer, and water-cooled or air-cooled condenser. Liquid refrigerant is distributed along the length of the evaporator to absorb enough heat from the water returning from the data center and circulated through the evaporator tubes to vaporize. The gaseous refrigerant is then drawn into the first stage of the compressor. Compressed gas passes from the multistage compressor into the condenser. Cooling



(a)

(b)

Fig. 1. (a) Thermal map of a data center showing racks arranged in rows and CRAC units, and (b) typical cooling infrastructure of a data center.

tower water circulated through the condenser tubes absorbs heat from the refrigerant, causing it to condense. The liquid refrigerant then passes through an orifice plate into the economizer. Flashed gases enter the compressor while the liquid flows into the evaporator to complete the circuit.

Starting and stopping a chiller is a complex, multistep process. Figure 2 shows the operational state diagram of a typical chiller. On power-on, the chiller waits for the compressors to start, after a prescribed delay. On startup, the chiller utilization varies to match the cooling load. Based on chiller technology, chiller compressors can throttle in discrete stages or continuously. Feedback control is used to maintain the outlet temperature, T_{out}, close to a user-specified set-point temperature.

We define some terms used in the context of a data center chiller unit.

—*IT cooling load.* This is the amount of heat that is generated (and thus needs to be dissipated) at a data center. It is approximately equivalent to the power consumed by the equipment since almost all of it is dissipated as heat. It is commonly specified in kilowatts (KW).

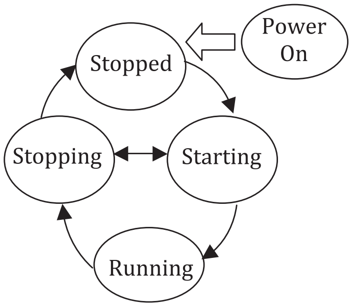


Fig. 2. Operational state diagram of a chiller unit.

- *COP*. The Coefficient Of Performance (COP) of a chiller unit indicates how efficiently the unit provides cooling, and is defined as the ratio between the cooling provided and the power consumed, that is,

$$\mathrm{COP}_i = \frac{L_i}{P_i}, \tag{1}$$

  where L_i is the cooling load on the ith chiller unit and P_i is the power consumed by it. In the data center studied in this work the typical values of COP for air-cooled chillers and water-cooled chillers are 3.5 and 6.5, respectively.
- *Chiller utilization*. This is the percentage of the total capacity of a chiller unit that is in use. It depends on a variety of factors, mainly, the mass flow rate of water that passes through a chiller and the degree of cooling provided, that is, the difference between the inlet and outlet temperatures ($T_{in} - T_{out}$). For a particular T_{out}, an administrator can control the utilization at a chiller through power capping or by changing the mass flow rate of water. The air-cooled chillers are operated in swing-mode to handle rapidly changing cooling load and their utilizations typically vary in the range 20–80%. On the other hand the water-cooled chillers are operated at high utilization $\approx$80% and handle the base cooling load.
- *Chiller power consumption*. This is simply the power consumed by a chiller unit. Although power meters that measure aggregate power consumption of data center infrastructure elements are usually available, meters that measure power consumed by an individual entity or a specific group (e.g., chillers) may not always be installed. In such cases, if the capacity of the unit and average COP are known, they, together with unit utilization, can be used to estimate power consumed. We have

$$P_i = \frac{U_i * C_i}{100 * \mathrm{COP}_i}, \tag{2}$$

  where P_i is the power consumed, U_i the utilization, and C_i the capacity, all pertaining to the i^{th} chiller unit.

2.3. Ensembles of Chiller Units

The number of chiller units required depends on the size and thermal density of a data center. While one unit may be sufficient for a small data center, several units operating as an ensemble may be required to satisfy the cooling demand of a large data center. Figure 3 shows an ensemble of chiller units that collectively provide cooling for a data center. Out of the five units shown, three are air-cooled while the remaining two are water-cooled. Also, to provide a highly available data center and ensure business continuity, sufficient spare capacity is usually provisioned to meet the cooling demand

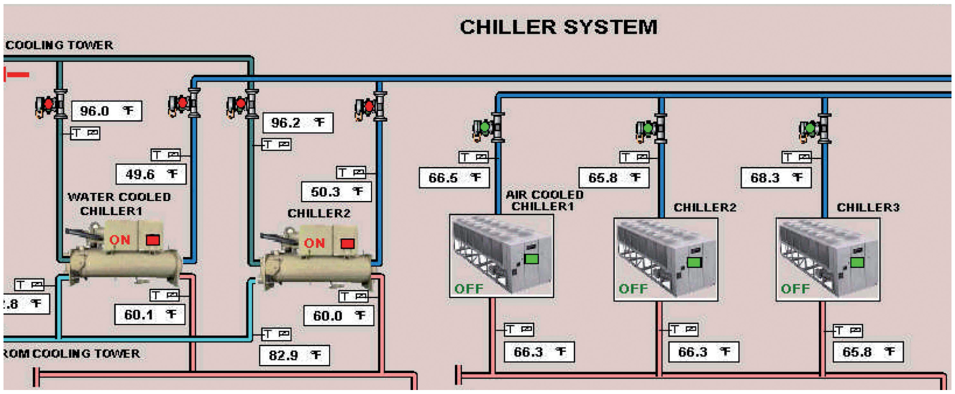


Fig. 3. Five chiller units work in tandem to provide cooling for a large data center.

in the event of one or more units becoming unavailable as a result of failure or required maintenance.

Although operating curves for individual chiller units exist, no model is available for operation of an ensemble, especially one consisting of heterogeneous units. Additionally, shift and/or drift of response characteristics with time further complicate their management. The operational goals are to satisfy the cooling requirements while minimizing the total power consumption of the ensemble and maximizing the average lifespan of the units. While multiple factors impact the lifespan of a chiller unit, an important one is: rapid and large oscillations in utilization value. High amplitude and frequent variations in utilization due to varying load or some failure condition result in decreased lifespan, and thus need to be minimized.

2.4. Chiller Management Issues

There are several issues that lead to inefficient chiller operation and these are some of the topics that motivate the design of CAMAS.

—*Short cycling*. Frequent start and stop cycles lead to fatigue of mechanical parts due to high torque requirements, and deterioration of electrical circuitry due to high inrush current. Moreover, load fluctuations due to cycling can also lead to drop in power factor and potential penalties from the utility. In case of data centers with on-site generation, such fluctuations can lead to reliability issues at the generators as well. Downstream of chillers, pump performance, and cooling tower efficiency can also be adversely affected. Typically chillers have an MTBF (Mean Time Between Failure) of 20,000 hours or more, which can reduce exponentially due to oscillations.

—*COP dependence on utilization.* Chillers show poor energy efficiency at low and high utilizations. Figure 4 shows a typical variation of efficiency (in terms of coefficient of performance (COP), see Eq. (1)) with utilization. These curves depend not only on the type of chiller but also on external factors, such as ambient temperature, supply temperature of the coolant, etc. Further, these curves may even shift with time over the typical 15 to 20 years lifespan of a chiller.

—*Complex and unknown dependencies between external variables and performance.* The performance of an ensemble of chillers depends on many factors, several of which are not explicitly monitored. For example, it depends on ambient temperature and humidity. Further, each installation of chillers is slightly different from other installations, requiring local domain experts to fine-tune the performance. Further, these relationships may show a drift with time.

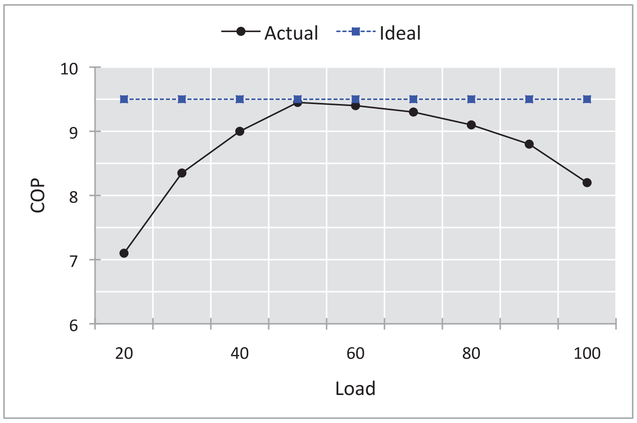


Fig. 4. Variation of chiller efficiency (COP) with utilization (load).

3. CAMAS: A FRAMEWORK FOR MINING CHILLER INSTALLATIONS

While administration of a single chiller unit is not complicated, configuring an ensemble of chillers for optimal performance is a challenging task, especially in the presence of a dynamically varying cooling load [Boucher et al. 2006]. Typically, heuristics and rules-of-thumb are used to make decisions regarding:

—Which chiller unit(s) should be turned on/off, and when?
—What utilization range should a particular unit be operated at?
—How should the ensemble react to an increase or decrease in cooling demand?

Any guidance regarding questions posed before while maintaining performance and optimizing the earlier stated goals will be invaluable to a data center facilities administrator.

The primary goal of CAMAS is to aid in precisely this objective. CAMAS links the multivariate, numeric, time-series data streams gathered from chiller units to high-level sustainability characterizations. As shown in Figure 5, CAMAS performs such transduction at many levels. First, it transduces multivariate time-series streams into symbolic event data from which frequent episodes are mined to yield motifs of chiller operation. We show how these motifs directly map to sustainability characterizations. In particular, we can describe states of chiller operations in terms of the motifs they exhibit and the efficiency regions they involve. Figure 6 describes an example session with CAMAS in this objective. This constitutes Part 1 of the framework (see Figure 5). In Part 2 of the framework, by taking external conditions into account, we demonstrate how we can use association analysis to find inefficient states of operation. In Part 3, we demonstrate how these ideas can be generalized further into a complete framework for reasoning about chiller states and their transitions. We demonstrate a Dynamic Bayesian Network (DBN) approach to infer probabilistic relationships between chiller variables. Each of these three aspects are covered in the following sections.

4. SUSTAINABILITY CHARACTERIZATION USING MOTIFS

Motifs are repetitive patterns of occurrence in time-series data. In understanding multivariate time-series data about chiller utilizations, we seek to identify motifs that underly how different chillers are involved in meeting the varying demand posed by data centers. We would like to identify regions of time-series progression that demonstrate better/improved sustainability measures than others.

