RC25585 (WAT1601-018) January 7, 2016
Mathematics

IBM Research Report

A Statistical Framework of Demand Forecasting for Resource-Pool-Based Software Development Services

Ta-Hsin Li
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598 USA

A Statistical Framework of Demand Forecasting for Resource-Pool-Based Software Development Services

Ta-Hsin Li*

December 15, 2015

*IBM T. J. Watson Research Center, Yorktown Heights, NY 10598-0218, USA (thl@us.ibm.com)

Abstract

To adapt to the fast-changing landscape of technology and the increasing complexity of skills needed as a result, an outcome-based delivery model, called crowdsourcing, emerges in recent years for software development. In this model, some of the work required by a large project is broken down into self-contained short-cycle components, and a resource pool of vetted freelancers is leveraged to perform the tasks. The resource pool must be managed carefully by the service provider to ensure the availability of the right skills at the right time when they are needed. This article proposes a statistical framework of demand forecasting to support the capacity planning and management of resource pool services. The proposed method utilizes the predictive information contained in the system that facilitates the resource pool operation through survival models, and combine the results with special complementary time series models to produce demand forecasts in multiple categories at multiple time horizons. A dataset from a real-world resource pool service operation for software development is used to motivate and evaluate the proposed method.

Key Words and Phrases: autoregressive, capacity planning, crowdsourcing, hierarchical, pipeline, predictive modeling, survival analysis, time series of counts

Abbreviated Title: Demand Forecasting for Resource-Pool-Based Software Development Services

Acknowledgment: The author thanks David Hoffman, Blain Dillard, and Yi-Min Chee for helpful discussions.

Version History: Version 1, December 5, 2014. Version 2, December 15, 2015

1 Introduction

With the rapid advance of computer technology, software development for business applications becomes increasingly complex. A single application often requires multiple technologies involving languages, tools, databases, frameworks, and platforms. Technological requirements also vary across different applications. Such complexity makes it very difficult for large enterprises to maintain an adequate and efficient workforce with the right skills at the right time when they are needed. The ever-changing landscape of technology only exacerbates the problem.

To mitigate the difficulty, many software development projects begin to adopt an outcome-based delivery model, called crowdsourcing, where a flexible workforce, or resource pool, of vetted freelancing professionals is leveraged to supplement the regular workforce and deliver some well-defined short-cycle work items or components (Howe 2008; Vuković 2009; Peng, Ali Babar, and Ebert 2014). A service provider enables the end-to-end process of request and delivery with an electronic platform (e.g., web portal), which we call the event management system. Typical steps of the request and delivery process include (a) creation of the requested work item, or event, by the client, also known as the event manager, with technological requirements and scheduled start date, (b) launch of the event by the service provider on the scheduled start date to allow the resource pool members to register with proposals, (c) evaluation of submitted proposals by the client to select a winner of the event (multiple winners are also allowed in some practice), (d) submission of the finished work by the winner, and (e) review of the finished work by the client with the options of accept, revise, or reject.

The service provider also manages the resource pool to ensure the availability of the right skills at the right time. When the resource pool cannot provide the skills needed by the anticipated demand, new members with the right skills must be recruited in order to prevent the work from going unstaffed. Over-recruiting relative to the demand should be avoided, because it not only increases the administrative cost but also reduces the effectiveness and commitment of the resource pool members when the work is not plentiful enough. Accurate demand forecasting plays a critical role in making resource management decisions.

Demand of different skills can fluctuate dramatically over time in different ways. Figure 1 shows the weekly time series of demand in two categories from a real-world resource pool operation which will be discussed shortly. A conventional method of forecasting demand series such as these is to employ time series models. Among the most popular ones are the autoregressive integrated moving-average (ARIMA) models and the exponential smoothing (ES) models (Box, Jenkins, and Reinsel 2008). The forecasts shown in Figure 1 with diamonds ($\diamond$) are produced by the ARIMA/ES method using SPSS Expert Modeler, a commercial-grade software package equipped with the desired capability of automatic data-driven model selection — not only within the ARIMA family and the ES family respectively, but between them as well (http://www-03.ibm.com/software/products/en/spss-forecasting).

In this article, we develop an alternative framework of demand forecasting. The basic idea is to leverage the predictive information in a pipeline and combine the results with special complementary time series models to produce the final forecast.

The pipeline in this context refers to the service provider's event management system that facilitates the creation and launch of work requests. The system mandates that every work item, before launched for registration, goes through a sequence of preparation stages in an irreversible order of maturity as more and more required information is provided by the client. The required information includes technological requirements which dictate the skills needed for the work item. It also includes the so-called scheduled start date which informs the system when to launch the work item for registration. Endowed with such information, the work items contained in the pipeline at the time of forecasting, which we call the *planned events*, can be readily projected as future demand. In fact, these projections are often reported by the system as demand outlook. We refer to this method of demand forecasting as the baseline method. It will be used to normalize other methods when comparing their accuracies. In Figure 1, the forecasts by the baseline method are shown with plus signs ($+$).

