Hierarchical Forecasting of Web Server Workload Using Sequential Monte Carlo Training

Tom Vercauteren, Pradeep Aggarwal, Xiaodong Wang, *Senior Member, IEEE*, and Ta-Hsin Li, *Senior Member, IEEE*

***Abstract*—Internet service utilities host multiple server applications on a shared server cluster (server farm). One of the essential tasks of the hosting service provider is to allocate servers to each of the websites to maintain a certain level of quality of service for different classes of incoming requests at each point of time, and optimize the use of server resources, while maximizing its profits. Such a proactive management of resources requires accurate prediction of workload, which is generally measured as the amount of service requests per unit time. As a time series, the workload exhibits not only short time random fluctuations but also prominent periodic (daily) patterns that evolve randomly from one period to another. We propose a solution to the Web server load prediction problem based on a hierarchical framework with multiple time scales. This framework leads to adaptive procedures that provide both long-term (in days) and short-term (in minutes) predictions with simultaneous confidence bands which accommodate not only serial correlation but also heavy-tailedness, and nonstationarity of the data. The long-term load is modeled as a dynamic harmonic regression (DHR), the coefficients of which evolve according to a random walk, and are tracked using sequential Monte Carlo (SMC) algorithms; whereas the short-term load is predicted using an autoregressive model, whose parameters are also estimated using SMC techniques. We evaluate our method using real-world Web workload data.**

***Index Terms*—Dynamic harmonic regression, seasonal time series, sequential Monte Carlo, Web-load prediction.**

I. Introduction

A WEB server farm is a cluster of servers shared by several Web applications and services and maintained by a host service provider. Usually, the owner of the Web applications pays the host service provider for the computing resources and, in return, gets a quality-of-service (QoS) guarantee, which promises a certain minimum level of resources and performance. Static allocation of resources at the server farm is not efficient since very often it results in either underutilization of resources (when the particular Web application is not actively being sought) or violation of QoS (when, for example, traffic for a particular website is very high and the allocated sources are insufficient to cater to the demands). Therefore, the server farm allocates the computing resources dynamically among the competing applications to meet the quality-of-service for different classes of service requests, while at the same time striving to maximize its own profits. The requirement for dynamic allocation of resources makes it necessary for the server farm to be able to predict the workload accurately, with a sufficiently long time horizon to ensure that adequate resources are allocated to the services in-need in a *timely* manner, while still achieving certain systemwide performance objective such as maintaining the QoS requirements of the entire server farm and maximizing the total revenue of the server farm under the QoS constraints. In a typical dynamic allocation scheme, each application is provided a certain minimum share of resources, and the remaining resources (servers and bandwidth) are dynamically allocated to the different active applications based on their instantaneous requirements and based on predefined policy in response to the workload changes. The prediction techniques/models are also helpful in preventing imminent service disruptions by anticipating potential problems due to heavy load on a particular website [1]–[5].

The server workload is usually measured in terms of the amount of services request per unit time (also called the request arrival rate). It can, for example, be the total number or size of the files requested per unit time, or it can be the total number of operations requested per unit time. A time series of such a workload is known to vary dynamically over multiple time scales, and therefore it is quite challenging to predict it accurately. In particular, the bursty nature and the nonstationarity of the server workload impose inherent limits on the accuracy of the prediction. Such a time series can, for example, be stationary but self-similar (i.e., the correlational structure remains unchanged over a wide range of time scales, resulting in long-range dependence, or in other words, it exhibits bursts wherein the workload remains above the mean for an extended duration at a wide range of time scales) and/or heavy-tailed over small duration (seconds or minutes) at a fine time granularity [6]–[9]; it can also exhibit strong daily and weekly patterns (seasonality), which change randomly over different times of the day and different days of the week, and can also show calendar effects (different patterns on weekends) [8]–[10]. It is this second type of data with seasonal variations that is key to the designing of dynamic resource allocation schemes and is the focus of the current paper.

The traditional linear-regression-based methods can give predictions with a limited accuracy, since the model can become inefficient in the presence of correlated error. In this paper, we follow the hierarchical approach proposed in [11] and [12] in which the time-series prediction is decomposed into two steps: first a prediction of the long-term component, which primarily captures the nonstationarity of the data, is performed, and then the residual short-term process, which captures both the long-term prediction error and the short-term component of the time series, is processed. As demonstrated by the results, the two-scale decomposition captures the underlying statistics

Manuscript received November 12, 2005; revised June 23, 2006. The associate editor coordinating the review of this paper and approving it for publication was Prof. Steven M. Kay.

T. Vercauteren is with INRIA, 06902 Sophia Antipolis, France.

P. Aggarwal and X. Wang are with Department of Electrical Engineering, Columbia University, New York, NY 10027-4712 USA (e-mail: wangx@ee.columbia.edu).

T.-H. Li is with Department of Mathematical Sciences, IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA.

Digital Object Identifier 10.1109/TSP.2006.889401

of the data fairly well. Additional components (e.g., weekly or monthly seasonality) increase the complexity of the algorithm as we have to estimate more parameters now. Furthermore, they also require much more training data. One of the main aims of this paper is to keep the computational complexity low, which our sequential Monte Carlo (SMC) algorithm-based two-scale scheme indeed achieves, without compromising the accuracy of the prediction. For example, the dynamic harmonic regression (DHR) model in [13] estimates the parameters by first running a two-step (prediction–correction) Kalman filter, followed by a fixed-interval smoothing algorithm. As the number of parameters increases, these steps become highly complex as they involve multiplication and inversion of high-dimensional matrices, whereas in the proposed SMC algorithm, the complexity increases only linearly with the number of parameters to be estimated. Previous literature, e.g., [11]–[13], also strongly mentions that not all the components of the unobserved component (UC) model are required for modeling the Web-server workload data and that the two-scale modeling is a good compromise between accuracy and computational complexity of the algorithm.

In this paper, the long-term component is modeled as a linear combination of certain basis functions with random amplitudes evolving with time, while the residual short-term process is modeled as a traditional autoregressive (AR) process. The short-term prediction is useful in predicting abnormalities in the workload data and to take care of rapid fluctuations, thereby giving the server farm management system sufficient time to prevent the possible disruption of services [11]. In this paper, in addition to the traditional short-term prediction, we also derive a scheme to predict the long-term component, utilizing the daily patterns in the workload time series. This not only provides ample time for advance planning, but also reduces the magnitude (and hence complexity) of the short-term adjustment that needs to be made in case of an imminent (potential) disruption. Moreover, the proposed method, in addition to providing predictions, can also be used to compute confidence bands simultaneously. This is of major interest in this setting since quantiles, as opposed to a simple prediction of the time series, can be used to support flexible (probability based) service-level agreements. Further, the proposed model allows the model parameters to change with time, thereby making itself capable of handling the nonstationarity in the data.

For the long-term component, we combine the DHR framework of [13] together with the filter bank approach of [14], to decompose the time series into seasonal components and only those basis functions that show high coherence across the periods are selected for long-term modeling and forecasting purpose. The use of highly coherent basis functions not only reduces the dimensionality of the problem, but also results in reliable long-term forecasting by using only the persistent components. This, in turn, results in reduction of the amount of training data required as well as making the model robust to the impact of noise and occasional corruption of training data. As mentioned before, the long-term component is represented as a linear combination of the basis functions, whose amplitudes are modeled as random processes under the state-space setting, the dynamic nature of which allows them to efficiently capture the trends and fluctuations of the data. Moreover, to take care of the calendar effect (e.g., the weekend data follows different trends as compared to the weekday data), a multiregime model is employed, in which the data belonging to different regimes is handled separately (using similar scheme nonetheless). The parameters for both the short-term model and the long-term model are estimated using the SMC methods, which are very powerful statistical tools for dealing with online estimation problems in dynamic systems (see [15]–[17] and references therein) and find applications in diverse fields.

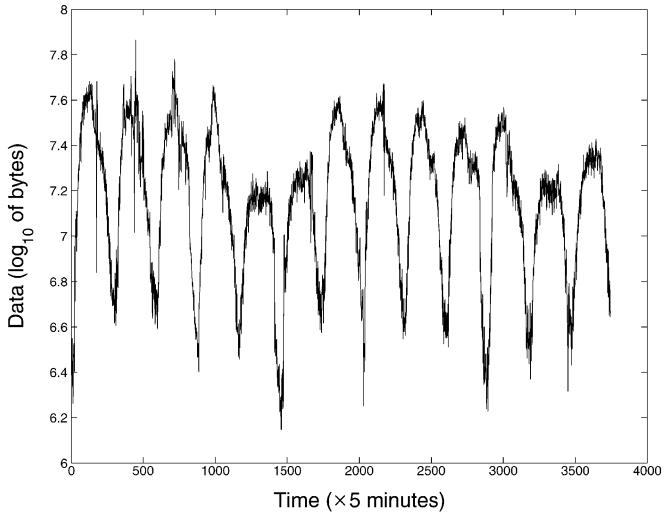


Fig. 1. Total file size (in $\log_{10}$ of bytes of HTTP requests), aggregated over $\Delta = 5$-min intervals, received at a Web server of an online retail store over 13 days.