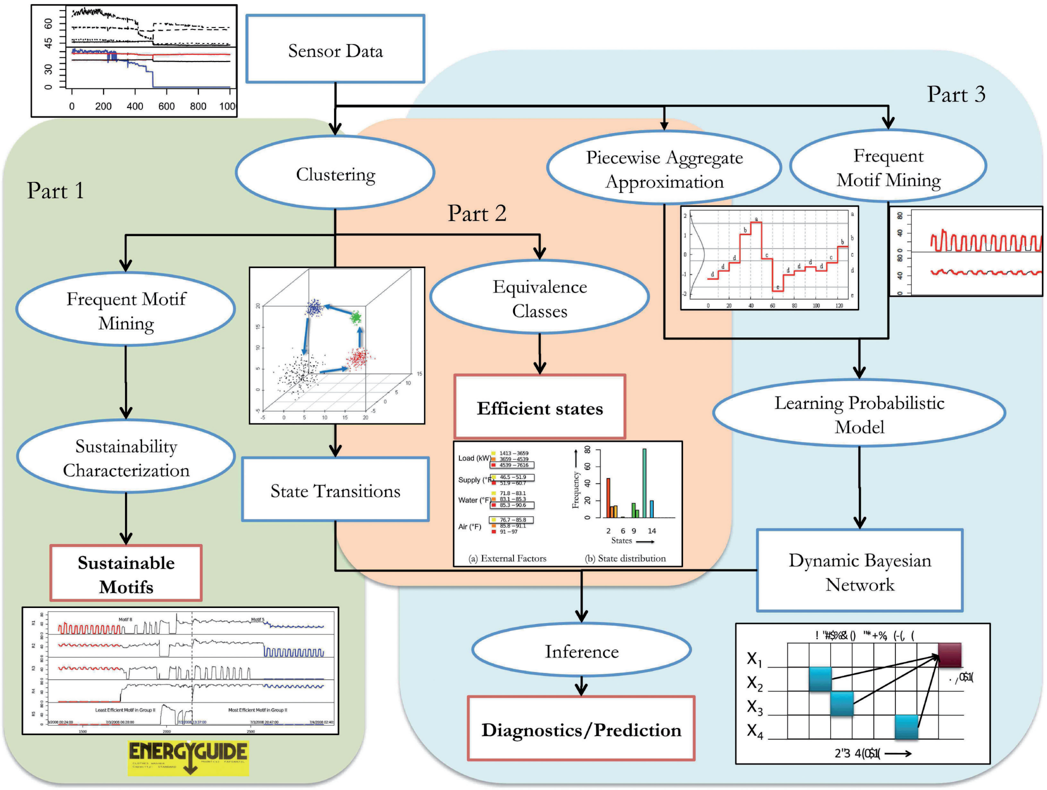


Fig. 5. CAMAS framework. Three important building blocks are illustrated: (1) Motif discovery and sustainability characterization, (2) modeling external factors and internal states, and (3) probabilistic modeling of system variables.

4.1. Approach

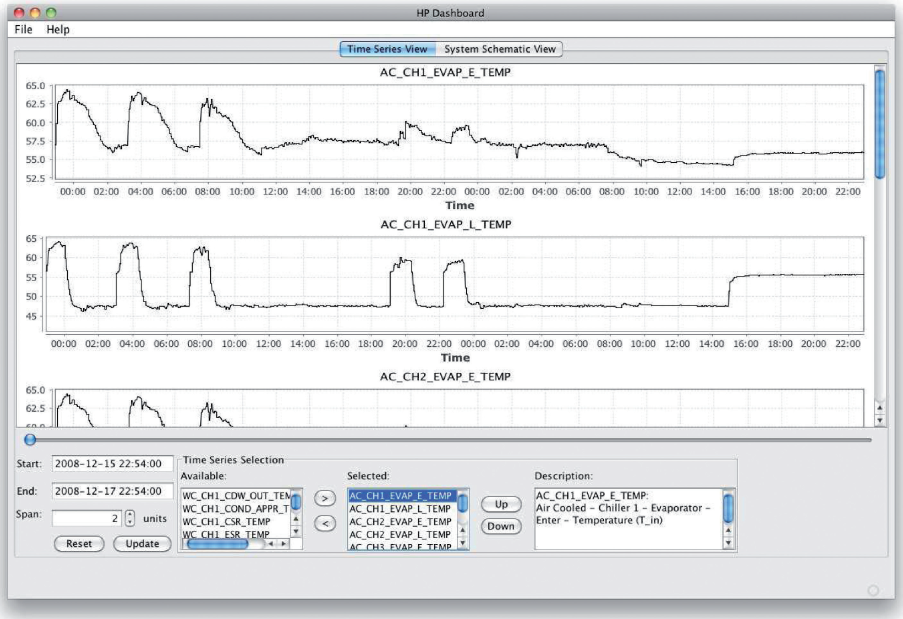
Prior work in multivariate change point detection (e.g., see Xuan and Murphy [2007]) has posited statistical models for behavior within a segment and tracks changes in model parameters to denote qualitative change points. Our task is exacerbated by the lack of adequate models to characterize chiller behavior and also because of the varying interpretations that are attachable to multivariate data.

First, a motif or a trend can manifest in a single series or in multiple series. Second, even when it manifests in multiple series, the specificity with which it manifests can be fixed or variable. For instance, a motif can be "three chiller units show oscillatory behavior" versus "first three chiller units show oscillatory behavior." Since we seek to mine motifs so that their presence/absence can be cross-correlated with the chiller design (e.g., air cooled versus water cooled), we seek motifs of the latter form.

Definition 4.1. A multivariate time-series $T = \langle \mathbf{v_1}, \ldots, \mathbf{v_m} \rangle$ is an ordered set of real-valued vectors. Each real-valued vector $\mathbf{v_i}$ represents utilizations across all the chiller units.

The preceding definition assumes a uniform sampling and hence the associated times are not explicitly recorded in the definition of T.

Definition 4.2. A segment of the time-series T is an ordered subset of consecutive vectors of T denoted by $T[i, j] = \langle \mathbf{v_i}, \mathbf{v_{i+1}}, \ldots, \mathbf{v_j} \rangle$, where $1 \le i < j \le m$.



Start: 2008-12-15 22:54:00

End: 2008-12-17 22:54:00

Span: 2 units

Time Series Selection

Available:
WC_CH1_CDW_OUT_TEM
WC_CH1_COND_APPR_T
WC_CH1_CSR_TEMP
WC_CH1_ESR_TEMP

Selected:
AC_CH1_EVAP_E_TEMP
AC_CH1_EVAP_L_TEMP
AC_CH2_EVAP_E_TEMP
AC_CH2_EVAP_L_TEMP
AC_CH3_EVAP_E_TEMP

Description:
AC_CH1_EVAP_E_TEMP:
Air Cooled – Chiller 1 – Evaporator – Enter – Temperature (T_in)

Fig. 6. A screenshot of CAMAS used to analyze multivariate time-series data.

Definition 4.3. A motif represents a set of segments of the time-series T, $\{T[i_1, j_1], T[i_2, j_2], \ldots, T[i_n, j_n]\}$, where $1 \le i_1 < j_1 \ldots \le i_{k+1} < j_{k+1} \ldots j_n \le m$, such that any pair of segments in the set satisfy a similarity requirement and $n \ge \theta$, where θ is a user-defined minimum count.

There are many possible instantiations of the similarity measure [Chiu et al. 2003; Yankov et al. 2007], each leading to a specific formulation of a motif. We will describe later in this section the specific similarity requirement adopted here.

We decompose our overall goal into motif mining and sustainability characterization stages. Although a streaming algorithm would be more suitable in the context of time-series data, the current implementation is intended more as a diagnostic tool than for prediction. We first transduce the continuous multivariate stream into a discrete symbol stream amenable for processing by episode mining algorithms. We perform a k-means clustering on these vectors and use the cluster labels as symbols to encode the time series. Observe that the multivariate series is now encoded as a single symbol sequence.

Already we have suitably transformed the multivariate numeric data to discrete symbols. We raise the level of abstraction further by doing a run-length encoding of the symbol sequence and noting where transitions from one symbol to another occur. This gives us a sequence of events for input to serial episode mining as illustrated next.

$$\begin{array}{rl} \text{Symbol Sequence}: & \text{d d d b a c c d d d d c b} \\ & \Downarrow \\ \text{Event Sequence}: & \langle(\text{d-b}, 4), (\text{b-a}, 5), (\text{a-c}, 6), (\text{c-d}, 8), \\ & \quad (\text{d-c}, 12), (\text{c-b}, 13)\rangle \end{array}$$

Frequent episode mining is now conducted over this sequence of transitions. We adopt the framework of serial episodes with interevent constraints. The structure of a

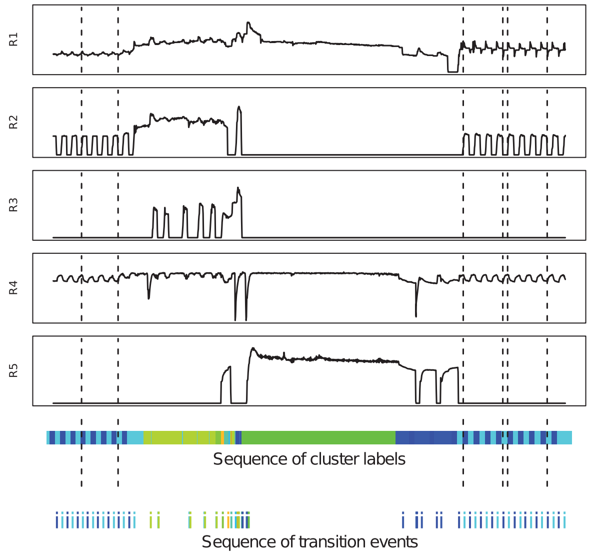


Fig. 7. Illustration of change detection in multivariate time-series data.