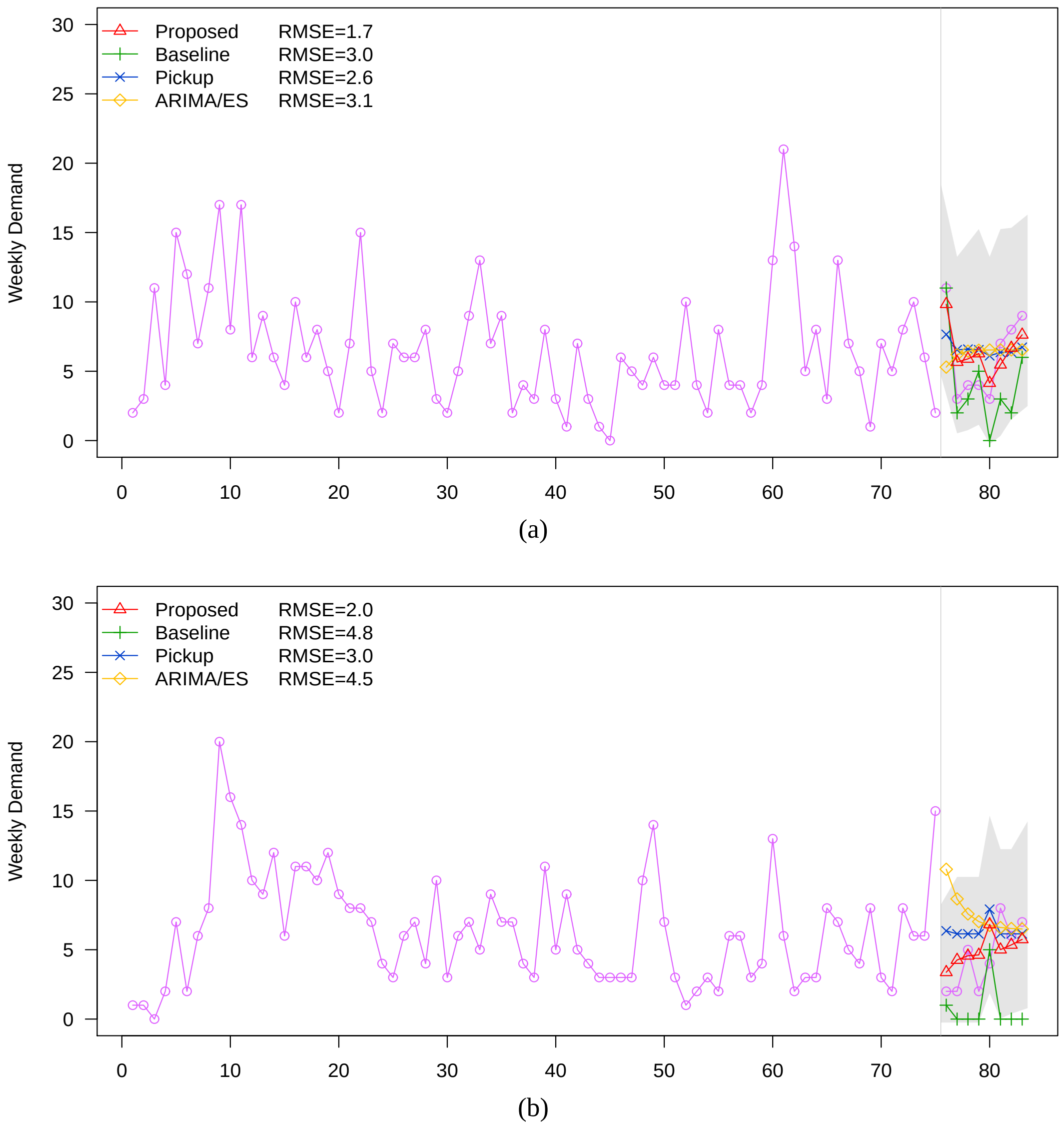


Figure 1: Weekly demand series and eight-week forecast for (a) Java and (b) Lotus Notes and Domino. ○, actual; △, forecast by the proposed method; +, forecast by the baseline method; ×, forecast by the pickup method; ◇, forecast by the ARIMA/ES method. Shaded area shows the 90% interval forecast by the proposed method. The vertical line separates the past from the future.

In reservation-based service industries such as airlines and hotels, the so-called pickup method has been successfully used for demand forecasting (L'Heureux 1986; Weatherford and Kimes 2003). In this method, future demand is predicted by suitably scaling and shifting the total bookings on hand using historical averages or regression techniques. The baseline forecast in our application is the counterpart of bookings on hand and therefore can be used as input to a pickup model for demand forecasting. This method produces the results shown in Figure 1 with crosses ($\times$).

The planned events in the pipeline are subject to revisions. The scheduled start date, in particular, can be changed for various reasons and therefore is not a perfect indicator of the actual launch time. The discrepancy between the scheduled start date and the actual launch time can be very large, ranging from days to weeks, especially for work items in early stages of preparation when the information entered into the system tends to be no more than a temporary place holder. To rely on the scheduled start date indiscriminately could yield erroneous demand forecasts. Improving such forecasts by employing an advanced statistical method to determine the actual launch time of planned events more reliably at any stage in the pipeline constitutes the first component of the proposed framework.