The remainder of the paper is organized as follows. Section II discusses the properties of the time series at hand and explains the hierarchical structure of our model. In Section III, we describe the dynamic harmonic regression based model for the long-term components and the long-term prediction using the SMC methods. Section IV deals with the modeling and prediction of the short-term component. Simulation results are presented in Section V; and Section VI concludes the paper.

II. Hierarchical Framework

We consider a typical Web-server farm, which records the number of requests at each server and aggregates them over small time intervals of length $\Delta > 0$ to obtain a time series. For example, Fig. 1 shows the server workload obtained by aggregating the hypertext transfer protocol (HTTP) service requests for a commercial website over $\Delta = 5$-min intervals over a 13-day period, giving a total of 288 intervals in a day. This is the same data as used in [12], and we will employ this time series throughout this paper as a working example to demonstrate the performance of the proposed algorithm. The time series is first converted into logarithmic domain to reduce the dependence of the local variability on the local mean of the untransformed data.

Clearly, the data is nonstationary in that the mean changes with time of day and day of week. It is also observed that the time series shows predominant daily patterns, varying randomly. Let p denote the sampling frequency (number of samples per period) of the daily pattern. For the example shown in Fig. 1, for $\Delta = 5$ min, $p = 288$. Note that for a given p, the time index for the observation in the rth time interval in the τth period is given by $t = \tau p + r$, $\tau = 0, 1, 2, \ldots$, and $r = 1, \ldots, p$. Although, several methods exist for modeling such a time series, we follow the hierarchical approach developed in [12],

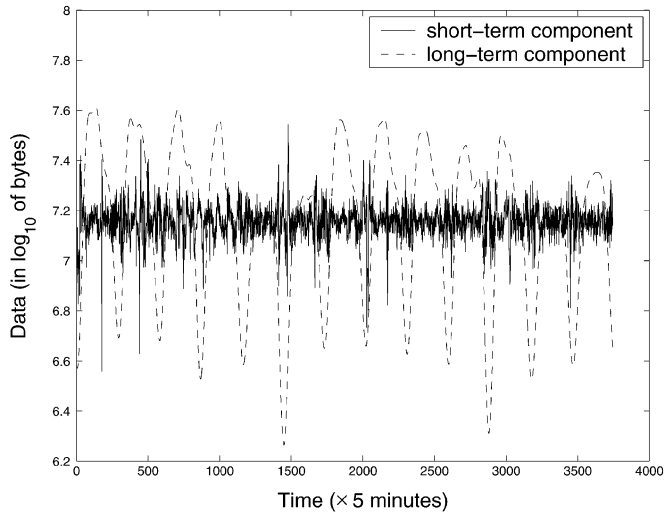


Fig. 2. Decomposition of the Web-server data into a long-term pattern and short-term random components.

which not only provides point predictions, but also simultaneous confidence bands, that can be used to support flexible (probability-based) service-level agreements. Furthermore, by allowing the model parameters to change with time, we can handle nonstationarity in both the long-term patterns as well as short-term fluctuations. Moreover, the hierarchical approach allows for easy diagnostic checking of model adequacy. Fig. 2 shows the hierarchical structure of the time series, where the data is decomposed into a periodic long-term component and a randomly fluctuating short-term component.

Let $y(t)$ denote the observed load (after taking logarithm) at a server at time t. In order to capture the seasonality in the data, we use the DHR model of [13]. The DHR model is a special type of the unobserved component model and can be used to capture several components such as trend, cyclical component, and seasonality. Stochastic time-varying parameters are used to characterize the various components of the DHR, thus allowing for nonstationarity in the resulting time series. In practice, not all components of the DHR are necessary, and in this paper, we focus on the seasonal component. In our hierarchical framework, the time series is first modeled as a combination of a periodic long-term pattern $p(t)$, and a more irregular short-term component $e(t)$:

$$y(t) = p(t) + e(t). \tag{1}$$

The periodic long-term pattern $p(t)$ is represented as a weighted sum of some p-periodic basis functions $\phi_j(t)$ as

$$p(t) = \sum_{j \in \mathcal{P}_t} a_j(t)\phi_j(t) \tag{2}$$

where $a_j(t)$ are stochastic time-varying parameters and $\{\phi_j, j \in \mathcal{P}_t\}$ forms a subset of some set of linearly independent basis functions $\{\phi_j, j \in \mathcal{P}\}$. In contrast with [13], our model also deals with a time-varying set of basis functions $\mathcal{P}_t$. At this point of the model, we are only interested in obtaining an estimation of the long-term component; the short-term component $e(t)$ will therefore be roughly modeled as a white noise term. To filter out the long-term component in (1), we choose the periodic basis functions to be sinusoidal waves whose frequencies are chosen based on the spectral properties of the time series [this is discussed further in Section 3-A-1)], giving us a harmonic regression (HR) on the long-term component. Moreover, an extra zero-frequency term is also added that can be considered as a stochastic time-variable "intercept" in the DHR.

Once the long-term estimation $\hat{y}_L$ has been performed, our hierarchical framework focuses on making an accurate d-step ahead forecast of the residual time series

$$z(t) = y(t) - \hat{y}_L(t \,|\, t). \tag{3}$$

This d-step-ahead forecast process is modeled as an AR process

$$z(t+d) = \sum_{i=1}^{q_t} b_i(t) \cdot z(t-i+1) + \varepsilon(t) \tag{4}$$

the parameters of which (order, coefficients, and noise characteristics) being stochastic time-varying parameters as in [18]. Furthermore, in order to accommodate for the shot noises in the data, we use heavy-tailed distributions for the noise term $\varepsilon(t)$.

The observation models (1), and (4), together with their dynamically varying coefficients form two dynamic state spaces, both are tracked using SMC methods.

For the time series of service requests, it is typical to have weekday patterns behaving significantly differently from the weekend patterns. In this paper, a multiple-regime approach is employed, in which data belonging to different regimes are modeled separately to take advantage of the within-regime resemblance, taking care of the between-regime change at the same time. The data belonging to the same regime is cascaded to obtain a set of new time series, one for each regime. For our example of Fig. 1, all weekday data can be collected together to form a weekday time series and all weekend data can be collected to form a new weekend series. Each time series is then modeled by (1) and (4). Each regime has its own set of parameters and some of them may be shared across the regimes to ensure smooth transition when the regime shifts. Note that sometimes, under the assumption that the short-term component do not change substantially with regime shifts, it is more convenient and justifiable to merge the short-term components in all regimes to obtain a single time series for modeling and prediction.

The remainder of the paper discusses in detail the modeling, parameter estimation, and prediction based on the model described above.

III. Long-Term Modeling and Prediction

Since we deal with different regimes separately, it is sufficient to consider only a single regime and assume that the statistical properties do not change abruptly with regime shift. We also assume that the type of basis functions $\phi_j(t)$ in (2) are known *a priori* (e.g., sinusoids, wavelets, etc.) and that the largest allowed set of basis functions $\mathcal{P}$ does not change with time.

Selection of the basis set $\mathcal{P}$ serves a dual purpose. First, it reduces the dimensionality of the problem, hence reducing the computational complexity as well as storage requirement associated with the training, modeling, and prediction. Indeed, the higher the dimension is, the greater the number of parameters to be estimated is. The need for reduction in dimension becomes more important in Web-server management as compared

to, say, economic forecasting [19] or electric utility management [20]–[22], due to the sheer number of parameters that are to be estimated, resulting from the much finer time granularity (minutes rather than hours or days). For example, for $\Delta = 5$ min, we get a sampling frequency of $p = 288$ steps and for $\Delta = 1$ min, $p = 1440$, as compared with $p = 24$ for hourly data with daily seasonality, and $p = 12$ for monthly data with yearly seasonality. The second advantage of selecting $\mathcal{P}$ lies in the fact that it makes the modeling and prediction more robust to estimation errors as compared with the model with a large basis set $\mathcal{P}$. Statistical theory of regression analysis [23] asserts that in the presence of inherent statistical error in parameter estimation, the mean-square error associated with both modeling and prediction can be reduced by simply dropping the "minor" components even if they exist in reality.

We will see in Section 3-A-1) that only the first few frequencies (including the zero-frequency term) affect the seasonal variations to any significant extent and that only a fixed number of sinusoids need to be in $\mathcal{P}$. It turns out that usually even within this fixed subset, only a few among those chosen frequency components are significant in representing the model at a particular time t, while the remaining ones carry relatively small weights and, hence, can be discarded without significant loss in performance. However, the important subset of $\mathcal{P}$ can change with time. Therefore, instead of accommodating all of them in our model, we can reduce the dimensionality further by dynamically selecting the frequencies from the set $\mathcal{P}$, as the system evolves. We follow the jump Markov framework of [18], which is close to the resampling-based shrinkage method proposed in [24] in the context of blind detection in fading channels.