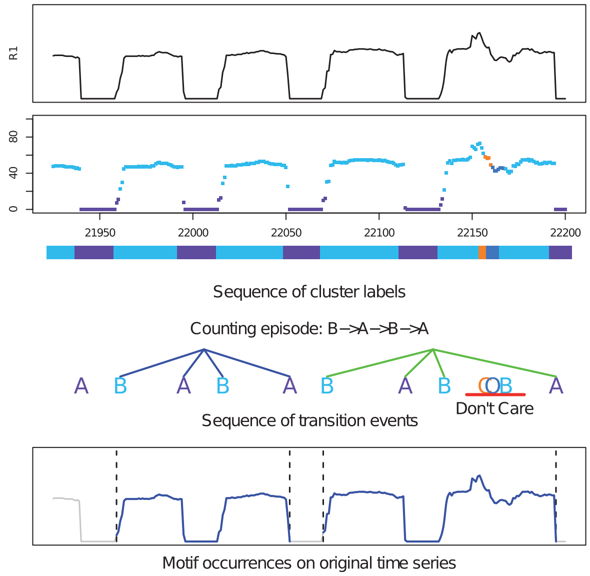


Fig. 8. Illustration of motif mining in a single time-series using frequent episodes.

serial episode α is given as

$$\alpha = \langle E_1 \overset{(0,d_1]}{\rightarrow} E_2 \ldots \overset{(0,d_{n-1}]}{\rightarrow} E_n \rangle. \tag{3}$$

Here $E_1, \ldots, E_n$ are the event types participating in the episode α and, for our domain, these event types are cluster symbol indices. Note that a serial episode requires a total order among the events. Each pair of event types in α is associated with an interevent constraint. For example, the pair $E_1 \rightarrow E_2$ is associated with $(0, d_1]$ such that in an occurrence of α, event E_2 occurs no later than time d_1 after event E_1.

Referring back to our definition of a motif, we see that the similarity requirement is thus every pair of segments must have an occurrence of the same serial episode defined over transition events. A key feature of episode mining is that the event occurrences can be interspersed with “don’t care” states and this enables us to uncover expressive patterns in the sequential data. While other approaches exist to accommodate don’t care states [Chiu et al. 2003], our algorithm is deterministic and guarantees finding repeating patterns in the discrete domain with a given support threshold. However, some patterns in the raw time-series can go unnoticed due to coarse discretization.

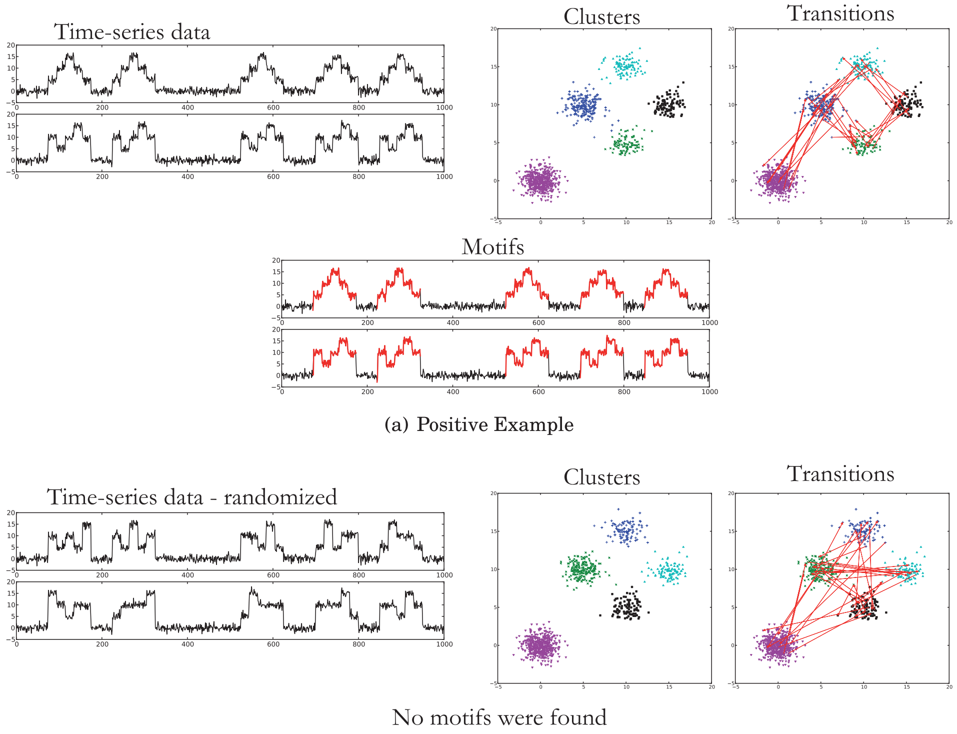
4.2. Algorithms

The mining process follows the level-wise procedure ala *Apriori*, that is, candidate generation followed by counting. 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 the another n-node frequent episode at a given level to generate candidates for the next level. The time complexity of the candidate generation process is $O(m^2 n)$, where n is the size of each frequent episode in the given level, m is the number of frequent episodes in that level, since all pairs of frequent episodes need to be compared for a prefix-suffix match.

The algorithm for counting the set of candidates episodes is given in Algorithm 1. The count or frequency measure is based on nonoverlapped occurrences [Laxman et al. 2005]. Two occurrences of an episode are said to be nonoverlapped if the events in one occurrence appear between the events in the other occurrence. This notion most naturally eliminates the problem of trivial matches highlighted in Patel et al. [2002] where a match is found between two slightly shifted segments of the time series. Algorithm 1 takes as input the event-sequence and a set of candidate episodes and returns the set of frequent episodes for a given frequency threshold θ. The algorithm counts the maximum number of nonoverlapped occurrences of each episode with the interevent time constraint $(0, T]$. This approach also allows repeated symbols or events in the episodes.

ALGORITHM 1: Counting occurrences of serial episodes with interevent time constraint $[0, T)$

```
Input: 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 $S = \{(E_i, t_i)\}$.
Output: 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 $\alpha.count \geq \theta\}$
```



(a) Positive Example

(b) Negative Example

Fig. 9. Illustration of the proposed motif mining algorithm using example data.

To illustrate how the overall multivariate motif mining approach works, we use some synthetic examples. In Figure 9(a), we demonstrate a multivariate time series consisting of two components. A motif (corrupted by Gaussian noise) is embedded in this data. The motif comprises of a triangular waveform in the first series and simultaneously exhibits an “H”-shaped waveform in the second time series. This motif repeats five times in the given data.

First, the time information is stripped from the data and clustering is performed with number of clusters, $k = 5$. The cluster transitions are overlaid in the plot. These transitions are used to encode the multivariate numeric data into symbolic form and serve as the input to frequent episode mining. One episode is discovered which occurs five times and this episode is unpacked into the original data and overlaid on the dataset as shown. This serves as an example of a motif that we embedded and were able to uncover.

Next in Figure 9(b), we take the preceding time-series data and randomize the time points so that the higher-order information necessary to exhibit the motif is destroyed. The clusters still remain the same but there are no frequent episodes discovered in the data, and hence there are no motifs.

The worst-case time complexity of the counting algorithm is given by $O(lnm)$, where l is the number of events in the data sequence, m is number of candidate episodes, and n is the size of the episode. The algorithm makes one pass of all the events in the event sequence and every time an event that belongs to an episode is seen, the data structure s for the episode is updated. A hash map is used to efficiently locate only a subset of relevant episodes for each event seen in the event sequence. Since our method allows repeated symbols, in the case of such episodes the same event can update s structure at most n times. Therefore if the level-wise growth of candidates is sufficiently arrested by a suitably chosen threshold, the algorithm scales linearly with data size.

Recall here that the events in mined frequent episodes correspond to transitions from one symbol to another. Our hypothesis here is that if motif occurrences are matched at transitions under an intertransition gap constraint, then the corresponding time-series subsequences will match under a suitable distance metric. In addition the episode mining framework allows for robustness to noise and scaling. The distance metric under which such motifs can be shown to be similar needs further investigation. Nonetheless this technique is found to be very effective in unearthing similar time-series subsequences in real datasets.

4.3. Sustainability Characterization

It is difficult (and subjective) to compare two motifs in terms of their sustainability impact by inspecting them visually. Therefore, it is necessary to quantify the sustainability of all motifs by computing a sustainability metric for them. This would enable quantitative comparisons between motifs; their categorization as “good” or “bad” from the sustainability metric point-of-view; and, furthermore, this information could be used to provide guidance to an administrator or a management system regarding the most “sustainable” configurations of the chiller ensemble under a particular load. There are several sustainability metrics, such as power consumed, carbon footprint, and exergy loss [Shah et al. 2008], where exergy is defined as the energy that is available to be used. Note that typically optimizing a sustainability metric, such as power consumed, also minimizes the total cost of operation.

In this article, we estimate two sustainability metrics for each motif: (1) the average COP of the motif; and, (2) a metric reflecting the frequency and amplitude of oscillations in utilization values. The average COP is calculated using Eq. (1), where the load during motif i, averaged over all its occurrences, is used as L_i and the averaged power consumption of the motif as P_i. The power is estimated using Eq. (2) with an averaged constant COP of 3.5 for the air-cooled and 6 for the water-cooled chillers. The COP of a motif quantifies the cooling effectiveness of the ensemble during that motif. In order to estimate the frequency of oscillations of a motif, we compute the number of mean-crossings, that is, the number of times the utilization crosses the mean value. This is very similar to number of zero-crossings that is commonly used in speech processing for estimation of frequency. This, together with standard deviation of a motif, allows oscillatory behavior to be compared.

4.4. Results

We applied our motif mining methodology to chiller data obtained from a large HP production data center covering 70,000 square feet with 2000 racks of IT equipment. Its cooling demand is met by an ensemble of five chiller units. The ensemble consists of two types of chillers: three are air-cooled and the remaining two are water-cooled. The data characteristics are given in Table II. We mined 20 clusters from this data, with a view toward identifying well-separated clusters.