The pipeline at the time of forecasting does not contain any future work items that may contribute to the demand at a forecasting horizon. For example, the work items created in the first week after the time of forecasting may contribute to the demand in the second week. We call these work items the *unplanned events*. The unplanned events can take a bigger share in the total demand as the forecasting horizon increases. It is especially the case when the system has no requirement on the minimum lead time — the time elapsed between the creation and the launch of a work item — as many work items may be created on short notice. Predicting the contribution of unplanned events in the total demand at each forecasting horizon to complement the demand predicted from the planned events constitutes the second component of the proposed framework.

It is assumed in this article that demand forecasting takes place on a weekly basis and work items are classified into predefined demand categories based on their technological requirements. The objective of demand forecasting is to predict the number of work items, or events, that will be launched from each demand category in each of the consecutive coming weeks (up to an upper bound). The time series shown in Figure 1 represent such weekly demand from two categories and the corresponding eight-week forecasts at the end of week 75.

Under the proposed framework, statistical survival functions for the lifetime of events in the pipeline are employed to determine the actual launch time of planned events in forecasting their contribution to the future demand. To complement the pipeline forecast, special time series models, including those for time series of counts, are employed to predict the contribution of unplanned events to the total demand. The results of this method are shown in Figure 1 with triangles ($\triangle$).

Survival functions have been used as a forecasting tool in many applications. For example, Read (1997) employs survival functions to forecast the attritions of U.S. Army personnel. Malm, Ljunggren, Bergstedt, Pettersson, and Morrison (2012) use survival functions to forecast replacement needs for drinking water networks. Canals-Cerdá and Kerr (2014) discuss the use of survival functions to forecast credit card portfolio losses. The present application is served by a nonparametric hierarchical survival model that predicts the remaining lifetimes of planned events in the pipeline. The hierarchy is constructed by taking advantage of the personal behavior of event managers in scheduling the start date while overcoming the difficulty of data disparity.

Time series models have been considered in Lee (1990) for predicting the daily bookings of a passenger flight leading to the departure day. We employ time series models to predict the demand from unplanned events to complement the demand from planned events predicted by the survival analysis method. In addition to the ordinary (Gaussian) autoregressive models, we investigate the performance of linear and log-linear Poisson autoregressive models which are specifically designed for time series of counts (Zeger and Qaqish 1988; Fokianos and Tjøstheim 2011; Christou and Fokianos 2015). We also consider horizon-specific autoregressive models which are tailored to the special horizon-dependent structure of the demand from unplanned events in relation with the demand from planned events.

To develop and maintain a demand forecasting tool for operational use, the service provider has some choices to make on the spectrum of cost and complexity. The ARIMA/ES method and the pickup method are viable choices because they only require the collection and storage of weekly demand series and the straightforward application of existing software. To compete with them, the alternative method not only needs to achieve higher accuracy but also has to keep its complexity within a suitable level.

The dataset that we use in this article to motivate and validate the proposed method is provided by a multinational corporation that offers resource pool services for software development. It is a random sample of the events they managed in two years. It is derived from a database of weekly snapshots of the pipeline, thus representing a discretized evolutionary history of each event over time. Captured in the dataset are two stages of event preparation, called *scheduled* and *scheduled-ready*, with the latter representing greater maturity, which are automatically designated to an event based on the information provided by the event manager. There is also a preliminary draft stage which is ignored in our analysis because events in this stage lack sufficient and reliable information for demand forecasting. Among the attributes associated with an event in the dataset are the stage of preparation and the scheduled start date and the required technologies. These attributes are dynamic and subject to change during the course of evolution leading to the launch of the event. Other useful attributes are the event identification number and the name of the event manager, which do not change over time. Examples of required technologies are *InfoSphere DataStage*, *Java*, *Lotus Notes and Domino*, and *SAP*. A single event may require one or more technologies. By means of a predefined mapping, the required technologies determine the demand category of the event for the purposes of demand forecasting. Figure 1 shows the weekly demand series in categories (a) Java and (b) Lotus Notes and Domino. For reason of confidentiality, we will not divulge the complete list of technologies or the real scale of demand in this article.

It is worth mentioning that information regarding the underlining project of each work item is not available in the dataset. This is not uncommon in resource pool services, especially for those run by third-party providers, because such information is often considered internal and confidential by the client. More details about the dataset will be given later during the analysis.

The remainder of this article is organized as follows. In Section 2, we discuss the prediction of demand from planned events. In Section 3, we discuss the prediction of demand from unplanned events. In Section 4, we discuss the total demand forecast. Concluding remarks and discussions are given in Section 5.

2 Demand from Planned Events

At the time of forecasting, planned events are work items in the pipeline with adequate information including the required technologies and the scheduled start date. Equipped with these attributes, it is a simple matter of counting to determine how many planned events in a given demand category will be launched in a coming week. For example, the total number of Java events with the scheduled start date falling in the 2nd week from the time of forecasting constitutes the demand for Java skills in that week. We call this method of demand forecasting the *baseline method*. So called because it is often implemented in the system to provide a quick outlook of future demand, as is the case with the supplier of the dataset for the present study.