Let us now write down the state-space form we use for the long-term model

$$\begin{aligned} y(t) &= \sum_{j\in\mathcal{P}_t} a_j(t)\phi_j(t) + e(t) \\ a_j(t) &= a_j(t-1) + v_j(t), \quad \forall j \in \mathcal{P} \end{aligned} \tag{5}$$

where $\boldsymbol{v}(t) = \{v_j(t), j \in \mathcal{P}\}$ is the process noise, $e(t)$ is the measurement noise, and $y(t)$ is the observed data. The second equation in (5) represents the first-order Markov transition process, which is assumed to generate the coefficients vector $\boldsymbol{a}(t) = \{a_j(t), j \in \mathcal{P}\}$. The vector $\boldsymbol{v}(t)$ represents temporally uncorrelated Gaussian disturbances with zero mean, and covariance matrix $\boldsymbol{Q}_v$, i.e.,

$$\mathbb{E}[\boldsymbol{v}(t)] = \boldsymbol{0} \quad \text{and} \quad \mathrm{Cov}[\boldsymbol{v}(t)] = \boldsymbol{Q}_v. \tag{6}$$

The zero-mean noise in the state equation stems from the assumption that the long-term behavior is periodic with slow variations, for which, the incremental mean of the coefficients is close to zero. The first equation in (5) represents the measurement equation, where $e(t)$ is the temporally uncorrelated Gaussian disturbance with zero mean and variance σ_e^2, i.e.,

$$\mathbb{E}[e(t)] = 0 \quad \text{and} \quad \mathrm{Var}[e(t)] = \sigma_e^2. \tag{7}$$

The initial state vector $\boldsymbol{a}(0)$ is assumed to be Gaussian distributed with mean $\bar{\boldsymbol{a}}(0)$ and covariance matrix $\boldsymbol{P}_a(0)$, i.e.,

$$\mathbb{E}[\boldsymbol{a}(0)] = \bar{\boldsymbol{a}}(0) \quad \text{and} \quad \mathrm{Cov}[\boldsymbol{a}(0)] = \boldsymbol{P}_a(0) \tag{8}$$

which are computed as the respective mean and covariance of the coefficients of the harmonics included in the regression, obtained from the training data. Further, the disturbance $e(t)$ and $\boldsymbol{v}(t)$ are assumed to be uncorrelated with each other in all time periods and uncorrelated with the initial state, i.e.,

$$\mathbb{E}\,[e(s)\boldsymbol{v}(t)] = \boldsymbol{0} \quad \text{for all } s, t,$$

and

$$\mathbb{E}[e(t)\boldsymbol{a}(0)] = \boldsymbol{0}, \quad \mathbb{E}[\boldsymbol{v}(t)\boldsymbol{a}^T(0)] = \boldsymbol{0} \quad \text{for all } t. \tag{9}$$

In (5), we chose to use a dynamic subset of harmonics $\mathcal{P}_t$. This subset is assumed to follow a first-order discrete Markov model

$$\Pr(\mathcal{P}_{t+1} = \rho_j \,|\, \mathcal{P}_t = \rho_i) \triangleq \pi_{ij} \tag{10}$$

where the ρ_i are some subsets of $\mathcal{P}$.

In what follows, we explain how the fixed parameters of our model and the set of basis functions $\mathcal{P}$ are determined from the historical observations of $y(t)$, and in Section III-B, we demonstrate how SMC methods can be employed to track this state-space model and select the subset $\mathcal{P}_t$ dynamically.

A. Determination of the Fixed Parameters

We use the analysis filter bank approach proposed in [14] to predetermine the basis set $\mathcal{P}$ and guide our choice of fixed parameters (priors, variances). The aim is to decompose the time series into seasonal components and consider only those components that are highly coherent across the period, as well as having high energy, and hence are important to modeling and prediction. In order to do this, we consider, at each time step, a single time period ending at the given time step and pass it through a filter bank. The resulting series of coefficients can then be analyzed.

Let $\boldsymbol{y}(t) = [y(t-p+1), y(t-p+2), \ldots, y(t)]^T$ be the data at hand. Then, from (2), using the full basis, we can write

$$\boldsymbol{a}^{\mathrm{fil}}(t) = \boldsymbol{\Phi}^{-1}\boldsymbol{y}(t), \quad t = 1, \ldots, n \tag{11}$$

where $\boldsymbol{a}^{\mathrm{fil}}(t) = [a_1^{\mathrm{fil}}(t), \ldots, a_p^{\mathrm{fil}}(t)]$ are the coefficients associated with the complete basis decomposition, $\boldsymbol{\Phi} = [\phi_1, \ldots, \phi_p]$ is the matrix of all the basis functions, and its inverse has an analysis filter bank interpretation. In other words, denoting the jth row of $\boldsymbol{\Phi}^{-1}$ by $\boldsymbol{\Psi}_j^T = [\psi_j(p-1), \ldots, \psi_j(0)]$, (11) can be written as

$$a_j^{\mathrm{fil}}(t) = \sum_{i=0}^{p-1} \psi_j(i)y(t-i), \quad j = 1, \ldots, p \tag{12}$$

which is nothing but the output obtained on passing $y(t)$ through a filter bank consisting of p finite-impulse-response (FIR) filters, with $[\psi_j(p-1), \ldots, \psi_j(0)]$ being the impulse response of the jth filter. After having obtained the analysis filter bank output $a_j^{\mathrm{fil}}(t)$ defined in (12), the data $y(t)$ can be reconstructed according to

$$y(\tau) = \sum_{j=1}^{p} a_j^{\mathrm{fil}}(t)\phi_j(\tau), \quad \tau = t-p+1, \ldots, t \tag{13}$$

which can be considered as the decomposition of y into p component waveforms, whose shapes are determined by the basis functions ϕ_j.

1) Choice of Basis Set: Clearly, with p being very large ($p = 288$ for our example), we aim to reduce the dimensions of the filter bank and chose $\mathcal{P}$ by analyzing $\boldsymbol{a}^{\text{fil}}(t)$. In [12], two measures on the component waveforms are suggested to quantify the behavior of $\boldsymbol{a}^{\text{fil}}(t)$ to aid in the selection of $\mathcal{P}$, namely, the *coherence* measure and the *energy* measure. The *coherence* measure is defined as

$$\hat{c}_j = \frac{\hat{\mu}_j^2}{\hat{\mu}_j^2 + \hat{\sigma}_j^2} \tag{14}$$

where $\hat{\mu}_j^2$ is the sample mean and $\hat{\sigma}_j^2$ is the sample variance of $\{a_j^{\text{fil}}(t)\}_{t=1}^n$, obtained as

$$\hat{\mu}_j = \frac{1}{n}\sum_{t=1}^{n} a_j^{\text{fil}}(t), \quad \text{and} \quad \hat{\sigma}_j^2 = \frac{1}{n}\sum_{t=1}^{n}\left(a_j^{\text{fil}}(t) - \hat{\mu}_j\right)^2. \tag{15}$$

The kth waveform is said to be completely coherent if $\hat{c}_j = 1$, and incoherent if $\hat{c}_j = 0$. We seek to include highly coherent waveforms (waveforms with high values of $\hat{c}_j$) in $\mathcal{P}$ as they have long-lasting effects, making them good candidates for long-term forecasting. The *energy* measure of the component waveforms is defined as

$$\hat{E}_j = \frac{1}{n}\sum_{t=1}^{n}\left(a_j^{\text{fil}}(t)\right)^2 = \hat{\mu}_j^2 + \hat{\sigma}_j^2. \tag{16}$$

High-energy components along with high-coherence component are crucial to effective modeling of $y(t)$ and are included in $\mathcal{P}$. For future reference, let the number of basis functions included in $\mathcal{P}$ be K.

In this paper, we take sinusoids as the basis functions and perform short-term Fourier transform (STFT) on the weekday data $y(t)$ of our example in Fig. 1. We show the *coherence* measure and the *energy* measure for the first 30 frequencies and the DC (zero-frequency) component in Figs. 3 and 4, respectively. It is clear from the two figures that only the fundamental frequency term ($\omega_0 = 2\pi/p$), its first few harmonics, and the zero-frequency term have sufficiently high coherence as well as energy measure, while the rest of them appear to be insignificant in comparison. We select the first five frequencies (the fundamental frequency and its first four harmonics) together with DC (zero-frequency) to form $\mathcal{P}$. Since each frequency corresponds to two waveforms (a sine and a cosine), the dimension of $\mathcal{P}$ is $K = 11$. Thus, we achieve our goal of reducing the dimensionality of the model, by bringing it down from $p = 288$ to 11.

2) Choice of Fixed Parameters: The initial state vector mean $\bar{\boldsymbol{a}}(0)$ and covariance matrix $\boldsymbol{P}_a(0)$ are computed as the respective mean and covariance of the output of the analysis filter bank on the training data.

For the dynamic selection of the basis set $\mathcal{P}_t$, let P_i and P_d be the probabilities of increasing and decreasing the order of the harmonic regression by one, respectively. The introduction of these probabilities offers flexibility in changing the harmonic regression order, thus allowing the algorithm to adaptively adjust according to the data. We choose some small probabilities and favor parsimonious models by letting $P_i < P_d$ to limit the increase in dimensionality.

The noise variances can be estimated by looking at the residuals. We choose a larger variance for the zero-frequency term as compared with the variance of the residual time series to be able to accommodate the outliers.