4.4.1. Motifs. All the parameters using in mining motifs in multivariate time-series data is shown in Table I. In all, 22 motifs were discovered in the chiller utilization data whose qualitative properties are summarized in Table V (more on this later). From a quantitative point of view these motifs can be clustered into groups based on load. One such group (Group II) is depicted in Table III with other quantitative measures. Although each group has very similar load levels, the COP within a group varies with

Cluster	Power	Avg Util	Dev.	#Samples	Color
ID	(in tons)	(in %)			
4	944.38	185.34	187.88	681	🟥
14	859.10	168.60	204.35	699	🟥
20	832.14	163.31	151.58	781	🟧
18	829.87	162.86	151.09	772	🟧
12	768.14	150.75	47.68	1485	🟨
19	706.24	138.60	166.98	1382	🟩
1	666.01	130.71	109.98	653	🟩
5	659.68	129.47	115.78	1299	🟩
8	597.20	117.20	40.71	3723	🟩
6	583.26	114.47	143.87	667	🟩
9	479.37	94.08	81.38	949	🟩
16	464.30	91.12	164.41	562	🟩
17	456.84	89.66	71.88	1504	🟦
15	431.86	84.75	80.54	975	🟦
11	395.65	77.65	215.82	1250	🟦
7	348.92	68.48	59.97	2330	🟦
3	326.92	64.16	58.03	4313	🟦
2	325.93	63.97	50.96	2280	🟦
13	312.77	61.38	164.51	970	🟪
10	268.89	52.77	21.08	1388	🟪

Fig. 10. Characterization of operating states of the ensemble of chillers using clustering.

Table I. Parameters Used in Mining Time-Series Motifs

Parameter	Value
Number of clusters	20
Support Threshold	10 or more occurrences
Inter-event gap lower-bound	5 min
Inter-event gap upper-bound	60 min

Table II. Dataset I

Number of data points	Date range	Number of Air-cooled Chillers	Number of Water-cooled Chillers

28,663
(≈480 hours)
	July 2, 2008 to July 7, 2008;
	Nov 27, 2008 to Nov 30,2008;
	Dec 16, 2008 to Dec 26, 2008
		3	2

the motifs. In Group II, for instance, motif 8 has a COP of 4.87, while motif 5 is significantly more efficient at a COP of 5.4. This information provides key insights to an administrator regarding motifs that are more energy efficient compared to others. Furthermore, this information could be codified into rules for setting chiller ensemble configuration based on the current load.

Table IV shows the most efficient and the least efficient motifs for Group II based on power consumption, determined by the motif's COP value. Also shown are the potential power savings assuming that the least efficient motif could be transformed into the most efficient one.

Table III. Summary of the Quantitative Measures Associated with Motifs

Motif	Load (KW)	Power (KW)	COP	MC	Std	Span (min)	N
Group II							
8	**2028**	**417**	**4.87**	**.042**	**5.05**	**50**	**10**
2	2073	400	5.18	.048	4.28	113	17
5	**2076**	**384**	**5.4**	**.048**	**4.25**	**49**	**35**
4	2083	387	5.38	.048	4.31	82	23
6	2119	422	5.02	.044	4.08	51	23
3	2121	414	5.13	.042	4.31	119	11
1	2123	395	5.38	.046	4.39	172	10

Shown here are statistics for motif in one of the load groups.
MC – number of mean-crossings per minute.
N – number of occurrences of the motif.

Table IV. The Most and Least Efficient Motif for Each Group and the Potential Power Savings if the Operational State of the Chiller Ensemble Could be Transformed from the Least to the Most Efficient Motif

	Load (KW)		Most Efficient	Least Efficient	Potential Power Savings	
	Ave.	Std	Motif	Motif	KW	%
Group II	2089	35	5	8	41	9.83%

Table V provides a qualitative description of all the motifs. They are grouped together manually based on similar utilization behavior of each of the chillers in the ensemble. Two factors are considered: the average utilization during the motif, and the oscillatory behavior. The utilization level is labeled as low (L), medium (M), or high (H) based on the average utilization ranges of (0, 35), (35, 65), or (65, 100), respectively. The oscillatory behavior is marked (1) none (N) – which indicates steady values with minor variations; (2) small (S) – which indicates the values show oscillations that have a small amplitude; and finally, (3) large (L) – which indicates oscillations with a large amplitude. Furthermore, units that are not operating are marked "OFF".

4.4.2. Comparing Motifs 5 and 8. We investigate now in more detail the differences between motifs 5 and 8 of Group II. While both motifs 5 and 8 have three chillers turned on, they are of different types. In motif 8, all three operating chillers (C1, C2, and C3) are air-cooled. In motif 5, two air-cooled (C1 and C2) and one water-cooled chiller (C4) are running. The qualitative behavior of the chillers in the two motifs are shown in Table V. In motif 5, one chiller runs at high utilization (C4 at 66.5%), while the other two run at low utilizations (11.3% and 33.8%). In motif 8, one chiller runs at low utilization (17.63) while the other two operate at the medium range (49.1% and 44.3%). The amount of oscillatory behavior in the two motifs is about the same as indicated by the average rate of mean-crossings and standard deviation in Table III. In both the motifs, one air-cooled chiller shows large oscillations.

4.4.3. Economical Incentives for Sustainable Operation. Business growth is an important factor in planning for investment in infrastructure. Chillers, of the kind described in this article, can cost upwards of $150k and typically depreciate over 15–20 years. In the absence of intelligent management, the annual operating cost, even at 50% load, can be equally high. Maximizing utilization and efficiency is crucial to achieve a favorable return on investment. Operating chillers among favorable motifs that satisfy these conditions will enable IT administrators to sustain growth in business without major capital or maintenance costs. It will not only prolong the useful life of the existing equipment but postpone expensive retrofits and further infrastructure investments.

The cost savings by operating in favorable motif regimes can be quite easily characterized by multiplying the kW savings by $0.11 \times 24 \times 365$ to obtain the annual savings in

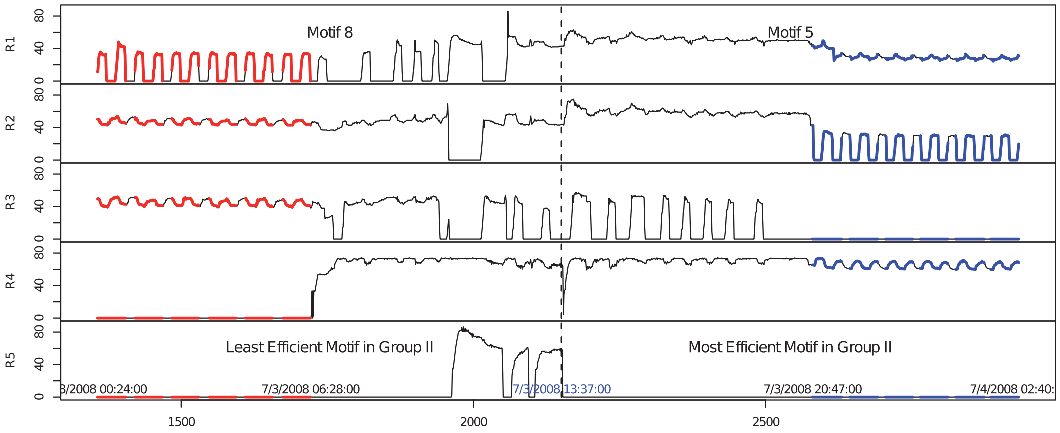


Fig. 11. Two motifs with similar load but widely varying levels of efficiency. Observe how the least efficient motif transitions to the most efficient motif in the given time period.

Table V. Summary of the Qualitative Behavior of all the Motifs

Motifs	Num of Units ON	Chiller behavior									
		C1		C2		C3		C4		C5	
		Util.	Oscil.	Util.	Oscil.	Util.	Oscil.	Util.	Oscil.	Util.	Oscil.
7, 15	2	M	L	OFF		H	S	OFF		OFF	
13	2	L	L	OFF		M	S	OFF		OFF	
1,2,4,5,9,10	3	L	N	L	L	OFF		H	N	OFF	
3,6,11,12	3	OFF		L	N	L	L	H	N	OFF	
8,14	3	L	L	M	N	M	S	OFF		OFF	
16	3	L	S	M	N	M	S	OFF		OFF	
17	4	M	N	M	L	OFF		H	N	H	S
18	4	M	N	M	S	L	L	H	N	OFF	
19	4	M	S	L	L	OFF		H	N	H	L
20	4	M	N	L	L	L	L	H	N	OFF	
21	4	OFF		M	S	L	L	H	N	H	L
22	4	OFF		H	N	M	L	H	N	H	S

The ones showing similar behavior are grouped together.
Utilization (Util.) – Low (L), Medium (M), High (H).
Oscillations (Oscil.) – None (N), Small (S), Large (L).

dollars ($). Here, $0.11 is assumed to be the cost per KWh (kilo watt hour) of electricity. In this case switching from motif 8 to motif 5 gives us a nearly $40,000 in potential annual savings! Extrapolating this cost saving to other similar motifs gives us an idea of the utility of data mining algorithms in helping achieve cost effectiveness.

4.4.4. Carbon Footprint Calculation. Saving 1kWh of energy is equivalent to preventing release of 0.8 kg of carbon dioxide into the atmosphere. Based on the energy savings number, we can calculate the reduced carbon footprint.

$$\begin{aligned}\text{Saved Power} &= 41\text{kW}\\ \text{Saved Energy (annually)} &= 41\text{kW} \times\ 8760\text{ hrs}\\ \text{Reduction in Carbon footprint} &= 41\text{kW} \times\ 8760 \times\ 0.8\\ &= 287328\text{ kg CO}_2\end{aligned}$$

Observe that this is just the operational footprint; there is also an “embedded carbon footprint” of the chiller unit (as added in its manufacturing process). By maximizing operational life and utilization, we are managing this embedded carbon in the equipment as well. In other words, we are limiting the increase in embedded carbon in the environment while delivering the cooling required.

5. MODELING THE INFLUENCE OF EXTERNAL VARIABLES

While motifs are very useful in assessing repetitive patterns of occurrence, they constitute only a relative minority of the state space that the chiller ensemble operates in. Another interesting goal is to characterize the regions of state space occupied by the chiller ensemble and correlate them to the external conditions that the data center, and in particular, the chiller ensemble is subjected to. Once again, we focus on the utilization vectors of the same chiller system. But after a recent upgrade, the installation now has seven chillers instead of five as described earlier: five air-cooled and two water-cooled. We demonstrate the use of association analysis techniques to identify key conditions that underlie sources of inefficiency.