2.1 Survival Analysis Method

The baseline method has some serious shortcomings. For one, it relies solely on the scheduled start date to determine when a planned event will be launched. Unfortunately, the scheduled start date, entered into the system by the event manager during the preparation stages, is not entirely reliable as an indicator of the actual launch time, because the event could be rescheduled at any time (with no penalty) until it is actually launched.

As an example, Figure 2 shows the histogram of the discrepancy between the scheduled start date and

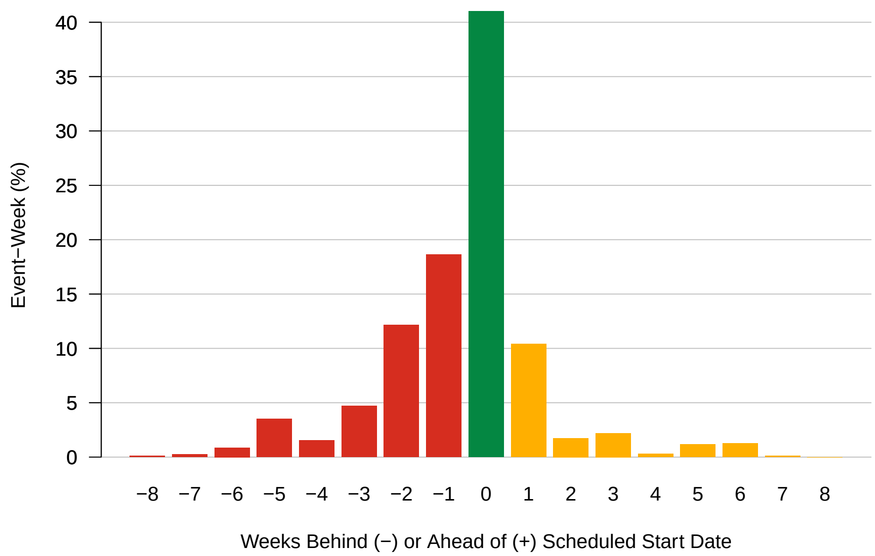


Figure 2: Histogram of discrepancy between scheduled start date and actual launch time. Negative value means actual launch time is behind scheduled start date; positive value means actual launch time is ahead of scheduled start date.

the actual launch time of 272 events created by an event manager. In this case, the scheduled start date incorrectly predicts the actual launch time for nearly 60% of the event weeks (an event week is defined as an event spending one week in the pipeline). The discrepancy can be very large: some events are postponed by as much as 8 weeks and some others are advanced by as much as 6 weeks. Due to the prevalence of such errors, relying on the scheduled start date for demand forecasting often produces poor results.

An event stays in the pipeline for a certain amount of time until it is launched. The actual launch time of a planned event can be formulated as the remaining lifetime of the event in the pipeline. Therefore, it is quite natural to model the entire lifetime of an event in the pipeline as a random variable to account for its uncertainty, and employ survival functions to describe the actual launch time in probabilistic terms.

To be more specific, let W denote the total number of weeks, measured in accordance with the weekly snapshot schedules, that an event spends in the pipeline before it is launched. Regarding W as an integer-valued nonnegative random variable, the survival function of W is defined by

$$S(\tau) := \Pr(W > \tau) \qquad (0 \leq \tau < \infty).$$

For a planned event that has been in the pipeline for a weeks at the time of forecasting, the probability that the event will be launched in the coming week h is given by

$$p(h|a) := \Pr(W = a+h-1 | W \geq a) = \frac{S(a+h-2) - S(a+h-1)}{S(a-1)} \qquad (h = 1, 2, \ldots). \tag{1}$$

The function $p(\cdot|a)$ is nothing but the probability mass function of an integer-valued positive random variable that represents the week in which the event will be launched. It can be regarded as a probabilistic prediction of the launch time over the coming weeks. It is interesting to observe that

$$p(h|a) = \{1 - \lambda(a)\}\{1 - \lambda(a+1)\} \cdots \{1 - \lambda(a+h-2)\}\lambda(a+h-1),$$

where $\lambda(a) := \{S(a-1) - S(a)\}/S(a-1) = p(1|a)$ is called the hazard rate at the age of a.

Suppose the pipeline contains n planned events in week t, which is the time of forecasting. Let c_i denote the demand category of event i ($i = 1, \ldots, n$), $S_i(\cdot)$ and a_i denote the survival function and the age of event i, and $p_i(\cdot|a_i)$ denote the corresponding prediction function for the launch time of event i. Then, the number of planned events that will be launched in week $t+h$ as demand in category c, denoted by $D(t+h|t,c)$, can be modeled as

$$D(t+h|t,c) = \sum_{i=1}^{n} B_i(h,c), \tag{2}$$

where $B_1(h,c), \ldots, B_n(h,c)$ are Bernoulli random variables with success probability given by

$$\phi_i(h,c) := p_i(h|a_i) I(c_i = c). \tag{3}$$

The expected value of $D(t+h|t,c)$ in (2) takes the form

$$\pi(t+h|t,c) := E\{D(t+h|t,c)\} = \sum_{i=1}^{n} \phi_i(h,c). \tag{4}$$

This constitutes the best point forecast for the number of planned events in category c to be launched in week $t+h$ based on the minimum mean-square error (MMSE) criterion.