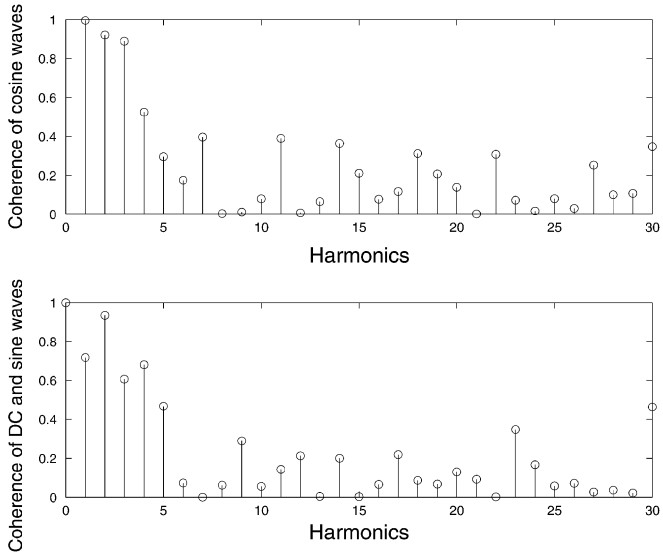


Fig. 3. Coherence measures for the first 30 frequencies of the cosine and sine waves and the DC component.

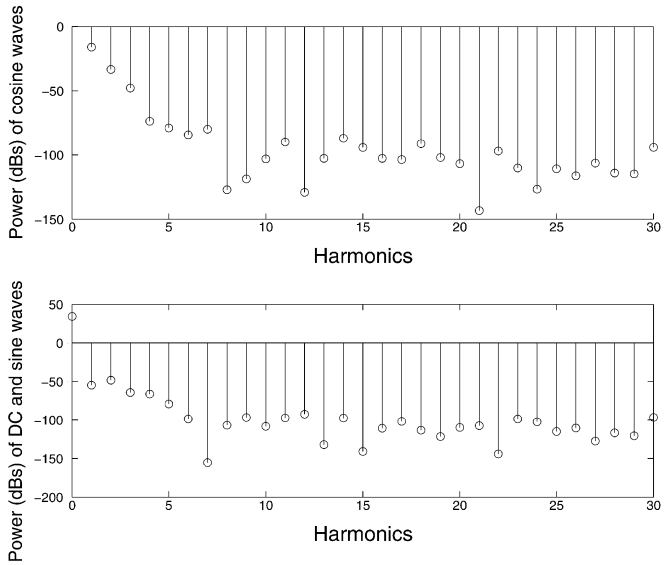


Fig. 4. Energy measures (in decibels) for the first 30 frequencies of the cosine and sine waves and the DC component.

Once the model in state-space form has been identified, and the corresponding parameters assumed accordingly, SMC can be applied for recursively calculating the optimal estimate of the state vector $\boldsymbol{a}(t)$, based on the information up to time t.

B. Online Estimation and Prediction by SMC

We track the model based on the set of available historical data using the SMC technique [15], [18], [25], [26]. Let $x_{r:s}$ denote the vector of values of a variable x from time r to time s. Our aim is to obtain an online Monte Carlo approximation of the target distribution $p(\boldsymbol{a}(0:t), \mathcal{P}_{0:t}|y(1:t))$. With this goal, the SMC method keeps M sample streams

$(\boldsymbol{a}^{(m)}(0:t), \mathcal{P}_{0:t}^{(m)})$, together with the associated importance weight $\omega_t^{(m)}, m = 1 \ldots, M$, such that

$$p(\boldsymbol{a}(0:t), \mathcal{P}_{0:t} \,|\, y(1:t)) \approx \sum_{m=1}^{M} \omega_t^{(m)} \delta\left[\boldsymbol{a}(0:t) - \boldsymbol{a}^{(m)}(0:t), \mathcal{P}_{0:t} - \mathcal{P}_{0:t}^{(m)}\right] \quad (17)$$

where $\delta[\cdot]$ is a Dirac function (written with brackets instead of the conventional subscript to ease the reading).

We progress sequentially through each stream by extending at time t, the past particles $(\boldsymbol{a}^{(m)}(0:t-1), \mathcal{P}_{0:t-1}^{(m)})$, by sampling $(\boldsymbol{a}^{(m)}(t), \mathcal{P}_t^{(m)})$ according to a so-called trial distribution (the probability distribution that is employed to draw samples of the particles by virtue of it being easier to draw samples from as compared to the actual probability distribution of interest)

$$q\left(\boldsymbol{a}(t), \mathcal{P}_t \,\middle|\, \boldsymbol{a}^{(m)}(0:t-1),\ \mathcal{P}_{t-1}^{(m)}, y(1:t)\right). \quad (18)$$

The discrepancy thus induced is corrected by the importance weight $\omega_t^{(m)}$ associated with each stream that can then be recursively updated as (19), shown at the bottom of the page. The simplest choice for the trial distribution is to take the transition probability. With this choice, the weight update equation (19) reduces to $\omega_t^{(m)} = \omega_{t-1}^{(m)} \cdot p(y(t)|\boldsymbol{a}^{(m)}(t), \mathcal{P}_t^{(m)})$. It is shown in [18] that one can actually sample from such a distribution by first sampling the discrete variable $\mathcal{P}_t^{(m)}$ and then sample the corresponding coefficients $\boldsymbol{a}^{(m)}(t)$. More accurate (but also more computationally expensive) alternatives for the trial distribution are shown in [18] using techniques such as the auxiliary particle filter and the unscented transform. While these alternative methods have more computational requirements per stream, it may also turn out that the number of particles required with them is lesser as compared to the case of sampling using simply the transition probability depending on the nature of the problem. Throughout the rest of the paper, we use the transition probability as the sampling density to explain the algorithm.

The estimated long-term component at time t is then given by

$$\hat{y}_L(t|t) = \sum_{m=1}^{M} \left(\sum_{j \in \mathcal{P}_t^{(m)}} a_j^{(m)}(t) \phi_j(t) \right) \cdot \tilde{\omega}_t^{(m)} \quad (20)$$

where $\tilde{\omega}_t^{(m)} = (\omega_t^{(m)})/(\sum_{m=1}^{M} \omega_t^{(m)})$ is the normalized importance weight corresponding to the mth stream. Since the coefficients $\boldsymbol{a}$ are assumed to follow a random walk (5) and e is a white noise, the d-step ahead predicted value using the long-term model is given by

$$\hat{y}_L(t+d|t) = \sum_{m=1}^{M} \left(\sum_{j \in \mathcal{P}_t^{(m)}} a_j^{(m)}(t) \phi_j(t+d) \right) \cdot \tilde{\omega}_t^{(m)}. \quad (21)$$

C. Resampling-Based Adaptive Shrinkage of the Basis Functions

The importance weights measure the quality of the Monte Carlo samples. As we proceed with the algorithm, the weights progressively get smaller and smaller, and after a while, only a few of the streams carry significant weights, while the rest of the samples become ineffective. To avoid this problem, resampling [15], [27] is performed when the effective sample size (ESS) goes below a certain predetermined threshold. The ESS is a measure of the overall quality of the samples and is defined as

$$\text{ESS} \triangleq \frac{M}{1 + v_t^2} \quad (22)$$

where v_t is the coefficient of variation of the importance weights and is given by

$$v_t^2 = \frac{1}{M} \sum_{m=1}^{M} \left(\frac{\omega_t^{(m)}}{\tilde{\omega}_t^{(m)}} - 1 \right)^2. \quad (23)$$

For a detailed treatment of ESS and resampling, see [15], [17], [25], [27], and references therein.

At the beginning of the SMC procedure, for each of the Monte Carlo samples, the sinusoids to be included are randomly drawn with probability proportional to their respective coherence values. At time $(t-1)$, let $\mathcal{P}_{t-1}^{(m)} \subseteq \mathcal{P}$ denote the set of sinusoids being used by the mth sample stream. Since we introduced probabilities P_i and P_d of increasing or decreasing the order of the harmonic regression, the algorithm will adaptively adjust according to the data. At time t, the set $\boldsymbol{S}$ of m samples can be divided into three subsets: $\boldsymbol{S}_0$, whose harmonic regression order is left unchanged, $\boldsymbol{S}_{+1}$, whose harmonic regression order is incremented by unity (up to a maximum of K), and $\boldsymbol{S}_{-1}$, whose order is decreased by unity (down to a minimum of 0). A particular sample finds place in the subsets $\boldsymbol{S}_{+1}, \boldsymbol{S}_{-1}$ and $\boldsymbol{S}_0$ with probabilities P_i, P_d, and $(1 - P_i - P_d)$, respectively. Thus, we obtain new set of basis functions $\mathcal{P}_t^{(m)} \subseteq \mathcal{P}$, associated with the mth Monte Carlo stream, at time t. Following this step, the samples and the importance weights are updated using (18) and (19), respectively. At time t, for the particles in the

$$\omega_t^{(m)} = \frac{p\left(\boldsymbol{a}^{(m)}(0:t), \mathcal{P}_{0:t}^{(m)} \,\middle|\, y(1:t)\right)}{q\left(\boldsymbol{a}^{(m)}(0:t), \mathcal{P}_{0:t}^{(m)} \,\middle|\, y(1:t)\right)} \propto \omega_{t-1}^{(m)} \cdot \frac{p\left(y(t) \,|\boldsymbol{a}^{(m)}(t),\ \mathcal{P}_t^{(m)}\right) p\left(\boldsymbol{a}^{(m)}(t), \mathcal{P}_t^{(m)} \,\middle|\, \boldsymbol{a}^{(m)}(t-1), \mathcal{P}_{t-1}^{(m)}\right)}{q\left(\boldsymbol{a}^{(m)}(t), \mathcal{P}_t^{(m)} \,\middle|\, \boldsymbol{a}^{(m)}(0:t-1), \mathcal{P}_{t-1}^{(m)}, y(1:t)\right)}. \quad (19)$$

set $\boldsymbol{S}_{+1}$, which did not have the corresponding particles at time $t-1$, the coefficients are drawn from a zero-mean Gaussian distribution. We then check for the resampling condition, i.e., ESS $< M/\gamma$ ($\gamma = 10$ in this paper), and if required, perform resampling. This resampling step combined with randomly changing the regression order of the sample stream is the key step in achieving our goal of adaptive shrinkage of the basis functions. Using this step, the samples with proper number of harmonics, effectively modeling the observed data [i.e., samples with lower least-square error, and thus, having relatively higher importance weights in (19)] are replicated, and the samples with improper harmonic regression order (and, hence, larger least-square error) are discarded. Thus, we observe that the time series is effectively modeled by a mixture of harmonic regressions, with different orders, and the mixture distribution evolves dynamically during the SMC procedure.

