An Approach to Predictive Detection for Service Management

Joseph L. Hellerstein[1], *Fan Zhang*[2], *and Perwez Shahabuddin*[2]

[1] *IBM T.J. Watson Research Center*
Hawthorne, New York, USA
hellers@us.ibm.com

[2] *Industrial Engineering and Operations Research*
Columbia University
New York City, New York USA
{fzhang, perwez}@ieor.columbia.edu

Abstract

Service providers typically define quality of service problems using threshold tests, such as "Are HTTP operations greater than 12 per second on server XYZ?" This paper explores the feasibility of *predicting* violations of threshold tests. Such a capability would allow providers to take corrective actions in advance of service disruptions. Our approach estimates the probability of threshold violations for specific times in the future. We model the threshold metric (e.g., HTTP operations per second) at two levels: (1) nonstationary behavior (as is done in workload forecasting for capacity planning) and (2) stationary, time-serial dependencies. Using these models, we compute the probability of threshold violations. We assess our approach using measurements of HTTP operations per second collected from a production web server. These assessments suggest that our approach works well if (a) the actual values of predicted metrics are sufficiently distant from their thresholds and/or (b) the prediction horizon is not too far into the future.

Keywords

Proactive Management, Service Level Agreement, Forecasting Models, Time-series Models

1 Introduction

The tremendous growth in network-based services has greatly increased the pressure on administrators to rapidly detect and resolve service problems. Typically, problem detection is done by specifying threshold tests. Examples include: "Do ping response times exceed .7 seconds?" and "Are HTTP operations greater than 12 per second on server XYZ?" Unfortunately, once detection occurs, there is often little time to take correction actions. This paper describes an approach to predicting threshold violations, thereby enabling administrators to take corrective action in advance of wide-spread service disruptions.

Threshold tests (hereafter, just thresholds) are specified by a predicate consisting of: a measurement variable or function of measurement variables (e.g., LAN utilization over the last five minutes), a comparison operator (e.g., greater than), and a threshold value (e.g., 30%). A threshold violation occurs if this such a predicate is satisfied. Typically, this results in an alarm, such a flashing red light on an operator's display.

We are interested in predictive detection. By this, we mean predicting that a threshold violation will occur. In our view, such predictions have two components: (1) the *probability* that the threshold will be violated and (2) the *time* at which it is anticipated that the threshold will be violated. An example of predictive detection is "There is a 60% probability that in 5 minutes the requests for hypertext transfer protocol (HTTP) operations on web server XYZ will exceed 12 per second." By knowing the probability of a threshold violation, service providers can assess the seriousness of the situation. By knowing the time frame, service providers can determine what level of intervention is possible.

We investigate predictive detection in the context of a web server. In particular, we extend our work in [6] that considers data collected over eight months (June, 1996 through January, 1997) from a production web server at a large corporation using the collection facility described in [2]. Each observation in the data set contains approximately twenty variables that are aggregated over five minute intervals; 288 five minute intervals are reported for each day. We focus on HTTP operations per second, which is the measurement variable **httpop/s**. HTTP operations include gets, posts, and the rate at which cgi (common gateway interface) scripts are initiated.

Many researchers have investigated the detection of service degradations. Central to this is characterizing normal, nonstationary behavior, as is done in [12] (who uses ad hoc models to estimate weekly patterns), [9] (who employs more formal time series methods), and [10] (who uses knowledge of the functional relationship between inputs and outputs to detect changes in system operation). Work on predictive detection has been much more limited. Statistical process control (SPC) employs warning limits on measurement variables to provide advance notice of threshold violations (e.g., [13]). However, violating a warning limit gives little insight into when (or if) a threshold violation will occur. Research in data mining has addressed ways to anticipate the next pat-

tern in a sequence (e.g., [3]). But these efforts do not provide the anticipated time of this occurrence, and the approaches employed are more suited to categorical data (e.g., event type) than to the continuous variates considered in our work (e.g., httpop/s). References [8] and [14] describe techniques for detecting changes in networks that are leading indicators of service interruptions. This work complements ours in that if these indicators can be expressed as threshold tests, then our work provides a way to predict service interruptions. Our efforts follow the lines of workload and resource usage forecasting in capacity planning, which employs curve extrapolation to predict threshold violations (e.g., [11]). In essence, we extend these approaches to consider stationary, time-serial dependencies and the probability of threshold violations.

This paper explores the feasibility of predicting threshold violations by studying httpop/s in a production web server. Section 2 describes the model we use for the normal behavior of httpop/s. Section 3 uses this model to develop our approach to predictive detection. Section 4 assesses its effectiveness. Our conclusions are contained in Section 5.

2 Model of Normal Behavior

This section summarizes our model of normal behavior of httpop/s. More details can be found in [6].

