RC25431 (WAT1312-014) December 9, 2013
Computer Science

IBM Research Report

A Framework for Predicting Services Delivery Efforts Using IT Infrastructure-to-Incident Correlation

Joel W. Branch, Yixin Diao, Larisa Shwartz
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
USA

A Framework for Predicting Services Delivery Efforts Using IT Infrastructure-to-Incident Correlation

Joel W. Branch, Yixin Diao, and Larisa Shwartz
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
{branchj, diao, lshwart}@us.ibm.com

***Abstract*—Predicting IT infrastructure performance under varying conditions, e.g., the addition of a new server or increased transaction loads, has become a typical IT management exercise. However, within a services delivery context, enterprise clients are demanding predictive analytics that outline future "costs" associated with changing conditions. The *services delivery staffing* costs incurred in addressing problems and requests (arriving in the form of incident and other problem tickets) in the managed environment is especially of high importance. This paper describes a framework and analytical study addressing such cost prediction. Specifically, a novel approach is described in which (1) a framework combining various analytical models is proposed to predict services delivery staffing requirements under changing IT infrastructure characteristics and conditions, and (2) machine learning techniques are used to predict service delivery workloads (measured by ticket volumes) based on managed server characteristics. Detailed descriptions of the workload prediction techniques, as well as an evaluation using data from an actual large service delivery engagement, are presented in this paper.**

Keywords—IT management costs, IT ticket analysis, prediction, services delivery.

I. Introduction

Technological advances, liberalization, and evolving business trends have resulted in an increasingly significant role for customers and users of information technology (IT) services. Customers, particularly corporate customers, for whom responsive and efficient IT capabilities are essential to their core businesses, are becoming more demanding and knowledgeable concerning the services they purchase. They must see the benefit of subscribing to a new service or to a new feature in an existing service, and it must be at the price that they are prepared to pay. At the same time more and more organizations are becoming more dependent on IT to run their core business. Enterprise customers expect a high level of convenience and dynamic response in services tailored to their specific requirements. There are not only more services to choose from, but also a greater variety, being of low-cost, high-volume, as well as high quality customized services. In this competitive environment, customer requirements regarding the control that they have on the services they purchase may well influence the purchasing decision. A basic requirement is therefore access to *status*, *performance*, *fault*, and *accounting statistics* of these services. Some also want more active control over their services. They may wish to be able to change their service configurations easily and quickly. In order to keep costs within acceptable bounds, they may want to understand how changing such configurations would affect their costs going forward. This includes understanding not only cost of hardware and software, but also the cost of service management including *staffing cost*. Staffing cost management is of particular interest since among other costs, it is the most flexible, and hence can be an essential point of competitive differentiation among IT service providers.

These more sophisticated IT customer expectations have also surfaced in a growing era of simplified "do it yourself" IT management. Specifically, enterprise customers have access to a growing array of (largely) cloud-based services (e.g., *platform as a service*, *software as a service*, etc.) that lower the barrier to adoption of new technologies, including greater in-house management of such technologies and virtualized infrastructures. Within this new reality, it has become more important for IT service providers to address inquiries regarding innovation and value surrounding their service offerings. Increasingly, the solution has been to offer unique analytical capabilities within their IT service offerings.

Many research activities have surfaced investigating the integration of analytical capabilities into services delivery solutions. Timely examples include the use of analytics to increase services delivery efficiency. Specific examples include extra cost and time avoidance tasks such as automating the detection of conflicting IT change plans [10] and automating the recommendation of best resolution actions for IT monitoring events [19]. Our own recent experiences with services delivery clients have revealed an increasing demand for *predictive* analytics for providing answers to various "what if" scenarios. An example scenario includes analyzing "*what* improvements in my infrastructure performance and costs will I gain *if* I invest in specific additions or enhancements." Understanding answers to such questions *before* contracting related services delivery activities has become more important in the midst of increasing market competition and stagnant IT budgets. Related to previously stated comments, prediction of service delivery efforts and staffing costs has become essential. In services delivery, effort can be measured largely by analyzing properties of tickets submitted to a delivery workforce requesting resolution of some incident (e.g., a slow or unresponsive service). Such tickets drive a large share of a services delivery workforce's activities, or *effort*.

This paper addresses the aforementioned client demands by introducing a framework for predicting services delivery staffing requirements (largely dependent on required effort) given IT infrastructure characteristics and projected transformations. Given financial characteristics about a specific services delivery workforce and client engagement agreements (e.g., costs of staffing and service level agreement penalties), such requirements and efforts can then be used to predict services delivery *costs*. Our contribution is two-fold. One, we present a comprehensive framework for predicting services delivery staffing requirements based on IT infrastructure loads using a combination of different predictive analytical models. Two, and the greater focus of the paper, we conduct a deep analysis of services delivery effort prediction, specifically using support vector regression to model the correlation between IT infrastructure characteristics and incident ticket volumes and properties. We show through experimentation using real-world data that such an approach to effort prediction, as required by the overall solution framework we present, is quite promising. To the best of our knowledge, both the proposed solution framework and study of effort prediction using regression are novel contributions to the literature and IT services delivery practitioner community.