5.1. Approach

We discretize the utilization state space of chillers by clustering using seed prototype vectors. The seed prototypes are obtained by combining low, medium, and high utilization values for individual chiller units to obtain a total of $3^7 = 2187$ prototype utilization vectors. These prototypes are used to seed clusters for a k-means algorithm and clusters that are assigned zero members are removed from the analysis. The resulting clusters are still too numerous for human consumption; consequently, we applied hierarchical clustering with distance between pairs of clusters as the average distance between all pairs of points across the two clusters. The agglomerative process of merging clusters was stopped at a level where there was sufficient separation between the clusters being merged. This ensures that the states represent qualitatively distinct regions. The prototypes for the final states are shown in Figure 12. Each state is annotated with the percentage fraction of total time spent by the system in that state, the average time spent there before moving into a different state, and the average power consumed. The states are colored using the rainbow palette based on the power consumed in a state, red being the highest power consuming state and blue the lowest power state.

As a preliminary analysis, we can organize the states from Figure 12 in many different ways and assess their characteristics. For example, we can group the states based on power consumption and investigate the fraction of time that is spent in, say, high power states. We can assess the states by asking where the system spends most of its time and determine the profile of these states in terms of the different chillers. This can be used to validate if the load balancing objectives of the system are being met. Finally, we can also investigate which chillers are on/off in different states. In states 1, 5, and 8 none of the water cooled chillers is running. This is unusual for this installation since the scheduling policy mandates the use of at least one water-cooled chiller at all times to handle the base cooling load.

Given the same external conditions, the chiller ensemble is expected to exhibit similar power consumption characteristics. For example, for a given cooling load the chiller system should ideally consume the same amount of power. In practice, the system may not be able to operate in the most optimal mode to meet the required load under given external conditions. In CAMAS, we analyze the distribution of states observed under given external conditions and investigate its characteristics.

The external conditions known to influence the operations of chiller are cooling load, supply water temperature, ambient air temperature, and ambient water temperature. The cooling load is the total cooling requirement of the data center and office spaces that are served by the chiller ensemble. The supply water temperature is the temperature at which chiller water is supplied to the air-conditioning units. The set point of the supply water temperature impacts the coefficient of performance (COP) of the chillers.

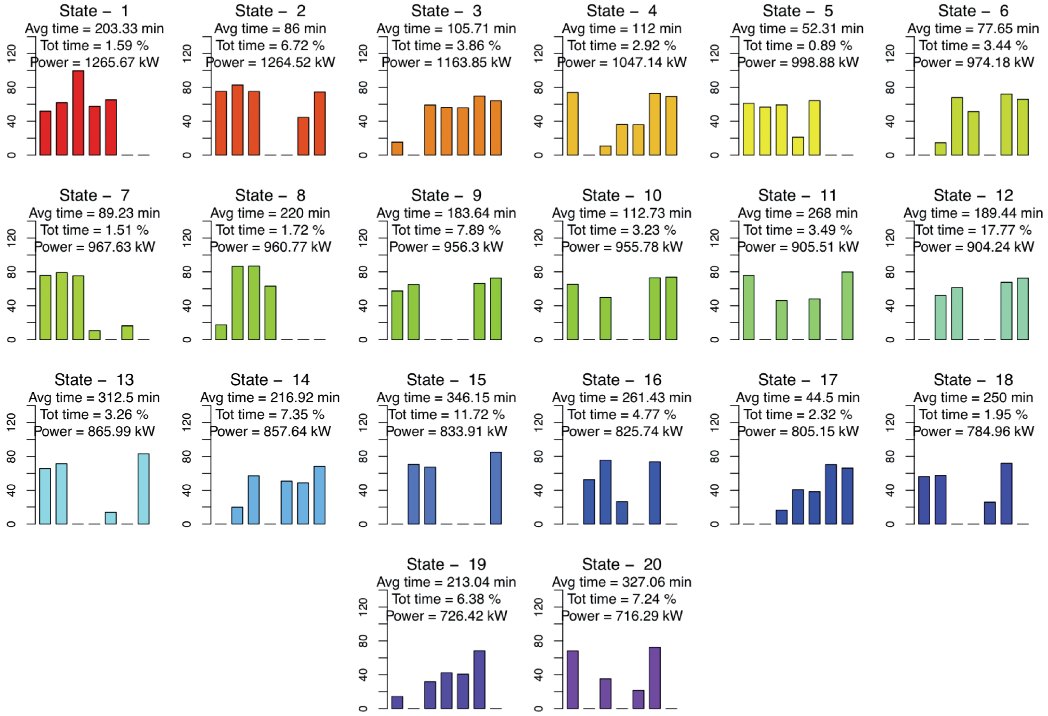


Fig. 12. Prototype vectors of the 20 discrete states of the chiller ensemble. Each bar represents a particular chiller and its mean utilization level in a given state of the system. The first five bars in each plot indicate the air-cooled chillers and the last two indicate the water-cooled ones.

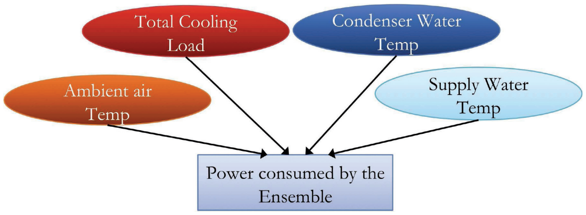


Fig. 13. External factors affecting chiller performance.

Table VI. Dataset II

Number of data points	Date range	Number of Air-cooled Chillers	Number of Water-cooled Chillers
77,356	Apr 15, 2010 to July 14, 2010	5	2

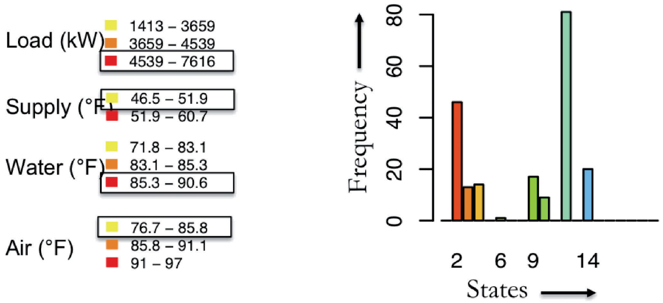
Similarly ambient air temperature and ambient water temperature (for water-cooled chillers only) affect COP. Once these external factors are held fixed, the distribution of states exhibited by the system exposes inefficiencies in operation; in particular, it can help identify high power states that can potentially be substituted by low power states (e.g., where fewer chillers are operating) to meet the same load under similar external conditions.

5.2. Results

The details of the data used in this analysis is given in Table VI. For the purposes of this analysis, we discretize the time-series for load, supply temperature, ambient

Table VII. Parameters Used in Analysis of the Effect of External Factors

Parameter	Value
Number of clusters(discrete states)	20
Number of (equal-frequency) bins	
Load	3
Supply Temperature	2
Ambient-water Temperature	3
Ambient-air Temperature	3



(a) External Factors (b) State distribution

Fig. 14. Frequency distribution of different states under a given external condition.

water temperature, and ambient air temperature into three, two, three, and three levels, respectively (see Table VII). We use equal frequency bins for each level. Note that only two levels of discretization are used for supply temperature since the dynamic range of this variable is very narrow. Our hypothesis is that, under the same operating conditions, if we observe multiple states with widely varying power consumption profiles then this is an indicator of inefficiencies that exist in the system. It should be potentially possible to move the system from an inefficient high power state to a more efficient state to meet the same operation conditions.

Consider Figure 14 which depicts the distribution of states for a given setting of external conditions (i.e., load is high, supply temperature is low, ambient water temperature is high, and ambient air temperature is low). Under these conditions, notice that State 2 (with average power consumption = 1256kW) and State 12 (with average power = 904kW) are observed with relatively high frequency. In this example if we are able to switch from State 2 to State 12, the energy savings over a three-month period is roughly about 20,336 kWh. More examples of such skewed distributions are shown in Figure 15.

Furthermore, we can use the external factors as a handle to reason about how the system can be moved from its current state to a more desired state. For example, we can investigate if the system can be moved from State 2 to State 12, by changing some of the external conditions. The example in Figure 16 shows such a scenario where load decreases with other factors remaining the same. We see that the high power states are no longer seen. Here the focus is on changes in state distribution given a change in external conditions.

These examples illustrate that there are numerous opportunities over the course of a chiller ensemble operation where changes in modes of operation (states) would result in energy savings. Our next step is to use these ideas to develop a comprehensive framework for reasoning about chiller states and transitions.

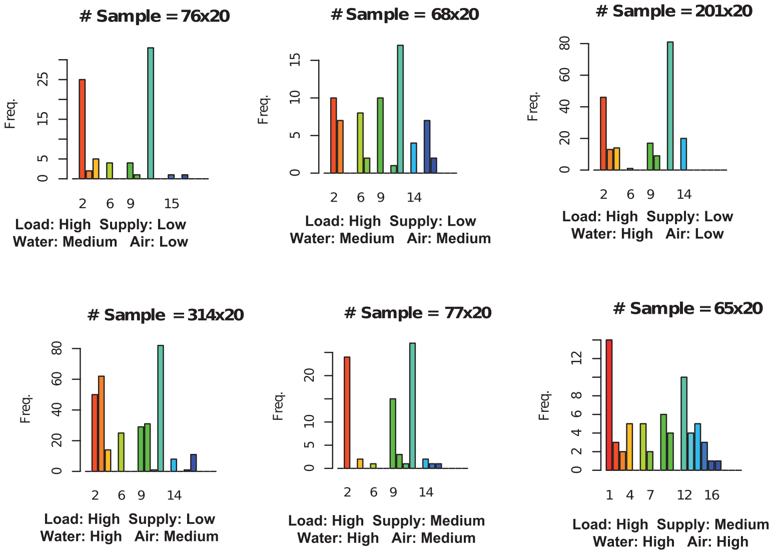


Fig. 15. More examples of skewed state distributions. It can be seen that there are high power states and low power states under similar external conditions.