The model consists of two submodels. The first addresses time-varying or nonstationary behavior of the measurement variable, as is done in workload forecasting for capacity planning. The second addresses stationary, time-serial dependencies once the effect of the first model have been removed.

We begin by modeling nonstationary behavior. An example of such behavior is shown in part (a) of Fig. 1, which plots two weeks of httpop/s collected from the web server we study. Observe the pronounced and consistent trend based on time-of-day.

Our approach is similar to that used in workload forecasting for capacity planning. We consider the factors time-of-day, day-of-week, and month. Let S_{ijkl} be the random variable for httpop/s in the i-th five minute interval (e.g., $i = 2$ is the interval [12:05am, 12:10am)), the j-th day of the week (e.g., $j = 3$ is Wednesday), the k-th month (e.g., $k = 1$ is June), and the l-th instance in the data (since several i and j occur within the same k). Let s_{ijkl} be the observed value of S_{ijkl}. Fig. 1 part (a) plots s_{ijkl} for a work week (Monday through Friday) in June of 1996 and a work week in November of 1996. The x-axis is time in hours, and the y-axis is httpop/s.

We proceed by using notation from analysis of variance (ANOVA, e.g., [4]). We partition S_{ijkl} into five components: the overall mean (μ), the deviation from the mean due to the i-th time-of-day value (α_i), the deviation due to the j-th day of the week (β_j), the contribution of the k-th month (γ_k), and a random variable that quantifies statistical errors (Y_{ijkl}). By definition, $\sum_i \alpha_i = \sum_j \beta_j = 0$. (The γ_k are obtained using least squares regression and so do not

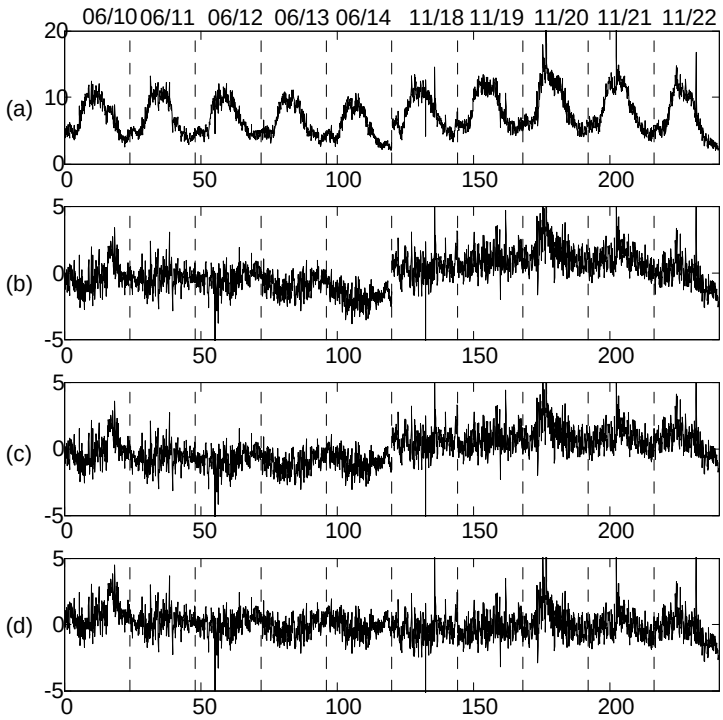

Figure 1: Illustrate Data for Model of Nonstationary Behavior. x-axis is time in hours; y-axis is httpop/s. (a) is raw data; (b) removes $\hat{\mu}$ and $\hat{\alpha}_i$; (c) also removes $\hat{\beta}_j$; (d) also removes $\hat{\gamma}_k$

---

necessarily sum to 0.) Our submodel for nonstationary behavior is:

$$S_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + Y_{ijkl}. \tag{1}$$

We evaluate the quality of this model in two ways. First, we note that the model accounts for 64.18% percent of the variability in the data, which is fairly good for models of production computer systems. Second, we examine the **residuals**, what remains after the effects of the model have been removed. In essence, the residuals estimate of Y_{ijkl}. Let, $\hat{\mu}$, $\hat{\alpha}_i$, $\hat{\beta}_j$, and $\hat{\gamma}_k$ be estimates of the model parameters as obtained using the techniques described in [6]. Then,

$$\hat{Y}_{ijkl} = s_{ijkl} - \hat{\mu} - \hat{\alpha}_i - \hat{\beta}_j - \hat{\gamma}_k.$$

These residuals are plotted in Fig. (1), part (d). Observe that little remains in the way of systematic behavior. That is, our model has done a good job of