IV. Short-Term Modeling and Prediction

The long-term estimation error $z(t) = y(t) - \hat{y}_L(t|t)$ is employed as the raw data for the d-step-ahead prediction of the short-term component, which covers both the short-term fluctuations and the long-term prediction error. As mentioned earlier, it can either be separated into different regimes (weekdays and weekends for our example) or be considered as a single time series. We employ an autoregressive (AR) model, which is simple and effective in time-series modeling for such data. However, since Web server load time-series structures evolve with time and show some nonstationarity, we will allow the order q_t to evolve dynamically within a given range $\mathcal{Q}$ and the coefficients $\boldsymbol{b}(t) = \{b_i(t), i \in \mathcal{Q}\}$ of the AR model to vary with time. Since we focus on the d-step-ahead prediction, we will employ a dth-order AR model, which is known to provide more accurate and robust estimators [12].

Our short-term model can be cast into the following state-space model:

$$\begin{aligned} z(t) &= \sum_{i=1}^{q_t} b_i(t) \cdot z(t-d-i+1) + \varepsilon(t) \\ b_i(t) &= b_i(t-1) + w_i(t), \quad \forall i \in \mathcal{Q} \end{aligned} \tag{24}$$

where $\varepsilon(t)$ is the observation noise term, and $\boldsymbol{w}(t) = \{w_i(t), i \in \mathcal{Q}\}$ is the process noise. Similar models have been proposed in [28] and [29], using the Markov Chain Monte Carlo (MCMC) methods. In order to model the bursts in the data, we use a heavy-tail distribution, such as a t-distribution, to model the observation noise density.

As was done in the long-term model case, the order q_t and the coefficients $\boldsymbol{b}(t)$ in the regression are tracked using SMC. For the short-term process, we are particularly interested in having an accurate prediction. Since the accuracy of the SMC tracking depends on the use of a good observation noise model, here we also track the variance $\sigma_\varepsilon^2(t)$ of $\varepsilon(t)$. This is done as in [18], by modeling the evolution of the log-variance

$$\log \sigma_\varepsilon^2(t) = \log \sigma_\varepsilon^2(t-1) + u(t) \tag{25}$$

where

$$\mathbb{E}[\boldsymbol{w}(t)] = \boldsymbol{0} \quad \text{and} \quad \mathrm{Cov}[\boldsymbol{w}(t)] = \boldsymbol{Q}_w$$

and

$$\mathbb{E}[u(t)] = 0 \quad \text{and} \quad \mathrm{Var}[u(t)] = \eta. \tag{26}$$

The sample streams are initialized by drawing samples of $\boldsymbol{b}(0)$ from a zero-mean Gaussian distribution with covariance $\boldsymbol{P}_b(0)$. Similarly, the initial set of samples for the noise parameter $\log \sigma_\varepsilon^2(0)$ is drawn from the Gaussian density with mean and variance $\mu_n(0)$, and $\sigma_n^2(0)$ respectively, i.e.,

$$\begin{aligned} \boldsymbol{b}(0) &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{P}_b(0)) \\ \log \sigma_\varepsilon^2(0) &\sim \mathcal{N}\left(\mu_n(0), \sigma_n^2(0)\right). \end{aligned} \tag{27}$$

The objective of the SMC algorithm here is to find an estimate of the coefficients $\boldsymbol{b}(t)$ of the underlying AR process and the log-variance parameter $\log \sigma_\varepsilon^2(t)$ of the noise, based on the available short-term process $z(1:t)$. The target distribution $p(\boldsymbol{b}(0:t), q_{0:t}, \log \sigma_\varepsilon^2 | z(1:t))$ can be factored as in Section III-B, allowing for a recursive weight update.

Under this model, the d-step-ahead prediction of z, based on the knowledge of $\{z(1), \ldots, z(t)\}$, is given by

$$\hat{z}(t+d|t) = \sum_{m=1}^{M_s} \tilde{\omega}_t^{(m)} \sum_{i=1}^{q_t^{(m)}} b_i^{(m)}(t\,|\,t) \cdot z(t-i+1) \tag{28}$$

where $\tilde{\omega}_t^{(m)}$ are the SMC weights similar to (21) but referring, of course, to the Monte Carlo samples for the short-term state-space.

Finally, the short-term prediction can be combined with the long-term prediction to obtain a complete d-step-ahead forecast as

$$\hat{y}(t+d\,|\,t) = \hat{y}_L(t+d\,|\,t) + \hat{z}(t+d\,|\,t). \tag{29}$$

A. Adaptive AR Order Selection

Extending the idea of adaptive shrinkage of the harmonic regression discussed in Section III-B, we select the order of the AR process modeling the short-term component adaptively via resampling and also keep the provision of increasing or decreasing the order by introducing very small probabilities P_{in}, and P_{de}, which represent the probability of increasing and decreasing the order of the regression respectively. We again favor parsimonious representations by letting $P_{\mathrm{in}} < P_{\mathrm{de}}$. At the beginning of the SMC procedure for the short-term component, the order $q_0^{(m)}$ for the mth sample is randomly selected with uniform probability from a set $\mathcal{Q}$ ($\mathcal{Q} = \{3, 4, 5\}$ in this paper). Let $q_{\min}$ and $q_{\max}$ denote the minimum and maximum allowed order, respectively. At time $(t-1)$, let $q_{t-1}^{(m)} \in [q_{\min}, q_{\max}]$ denote the regression order used by the mth sample stream. Then, at time t, as was done in the harmonic regression case in Section III-B, the set $\boldsymbol{S}$ of m samples is divided into three subsets: $\boldsymbol{S}_0$, whose regression order is left unchanged, $\boldsymbol{S}_{+1}$, whose regression order is incremented by unity (up to a maximum of $q_{\max}$), and $\boldsymbol{S}_{-1}$, whose order is decreased by unity (down to a minimum of $q_{\min}$). A particular sample finds place in the subsets $\boldsymbol{S}_{+1}, \boldsymbol{S}_{-1}$ and $\boldsymbol{S}_0$ with probabilities $P_{\mathrm{in}}, P_{\mathrm{de}}$, and $(1 - P_{\mathrm{in}} - P_{\mathrm{de}})$, respectively. For the particles in the set $\boldsymbol{S}_{+1}$ at time t, which did not have the corresponding particles at time $t-1$, a zero-mean Gaussian distribution is used to draw the coefficients from. The samples and weights are then updated and resampling condition is checked. If required, resampling is performed, replicating the samples with proper regression order, and annihilating the samples with improper order. Thus, instead of keeping a fixed regression order, we let it evolve during the SMC procedure and allow different streams to have different orders.

B. Computation of Confidence Bands

The SMC algorithm described above inherently provides a way of computing confidence bands since it carries information about the complete probability density function of the variables. In our hierarchical framework, the long-term estimation is subtracted to the time series in order to form the short-term process. Once the long-term estimation is done, the entire randomness (error in the long-term prediction and remaining fluctuations) is thus carried by the short-term process $z(t)$. It is therefore only necessary to find the confidence-band associated with this short-term process.