The remainder of the paper is organized as follows. Section II compares our work to prior art. Section III describes the overall solution framework. Section IV describes details of our approach to services delivery effort prediction. Section V provides an evaluation of our approach and Section VI concludes the paper with a statement of future work.

II. Related Works

As will be described in Section III, our overall prediction framework is based on IT performance prediction, service delivery effort prediction, and staffing modeling and optimization components. The study of various subjects in IT performance prediction is quite mature, and much of it can be traced back to fundamental queuing theory concepts [4]. More recent investigations have contributed a keener understanding of what affects performance in increasingly complex and customized IT infrastructures. For instance, [20] describes the use of profit-based performance criteria to model how workload variations and other resource management policies affect application response times in cloud environments. In [21], the distributed key-value storage system of the Spotify® infrastructure is analyzed for performing response time prediction. Insight from such empirical studies can benefit our work in the future. However, such studies do not extend into the prediction of service delivery staffing workloads, which is a central focus of our overall research agenda.

There also exists work in specifically predicting and classifying aspects of IT management tickets. For example, earlier work described in [14] addressed the classification of maintenance request tickets for automated dispatch to the appropriate service delivery personnel. Various techniques (e.g., support vector machines, classification trees, etc.) were used with encouraging results. A different approach was proposed in [7], which proposed a crowd-sourcing based approach to ticket classification. We do not focus on ticket classification, but it can be used in our solution framework since one cannot always expect essential ticket data (e.g., severity and complexity attributes) to be supplied. Similarly beneficial is the work described in [18]. There, the authors analyze the relationship between IT agent monitoring policies and the volume and types of tickets produced, with the aim of reducing the amount of non-actionable tickets resulting from overly-sensitive monitoring policies. Most recently, the authors of [3] studied the correlation between various server properties and ticket volumes to understand how modernizing specific server attributes (e.g., operating systems) can reduce error-prone operation. Our works are similar, though the goals differ, and the authors employ random forest techniques for *binary* classification, whereas we require regression analysis for our framework. To the best of our knowledge, the only (recent) prior work addressing service delivery *cost* prediction is described in [6]. However, the authors focus on IT project management costs as opposed to service delivery staffing costs associated with incidents and service level agreements.

Regarding service delivery workforce modeling, simulation-based approaches are usually preferred which suggest optimal staffing solutions after considering the complexities of the real-world system such as the non-stationarities in the arrival rates and the interactions between decisions made in different periods. [1] considers a multi-period problem of determining optimal staffing levels while meeting service level requirements. The authors solve a sample mean approximation of the problem using a simulation-based analytic center cutting plane method and assuming that the service level functions are pseudo-concave. [9] uses stochastic approximation to determine optimal staffing levels, assuming that the service level functions are convex. [16] considers a two stage approach for determining optimal staffing levels in a call center environment. In the first stage they solve for the staffing levels by using per period attainment as an approximation for the true service level attainment. In the second stage, the simulation is used to evaluate true system performance and service level attainment. [8] uses a simulation-optimization approach to minimize the total staffing related variable cost while considering the contractual service level constraints, the skills required to respond to different types of service requests, and the shift schedules that the service agents must follow. We specifically plan to leverage the contributions described in [8] in our overall solution framework.

III. Services Delivery Effort and Staffing Requirements Prediction Framework

The overall solution framework we propose facilitates the prediction of IT services delivery effort and staffing requirements based on the analysis of IT infrastructure characteristics, current behavior, and predicted behavior. Given our experience, the prediction we focus on can proceed from multiple aspects of IT infrastructure analysis. Here, we explain our view of a comprehensive solution framework and identify its optional and required components. **Error! Reference source not found.** illustrates this framework which consists of the following several components: *server performance prediction*, *services delivery effort (or workload) prediction*, and *service delivery staffing requirements prediction*. As a