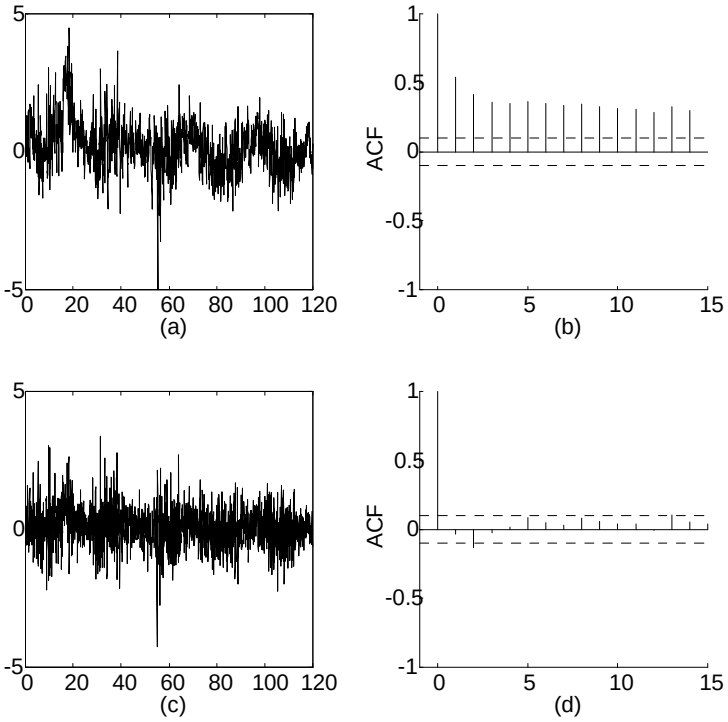

Figure 2: Autocorrelations of Data Used to Illustrate Building the Characterization Model: (a) residuals of Eq. (3) model for First Week in Fig. 1; (b) ACF of residuals in (a); (c) Residuals after AR(2) model (Eq. (4)) is applied to (a); (d) ACF of residuals in (c).

removing predictable patterns.

Although Eq. (1) does well with removing nonstationary effects, it turns out that a somewhat subtle behavior remains. These are stationary, time-serial dependencies. Such dependencies can be detected by plotting the autocorrelation function (ACF) of the residuals.

Fig. 2 part (a) plots the residuals of Eq. (1) for the week of June 10 (i.e., the first half of part (d) of Fig. 1). Part (b) of this figure plots the ACF of these residuals. The y-axis is the correlation value (which lies between -1 and 1); the x-axis is the lag (number of time intervals) between measurements that are used for that correlation. ACF values that lie between the dashed-lines are statistically identical to zero (i.e., would not reject the hypothesis of 0 autocorrelation at a significance level of 5%). The correlation at lag 0 is always one since this is the variable correlated with itself at the same lag. Note that in

part (b) of the figure, all correlations are above the dashed line. This suggests that the data contain significant time-serial dependencies.

To remove these dependencies, we employ the following model:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + U_t, \tag{2}$$

where the U_t are independent and identically distributed (i.i.d.) random variables with mean 0 and variance σ_u^2. Eq. (2) is a second order autoregressive model (AR(2)); ϕ_1 and ϕ_2 are parameters of the model. The model parameters are estimated from the data using standard techniques [1]. Fig. 2 part (c) plots the residuals of Eq. (2), that is: $u_t = y_t - \hat{\phi}_1 y_{t-1} - \hat{\phi}_2 y_{t-2}$, where $\hat{\phi}_m$ is an estimator of ϕ_m. In our data, $\hat{\phi}_1 = .4632$ and $\hat{\phi}_2 = .2111$. Part (d) displays the ACF of the residuals of Eq. (2). Observe that almost all correlation values lie within the dashed lines. This suggests that the autocorrelations have been removed, which is consistent with the U_t being independent random variables. From the u_t, we can compute $\hat{\sigma}_u^2$, the unbiased estimator of σ_u^2 (e.g., as in [4]). In our data, $\hat{\sigma}_u^2 = .011109$.

3 Approach to Predictive Detection

This section describes our approach to predictive detection. We begin with an overview of the approach, followed by a detailed description of its operation. We conclude with several observations about the characteristics of our approach.

Our approach to predictive detection extends existing techniques employed in workload forecasting for capacity planning. These techniques model the nonstationary behavior of the mean of a metric (e.g., httpop/s), as in Eq. (1). We do the same. In addition, we consider stationary, time-serial dependencies that remain after the effects of nonstationarities are removed. Such time serial dependencies are modelled using Eq. (2). Based on these two models, we estimate the probability of a threshold violation.

We use the model of nonstationary behavior (a modification of Eq. (1)) to transform the time-varying measurements of the s_t into stationary y_t. Since we transform the measurements, we must transform the thresholds in the same way. An implication of the latter is that transformed thresholds vary with time (since the transformation is time-indexed). We use th_t to denote the transformed value of T at time t.