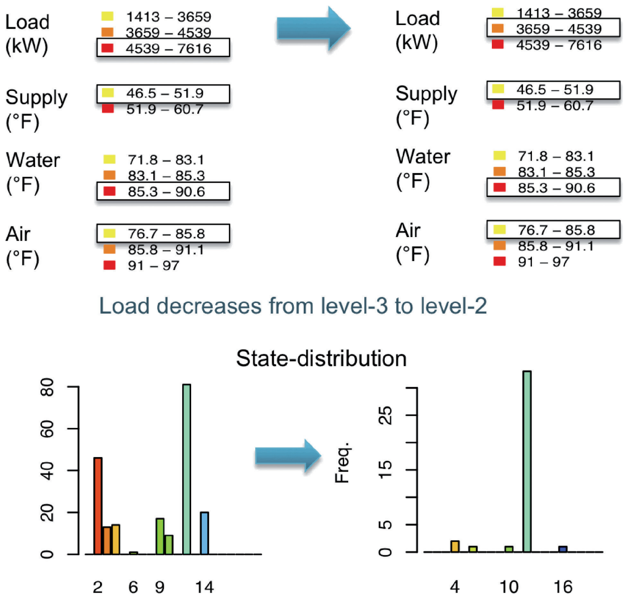


Fig. 16. Change in state distribution with change in load.

6. REASONING ABOUT STATES AND TRANSITIONS

In this section, we demonstrate the overarching framework of CAMAS to help monitor and analyze a chiller ensemble, while keeping the formulation general enough so that it could be extended to any similar infrastructure.

6.1. Approach

As shown in Figure 5, data reduction is performed in a few different ways. The raw time-series data is compressed using piece-wise aggregate approximation following discretization using equal frequency bins. This helps capture the average dynamics of the variables. A higher-order aspect of the time series involves repeating or oscillatory behavior. This is inferred by mining frequently repeating motifs or patterns in the time series, as described in Section 4. This information is integrated with the average behavior by recording the windows in which a time series exhibits motif patterns. Finally, it is also pertinent to mine relationships involving “control” actions that were taken during the operation of the chiller units. In this article, we focus on ON/OFF actions.

A graphical model in the form of a Dynamic Bayesian Network (DBN) is learnt from the preceding data. This model captures the dependencies between the variables in the system over different time lags. Here, we focus on the target variable of utilization and seek to identify suitable parents for modeling in the DBN. Unlike classical methods to learn BNs [Friedman et al. 1999], we demonstrate a powerful approach to learn DBNs by creating bridges to the frequent episode mining literature [Patnaik et al. 2009a]. To apply the learned DBN, we define states of the system by clustering together the combined utilization of the chiller units. This allows the operation of the chiller ensemble to be represented as a sequence of state transitions. We now use the dependencies and (conditional) independency relationships found in learning the graphical model to find the most probable explanation behind the state transitions. This framework can then be applied for activities like data center diagnostics, performance improvement, load balancing, and preventive maintenance.

As is well known, a Bayesian Network (BN) is a graphical model denoted by $B = (G, P)$ where G is a Directed Acyclic Graph (DAG) and P is a set of conditional probability distributions. The graph $G = (V, E)$ consists of a set of nodes V representing the random variables $\{X_1, \ldots, X_N\}$ in the system and a set of directed edges E. Each directed edge in E, denoted by $i \rightarrow j$, indicates that random variable X_i is a parent of random variable X_j. The conditional probabilities in P are used to capture statistical dependence relationships between child nodes and parent nodes. In particular, given a random variable X_i in the graph, we denote by $\text{par}(X_i)$ the set of random variables that are parents of X_i . The statistical dependence between X_i and its parent nodes $\text{par}(X_i)$ is captured by the conditional probabilities $P(X_i|\text{par}(X_i)$.

To model a discrete-time random process $\mathbf{X}(t), t = 1, \ldots, T$; $\mathbf{X}(t) = [X_1(t)X_2(t) \cdots X_N(t)]$ as studied here, we use the more expressive formalism of dynamic Bayesian networks. In particular, we focus on time-bounded causal networks, where for a given $w > 0$, the nodes in $\text{par}(X_i(t))$, parents for the node, $X_i(t)$, belong to a w-length history window, $[t - w, t)$. Note that parent nodes cannot belong to the current time slice t for $X_i(t)$.

This assumption limits the range-of-influence of a random variable, $X_k(t)$, to variables within w time slices of t and also indicates that the random variables $X_i(t)$ and $X_j(t)$ are conditionally independent given their corresponding parent sets in the history window. Further, we also assume that the underlying data generation model is stationary, so that joint statistics can be estimated using contingency tables.

The learning of network structures involves learning the parent set, $\text{par}(X_i(t))$, for each $X_i(t)$, $i = 1, \ldots, N$. In this work we assume that there are no spurious independencies in the data, that is, if a random variable $X_j(t - \tau)$ is a parent of $X_i(t)$, then the mutual information $I(X_i(t); X_j(t - \tau)|\mathcal{S})$ conditioned on a subset $\mathcal{S} \subseteq \text{par}(X)$ is always greater than zero. Moreover time-bounded causality enables us to learn the parents of each node $X_i(t)$ independent of any other node in the same time slice. We use a greedy approach to learn the parent set of each node $X_i(t)$. And proceed by adding a node which

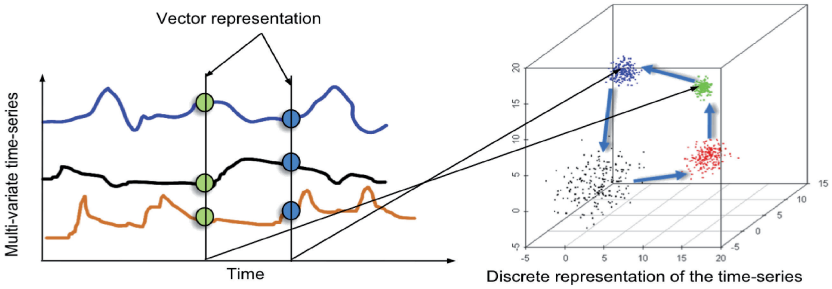

Vector representation
Multi-variate time-series
Time
Discrete representation of the time-series

Fig. 17. Illustration of rendering multivariate time-series data as a sequence of state transitions using clustering. The arrows are drawn between corresponding vectors on the time-series plot and the 3D plot.

has the highest conditional mutual information to the parent set of $X_i(t)$ as shown in Eq. (4).

$$\text{par}^{i+1}(X_i(t)) \leftarrow \text{par}^{i}(X_i(t)) \cup \arg\max I(X_i(t); X_j(t-\tau)|par^{i}(X_i(t))) \qquad (4)$$

The search is continued until the number of nodes in the parent set is k (where k is an user-defined parameter) or the conditional mutual information drops to zero. The structure learning is then followed by maximum likelihood estimation of the conditional probability tables.

6.2. State Transition Modeling

In analyzing complex systems it is usual to try and summarize the operating characteristics into a finite set of states where the systems spends most of its time. The utilization information of the chiller ensemble is important from both the aspect of efficiency and sustainability. Therefore we define the states of our system in terms of the combined utilization of all the chiller units.

In order to obtain a finite set of states, we fist perform a k-means clustering on the utilization vectors and use the cluster labels as symbols to encode the multivariate time series of the combined utilization. Thus the multivariate series of utilizations is now encoded as a single long state sequence.

Over the state sequence the points of interest are the times where the system moves from one state to another. The exact transition times can be affected by the clustering scheme used (e.g., number of clusters) but on the average they capture changes in the operating characteristics.

An interesting question to ask now is: What causes the state transitions to take place? In the context of our modeling, this question translates to: what factors in the system cause the utilization of the chiller to go up or down? This can be answered by observing the changes that take place in the system around the state transitions. The problem with this approach is the lack of availability of sufficient data around the change points. We propose an alternative approach where we decompose the question by asking what causes each of the changes in the chiller units. From the graphical model already learnt we know the set of variables in the system that directly affect the utilization. These variables belong to the parent set of the utilization nodes. The task of finding the most probable causes for a transition amounts to evaluating the following probability distribution:

$$\Pr(\mathcal{S}) = \prod_{i \in \text{UtilizationVariables}} \Pr(\mathcal{S}_i | X_i(t) = a_i, X_i(t-1) = b_i), \qquad (5)$$

Table VIII. Dataset III

Number of data points	Date range	Number of Air-cooled Chillers	Number of Water-cooled Chillers
28,289 (≈ 576 hrs)	Oct 21, 2009 to Nov 13, 2009	3	2

Table IX. Parameters Used in State Transition Modeling

Parameter	Value
Number of clusters k (discrete states)	10
Window size (for piece-wise avg. aggregation)	15 min
Number of history windows (used in DBN learning)	5

Table X. A Few Important System Variables in the Chiller Ensemble Data

System Variable	Description
AC_CH(i)_EVAP_E_TEMP	Temperature of water entering air-cooled chiller i
MAIN_HEADER_TEMP_1_F	Water temperature at distribution header
WC_CH(i)_IN_TEMP	Temperature of water entering water-cooled chiller i
WC_CH(i)_OUT_TEMP	Temperature of water leaving water-cooled chiller i
AC_CH(i)_RLA	Percentage utilization of air-cooled chiller i
WC_CH(i)_RLA	Percentage utilization of water-cooled chiller i

where $\mathcal{S}_i = \text{par}(X_i(t)) \setminus X_i(t-1)$ and $\mathcal{S} = \cup \mathcal{S}_i$. The most likely values that $\mathcal{S}$ takes can be considered the best explanation of the transition. Here t is the time at which a state transition occurs, a_i, b_i are the discrete values the utilization variable takes before and after the state transitions. These can be approximated by the cluster centers of each cluster used to define a state.

6.3. Results

We applied our methods to chiller data obtained from a the same data center described in Section 4.4. All the parameters used in state transition modeling using DBNs are listed in Table IX.

During the period from October 21, 2009 to November 13, 2009 the cooling demand of the data center was met by an ensemble of five chiller units (see Table VIII). The ensemble consists of two types of chillers: three air-cooled and the remaining two water-cooled. The data collected over this period totaled to over 576 hours of data consisting of over 47 variables.

6.3.1. Graphical Model. For learning the graphical models, the raw time-series data was aggregated over windows of 15 minutes and augmented with results from motif mining conducted over the same data. The motif information is in form of a binary valued sequence for each time series. The binary values indicate whether or not motif occurrences were seen in each 15 minute window. A finer-grained representation of the motifs will be explored in the future.