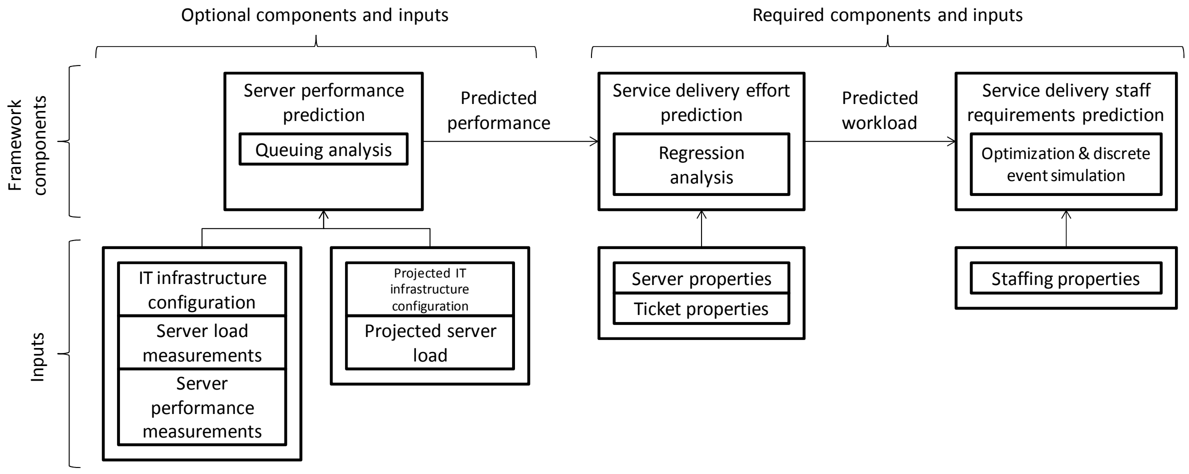
Optional components and inputs
Required components and inputs
Framework components
Inputs
Server performance prediction
Queuing analysis
Predicted performance
Service delivery effort prediction
Regression analysis
Predicted workload
Service delivery staff requirements prediction
Optimization & discrete event simulation
IT infrastructure configuration
Server load measurements
Server performance measurements
Projected IT infrastructure configuration
Projected server load
Server properties
Ticket properties
Staffing properties

Fig. 1. Overall service delivery effort and staffing requirements prediction framework.

note, we will use the terms *effort* and *workload* interchangeably for the remainder of the paper.

The first starting point for overall prediction is the analysis of server behavior and connectivity and understanding how they affect server *performance*. This is because IT system performance has a significant affect on the volume of workload submitted to a services delivery staff. For example, monitoring agents observing high CPU utilization often automatically generate associated incident tickets. The *server performance prediction* component implements techniques for the prediction of IT performance metrics. As shown in **Error! Reference source not found.**, this component's inputs include *IT infrastructure configuration data*, *server load measurements*, and *server performance measurements*. IT infrastructure configuration data describes machines' resources (e.g., CPU speed and memory) and properties of their connectivity (e.g., network architecture and link speeds). *Server load* and *performance measurements* are correlated with each other and may include measurements such as user transaction rates and CPU utilization, respectively. The combination of such inputs naturally attracts the application of queuing network analysis [4] to build a prediction model, especially if the prediction of performance is based on altering a network of dependent machines. This applies in our case since our work is motivated by "what if" scenarios that serve to analyze efforts and costs based on adding additional IT assets and services. However, modeling a specific IT system for accurate performance analysis using queuing models can be overwhelming, hence impeding the large-scale deployment of such a solution. Hence, we propose the use of various queuing parameter estimation techniques, designed specifically for modeling tiered IT systems, which are described in [13] and [18]. The remaining inputs to this component, *projected IT infrastructure configuration* and *projected server load*, are then used as inputs into the model to predict performance (the output of this component) in the "what if" scenario.

We frame server performance prediction as optional mainly because of data availability challenges. In a growing number of strategic outsourcing engagements, multiple vendors may be responsible for different aspects of system and services management, thereby splintering data governance. Therefore, a vendor charged with service delivery effort prediction may not have access to details such as network topologies and transaction activities. We did not have access to such data for this publication; hence, we reserve the detailed integration of IT performance prediction into our framework and analysis for future investigations.

The second starting point for effort prediction (the one on which we focus in this work) is the analysis of the relationship between managed server properties and submitted tickets. The premise here is that problematic IT system behavior is not always solely dependent on load-based metrics, but also on *static* properties of IT systems. For example, the currency of a server's operating system might be a dominant factor in the occurrence of problematic operation, and hence might affect the generation of tickets. The *service delivery workload prediction* component is responsible for such analysis. As shown in **Error! Reference source not found.**, inputs here include *server properties* and *ticket properties*. The former includes typical static properties of a server (e.g., operating system, CPU speed, etc.) as well as properties of applications that the server hosts. Ticket properties, the requirements of which are largely driven by service delivery workforce optimization, may include items such as incident severity and ticket submission time. As will be explained in detail later, we use support vector regression to implement service delivery workload prediction, since the problem can be mapped to one of regression. As shown in Fig. 1, the predicted performance of one or more servers associated with the other inputs for this component can also be used as an input into the (regression) model, providing more insight into the overall prediction. Prediction based solely on server and ticket properties, however, remains quite useful since one can still analyze effort and staffing costs if similar new servers are added to the IT infrastructure.