Let $\hat{y}_t(h)$ denote a prediction that is h time units in the future, where h is the **prediction horizon**. $\hat{y}_t(h)$ is computed as $E[Y_{t+h} \mid Y_t = y_t, Y_{t-1} = y_{t-1}]$, which is the expected value of Y at time $t+h$ given values of y_t and y_{t-1} (which are computed from s_t and s_{t-1}). Because of random variations, $\hat{y}_t(h)$ may be larger than th_t, but the actual value of y_{t+h} may be less than th_t. For this reason, our approach calculates $\hat{P}_t(h) = P[Y_{t+h}$ violates a threshold $\mid Y_t = y_t, Y_{t-1} = y_{t-1}]$. $\hat{P}_t(h)$ incorporates information about both the expected value

and variance of Y_{t+h}. $\hat{P}_t(h)$ is computed for $h = 1, 2, ..., H$ time units into the future.

In general, there may be several threshold tests for a measurement variable or function of measurement variables. Following the conventions of statistical process control, we consider two thresholds for s_t. T is an upper threshold, which means that an alarm is raised if $s_t > T$. T' is a lower threshold; an alarm is raised if $s_t \leq T'$. We assume that $T' \leq T$.

Before continuing, we define more precisely what we mean by *transformed* values of s_t. A common difficulty with measurements of queueing systems is that the variance of metrics increases as their mean increases. Such metrics cannot be modelled as identically distributed Gaussian (normal) random variables, which in turn makes it difficult to calculate the probability that the metric violates a threshold. We address this difficulty in a standard way by modifying Eq. (1) so as to stabilize the variance of the residuals. Thus, instead of Eq. (1), we use

$$\ln(S_{ijkl} + 1) = \mu + \alpha_i + \beta_j + \gamma_k + Y_{ijkl}. \tag{3}$$

(The 1 is added to ensure that transformed values are non-negative.) By transforming the s_t, we mean computing the residuals of the above equation from the measurements. Our analysis of these residuals for httpop/s in the web server data suggests that it is reasonable to assume that the U_t are i.i.d. Gaussians, where these U_t are obtained by applying Eq. (2) to the Y_{ijkl} in Eq. (3). Further, it turns out that using Eq. (3) instead of Eq. (1) does not change the values of ϕ_1, ϕ_2 in for our data.

We now describe our algorithm for predictive detection. We consider a single routine, predict, that estimates the probability of threshold violations. predict assumes that estimates have been obtained for μ, α_i, β_j, γ_k, ϕ_1, and ϕ_2. It is also assumed that there is a separate process that uses Eq. (3) to transform s_t into y_t using estimates of these parameters.

The predict routine is invoked on demand, such as to update an operator display. predict takes as input the current time index (t), the lower threshold (T'), the upper threshold (T), and the maximum time horizon (H). predict returns $\hat{P}_t(1), ..., \hat{P}_t(H)$–the probabilities of violating either T' or T at each time horizon h.

Fig. 3 details the four steps in predict. Step one estimates the value of future measurements by computing $\hat{y}_t(h)$, where $\hat{y}_t(h) = E[Y_{t+h} \mid Y_t = y_t, Y_{t-1} = y_{t-1}]$ (e.g., see [1]). Step two estimates the variance of the future measurement given y_t, y_{t-1}; that is,

$$\sigma_t^2(h) = Var(Y_{t+h} \mid Y_t = y_t, Y_{t-1} = y_{t-1}).$$

Step three estimates the probabilities of violating the thresholds based on these estimates. Steps one, two, and three are repeated for $h = 1$ through H. Step four returns the probabilities of threshold violations.

Steps two and three require some additional explanation. For step two, we explain the computation of $\sigma_t^2(h)$. We begin by expressing Y_t as a function of

predict(t, T',T, H) returns($\hat{P}_t(1),...,\hat{P}_t(H)$)

1. Estimate the forecast means for $1,2,...H$

$$\begin{aligned}\hat{y}_t(1) &= \hat{\phi}_1 History[1] + \hat{\phi}_2 History[2] \\ \hat{y}_t(2) &= \hat{\phi}_1 \hat{y}_t(1) + \hat{\phi}_2 History[1] \\ &\cdots \\ \hat{y}_t(H) &= \hat{\phi}_1 \hat{y}_t(H-1) + \hat{\phi}_2 \hat{y}_t(H-2)\end{aligned}$$

2. Estimate the forecast variances for $h = 1,2,...H$

$$\hat{\sigma}_t^2(h) = \hat{\sigma}_u^2[\sum_{n=0}^{h-1} \frac{(A_1^{n+1} - A_2^{n+1})^2}{(A_1 - A_2)^2}]$$