The SMC procedure provides the following approximation of the distribution of the stochastic time-varying parameters at time t:

$$p\left(\boldsymbol{b}(0:t), q_t, \log\sigma_\varepsilon^2(t)\middle|\, z(1:t)\right) \approx \sum_{m=1}^{M_s} \tilde{\omega}_t^{(m)} \delta\left[\boldsymbol{b}^{(m)}(t), q_t^{(m)}, \log\sigma_\varepsilon^{2(m)}(t)\right]. \quad (30)$$

For a given $\sigma_\varepsilon^2(t)$, it is also possible to approximate the density of the noise $\varepsilon(t)$ using Monte Carlo technique by sampling N_n samples $\varepsilon_\sigma^{(n)}(t)$ from a t-distributed density with variance $\sigma_\varepsilon^2(t)$ and using another Dirac representation

$$p\left(\varepsilon(t)\,|\log\sigma_\varepsilon^{2(m)}(t)\right) \approx \sum_{n=1}^{N_n} \delta\left[\varepsilon_\sigma^{(n)}(t)\right]. \quad (31)$$

By plugging (30) and (31) into our observation equation (24), we obtain an approximation of the d-step-ahead predictive distribution of the server load as

$$p(z(t+d)\,|\,z(0:t)) \approx \sum_{m=1}^{M}\sum_{n=1}^{N_n} \tilde{\omega}_t^{(m)} \delta\left[\sum_{i=1}^{q_t^{(m)}} b_i^{(m)}(t) z(t-d-i+1) + \varepsilon_\sigma^{(n)}(t)\right]. \quad (32)$$

Using the $M \cdot N_n$ samples $z^{(m,n)}(t+d|t) = \sum_{i=1}^{q_t^{(m)}} b_i^{(m)}(t) \cdot z(t-d-i+1) + \varepsilon_\sigma^{(n)}(t)$, and their approximate distribution obtained in (32), we can extract any measure related to the d-step-ahead predictive distribution of the server load such as the confidence bands as explained next.

Let α denote the intended confidence level. We look for a α-confidence band, which is symmetric and centered around the predicted value $\hat{z}(t+d\,|\,t)$. This can also be formulated as finding the smallest radius $\theta_\alpha(t)$ such that $[\hat{z}(t+d\,|\,t) - \theta_\alpha(t), \hat{z}(t+d\,|\,t) + \theta_\alpha(t)]$ contains a ratio α of the weights $\tilde{\omega}_t^{(m,n)} = \tilde{\omega}_t^{(m)}$ of the $M_s \cdot N_n$ samples. This can be done simply by ordering the samples $z^{(m,n)}(t+d\,|\,t)$ according to their absolute difference with respect to the predicted mean $\hat{z}(t+d\,|\,t)$ and iteratively adding the closest sample until we get a total weight that is above the ratio α.

The final confidence band is then simply obtained by shifting the above confidence-level by the long-term prediction, giving $[\hat{z}(t+d\,|\,t) - \theta_\alpha(t), \hat{z}(t+d\,|\,t) + \theta_\alpha(t)]$. This is clearly much simpler in contrast to [12], which requires cumbersome smoothing and model-fitting procedures to compute the confidence band. Finally, we summarize the SMC-based hierarchical Web server workload prediction algorithm in Algorithm 1.

Algorithm 1: SMC-based Hierarchical, d-Step-Ahead Web Server Workload Prediction Algorithm

1: Perform STFT on the training data and select the basis set $\mathcal{P}$ using the *coherence* (14) and *energy* (16) measures;
2: Initialize the importance weights corresponding to the long-term and short-term components as $\omega_{0,L}^{(m)} = 1$, and $\omega_{0,s}^{(m)} = 1, m = 1, \ldots, M$ respectively;
3: Initialize the long-term state-vector $\boldsymbol{a}(0)$ by drawing M samples from the Gaussian distribution with parameters given in (8);
4: Initialize the short-term state-vector $\boldsymbol{b}(0)$ and noise parameter $\log\sigma_\epsilon^2(0)$ by drawing M samples according to (27);
5: **for** $t = 1, \ldots, d + \mathcal{Q} - 1$ **do**
6: Increase or decrease the DHR order of the mth stream with probability P_i and P_d respectively, to obtain the basis set $\mathcal{P}_t^{(m)}$;
7: Draw M samples of $\boldsymbol{a}(t)$ according to the trial distribution (18);
8: Update the importance weight $\omega_{t,L}^{(m)}$ according to (19), $m = 1, \ldots, M$;
9: Compute the long-term estimate $\hat{y}_L(t\,|\,t)$ and the d-step ahead long-term prediction value $\hat{y}_L(t+d\,|\,t)$ according to (20), and (21) respectively;
10: Perform resampling if required by checking for the resampling condition in (22);
11: **end for**
12: **for** $t = d + \mathcal{Q}, \ldots, T$ **do**
13: Repeat steps 6–10 to obtain long-term estimate $\hat{y}_L(t\,|\,t)$, and long-term prediction $\hat{y}_L(t+d\,|\,t)$;
14: Increase or decrease the AR order of the mth stream with probabilities P_{in}, and P_{de} respectively to obtain $q_t^{(m)}$;
15: Draw M samples of $\boldsymbol{b}(t)$ and $M \cdot N_n$ samples of $\log\sigma_\epsilon^2(t)$ according to the the model in (24), and (25) respectively, and update the corresponding importance weights $\omega_{t,s}^{(m)} = 1, m = 1, \ldots, M$;
16: Perform resampling of the samples corresponding to the short-term model if required;
17: Compute the short-term prediction $\hat{z}(t+d\,|\,t)$ according to (28), and add it to the long-term prediction $\hat{y}_L(t+d\,|\,t)$ to obtain the final forecast $\hat{y}(t+d\,|\,t)$ according to (29);
18: Compute the confidence band as described in Section IV-B;
19: **end for**

V. Numerical Results

We use the example introduced in the beginning of this paper and present the performance of the proposed algorithm. The data

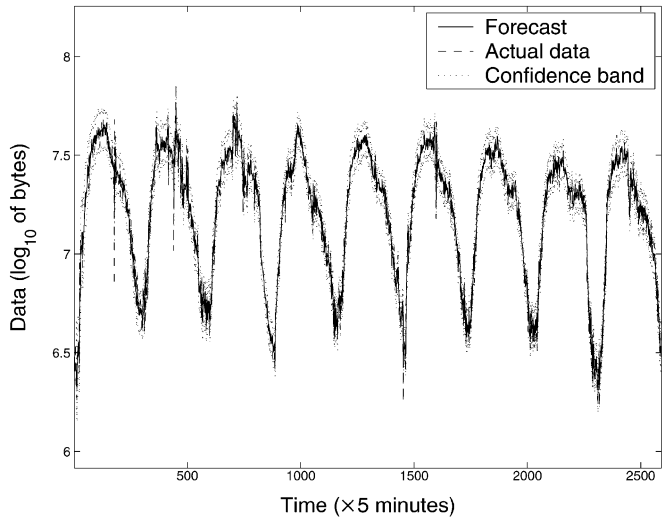


Fig. 5. 5-min-ahead prediction for the weekday data, with a 90% confidence band. RMSE of prediction is equal to 0.0586. Actual coverage of the confidence band is equal to 89.8%. Median width of the confidence band is equal to 0.1692.

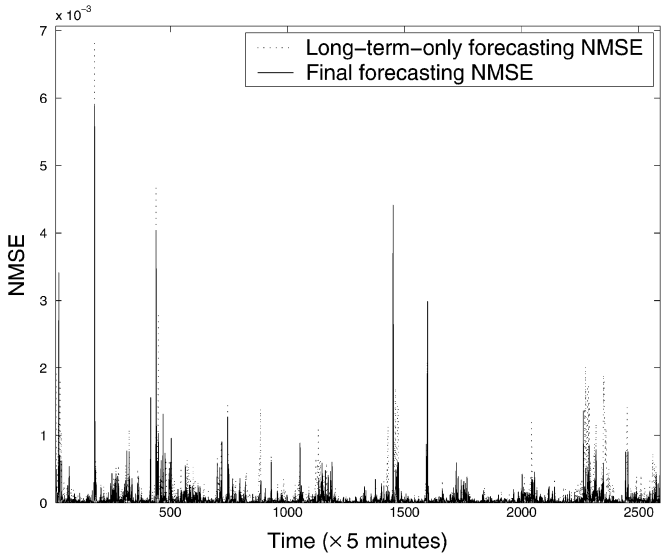


Fig. 6. NMSE of the 5-min-ahead prediction compared with the NMSE of the long-term-only forecast. The final forecast is approximately two times better than the long-term-only forecast in term of NMSE.

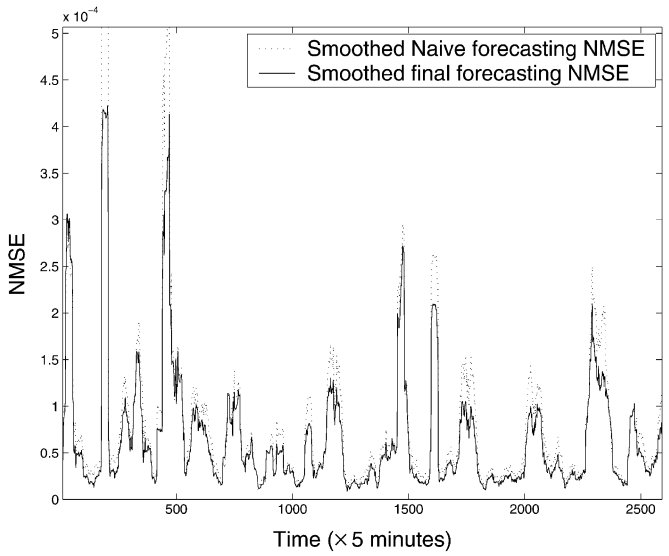


Fig. 7. NMSE of the 5-min-ahead prediction compared with the NMSE of the naive forecast. The final forecast is approximately 1.3 times better than the naive forecast in term of NMSE.

sets are the same as employed in [12] and are taken from actual Web servers. We employ $M = 500$ Monte Carlo samples for both the long-term as well as short-term prediction. The probabilities to increase (P_i) and decrease (P_d) the long-term regression order was chosen to be small, with $P_i = 10^{-4}$ and $P_d = 5 \cdot 10^{-4}$, respectively. Similarly, for the short-term model, we chose $P_{\text{in}} = 10^{-4}$ and $P_{\text{de}} = 2 \cdot 10^{-4}$. This is in accordance with our desire to keep the model parsimonious by having the probability of decreasing the regression order larger than the probability of increasing the regression order for both the components. All the results are obtained by averaging over 20 runs of the predictor.