The third component in this framework, *service delivery staff requirements prediction*, estimates various requirements, namely labor costs, based on predicted service delivery workloads using a simulation-optimization approach [8]. Particularly, a discrete event simulation model is used to model the complex service delivery environment in detail. This includes (1) service requests as characterized by arrival time, customer, work type, tooling, severity, complexity, and service time, (2) service delivery units as characterized by number of agents, shift schedule, skill level, customer familiarity, and tool familiarity, and (3) dispatching engine that specifies the order of which the arrived service request will be processed by the service agent. Based on the simulation model, the staffing optimization method minimizes the staffing cost subject to two types of constraints: *service level constraints* and *staffing coverage constraints*. Service level constraints represent the service attainment as computed by the simulation model described above and the service level objectives that must be satisfied. In a service delivery environment, the service level objectives typically takes on a form such as "no more than 5% of priority 1 incidents reported each month can be resolved in more than 2 calendar hours." Staffing coverage constraints represent the restrictions on the staffing assignment. This includes restrictions on the number of agents within each service delivery unit and constraints on the number of agents that must work in a given shift. Section IV will describe the exact input required from the services delivery effort prediction component. All remaining inputs are constraints of a service delivery organization that do not need to be predicted, but are assumed.

IV. Service Delivery Effort Prediction

As mentioned in Section I, the analytical focus of this paper entails an investigation of the ability to predict services delivery effort, as characterized by various ticket properties. Furthermore, given the type of data that was available for this investigation, we focus on workload prediction irrespective of server performance. This section describes details of our analytical approach. The actual results of our evaluation follow in Section V.

A. Server and Ticket Data

The server and ticket data we used for this investigation was obtained from a services delivery engagement involving at least 11,000 managed servers (known at publication time) from a large financial services company. The server data describes configurations of production *open system* (as opposed to *midrange* and *mainframe*) physical and virtual machines. The server attributes we used (and the ones to which we had access) are listed in **Error! Reference source not found.**The *CPU speed* and *memory size* are both continuous attributes; all others are nominal. The cardinality of the domain of the nominal attributes varies (between approximately 4 and 30 unique values) depending on the time frame of ticket data we use in our analysis. The *application codes* were obtained from tickets associated with a server, i.e., they identify what application is exhibiting the non-ideal behavior described in the ticket. Using this approach to map applications to servers may not produce a *complete* list of a server's applications. However, since we used four continuous months of ticket data, the approach should produce a sufficient list of the *problematic* applications, which is our focus anyway.

TABLE I. Server Properties used for Predictive Modeling

Attribute	Description
Classification	Describes the server's operating system.
Manufacturer	Describes the server's manufacturer.
Architecture	Describes the high-level architecture of the CPU (e.g., Intel®, Sun®, PowerPC®).
CPU type	Describes the specific type/model of the CPU (e.g., Intel® Xeon®, UltraSPARC, Quad-Core AMD® Opteron®).
CPU speed	CPU's speed expressed in hertz (Hz).
Memory size	Memory size expressed in bytes (B).
Application codes	Unique identifiers for business applications hosted on the server.

TABLE II. Ticket Properties used for Predictive Modeling

Attribute	Description
Severity	A rating of the ticket's importance; values include *low*, *normal*, and *critical*.
Complexity	Describes the difficulty of problem described in the ticket; values include *easy*, *normal*, and *hard*.
Creation time	Time and date when the ticket was created.
Completion time	Time and date when the ticket when work on the ticket commenced.

We used ticket data, which was generated in response to server-specific behavior, over a four month period. The ticket properties we used are described in TABLE II. The ticket data set did not originally include the *complexity* attribute. We derived this attribute value using historical data (from past IBM® service delivery engagements) containing correlations between ticket service times and complexities. We calculated service times using the *creation* and *completion times* from the tickets. We note that this may be an imperfect way of calculating actual time worked on a ticket, but more fine-grained data is rarely available and we believe the estimation is sufficient for our studies.

B. Regression Modeling Overview

For analyzing the correlations between server properties and services delivery workloads, we used support vector machine based regression (SVR), which is associated with the class of kernel-based learning techniques [2][5]. Kernel-based learning techniques have several notable advantages. Perhaps the most notable benefit is that they have an ability to generate non-linear decision boundaries using techniques originally designed for linear classifiers. This is advantageous for us since given the combination of nominal and continuous server configuration properties, clear linear relationships with workload, or ticket, data cannot be expected.

We present a simplified explanation of support vector regression here and refer the reader to works such as [5] for an exhaustive explanation. To start, consider a simple two-class classification problem using linear models of the following form:

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + b\,, \tag{1}$$

where $\Phi(\mathbf{x})$ is a fixed feature-space transformation, b is a bias parameter, and $\mathbf{w}$ is a normal vector to a hyper-plane (or decision boundary) dividing data points of the different classes. Support vector machines attempt to find the *maximum margin* around the boundary separating points of different classes. The maximum margin solution is found by solving the following:

$$\underset{\mathbf{w},b}{\arg\max}\left\{\frac{1}{\|\mathbf{w}\|}\min_{n}\left[t_n\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)+b\right)\right]\right\}, \tag{2}$$

where t_n is a data point of either class. Largely through the use of Lagrange multipliers, the following dual representation, also to be maximized, of (1) can be formulated as follows:

$$\widetilde{L}(\mathbf{a})=\sum_{n=1}^{N}a_n-\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}a_n a_m t_n t_m k(\mathbf{x}_n,\mathbf{x}_m), \tag{3}$$

subject to constraints,

$$a_n \geq 0, \quad n=1,\ldots,N\,, \tag{4}$$

$$\sum_{n=1}^{N}a_n t_n = 0\,, \tag{5}$$

where a_n is a Lagrange multiplier and $\mathbf{a}$=(a_1,…,a_N)$^{\mathrm{T}}$. For the linear case, the kernel function is defined by $k(\mathbf{x},\mathbf{x}')$= $\Phi(\mathbf{x})^{\mathrm{T}}\Phi(\mathbf{x}')$. The use of kernels in the dual representation of the problem is what enables the classifier to be applied efficiently to features spaces with dimensionality exceeding the number of data points.

As with any regression problem, we must consider minimization of an error function. Similar to the previous explanation of support vector classification, SVR employs the use of Lagrange multipliers to formulate regression as maximizing the dual formulation of an error function:

$$\begin{aligned}\widetilde{L}(\mathbf{a},\hat{\mathbf{a}})=&-\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n-\hat{a}_n)(a_m-\hat{a}_m)k(\mathbf{x}_n,\mathbf{x}_m)\\&-\varepsilon\sum_{n=1}^{N}(a_n+\hat{a}_n)+\sum_{n=1}^{N}(a_n-\hat{a}_n)t_n\end{aligned}\ , \tag{6}$$

subject to constraints,

$$0 \leq a_n \leq C, \tag{7}$$

$$0 \leq \hat{a}_n \leq C, \tag{8}$$

where ε controls sensitivity to error (i.e., zero error is given if absolute difference between target and predicted variables is less than ε and $\varepsilon > 0$) and C is the soft margin constant. Using operations on the Lagrangian, the regression model can then be represented as follows:

$$y(\mathbf{x})=\sum_{n=1}^{N}(a_n-\hat{a}_n)k(\mathbf{x},\mathbf{x}_n)+b\,. \tag{9}$$

Further explanation of how we perform regression modeling given our data set is presented in Section V.

TABLE III. SUPPORT VECTOR REGRESSION TRAINING PARAMETERS

Parameter		Value
Kernel type		Radial basis function (RBF), Gausian
SMO parameters	ε (from (6))	0.001
	Tolerance used for checking stopping criterion	0.001
Grid search SVR parameter optimization parameters	C (from (7) and (8)) search set	20, 40, 60, 80, 100, 120
	γ (from (10)) search set	0.1, 0.3, 0.5
	Optimization criterion	(1-abs(correlation_coefficient)) + root_relative_squared_error + root_absolute_error
Testing method		10-fold cross-validation

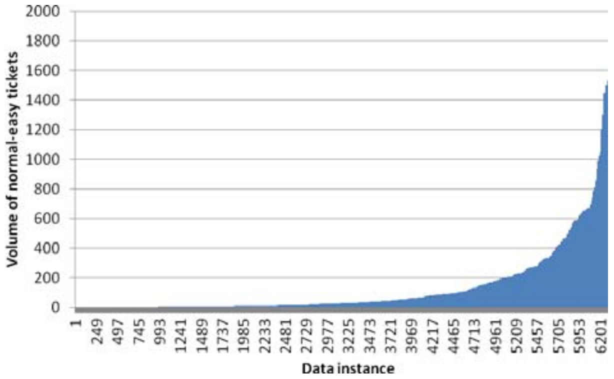

Fig. 2. Skewed distribution of number of *normal-easy* tickets among all data points.

V. EVALUATION

A. Approach

The goal of our evaluation was to asses if, under certain conditions, service delivery workload is correlated with server properties given real world information. If so, it means that prediction of service delivery workload is in fact feasible, supporting the usefulness of the framework presented in Section III.

Our evaluation first required some data pre-processing operations in order to perform more efficient and meaningful regression modeling. Specifically, we cleaned the server attribute data to reduce the occurrence of largely redundant feature values. For example, CPU types of servers reported as “Intel® Xeon®” or “Intel® Xeon® 2.33GHz” were simply replaced by “Intel® Xeon®,” especially since CPU speed is a

TABLE IV. REGRESSION MODEL TRAINING RESULTS FOR TICKET VOLUMES (4 MONTHS) BASED ON TICKET SEVERITY AND COMPLEXITY

Ticket severity-complexity combination	Training set size	SVM soft margin constant (C)	RBF kernel γ	Correlation coefficient	Mean absolute error (MAE)	Root mean squared error (RMSE)	Relative absolute error (RAE)	Root relative squared error (RRSE)
Low-easy	531	80	0.5	0.937	0.357	1.155	15.703%	36.240%
Normal-easy (64 < vol. <= 200)	1080	80	0.5	0.820	9.912	2.921	29.112%	58.28%
Normal-easy (vol. > 200)	1394	100	0.5	0.685	153.587	357.489	41.201%	74.471%
Critical-easy	735	120	0.5	0.709	74.325	224.587	40.529%	70.894%