If the events are launched independently, then the variance of $D(t+h|t,c)$ can be expressed as

$$\sigma_\pi^2(t+h|t,c) := V\{D(t+h|t,c)\} = \sum_{i=1}^{n} \phi_i(h,c)\{1-\phi_i(h,c)\}. \tag{5}$$

The independence assumption also implies that $D(t+h|t,c)$ has a Poisson binomial distribution from which an interval forecast can be easily derived. Incorporating dependencies in the model is expected to improve the accuracy of interval forecast and variance calculation, although the point forecast will not be affected.

Finally, we employ an allocation method to accommodate possible misclassification of events in (4). Let $r_i(c,c')$ denote the probability of event i being misclassified in category c' instead of c. Then, the modified success probability of $B_i(h,c)$ is given by

$$\phi_i(h,c) := p_i(h|a_i)r_i(c,c_i). \tag{6}$$

In practice, the allocation parameters $r_i(c,c')$ can be derived from historical data at suitably aggregated levels such as the preparation stage and the age of the event.

The survival analysis method relies solely on the observed lifetimes of events to determine the launch time. This does not mean that the scheduled start date, which feeds the baseline method, should be completely ignored. In general, the scheduled start date is noisy, but the amount of noise can vary among demand categories, for example. In the cases where the scheduled start date has little noise, the baseline forecast can be more accurate than the survival-based forecast. The opposite is generally true where the scheduled start date has too much noise. The question is how to combine these forecasts appropriately to produce better results. Toward that end, we develop two hybrid methods by incorporating the scheduled start date at different levels: the event level and the demand category level.

The first hybrid method identifies the so-called trusted event managers by the criterion that the events they manage are launched mostly according to the scheduled start date. For each event of a trusted event manager, the survival-based prediction of launch time is replaced by the scheduled start date of the event.

The second hybrid method combines the baseline forecast, denoted by $\pi_B(t+h|t,c)$, with the survival-based forecast or the forecast by the first hybrid method, both denoted by $\pi_S(t+h|t,c)$ for simplicity. The resulting forecast takes the form

$$\pi_R(t+h|t,c) := \alpha(h,c)\pi_S(t+h|t,c) + \beta(h,c)\pi_B(t+h|t,c), \tag{7}$$

where the coefficients $\alpha(h,c)$ and $\beta(h,c)$ are derived from historical data by least-squares regression. Note that the coefficients are allowed to vary with h in order to accommodate the situation where the benefit of the baseline forecast depends on the forecasting horizon.

The hybrid method in (7) is interpretable as a generalization of a Bayesian estimator. Indeed, the baseline forecast $\pi_B(t+h|t,c)$ can be expressed in the form of (4) with $\phi_i(h,c)$ replaced by $B_{i0}(h,c)$, which equals 1 if event i is in category c and the scheduled start date falls in week h, and equals 0 otherwise. Therefore, the regression model (7) can be rewritten as

$$\pi_R(t+h|t,c) = \sum_{i=1}^{n} \{\alpha(h,c)\phi_i(h,c) + \beta(h,c)B_{i0}(h,c)\}.$$

If the coefficients are constrained to take nonnegative values and sum up to 1, then the term $\alpha(h,c)\phi_i(h,c) + \beta(h,c)B_{i0}(h,c)$ can be interpreted as the Bayesian MMSE estimator (i.e., the posterior mean) of the Bernoulli random variable $B_i(h,c)$ given the binary observation $B_{i0}(h,c)$, with the survival-based forecast $\phi_i(h,c)$ serving as the mean of a Beta prior.

The allocation method in (6) can also be improved by incorporating the concept of trusted event managers. In this case, the trusted event managers are identified by sufficiently high rates of no change to the demand category for the events they manage. Exemption from allocation is granted to all events managed by trusted event managers. This can be done by setting $r_i(c,c') = 0$ in (6) for $c' \neq c$ if event i of category c' qualifies for the exemption.

2.2 Survival Function Modeling

To implement the survival analysis method in practice, one needs to designate a survival function for each event in the pipeline. Survival functions can be derived from historical data by statistical techniques. The simplest model is to apply a grand survival function to all events. However, a stratified approach based on a suitable segmentation of the event population is more appropriate to accommodate the expected heterogeneous characteristics of the events.

Because the event managers are responsible for creating and scheduling events in the pipeline, their individual behavior, dictated by personal preferences and the dynamics of underlying projects, should have a direct impact on the lifetime of the events they manage. It is desirable that the event population be segmented by event manager so that a personalized survival model can be developed. In practice, this idea runs immediately into the obstacle of data disparity: segmentation by event manager inevitably leads to a small number of segments with sufficient data points and a large number of segments with few data points. Training an exclusive survival function for each segment is likely to result in unreliable models with poor predictive capability due to their excessive statistical uncertainty.