As can be seen in Fig. 1, the data not only shows daily patterns, but also exhibits significant difference in the weekday and weekend patterns, making multiple-regime analysis suitable. We cascade all the weekday data together to obtain the weekday regime. The daily pattern in the weekday data can be clearly seen and is accommodated in the long-term component. The short-term component shows randomness and fluctuations. Fig. 5 shows the 5-min-ahead prediction employing the proposed algorithm. The root mean-square error (RMSE) is equal to 0.0586, which is slightly better than 0.0590 obtained by the algorithm in [12], and a confidence level of 89.8% is achieved, while the intended level was 90%. The median width of the confidence band comes out to be 0.1692, whereas [12] yields a somewhat tighter confidence band with a median width of 0.1618. The naive forecasting is obtained by repeating the first observation for the prediction horizon and shifting the rest of the series forward. As expected, the performance of naive forecasting is poor. Fig. 6 compares the normalized mean-square error (NMSE) of the final horizon forecast with the long-term-only forecast, which is obtained from the long-term prediction model of (21). The mean-square error (MSE) is normalized by dividing it with the variance of the observed data $y(t)$. The NMSE of the final forecast is 0.0331, which turns out to be about half of that of the long-term-only forecast, which is 0.06042. Fig. 7 shows the smoothed NMSE performance for naive forecast and final forecast. The final forecast gives a smoothed NMSE which is approximately 76% of the smoothed NMSE obtained from naive forecast.

Similarly, Fig. 8 illustrates the performance of the algorithm for a 20-min-ahead prediction horizon. It achieves a confidence level of 89.13%, for an intended level of 90%, with RMSE equal to 0.0712, which is better than the RMSE of 0.726 obtained for the same prediction horizon in [12]. We also get a tighter confidence band, with its median width level being equal to 0.212, compared against a median width level of 0.2259 obtained in [12]. Fig. 9 shows the smoothed NMSE of the final forecast and the long-term-only forecast. The long-term-only forecast has an NMSE of 0.0641, as compared with 0.0495 of the final forecast.

We also observed that the adaptive regression order selection scheme results in reduction in the complexity of the algorithm (as compared with keeping the regression order fixed). For

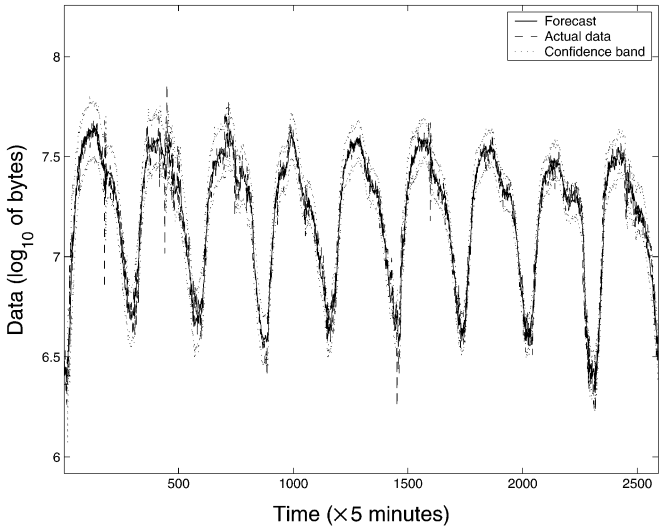


Fig. 8. 20-min-ahead prediction for the weekday data, with a 90% confidence band. RMSE of prediction is equal to 0.0712. Actual coverage of the confidence band is equal to 89.13%. Median width of the confidence band is equal to 0.212.

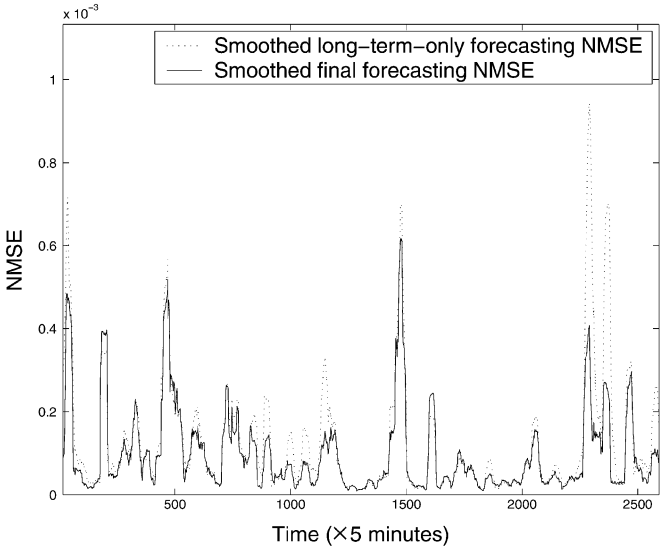


Fig. 9. Smoothed NMSE of the 20-min-ahead prediction compared with the smoothed NMSE of the long-term-only forecast. The final forecast is approximately 1.3 times better than the long-term-only forecast in term of NMSE.

the short-term component, the average regression order comes out to be approximately 4 for the 5-min-ahead as well as the 20-min-ahead prediction, while at the same time, we could actually have the regression order up to 8, allowing better modeling. Similarly, the average order of the harmonic regression comes out to be approximately 6 for the 5-min-ahead prediction, and around 7.6 for the 20-min-ahead prediction, which is well below 11, and way below the original 288. This shows that the adaptive selection of regression order indeed reduces the complexity of the algorithm and makes it suitable for applications involving fast learning and real-time prediction.

To demonstrate the advantage of selecting the basis subset adaptively at each instant from a set of high *energy* and high *coherence* waves over keeping the basis set fixed, we ran the algorithm for a 5-min-ahead prediction with a fixed harmonic

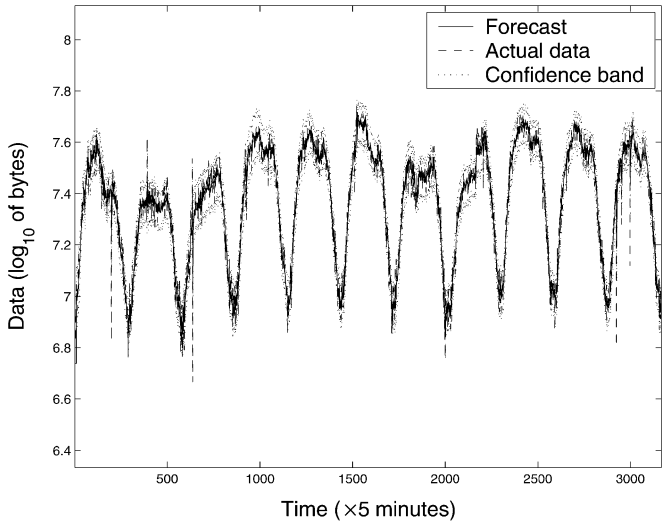


Fig. 10. Another example at a different commercial website. 5-min-ahead prediction for the weekday data, with a 90% confidence band. RMSE of prediction is equal to 0.0439. Actual coverage of the confidence band is equal to 89.7%. Median width of the confidence band is equal to 0.1182.

regression of order 41 (first 20 sines, first 20 cosines, and the DC (zero-frequency) component) for the long-term model, and suppressed the order-changing moves; the rest of the parameters remaining same. The RMSE for the 5-min-ahead prediction in this setting turned out to be 0.0593, which is slightly worse than 0.0586, which was obtained with adaptive order selection. This clearly implies that any improvement in accuracy by employing a larger (and fixed) number of harmonics is more than compensated by the estimation errors in the value of the coefficients. Thus, it is better to have a smaller basis set with harmonics that contribute significantly to the model (which we determine on the basis of the *energy* and the *coherence* values).

Moreover, the algorithm is quite robust to the parameter values, and different values of the order-changing parameters (P_i, P_d, P_{in} and P_{de}), and different initial regression order for both the long-term ($|\mathcal{P}_0|$) as well as short-term model ($\mathcal{Q}_0$) do not affect the performance of the algorithm significantly.

Fig. 10 shows simulations on another set of data from a different commercial website, which also shows the superior performance of the proposed algorithm over that of [12]. For a 5-min-ahead prediction, we obtain an RMSE of 0.0439 with the actual convergence of the confidence band equal to 89.7%, and median width 0.1292. On the other hand, for the same data, the algorithm in [12] (Fig. 15) yields an RMSE of 0.0464 and the actual confidence band coverage equal to 86%, although the median width is slightly better there at 0.1132 as compared with 0.1182 of the proposed algorithm.