TABLE V. REGRESSION MODEL TRAINING RESULTS FOR MEAN SERVICE TIMES (4 MONTHS) BASED ON TICKET SEVERITY AND COMPLEXITY

Ticket severity-complexity combination	Training set size	SVM soft margin constant (C)	RBF kernel γ	Correlation coefficient	Mean absolute error (MAE)	Root mean squared error (RMSE)	Relative absolute error (RAE)	Root relative squared error (RRSE)
Low-easy	531	60	0.5	0.837	0.821	2.861	25.016%	56.945%
Normal-easy (0 < s.t. < = 0.52)	1454	40	0.5	0.720	0.057	0.104	45.098%	70.494%
Normal-easy (0.52 < s.t. <= 1.35)	1629	20	0.5	0.662	0.111	0.190	52.144%	76.524%
Normal-easy (1.35 < s.t. <= 3.80)	1781	60	0.5	0.784	0.250	0.458	40.700%	63.346%
Normal-easy (s.t. > 3.80)	1584	80	0.5	0.706	0.892	1.912	42.775%	73.566%
Critical-easy	735	40	0.5	0.603	1.867	4.408	43.850%	87.6233%

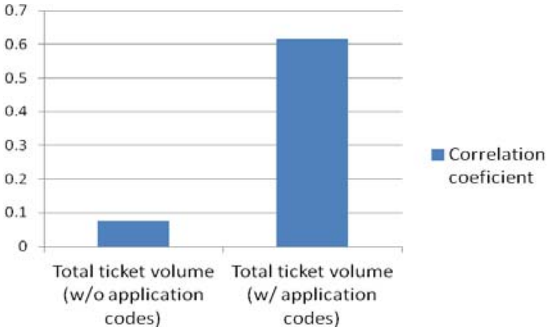

Fig. 3. SVR training performance comparison for dependent variable *total ticket volume* for January (C=20.0, γ=0.5).

separate attribute. Such variations can be attributed to the fact that different operating systems uniquely report their hardware specifications to IBM® TADDM. For some training exercises, we also pruned data instances in order to avoid affects of severe data skew as well as speed up the training process. For instance, as illustrated in Fig. 2, in the data set for tickets with *normal-easy* severity-complexity combinations, nearly 4000 of the 6677 total data points (nearly two-thirds) map to no more than 64 tickets each. Since the remainder of the data exhibits less dominance around any given volume range and also maps to greater ticket volumes (which is of greater interest for predicting workloads upon service delivery staff), we simply pruned the data points in question. We also pruned data points associated with servers that produced an anomalous number of tickets. Specifically, we omitted any server with a total volume of tickets greater than 2σ+*med*, where σ and *med* are the standard deviation and median values, respectfully of total ticket volume value for data set.

We used the WEKA data mining software toolkit to facilitate our evaluations [11]. As previously mentioned, we used SVR for our modeling. We chose the radial basis function (RBF), a Gaussian non-linear kernel for our SVM, which assumes the following definition:

$$k(\mathbf{x}_n, \mathbf{x}_m) = \exp\left(-\gamma\|\mathbf{x}_n - \mathbf{x}_m\|^2\right), \quad \text{for } \gamma > 0. \qquad (10)$$

After preliminary trials, we found the RBF kernel to yield better results than that of a linear (or polynomial) kernel. This is not too surprising given the combination of fairly high dimensionality and nominal features in our data set.

Further training details are as follows. The sequential minimal optimization (SMO) method was used to train the SVM for regression purposes [15]. The parameters used in the SMO-based training are listed in TABLE III. Also, we used a grid-based search technique for optimizing parameter values for the SVM and RBF kernel per evaluation experiment. The

TABLE VI. REGRESSION MODEL TRAINING RESULTS FOR MONTHLY TICKET VOLUMES FOR CRITICAL-EASY TICKETS

Month	Training set size	SVM soft margin constant (C)	RBF kernel γ	Correlation coefficient	Mean absolute error (MAE)	Root mean squared error (RMSE)	Relative absolute error (RAE)	Root relative squared error (RRSE)
January	264	100	0.5	0.811	19.638	59.935	25.815%	59.283%
February	322	100	0.5	0.794	14.062	59.172	18.403%	62.313%
March	288	100	0.5	0.761	21.756	70.589	29.170%	67.049%
April	175	20	0.5	0.560	1.949	5.619	43.471%	88.750%

parameter value ranges used in the grid search are listed in TABLE III. Last, training results were evaluated using 10-fold cross validation.

For SVR training, we use the server properties in TABLE I as the independent variables. The dependent variables reflect the "non-assumed" requirements for service delivery workforce optimization. Hence, the ticket attributes in TABLE II are used to define the following dependent variables:

- Ticket volume for severity *i* and complexity *j*,
- Mean ticket service time (days) for severity *i* and complexity *j*.