As a practical starting point, we present a hierarchical approach to tackle the problem (see Section 5 for comments on alternative methods). The approach yields a class-based model where some personalized

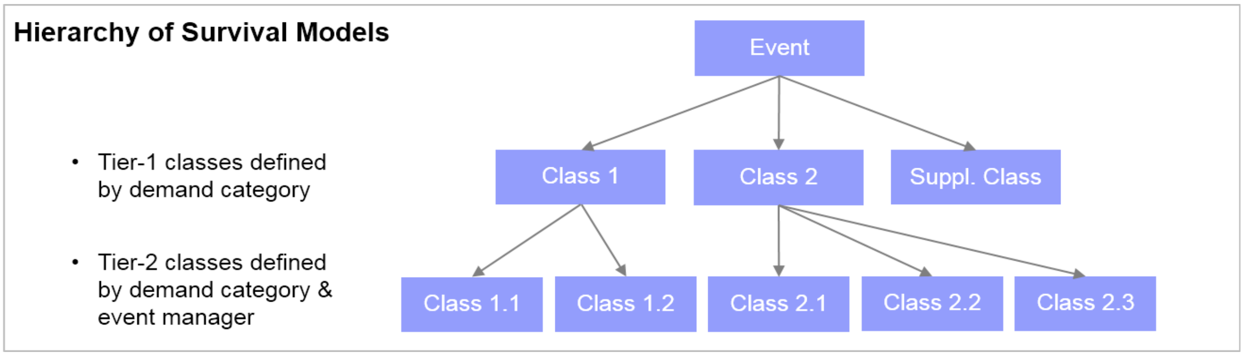


Figure 3: An example of hierarchical survival models with 2 tier-1 classes and 5 tier-2 classes plus a supplementary class for the remaining event segments.

segments retain their own survival functions whereas the others share common survival functions in the hierarchy.

Specifically, we consider a hierarchy with two tiers of event classes as illustrated by Figure 3. Tier-1 classes are defined by selected demand categories. This is a natural choice because demand forecasting is carried out at the granularity of demand categories. Tier-2 classes are defined by selected event managers under each selected demand category, representing a personalized refinement of the tier-1 parent classes.

The hierarchy of event classes enables a hierarchical approach to designation of survival functions for planned events in demand forecasting: First, determine whether or not a given event belongs to a tier-2 class; if yes, use the survival function associated with the tier-2 class; otherwise, determine whether or not the event belongs to a tier-1 class; if yes, use the survival function associated with the tier-1 class; if both tests fail, use the survival function associated with the supplementary class. In other words, the survival function designated to a planned event for demand forecasting is one associated with the class in the deepest tier to which the event belongs.

Searching for the best classes over all possible subsets of event segments is prohibitively time-consuming. To circumvent the difficulty, we consider a sequential (suboptimal) selection procedure, such that the candidate segments are admitted into a given tier one after another, in descending order of sample size, until the out-of-sample predictive power of the resulting survival model is maximized. For tier-1 classes, the procedure is applied to the segments defined by demand category, and the unselected segments form a single supplementary class. For tier-2 classes, the procedure is applied to the segments defined by event manager under each demand category admitted into tier 1.

An alternative procedure is what we call forward selection. It is analogous to the forward selection procedure of stepwise regression, except that the criterion for enter is based on the out-of-sample predictive power rather than a significance test. In each iteration, the forward selection procedure picks the best segment from the pool of remaining candidates until the out-of-sample predictive power is maximized. It is more effective than sequential selection, but the computational complexity is considerably higher, as the number of trials increases quadratically rather than linearly with the number of candidates.

The out-of-sample predictive power is an appropriate criterion for survival modeling in the present application because demand forecasting is its ultimate objective. Specifically, we determine the out-of-sample predictive power by K-fold cross validation: The entire dataset is partitioned randomly into K equal-sized subsets. A survival model is trained in turn on the $K-1$ out of K subsets and tested on the remaining one using a prediction error metric. This generates K out-of-sample error measurements whose average serves as the modeling criterion.

A useful prediction error metric is what we call the Q-score, which measures the relative improvement by a method of interest over the baseline method in root mean-square error (RMSE) for forecasting the demand from planned events. Specifically, the Q-score is defined as

$$Q := 1 - \frac{\text{RMSE(method of interest)}}{\text{RMSE(baseline method)}}. \tag{8}$$

This metric can be calculated using the RMSE for a given demand category at a given forecasting horizon to obtain a category-and-horizon-specific Q-score. It can also be calculated using the average of the RMSE's across demand categories and forecasting horizons to produce an overall Q-score. Maximizing the overall

Q-score obtained from cross validation is the criterion of the sequential and forward selection procedures.

Finally, for greater versatility, we take the preparation stage into account in survival modeling. Owing to the irreversible order of maturity, events in a later stage tends to have less uncertainty regarding the lifetime than events in an earlier stage. A stage-dependent survival model is expected to provide more accurate demand forecast. The survival analysis method discussed above can be applied independently to each stage. The resulting stage-dependent survival functions describe the stage-specific lifetime—the time elapsed after an event enters the said stage until it is launched. When forecasting the demand, these functions are used to determine the launch time of planned events in a stage conditional on the time spent in that stage.