3. Estimate $P_t(h)$, the probability that the forecast value $\leq T'$ or $> T$ for $h = 1,2,...H$.

$$\hat{P}_t(h) = \Phi\left(\frac{\hat{y}_t(h) - th_{t+h}}{\hat{\sigma}_t(h)}\right) + \Phi\left(\frac{th'_{t+h} - \hat{y}_t(h)}{\hat{\sigma}_t(h)}\right)$$

   - (i',j',k') are the indicies for $t+h$
   - $th'_{t+h} = \ln(1+T') - \hat{\mu} - \hat{\alpha}_{i'} - \hat{\beta}_{j'} - \hat{\gamma}_{k'}$
   - $th_{t+h} = \ln(1+T) - \hat{\mu} - \hat{\alpha}_{i'} - \hat{\beta}_{j'} - \hat{\gamma}_{k'}$
   - $\Phi(x)$ is the CDF of the standard normal

4. Return($\hat{P}_t(1),...,\hat{P}_t(H)$)

Figure 3: Algorithm for Predictive Detection

U_t. Using the backshift operator B (i.e. $B^m X_t = X_{t-m}$) as in [1], we rewrite Eq. (2) as

$$(1 - \phi_1 B - \phi_2 B^2) Y_t = U_t. \tag{4}$$

That is,

$$Y_t = \frac{U_t}{(1-A_1B)(1-A_2B)},$$

where A_1, A_2 are the reciprocal of the zeroes in $1 - \phi_1 B - \phi_2 B^2 = 0$. For example, if $\phi_1 = .4632$ and $\phi_2 = .2111$, then $A_1 = -0.2829$ and $A_2 = 0.7461$. Using partial fraction expansion, we have:

$$\begin{aligned}Y_t &= \frac{A_1 U_t}{(A_1 - A_2)(1 - A_1 B)} - \frac{A_2 U_t}{(A_1 - A_2)(1 - A_2 B)} \\ &= \sum_{n=0}^{\infty} \frac{(A_1^{n+1} - A_2^{n+1}) B^n U_t}{A_1 - A_2}\end{aligned} \tag{5}$$

Note that for Y_t to have a finite variance (i.e., be stable), it is required that $\mid A_m \mid < 1$ [1]. Also note that $\hat{y}_t(h) \rightarrow 0$ as h becomes large. This follows from the computation of $\hat{y}_t(h)$ in Step 1 of Fig. 3 and the fact that $\mid \phi_m \mid < 1$ for httpop/s.

To compute $\sigma_t^2(h)$, we express Y_t in terms of $U_{t+1}, ..., U_{t+h}$. (We need not consider U_v, $v \leq t$ since the forecast is conditioned on y_t, y_{t-1}.) Note that $h-1$ is the power of B that selects U_{t+1} for Y_{t+h}. Further, recall that the U_t are independent and identically distributed. Hence,

$$\sigma_t^2(h) = \sigma_u^2[\sum_{n=0}^{h-1} \frac{(A_1^{n+1} - A_2^{n+1})^2}{(A_1 - A_2)^2}]. \tag{6}$$

We obtain $\hat{\sigma}_t^2(h)$ by substituting $\hat{\sigma}_u^2$ for σ_u^2.

We make a few observations about $\sigma_t^2(h)$. First, observe that for $h = 1$, $\sigma_t^2(1) = \sigma_u^2$. This makes sense since $Y_{t+1} = \phi_1 y_t + \phi_2 y_{t-1} + U_{t+1}$, and only U_{t+1} is a random variable (since the observations through t are assumed to be known). Second, note that $\sigma_t^2(h)$ is non-decreasing in h. This is also intuitive since we are less certain about a forecast the further it is in the future. Also, note that it is intuitive that $\sigma_t^2(h)$ converges to the variance of Y_t as $h \rightarrow \infty$.

Next, we explain the computation of $\hat{P}_t(h)$, the probability of violating a threshold at time $t+h$ given knowledge of the measurements through time t. A threshold violation occurs if either $S_t \leq T'$ or $S_t > T$. That is,

$$\hat{P}_t(h) = P(Y_{t+h} \leq th'_{t+h} \text{ or } Y_{t+h} > th_{t+h} \mid y_t, y_{t-1}), \tag{7}$$

where th'_{t+h} is T' transformed into y units. Since conditioning on y_t, y_{t-1} is assumed throughout, we do not explicitly express this in the sequel.