Since utilization of a chiller mostly determines its energy consumption, we have shown that portion of the learned dynamic Bayesian network in Figure 18. Three air-cooled chiller utilization variables: AC_CH1_RLA, AC_CH2_RLA, AC_CH3_RLA; and two water-cooled ones: WC_CH1_RLA and WC_CH2_RLA are shown together with their parents and grandparents. In the DBN, almost every node has itself from the previous time slice as one of its parents, which is expected since the data is continuous valued time-series data. In order to assist in creation of a causal Bayesian network, partial domain information was incorporated through constraints such as limiting nodes that could become parents. Three utilization variables show direct dependency on water temperature. This is, again, expected since the aim of the chiller ensemble is to maintain water temperature close to the set point.

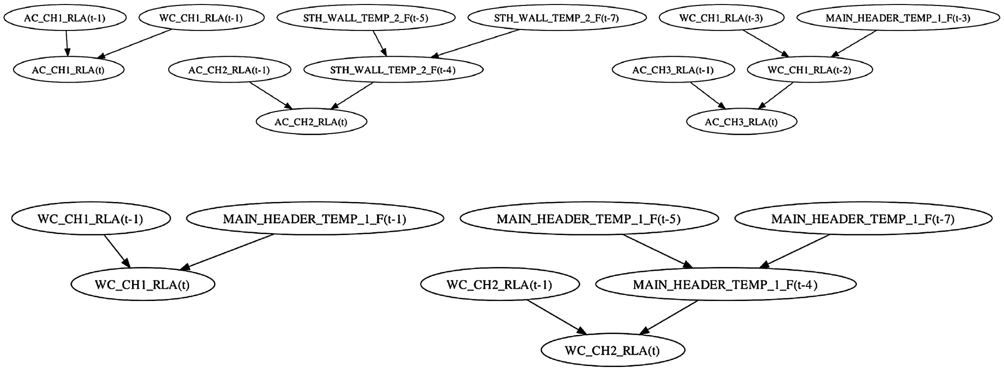


Fig. 18. Graphical model learned from the discretized time-series data. Shown are the parent nodes for only the utilization variables of the five chiller units.

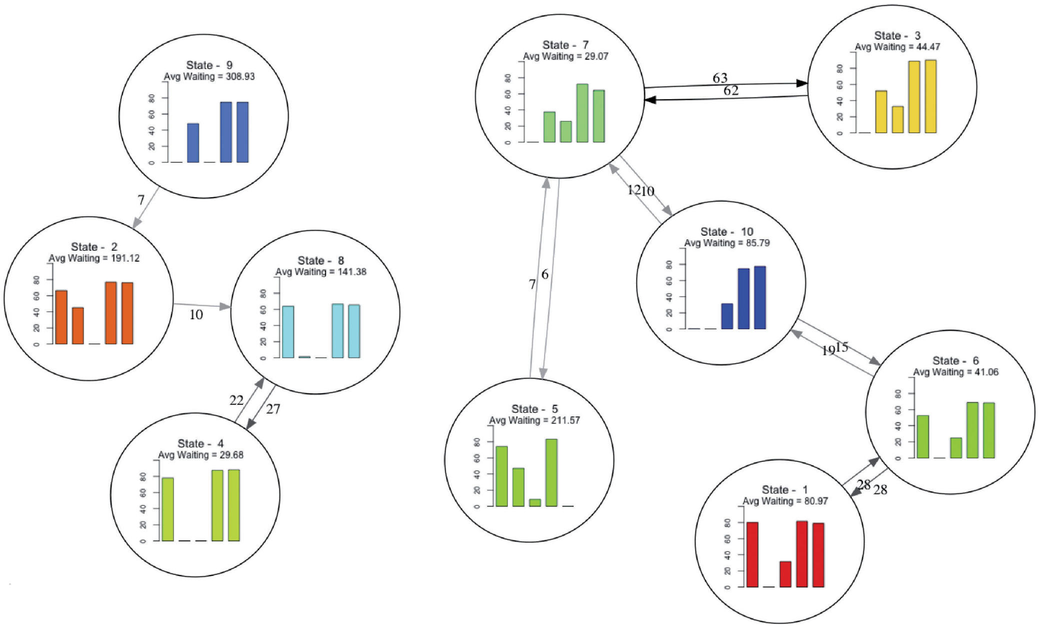


Fig. 19. Transitions in the state-based representation of the combined utilization of the chiller ensemble. Edges representing 5 or fewer transitions are not shown. And edges representing more than 5 transitions are labeled by the actual counts.

More interesting relationships are revealed by two air-cooled units (1 and 3) that directly depend on the utilization of a water-cooled chiller. This would indicate that these two chillers ramp up or down based on how the water-cooled chiller (which has higher capacity) is operating.

6.3.2. State Transitions. The operational states of the system, as shown in Figure 19, are obtained by clustering together the combined utilization of all the chillers in the ensemble. Here we use k-means clustering with $k = 10$ and organize the states in decreasing order of total power consumption. Since the objective of this work is to operate a chiller ensemble more energy efficiently, we color code the system states to reflect their power consumption. The color red indicates the maximum power consumption

Table XI. List of Most Likely Value Assignments of the Parent-Set of Node AC_CH1_RLA i.e. Utilization of Air-Cooled Chiller 1

$\text{par}(X_i(t))$	Delay	Value	$\Pr(\text{par}(X_i(t))\|X_i(t), X_i(t-1))$
WC_CH1_RLA	1	$(75.37, 77.38]$	0.27
WC_CH1_RLA	1	$(72.36, 75.37]$	0.18
WC_CH1_RLA	1	$(77.38, 79.71]$	0.18

(about 3298 KW) while the color blue indicates the least consumption (2148 KW). Also shown, for each state, are the average utilization values of the five chillers as a histogram with first three (starting from the left) being air-cooled ones and the last two being water-cooled units. The time spent by the system in each state is also listed (maximum in state 9, while least in state). The arrows show the transitions between states, with the gray-scale indicating the frequency of the transition (darkest implying most often).

The system states are mainly characterized by the number and kind of chiller units operating, their utilization levels, and power consumption. Some states are quite similar, for example, states 3 and 7, which have the same chiller units operating with not much difference in their utilization levels. Other states show marked difference, for instance, state 10 has three chillers operating (one air-cooled and two water-cooled) while state 6 has four working units (two air-cooled and two water-cooled). Note that consequently state 10 consumes less power than state 6. A data center chiller operator will be interested in understanding the variables that influence transitions from state 10 to 6.

Transition: State 10 → State 6. When the chiller ensemble makes a transition from state 10 to state 6, air-cooled chiller-1 turns on. The graphical model can be queried to provide the most probable explanation for this change. Using the model from Section 6.3.1, we estimate values of parent of utilization node when these state transitions take place, as listed in Table XI. Note that utilization levels of the parent (WC_CH1_RLA) are high when these transitions take place. These and other similar insights would facilitate more energy-efficient management of the chiller resources.

The preceding models can be generalized to incorporate system alarms as nodes in the graphical model with the objective of discovering the variables that most significantly influence a particular alarm. This causal inference would provide valuable information to a data center chiller operator on corrective steps needed to handle the alarm. Furthermore, operator actions would be added to the model to enable discovery of dependencies between state transitions and such actions. This would allow an operator to query what actions could be taken (if any at all) to move the system from a less energy-efficient state to a more efficient one.

7. RELATED WORK

7.1. Mining Systems and Installations

Many researchers have explored the use of data mining and machine learning to troubleshoot, analyze, and optimize computer systems and installations. There are a variety of projects with a diversity of foci, ranging from the mechanical equipment that power and cool the data center, to network-level diagnostics, to user-level applications and the system calls they make. For instance, modeling of rack-level temperature data specifically in relation to CRAC (computer room air conditioning) layout has been undertaken in Bautista and Sharma [2007] and Sharma et al. [2007]. Optimization opportunities at multiple levels of smart center architecture have also been studied in Sharma et al. [2008]. More recent work [Marwah et al. 2010] focuses on sensor data mining to identify anomalous and deviant behavior. Other related work includes the InteMon system from CMU [Hoke et al. 2006a, 2006b] that dynamically tracks correlations among multivariate time series [Papadimitriou et al. 2005], a characteristic of many sensor streams. Interactive visualizations for system management have also been investigated [McLachlan et al. 2008]. Diagnosing network-level traffic, for example, for volume anomalies, has been studied in Lakhina et al. [2004]. The performance of Internet-scale applications deployed over multiple data centers is characterized using automatically mined signatures in Bodik et al. [2008]. Online failure prediction with applications to network security is discussed in Gu et al. [2008]. Failure prediction using event logs has also been explored [Liang et al. 2007], especially in the context of IBM Blue Gene systems. Xu et al. [2009] use text analysis and mining capabilities, in conjunction with some source-code analysis, to study console logs from applications.

7.2. Approaches to Model Time-Series Data

There are numerous formalisms available to model time-series data and good surveys are in Ding et al. [2008] and Lin et al. [2003]. One broad class of representations conducts global decompositions of the time-series (e.g., PCA, DFT, wavelets) and aims to use only the most significant components to model the series. Alternatively, piece-wise representations are easier to compute and also lend themselves to a streaming mode of operation. SAX [Lin et al. 2007] performs a piece-wise aggregate approximation (the aggregate refers to the notion of modeling the given single time series by a linear combination of multiple time-series, each expressed as a box basis function) and symbolize the resulting representation so that techniques from discrete algorithms can be adapted toward querying, matching, and mining the time series. In our work, we have aimed to identify appropriate representations of multivariate time-series data that are efficient to compute as well as amenable to characterizing sustainability.

7.3. Finding Motifs in Time-Series

Motif mining is the task of finding approximately repeated subsequences in time series, and studied in various works, such as Chiu et al. [2003], Lin et al. [2002], Patel et al. [2002], and Yankov et al. [2007]. Mining motifs in symbolized representations of the time series can drawn upon 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, 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 nontrivial matches. A contrasting framework, referred to as frequent episode discovery, is an event-based framework that is most applicable to symbolic data that is not uniformly sampled [Laxman et al. 2005, 2008; Mannila et al. 1997; Patnaik et al. 2008]. This enables the introduction of junk, or “don’t care” states, into the definition of what constitutes a frequent episode. We have shown how by mining motifs in change point data, we can identify suitable oscillatory patterns and how they can be related to sustainability metrics.