Given the hierarchy of event classes, the corresponding survival functions can be derived from historical data using any number of statistical techniques (Lawless 2002; Kalbfleisch and Prentice 2002). A key requirement is the ability to accommodate right-censored events—the events that are still under preparation at the time of final snapshot. An important example of such techniques is the product-limit estimator, also known as the Kaplan-Meier estimator (Kaplan and Meier 1958). Being fully nonparametric, the Kaplan-Meier estimator has the utmost flexibility for fitting a wide variety of observed survival patterns without strong assumptions. This property is particularly desirable for large-scale modeling exercises such as ours where fine-tuning of each and every survival function with elaborate models is prohibitive.

Let $\tau_1 < \cdots < \tau_m$ be the distinct lifetimes of historical events in the class of interest. For $j = 1, \ldots, m$, let r_j denote the number of events in the said class with lifetime or censoring time greater than or equal to τ_j, and let d_j denote the number events in the said class with lifetime equal to τ_j. Then, the Kaplan-Meier estimator of the survival function $S(\tau)$, with τ being a continuous variable, can be expressed as

$$\hat{S}(\tau) := \prod_{j:\tau_j \leq \tau} (1 - d_j/r_j) \qquad (0 \leq \tau < \infty). \tag{9}$$

This formula defines a right-continuous and monotone-decreasing step function, starting with $\hat{S}(0) = 1$. For the weekly snapshot data, a drop in this function can only take place at positive integers $\tau = 1, 2, \ldots$, and the magnitude of the drop at τ represents the probability that an event spends τ weeks in the pipeline before it is launched in the following week.

The Kaplan-Meier estimator is not without limitations. For example, it tends to exhibit higher statistical variability in the right-hand tail where observations are sparse; it is also unable to extrapolate the survival function beyond the largest observation (except for setting it to zero). These shortcomings can have a negative impact on demand forecasting especially for events in advanced age. To mitigate the problem, one can model the lifetimes jointly across event classes under the semi-parametric Cox proportional hazards (PH) framework (Cox and Oakes 1984).

For example, consider a joint model of all survival functions in the two-tiered hierarchy. Let I_u be the event membership indicator for tier-1 class u and let I_{uv} be that for tier-2 class v under tier-1 class u. Then, the Cox proportional hazards model can be expressed as

$$\log\{S(\tau)\} = \log\{S_0(\tau)\} \exp\left\{ \sum_{u=1}^{k} \alpha_u I_u + \sum_{u=1}^{k} \sum_{v=1}^{k_u} \beta_{uv} I_{uv} \right\}, \tag{10}$$

where $S_0(\cdot)$ is a nonparametric baseline survival function, α_u and β_{uv} are the parameters which adjust the baseline survival function for different event classes, k is the number of tier-1 classes (minus 1 if the supplementary class is absent), and k_u is the number of tier-2 classes under tier-1 class u (minus 1 if it equals the number of segments under tier-1 class u).

Leveraging the data across all event classes allows the Cox estimator to overcome the aforementioned shortcomings of the Kaplan-Meier estimator. However, the assumption of proportional hazards imposes a limitation on its flexibility. Less restrictive models can be developed by considering each tier separately, resulting in a model of the form (10) without the β_{uv} terms for tier 1 and a similar model without the α_u terms for tier 2.

2.3 Case Study

The dataset discussed in Section 1 contains the weekly evolution history of 6,747 events in 126 demand categories. There are 6,278 events with records in stage 1 (scheduled) and 2,301 events with records in stage 2 (scheduled-ready). The fact that not all events have records in both stages is due to the limitation of the weekly sampling resolution. It is more pronounced for stage 2 because the sojourn times tend to be shorter in that stage. In survival modeling, the RMSE is calculated on the basis of weekly forecasts over 99 consecutive weeks for the number of planned events in each demand category that are launched in the next 1 through 8 weeks.

In this study, the allocation method defined by (6) is always applied to the survival-based forecasts. The trusted event managers are identified, separately for each stage, as one who manages at least 5 events and makes no revisions to the demand category. Their events are exempted from allocation. The remaining events are used to derive the fractions of event misclassification in each stage that serve as stage-dependent allocation parameters.

A 10-fold cross-validation method is used to calculate the out-of-sample Q-score as the criterion for construction of the two-tiered hierarchy. Tier-1 classes are obtained first, by using a one-tiered forecasting approach without requiring tier-2 classes. This step is carried out separately for each preparation stage. After tier-1 classes are determined, tier-2 classes are selected by using the two-tiered forecasting approach with fixed tier-1 classes.

As candidates for tier-1 classes, the event segments defined by demand category exhibit a great disparity in size. For stage-1 events, for example, the size ranges from 1 to 492 with a median of 25, indicating a very skewed distribution. The sequential selection procedure yields a survival model with 15 classes for stage-1 events and 8 classes for stage-2 events.