VI. Conclusion

We have proposed a novel scheme for the forecasting of a Web server workload time series which exhibits strong periodic patterns. A hierarchical framework is used to separately predict the long-term and the short-term components. Separating the time series into two components reduces the

data history required to train model, thereby reducing the impact of changes in the trend process. The long-term forecast is performed using dynamic harmonic regression, while the residual short-term component is tracked as an autoregressive process. The coefficients of both processes are tracked under a stochastic state-space setting, and the order of both regressions are adaptively selected by the SMC technique via resampling. Also, the predictions yield simultaneous confidence bands, which can be used to support probability-based service-level agreements and for optimal resource allocation. Simulation results also show that the algorithm is quite robust to the model parameters. Modeling the noise in the short-term model by a heavy-tailed distribution makes the algorithm robust to outliers in the data. Furthermore, the proposed model has the capability to automatically handle nonstationarity in both the long-term as well as the short-term data as it allows the model parameters to change with time.

REFERENCES

[1] V. A. F. Almeida and D. A. Menasce, *Capacity Planning for Web Services: Metrics, Models, and Methods*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[2] A. Chandra, W. Gong, and P. Shenoy, "Dynamic resource allocation for shared data centers using online measurements," in *Proc. 2003 ACM Joint Int. Conf. Measurement Modeling of Computer Systems (SIGMETRICS'03)*, Jun. 2003, vol. 31, pp. 300–301.

[3] R. P. Doyle, J. S. Chase, O. M. Asad, W. Jen, and A. M. Vahdat, "Model-based resource provisioning in a Web service utility," in *Proc. 2003 USENIX Symp. Internet Technologies Systems*, Seattle, WA, Mar. 26-28, 2003.

[4] D. P. De Farias, A. King, M. Squillante, and B. Van Roy, "Dynamic control of Web server farms," in *Proc. 2002 INFORMS Revenue Management Section Conf.*, New York, Jun. 13-14, 2002.

[5] D. P. Pazel, T. Eilam, L. L. Fong, M. Kalantar, K. Appleby, and G. Goldszmidt, "Neptune: A dynamic resource allocation and planning system for a cluster computing utility," in *Proc. 2002 IEEE/ACM Int. Symp. Cluster Computing Grid*, May 2002, pp. 48–55.

[6] M. E. Crovella and A. Bestavros, "Self-similarity in world wide Web traffic: Evidence and possible causes," in *Proc. 2002 IEEE/ACM Int. Symp. Cluster Computing Grid*, Dec. 1997, vol. 5, pp. 835–846.

[7] J. R. Gallardo, D. Makrakis, and M. Angulo, "Dynamic resource management considering the real behavior of the aggregate traffic," *IEEE Trans. Multimedia*, vol. 3, no. 2, pp. 177–185, June 2001.

[8] K. Kant and Y. Won, "Server capacity planning for Web traffic workload," *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 5, pp. 731–747, 1999.

[9] M. Arlitt and C. L. Williamson, "Internet Web servers: Workload characterization and performance implications," *IEEE Trans. Netw.*, vol. 5, no. 5, pp. 631–645, 1997.

[10] M. S. Squillante, D. D. Yao, and L. Zhang, "Web traffic modelling and Web server performance analysis," in *Proc. 1999 Conf. Decision Control*, Dec. 1999, pp. 4432–4439.

[11] D. Shen and J. L. Hellerstein, "Predictive models for proactive network management: Application to a production Web server," in *Proc. 2000 IEEE/IFIP Network Operations Management Symp.*, Apr. 2000, pp. 833–846.

[12] T. H. Li, "A hierarchical framework for modeling and forecasting Web server workload," *J. Amer. Stat. Assoc.*, vol. 100, no. 471, pp. 748–763, Sep. 2005.

[13] P. C. Young, D. J. Pedregal, and W. Tych, "Dynamic harmonic regression," *J. Forecast.*, vol. 18, pp. 369–394, 1999.

[14] T. H. Li and M. J. Hinich, "A filter bank approach for modeling and forecasting seasonal patterns," *Technometrics*, vol. 44, no. 1, pp. 1–14, 2002.

[15] R. Chen and J. S. Liu, "Sequential Monte Carlo methods for dynamic systems," *J. Amer. Stat. Assoc.*, vol. 93, pp. 1302–1044, 1998.

[16] X. Wang, R. Chen, and J. S. Liu, "Monte Carlo Bayesian signal processing for wireless communications," *J. VLSI Signal Process.*, vol. 30, no. 1–3, pp. 89–105, Jan.–Feb.–Mar. 2002.

[17] R. Chen, X. Wang, and J. S. Liu, "Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering," *IEEE Trans. Inf. Theory*, vol. 46, no. 6, pp. 2079–2094, Sep. 2000.

[18] C. Andrieu, M. Davy, and A. Doucet, "Efficient particle filtering for jump Markov systems—Applications to time-varying autoregressions," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1762–1770, Jul. 2003.

[19] A. C. Harvey, *Time Series Models*, 4th ed. New York: Harvester Wheatsheaf, 1993.

[20] G. Gross and F. D. Galiana, "Short-term load forecasting," *IEEE Proc.*, vol. 75, no. 12, pp. 1558–1573, Dec. 1987.

[21] I. Moghram and S. Rahman, "Analysis and evaluation of five short-term load forecasting techniques," *IEEE Trans. Power Sys.*, vol. 4, no. 4, pp. 1484–1491, Oct. 1987.

[22] F. J. Nogales, J. Contreras, A. J. Conejo, and R. Espinola, "Forecasting next-day electricity prices by time-series models," *IEEE Trans. Power Sys.*, vol. 17, no. 2, pp. 342–348, May 2002.

[23] D. C. Montgomery and E. A. Peck, *Introduction to Linear Regression Analysis*. New York: Wiley, 1992.

[24] D. Guo, X. Wang, and R. Chen, "Wavelet-based sequential Monte Carlo blind receivers in fading channels with unknown channel statistics," *IEEE Trans. Signal Process.*, vol. 52, no. 1, pp. 227–239, Jan. 2004.

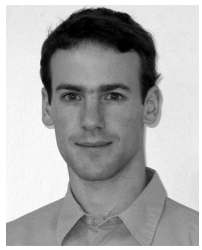
[25] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Stat. Comp.*, vol. 10, no. 3, pp. 197–208, 2000.

[26] A. Doucet and C. Andrieu, "Iterative algorithms for state estimation of jump Markov linear systems," *IEEE Trans. Signal Process.*, vol. 49, no. 6, pp. 1216–1227, Jun. 2001.

[27] R. Chen and J. S. Liu, "Mixture Kalman filters," *J. Amer. Stat. Assoc. (B)*, vol. 62, pp. 493–509, 2000.

[28] R. Prado, G. Huerta, and M. West, "Bayesian time-varying autoregressions: Theory, methods and applications," *J. Inst. Math. Statist. Univ. Sao Paolo*, no. 4, pp. 405–422, 2000.

[29] R. Prado and G. Huerta, "Time-varying autoregression with model order uncertainty," *J. Time Series Anal.*, vol. 23, no. 5, pp. 599–618, 2002.



Tom Vercauteren graduated from Ecole Polytechnique, Paris, France, in 2003 and received the M.S. degree in the Department of Electrical Engineering, Columbia University, New York, in 2004. He is currently working towards the Ph.D. degree in INRIA, Sophia-Antipolis, France.

His research interests are in the area of statistical signal processing.

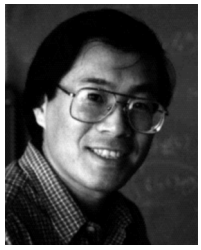
Pradeep Aggarwal received the B.Tech. degree in electrical engineering from the Indian Institute of Technology—Bombay, India, in 2004 and the M.S. degree in electrical engineering from Columbia University, New York, in 2006. He is currently working towards the Ph.D. degree in electrical engineering at Columbia University.

His research interests are in the area of communications and statistical signal processing.

Xiaodong Wang (S'98–M'98–SM'04) received the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ.

Currently, he is on the faculty of the Department of Electrical Engineering, Columbia University. His research interests fall in the general areas of computing, signal processing, and communications, and he has published extensively in these areas. Among his publications is a recent book entitled *Wireless Communication Systems: Advanced Techniques for Signal Reception* (Englewood Cliffs, NJ: Prentice-Hall, 2003). His current research interests include wireless communications, statistical signal processing, and genomic signal processing.

Dr. Wang received the 1999 NSF CAREER Award and the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the IEEE TRANSACTIONS ON INFORMATION THEORY.



Ta-Hsin Li (S'89–M'92–SM'04) received the M.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 1984 and the Ph.D. degree in applied mathematics from the University of Maryland, College Park, in 1992.

From 1992 to 1998, he taught at Texas A&M University, College Station, and the University of California, Santa Barbara, as an Assistant and Associate Professor of statistics. Since 1999, he has been with the IBM T. J. Watson Research Center, Yorktown Heights, NY. He also serves as Adjunct Professor at the Electrical Engineering Department of Columbia University, New York. His current research interests include time-series analysis, spectral and wavelet analysis, and statistical signal processing.

Dr. Li is a Fellow of the American Statistical Association (ASA). He currently serves as Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and for the *EURASIP Journal on Applied Signal Processing*.