7.4. Dynamic Bayesian Networks

As a class of probabilistic models, Dynamic Bayesian Networks (DBNs) are an outgrowth of Bayesian networks research, although specific forms of DBNs (e.g., HMMs) have a long and checkered history. Most examples of DBNs can be found in specific state space and dynamic modeling contexts [Wang et al. 2008]. In contrast to their static counterparts, exact and efficient inference for general classes of DBNs has not been studied well. Similar to our work, Nielsen and Jensen [2005] use a (dynamic) Bayesian network model of a production plant to detect faults and anomalies. Unlike this work, however, we show how such networks can be efficiently learned using frequent episode mining over sensor streams. Furthermore, our networks are defined over clustered representations of the time-series data rather than raw sensor signals.

8. DISCUSSION

We have demonstrated a powerful approach to data mining for data centers that helps situate trends gathered from sensor streams in the context of sustainability metrics useful for the data center engineer. In particular, we have attempted to raise the level of abstraction with which a data center administrator interacts with sensor streams. In CAMAS, we have demonstrated the ability to: (i) compose event occurrences to recognize episodes of recurring behavior, (ii) distinguish energy-efficient behavior patterns from others, and (iii) identify and suggest opportunities for better sustainable operation.

Our future work objectives fall into two categories. First, we would like to use temporal data mining to uncover a complete model of the data center cooling infrastructure which can then be tuned/controlled/optimized, thus making data mining an integral part of the data center architecture. To achieve this, we need to model the transfer function in a way that encapsulates workload changes, manual steering of chiller operation, and other intermittent transients and recoverable faults. One approach is to develop a hidden state transition model (e.g., a hidden process model [Shi et al. 2007]) where the actions by an operator or other changes in the operational environment denote the hidden states (since these are often not recorded) and the observed transitions could be the emissions emitted by the probabilistic model. As our second objective, we would like to analyze more components of a data center, including the IT and power subsystems. Although our current studies have focused on the chiller subsystem, we posit that our methods are general and can be targeted toward these other complex system modeling tasks.

Data Availability

All the datasets reported in this work are available for download at
http://nostoc.cs.vt.edu/files/camas-paper.

REFERENCES

Bautista, L. and Sharma, R. 2007. Analysis of environmental data in datacenters. Tech. rep., HP Labs.

Belady, C. 2007. In the data center, power and cooling costs more than the IT equipment it supports. *Electron. Cool. 13*, 1.

Bodik, P., Goldszmidt, M., and Fox, A. 2008. Hilighter: Automatically building robust signatures of performance behavior for small- and large-scale systems. In *Proceedings of the SysML Conference.*

Boucher, T., Auslander, D., Bash, C., Federspiel, C., and Patel, C. D., 2006. Viability of dynamic cooling control in a data center environment. http://escholarship.org/uc/Item/0wj7r61r.

Chiu, B., Keogh, E., and Lonardi, S. 2003. Probabilistic discovery of time series motifs. In *Proceedings of the International SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'03).* ACM, New York, 493–498.

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., and Keogh, E. 2008. Querying and mining of time series data: Experimental comparison of representations and distance measures. *VLDB J. 1*, 2, 1542–1552.

Friedman, N., Nachman, I., and Pe'er, D. 1999. Learning bayesian network structure from massive datasets: The "sparse candidate" algorithm. In *Proceedings of the 5th Conference on Uncertainty in Artificial Intelligence (UAI'99).* 206–215.

Gu, X., Papadimitriou, S., Yu, P. S., and Chang, S.-P. 2008. Toward predictive failure management for distributed stream processing systems. In *Proceedings of the 28th International Conference on Distributed Computing Systems (ICDCS'08).* IEEE Computer Society, Los Alamitos, CA, 825–832.

Hoke, E., Sun, J., and Faloutsos, C. 2006a. Intemon: Intelligent system monitoring on large clusters. In *Proceedings of the International Conference on Very Large Databases (VLDB'06).* VLDB Endowment, 1239–1242.

Hoke, E., Sun, J., Strunk, J. D., Ganger, G. R., and Faloutsos, C. 2006. Intemon: Continuous mining of sensor data in large-scale self-infrastructures. *SIGOPS Oper. Syst. Rev. 40,* 3, 38–44.

Kaplan, J. M., Forrest, W., and Kindler, N. 2008. Revolutionizing data center efficiency. Tech. rep., McKinsey CO.

Koomey, J. 2008a. Computing's environmental cost. *The Economist* (Print edition)

Koomey, J. 2008b. Power conversion in servers and data centers: A review of recent data and developments. In *Proceedings of the Applied Power Electronics Conference*.

Lakhina, A., Crovella, M., and Diot, C. 2004. Diagnosing network-wide traffic anomalies. In *Proceedings of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'04).* ACM, New York, 219–230.

Laxman, S., Sastry, P., and Unnikrishnan, K. 2005. Discovering frequent episodes and learning hidden markov models: A formal connection. *IEEE Trans. Knowl. Data Engin. 17*, 11.

Laxman, S., Tankasali, V., and White, R. 2008. Stream prediction using a generative model based on frequent episodes in event sequences. In *Proceedings of the International SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'08).*

Leake, J. and Woods, R. 2009. Revealed: The environmental impact of google searches. *Times Online*.

Liang, Y., Zhang, Y., Xiong, H., and Sahoo, R. 2007. Failure prediction in IBM BlueGene/L event logs. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'07).* 583–588.

Lin, J., Keogh, E., Lonardi , S., and Chiu, B. 2003. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'03)*. ACM, New York, 2–11.

Lin, J., Keogh, E., Lonardi , S., and Pat EL, P. 2002. Finding motifs in time series. In *Proceedings of the 2nd Workshop on Temporal Data Mining*. 53–68.

Lin, J., Keogh, E., Wei, L., and Lonardi, S. 2007. Experiencing SAX: A novel symbolic representation of time series. *Data Min. Knowl. Discov. 15*, 2, 107–144.

Mannila, H., Toivonen, H., and Verkamo, A. 1997. Discovery of frequent episodes in event sequences. *Data Min. Knowl. Discov. 1*, 3, 259–289.

Marwah, M., Sharma, R., Lugo, W., and Bautista, L. 2010. Anomalous thermal behavior detection in data centers using hierarchical pca. In *Proceedings of the 4th International Workshop on Knowledge Discovery from Sensor Data (SensorKDD'09).* ACM, New York.

Mclachlan, P., Munzner, T., Koutsofios, E., and North, S. 2008. Liverac: Interactive visual exploration of system management time-series data. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'08).* ACM, New York, 1483–1492.

Nielsen, T. D. and Jens EN, F. V. 2005. Alert systems for production plants: A methodology based on conflict analysis. *Symbol. Quant. Appro. Reason. Uncert*., 76–87.

Papadimitriou, S., sun, J., and Faloutsos, C. 2005. Streaming pattern discovery in multiple time-series. In *Proceedings of the International Conference on Very Large Databases (VLDB'05).* VLDB Endowment, 697–708.

Patel, P., Keogh, E. J., Lin, J., and Lonardi, S. 2002. Mining motifs in massive time series databases. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'02).* 370.

Patnaik, D., Laxman, S., and Ramakrishnan, N. 2009a. Discovering excitatory networks from discrete event streams with applications to neuronal spike train analysis. In *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM'09*). IEEE Computer Society, Los Alamitos, CA, 407–416.

Patnaik, D., Marwah, M., Sharma, R., and Ramakrishnan, N. 2009b. Sustainable operation and management of data center chillers using temporal data mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09).* ACM, New York, 1305–1314.

Patnaik, D., Marwah, M., Sharma, R., and Ramakrishnan, N. 2010. Data mining for modeling chiller systems in data centers. In *Advances in Intelligent Data Analysis IX*, P. Cohen et al., Eds. Lecture Notes in Computer Science, vol. 6065, Springer, 125–136.

Patnaik, D., Sastry, P., and Unnikrishnan, K. 2008. Inferring neuronal network connectivity from spike data: A temporal data mining approach. *Sci. Program. 16*, 1, 49–77.

Shah, A. J., Carey, V. P., Bash, C. E., and Patel , C. D. 2008. Exergy analysis of data center thermal management systems. *J. Heat Transfer 130*, 2, 021401.1–021401.10.

Sharma, R., Shih, R., Bash, C., Patel, C. D., Varghese, P., Mekanapurath, M., Velayudhan, S., and Kumar, M. 2008. On building next generation data centers: Energy flow in the information technology stack. In *Proceedings of the Bangalore Compute Conference.*

Sharma, R., Shih, R., Patel , C., and Sontag, J. 2007. Application of exploratory data analysis (eda) techniques to temperature data in a conventional data center. In *Proceedings of the ASME IPACK'07 Conference.*

SHI, Y., KLUSTEIN, M., SIMON, I., MITCHELL, T., AND BAR-JOSEPH, Z. 2007. Continuous hidden process model for time series expression experiments. *Bioinf. 23*, 13, 459–467.

VANDERBI LT, T. 2009. Data center overload. *New York Times*.

WANG, K., ZHANG, J., SHEN, F., AND SHI, L. 2008. Adaptive learning of dynamic bayesian networks with changing structures by detecting geometric structures of time series. *Knowl. Inf. Syst. 17,* 1, 121–133.

XU, W., HUANG, L., FOX, A., PATTERSON, D., AND JORDAN, M. I. 2009. Detecting large-scale system problems by mining console logs. In *Proceedings of the 22nd ACM Symposium on Operating Systems Principles.*

XUAN, X. AND MURPHY, K. 2007. Modeling changing dependency structure in multivariate time series. In *Proceedings of the International Conference on Machine Learning (ICML'07).* ACM, New York, 1055–1062.

YANKOV, D., KEOGH, E., MEDINA, J., CHIU, B., AND ZORDAN, V. 2007. Detecting time series motifs under uniform scaling. In *Proceedings of the International SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'07).* ACM, New York, 844–853.

Received October 2010; revised February 2011; accepted March 2011