Initially, we investigated service delivery effort prediction without considering application code data. This scenario did not yield favorable results, as illustrated in Fig. 3, which compares correlation coefficients for a case of SVR training with and without the application code attribute. Here, the inclusion of the application code yields a nearly 6-fold increase in the correlation coefficient. This justifies the inclusion of application code as an attribute, at least for our particular data set. After considering numerous unsuccessful options for incorporating application codes into the data set (which lead to huge increases in dimensionality), we decided on the following simple approach. For each application code associated with a given server, we copied the server's data point and appended the application code.

B. Results

The results of SVR training are shown in TABLE IV and TABLE V. Results are shown for various combinations of ticket severity and complexity, as required by staffing optimization analysis. Unfortunately, for our study, our data set did not contain significant numbers of tickets of *normal* or *hard* complexity. The data set size encountered while performing SVR training for the *normal-easy* combination was relatively large, and dramatically elongated the running time of the training process. Therefore, individual regression models were formulated over separate regions of the data set. For example, rows 3 and 4 of TABLE IV describe results for data points mapping to ticket volumes greater than 64 and up to 200 and greater than 200 tickets, respectively. Considering both the correlation coefficient as well as the error values, the majority of the results show that there are medium to strong relationships between server configurations (and hosted applications) and various ticket properties required for optimizing service delivery staffing requirements. This is especially true in predicting ticket service times (see TABLE V), where correlation coefficients ranged between 0.603 and 0.837, and all error-related statistics were kept at reasonable values. Furthermore, there was no apparent correlation between training quality and ticket severity, which is good since, given this data, prediction quality can be expected to remain satisfactory over other data.

The prediction quality for ticket volumes is not as consistently strong. Both the prediction quality for the normal-easy (for servers with greater than 200 tickets) and critical-easy scenarios exhibit substantial error rates. In investigating the potential reasons for such performance, we inspected the regression performance for *individual months* over the same data. For example, TABLE VI shows the training results for critical-easy tickets for four individual months. Compared to regression results for the same type of tickets over an aggregated four month period, these results for separate months are much better on average, with mostly better correlation coefficients and markedly better error statistics. For example, root mean squared error decreased by an average of 175.75 while the correlation coefficient increased by an average of 0.023. This suggests that trends in critical ticket generation are based on underlying monthly (or perhaps other cyclic) activities in an IT environment. A likely hypothesis is that such activities are related to server workloads (e.g., transaction rates and volumes) and maintenance tasks. In the case of the data we analyzed, from a financial services company, a significant share of IT workloads is related to cyclic financial market trends. Regarding maintenance activities, our experience has been that re-occurring "change freeze" periods (during which updates to servers and other components are banned due to auditing and other management tasks) often occur during the beginning of the calendar year. Again, this can affect periodic ticket generation rates (especially of the critical sort) and might account for the improved prediction quality over a narrower time windows, as shown in TABLE VI.

There are a couple of insights we gain from this evaluation. First, we observe that reasonable to strong prediction of service delivery effort using server and ticket properties is quite possible, especially if application data is available. Second, and what is particularly beneficial for service delivery workforce optimization, is that some of the prediction is valid over multi-month time periods (referring to TABLE IV and TABLE V). This is good since it gives an IT services provider sufficient time to reconfigure a service delivery workforce after predicting service delivery workloads in the form of tickets.

VI. Conclusion

This paper described an overall framework for predicting services delivery workforce requirements for efficiently resolving IT incidents given projected changes in system loads and infrastructure changes. The major analytical focus entailed modeling the correlation between infrastructure characteristics and ticket properties required for workforce staffing optimization. Given data from a real services delivery engagement and the use of support vector regression modeling, we showed that developing such accurate correlation models is quite feasible under certain circumstances (e.g., dependent on the temporal extent of prediction) and highlights the potential usefulness of the overall prediction framework.

Fully developing and evaluating the overall solution framework will require some future investigation. For instance, further experiments must be conducted using IT performance data (e.g., CPU utilization or application response times) which we did not have access to for this study. Identifying accurate correlation models using such data will enable us to develop a more detailed integration of queuing analysis (described in Section III) into our framework, further helping us understand how infrastructure changes affect delivery effort. Next, evaluation should be done using more data, especially related to different time periods of the year and even different services delivery engagements. In general, predictive modeling always benefits from more training data. However, given our services delivery focus, data from different parts of the year, and hence associated with varying workloads and potential IT transformations, will be especially helping in assessing the possible limits of services delivery effort prediction. Also, sophisticated data pruning techniques should be evaluated to potentially improve the regression models. For example, when considering performance measurements as independent feature variables in prediction analysis, one should ensure that the (dependent) ticket data is associated with performance-related incidents and requests. As previously mentioned, we expect that previous work in ticket classification such as that described in [7] can be leveraged for this task. Previous work in identifying non-actionable tickets may be potentially used in a similar manner [18]. In moving beyond the framework presented in this paper, we are also investigating the prediction of IT-centric activities based on environmental activities that are *external* to the confines of an IT infrastructure. This may be beneficial for proactively managing IT systems supporting more pervasive and mobile computing applications such as those dealing with city automation or social media.