We proceed in the standard manner by assuming that the estimated model parameters are in fact constants (e.g., [1]). Observe that from Eq. (5), Y_{t+h} is Gaussian since it is a linear combination of Gaussians. Further, the expected value of Y_{t+h} is $\hat{y}_t(h)$, and its variance is $\hat{\sigma}_t^2(h)$ (see Eq. (6)). Thus, $Z_{t+h} = \frac{Y_{t+h} - \hat{y}_t(h)}{\hat{\sigma}_t(h)}$ is a standard normal. So,

$$\begin{aligned}
\hat{P}_t(h) &= P(Y_{t+h} \leq th'_{t+h} \text{ or } Y_{t+h} > th_{t+h}) \\
&= 1 - P(\tfrac{th'_{t+h} - \hat{y}_t(h)}{\hat{\sigma}_t(h)} < Z_{t+h} \leq \tfrac{th_{t+h} - \hat{y}_t(h)}{\hat{\sigma}_t(h)}) \\
&= 1 - \left[\Phi\left(\tfrac{th_{t+h} - \hat{y}_t(h)}{\hat{\sigma}_t(h)}\right) - \Phi\left(\tfrac{th'_{t+h} - \hat{y}_t(h)}{\hat{\sigma}_t(h)}\right)\right] \\
&= \Phi\left(\tfrac{\hat{y}_t(h) - th_{t+h}}{\hat{\sigma}_t(h)}\right) + \Phi\left(\tfrac{th'_{t+h} - \hat{y}_t(h)}{\hat{\sigma}_t(h)}\right),
\end{aligned} \tag{8}$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal at x.

We make a few observations about this result. First, note that each variable in the arguments of Φ is determined entirely by only one of the submodels of

S_t. th_{t+h} is determined by the model of nonstationary behavior (Eq. (3)); $\hat{y}_t(h)$ and $\hat{\sigma}_t(h)$ are calculated using the model of stationary, time-serial dependencies (Eq. (2)). Further, recall that as h grows large, $\hat{y}_t(h) \rightarrow 0$ and $\hat{\sigma}_t(h)$ converges. Thus, for large h, the model of nonstationary behavior determines how $\hat{P}_t(h)$ changes with h.

Now consider the influence of the thresholds. If T becomes very large, then

$$\begin{aligned} \hat{P}_t(h) &= \Phi(-\infty) + \Phi\left(\frac{th'_{t+h} - \hat{y}_t(h)}{\hat{\sigma}_t(h)}\right) \\ &= \Phi\left(\frac{th'_{t+h} - \hat{y}_t(h)}{\hat{\sigma}_t(h)}\right). \end{aligned}$$

Similarly, if T' becomes very small, then $\hat{P}_t(h) = \Phi\left(\frac{\hat{y}_t(h) - th_{t+h}}{\hat{\sigma}_t(h)}\right)$. Further, if $T \approx T'$, then $\hat{P}_t(h) \approx 1 - \left[\Phi\left(\frac{th_{t+h} - \hat{y}_t(h)}{\hat{\sigma}_t(h)}\right) - \Phi\left(\frac{th_{t+h} - \hat{y}_t(h)}{\hat{\sigma}_t(h)}\right)\right] = 1$. That is, if the lower and upper thresholds are equal, we always violate a threshold.

Finally, note that the probability of a threshold violation at time $t+h$ depends on two factors. The first is the magnitude by which the predicted value violates the threshold. That is, $Max\{\hat{y}_t(h) - th_{t+h}, th'_{t+h} - \hat{y}_t(h)\}$. As this value increases, so does $\hat{P}_t(h)$. The second factor is $\hat{\sigma}_t(h)$, the standard deviation of the forecast. A larger $\hat{\sigma}_t(h)$ results in a smaller $\hat{P}_t(h)$.

4 Assessment of Approach

To gain insight into how well our approach works in practice, we study $\hat{P}_t(h)$ using the web data for a threshold that is fixed in s units. Our objective is to understand how $\hat{P}_t(h)$ varies with s_{t+h}, the value of the observation for which predictive detection is done. Throughout, we simplify matters by only considering an upper threshold, T; that is, $T' = -\infty$.

We compare $\hat{P}_t(h)$ with the ideal probability of a threshold violation. The latter is denoted by $P^*_t(h)$, where

$$P^*_t(h) = \begin{cases} 1 & \text{if } s_{t+h} > T \\ 0 & \text{otherwise} \end{cases}$$

Clearly, $P^*_t(h)$ can never be achieved in practice since it requires prior knowledge of future measurements! Thus, we do not expect to be anywhere close to this ideal. Rather, we use $P^*_t(h)$ as a reference from which values of $\hat{P}_t(h)$ are assessed.

In this study, $T = 7$. We construct confidence limits for $\hat{P}_t(h)$ ($\pm$ two standard deviations around the sample mean) by partitioning the s_{t+h} into intervals. Fig. 4 plots the results. There are ten plots, one for each value of h. In each, s_{t+h} is the x-axis and $\hat{P}_t(h)$ is the y-axis. The solid lines are mean

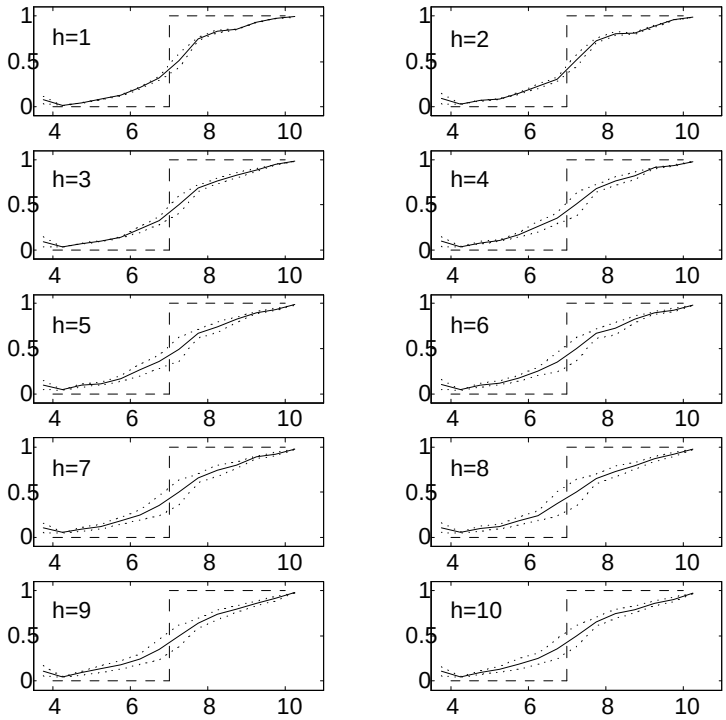

Figure 4: Assessment of the relative accuracy of $\hat{P}_t(h)$. The x-axis is s_{t+h}; the y-axis is probability. Dashed lines specify $P_t^*(h)$, the ideal estimate of the probability of exceeding the threshold. Solid lines are average $\hat{P}_t(h)$; dotted lines are $\pm$ two standard deviations around the average.

---

values of $\hat{P}_t(h)$; the dotted lines are the confidence limits. The dashed lines (which are the same in all ten plots) depict $P_t^*(h)$.

We use Fig. 4 to compare our approach with the ideal algorithm for predictive detection. Note that $\hat{P}_t(h)$ closely approximates $P^*(h)$ for values of s_{t+h} that are distant from T, such as $s_{t+h} < 6$ or $s_{t+h} > 8$. Put differently, in these ranges, $\mid P^*(h) - \hat{P}_t(h) \mid$ is small. This means that if $\hat{P}_t(h)$ is large, service providers can be certain that a threshold violation will occur in h time units. Conversely, if $\hat{P}_t(h)$ is small, service providers are assured that a threshold violation will not occur in h time units.

Now consider the effect of h on our ability to approximate $P^*(h)$. As h increases, s_{t+h} must be more distant from T in order to achieve the same value of $\mid P^*(h) - \hat{P}_t(h) \mid$. This is a consequence of two factors. First, the variance of

$\hat{Y}_t(h)$ increases with h, which causes $\hat{P}_t(h)$ to converge to .5. Second, for large h, $\hat{y}_t(h) \rightarrow 0$. Both effects increase $\mid P^*(h) - \hat{P}_t(h) \mid$.

There is another implication of these effects. As h increases, there is a diminished contribution of the model of time serial behavior to the calculation of $\hat{P}_t(h)$. To see this, observe how the $\hat{P}_t(h)$ curve in Fig. 4 flattens as h changes from 1 to 5. But from 6 through 10, the curve and its confidence intervals are relatively unchanged. This means that we are relying almost entirely on the model of nonstationary behavoir.

What happens if we want to detect small deviations from the threshold? From Fig. 4, we see that the $\hat{P}_t(h)$ confidence limits are widest when $s_{t+h} = 7 = T$. Thus, detecting small deviations from the threshold requires a very low model variance.

In addition to the assessments herein presented, we have: (a) compared $\hat{P}_t(h)$ with the measured fraction of the observations that violate thresholds and (b) used simulation to study the accuracy of our approach. These results suggest that our approach works quite well. Details can be found in [7].

5 Conclusions

This paper explores the feasibility of predicting threshold violations. Such a capability would be of immense benefit to service managers in that corrective actions could be taken before there are wide spread service disruptions.

We believe that predictive detection must include both the probability that the threshold will be violated and an occurrence time. The probability conveys the likelihood that corrective action is needed; the occurrence time constrains the actions that can be taken. While others have proposed approaches to proactive detection in networked systems (e.g., [8], [14]), our approach is the first that addresses predictive detection–providing the probability that a threshold will be violated at specific times in the future.