Acknowledgment

This investigation benefitted from the insight of many of the authors' talented and generous colleagues. Hence, the authors would like to thank the following for their assistance: Li Zhang, Aubrey Rembert, Niyu Ge, Ioana Giurgiu, Amit Paradkar, Emi Olsson, James Albrecht, and Nathan Smileye.

References

[1] J. Atlason, M. A. Epelman, and S. G. Henderson, "Optimizing call center staffing using simulation and analytic center cutting-plane methods," *Managment Science*, vol. 54, pp. 295–309, 2008.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*, M. Jordan, J. Kleinberg, and B. Schölkoph, Eds. New York: Springer, 2006.

[3] J. Bogojeska, D. Lanyi, I. Giurgiu, G. Stark, and D. Wiesmann, "Classifying server behavior and predicting impact of moderinzation actions," to appear in *IFIP/IEEE CNSM*, 2013.

[4] H. Chen and D. Yao, *Fundamentals of Queuing Networks: Performance, Asymptotics, and Optimization*, I. Karatzas and M. Yor, Eds. New York: Springer-Verlag, 2010.

[5] N. Cristianini and J. S.-T., *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge: Cambridge, 2000.

[6] B. L. Dalmazo, W. L. Cordeiro, L. Rabelo, J. A. Wickboldt, R. C. Lunardi, R. L. dos Santos, L. P. Gaspary, L. Z. Granville, C. Bartolini, and M. Hickey, "Leveraging IT project lifecycle data to predict support costs," in *Proc. of IFIP/IEEE IM*, pp. 249-256, 2011.

[7] Y. Diao, H. Jamjoom, and D. Loewenstern, "Rule-based classification in IT service management," in *Proc. of IEEE Cloud*, pp. 221-228, 2009.

[8] Y. Diao and A. Heching, "Staffing optimization in complex service delivery systems," In *Proceedings of International Conference on Network and Service Management*, 2011.

[9] Z. Feldman and A. Mandelbaum, "Using simulation based stochastic approximation to optimize staffing of systems with skills based routing," in *Proceedings of the 2010 Winter Simulation Conference*, pp. 3307–3317, 2010.

[10] S. Hagen and A. Kemper, "Towards solid IT change management: auomated detection of conflicting IT change plans," in *Proc. of IFIP/IEEE IM*, pp. 265-272, 2011.

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.-H. Witten, "The WEKA data mining software: an update," in *SIGKDD Explorations*, vol. 11, no. 1, 2009.

[12] L. Kleinrock, *Queueing Systems Volume II: Computer Applications*, John Wiley and Sons, 1976.

[13] Z. Liu, L. Wynter, C. Xia, and F. Zhang, "Parameter inference of queuing models for IT systems using end-to-end measurements," in *Performance Evaluation*, vol. 63, pp. 36-60, January 2006.

[14] G. A. D. Lucca, M. D. Penta, and S. Gradar, "An approach to classify software maintenance requests," in *Proc. of IEEE ICSM*, 2002.

[15] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods – Support Vector Learning*, B. Schoelkopf, C. Burges, and A. Smol,aEds. MIT Press, 1998.

[16] T. R. Robbins and T. P. Harrison, "A simulation based scheduling model for call centers with uncertain arrival rates," in *Proceedings of the 2008 Winter Simulation Conference*, pp. 2884–2890, 2008.

[17] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to the SMO algorithm for SVM regression," in *IEEE Transactions on Neural Networks*, vol. 11, no. 5, pp. 1188-1193, September 2000.

[18] L. Tang, T. Li, F. Pinel, L. Shwartz, and G. Grabarnik, "Optimizing system monitoring configurations for non-actionable alerts", in *Proc. of IFIP NOMS*, pp. 34-42, 2012.

[19] L. Tang, T. Li, L. Shwartz, and G. Grbarnik, "Recommending resolutions for problems identified by monitoring," in *Proc. of IFIP/IEEE IM*, pp. 134-142, 2013.

[20] Q. Wang, Y. Kanemasa, J. Li, D. Jayasinghe, M. Kawaba, and C. Pu, "Response time reliability in cloud environments: an empirical study of n-tier applications at high resource utilization," in *Proc. of IEEE SRDS*, pp. 378-383, 2012.

[21] R. Yanggratoke, G. Kreitz, M. Goldmann, and R. Stadler, "Predicting response times for the Spotify backend," in *Proc. of IFIP/IEEE CNSM*, pp. 117-125, 2012.

[22] L. Zhang, C. Xia, M. S. Squillante, W. N. Mills, "Workload service requirements analysis: a queuing network optimization approach," in *Proc. of the IEEE Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 23-32, 2002.