Our approach can be viewed as an extension of techniques for workload forecasting in capacity planning. These techniques model nonstationary behavior of metrics. We do the same. In addition, we model stationary, time-serial dependencies. We use both models to compute $\hat{P}_t(h)$, the probability of violating a threshold for a time horizon of h given that the current time is t. This is done in a manner that considers both lower and upper thresholds for measurement values.

We provide insight into our approach to predictive detection by applying it to measurements of a production web server. These results suggest that our approach works well if (a) the actual values of predicted metrics are sufficiently distant from their thresholds and/or (b) the prediction horizon is not too far into the future.

We show that for smaller values of h, modeling time-serial behavior provides considerable predictive benefit. However, as h increases, $\hat{P}_t(h)$ converges

to what would be obtained by only considering the model of nonstationary behavior.

Another insight is that our approach to predictive detection works best when s_{t+h} (the observed value at time $t+h$) is distant from the threshold value, either smaller or larger. This is a consequence of the fact that the variance of $\hat{P}_t(h)$ is largest when s_{t+h} is near the threshold. Also, for predictions made further into the future (i.e., larger values of h), s_{t+h} must be even more distant from the threshold since variance is nondecreasing in h.

While our results are encouraging, much work remains. Clearly, a broader range of measurement variables should be studied. Also, to be practical, parameter estimation should be on-line rather than off-line (via a fixed set of training data). Further, a variety of technical issues need to be addressed more carefully, such as the manner in which probabilities are estimated for thresholds that are at the tails of the distribution (which is very sensitive to deviations from a Gaussian distribution). Finally, some thought must be given as to how predictive detection should be employed in practice. Informing customers of *potential* service disruptions may or may not be a good idea, depending on their expectations. One possibility here is to describe predictive detection as a kind of weather report for future service delivery.

Acknowledgements

This work was supported in part by NSF Career Award Grant DMI-96-25291 and a grant from the IBM Corporation.

References

[1] **George E. P. Box and Gwilym M. Jenkins**: *Time Series Analysis Forecasting and Control*, Prentice Hall, 1976.

[2] **Adrian Cockcroft**: "Watching your Web server," SunWorld OnLine, http://www.sunworld.com/swol-03-1996/swol-03-perf.html, 1998.

[3] **Thomas G. Dietterich and Ryszard S. Michalski**: "Discovering Patterns in Sequences of Events," *Artificial Intelligence*, **25**, 187-231, 1985.

[4] **Wilfrid J. Dixon and Frank J. Massey**: *Introduction to Statistical Analysis*, McGraw-Hill Book Company, 1969.

[5] **N.R. Draper and H. Smith**: *Applied Regression Analysis*, John Wiley and Sons, 1968.

[6] **Joseph L. Hellerstein, Fan Zhang, and Perwez Shahabuddin**: "Characterizing Normal Operation of a Web Server: Application to Workload Forecasting and Problem Detection," *Proceedings of the 1998 Conference of the Computer Measurement Group*, Anaheim, California, December 7-11, 1998.

[7] **Joseph L. Hellerstein, Fan Zhang, and Perwez Shahabuddin**: "An Approach to Predictive Detection for Service Management," IBM Research Report, RC 21254, August 4, 1998.

[8] **C.S. Hood and C. Ji**: "Proactive Network Fault Detection.," *Proceedings of INFOCOM*, Kobe, Japan, 1997.

[9] **P. Hoogenboom and J. Lepreau**: "Computer System Performance Problem Detection Using Time Series Models," *Proceedings of the Summer USENIX Conference*, 15-32, 1993.

[10] **R. Isermann and B. Freyermuth**: "Process Fault Diagnosis Based on Process Model Knowledge," *Proceedings of 1989 ASME International Computers In Engineering Conference and Exposition*, July, 631-642, 1989.

[11] **G. Jay Lipovich**: "Fixing Capacity Planning's Achilles Heel: An Approach to Managing Forecast Accuracy," *Proceedings of the 1997 Conference of the Computer Measurement Group*, Orlando Florida, December 8-12, 1997.

[12] **Roy A. Maxion**: "Anomaly Detection for Diagnosis," *Proceedings of the 20th Annual International Symposium on Fault Tolerant Computing (FTCS)* 20, June 1990, pp. 20-27.

[13] **William I. Sikora**: "Response Time Measurement and SPC," *Computer Measurement Group Transactions*, pp.35-42, Summer, 1992.

[14] **Marina Thottan and Chuanyi Ji**: "Adaptive Thresholding for Proactive Network Problem Detection," *Third IEEE International Workshop on Systems Management*, Newport, Rhode Island, April 22-24, 1998, pp. 108-116.