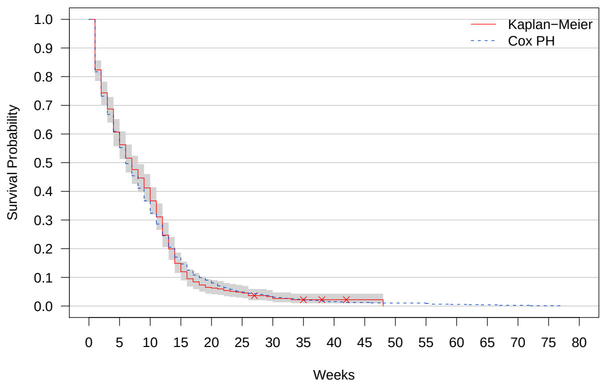
Figure 4 shows the survival functions by the Kaplan-Meier estimator in (9), obtained using the R function `survfit`, for two of the resulting tier-1 classes of stage-1 events. Figure 4 also shows the corresponding probability distributions of launch time, given by (1), for events of age 1, i.e., $p(h|1)$ $(h = 1, 2, \ldots)$. As can be seen, these event classes have distinct characteristics in launch time. For example, the launch time of Java events is most likely to fall into week 1 with roughly equal chances over the remaining weeks, whereas the launch time of SAP events is more likely to occur around week 5 or 6 with little chance of staying beyond week 10.

For comparison, Figure 4 also depicts the survival functions by the Cox proportional hazards estimator of the form (10) without the tier-2 terms, which are obtained using the R function `coxph`. This result demonstrates the Cox estimator's ability to extrapolate beyond the largest observation, which is desirable especially for the SAP events. However, the Cox estimator does not fit the observed lifetimes of SAP events very well, due to the restriction of proportional hazards.

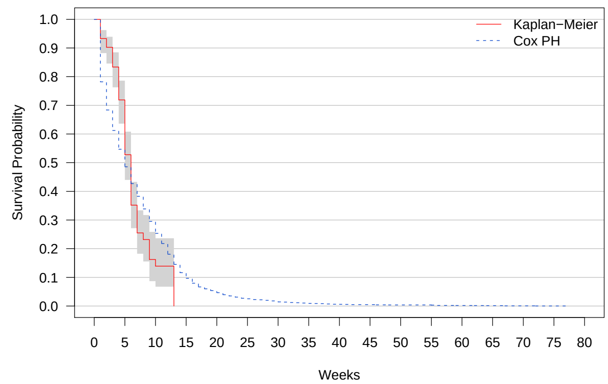
The candidates for tier-2 classes are even more unbalanced in size. Among the 1,279 segments of stage-1 events, the size ranges from 1 to 199 with a median equal to 2. The sequential selection procedure produces 92 tier-2 classes for stage-1 events and 6 for stage-2 events. The tier-2 classes constitute a personalized refinement of their parent classes in tier 1. For example, the Java class in tier 1 is further refined by 11 tier-2 classes, and the SAP class by 2, all corresponding to different event managers. Figure 5 depicts the survival functions of four tier-2 classes for Java events together with the survival function of the parent class. The admission of these tier-2 classes in the hierarchy is justified by their unique survival patterns.

To evaluate the predictive power of the survival analysis method, Table 1 contains the out-of-sample Q-score (in percentage) for three survival models: one which employs a grand survival function for all events (the no-tier model), one which employs tier-1 classes only (the one-tier model), and one which makes full use of the two-tiered hierarchy (the two-tier model). All survival functions are produced by the Kaplan-Meier estimator. The results show that the out-of-sample performance improves with the complexity of the survival model, and the personalized two-tier model achieves the highest accuracy.

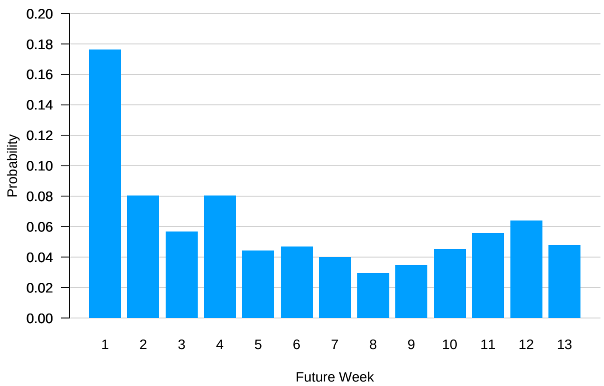
It is worth pointing out the Cox proportional hazards method produces inferior results. For example,



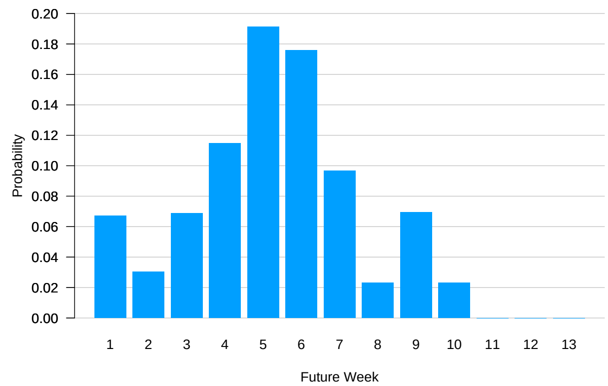
(a)



(b)



(c)



(d)

Figure 4: (a)(b) Survival functions for two tier-1 classes, by the Kaplan-Meier estimator (solid line) and the Cox proportional hazards estimator (dashed line), and (c)(d) the corresponding probabilities of launch time over the next thirteen weeks for events of age 1, by the Kaplan-Meier estimator. (a)(c) Java events. (b)(d) SAP events. Shaded area in (a)(b) shows the 95% confidence band of the Kaplan-Meier estimates.

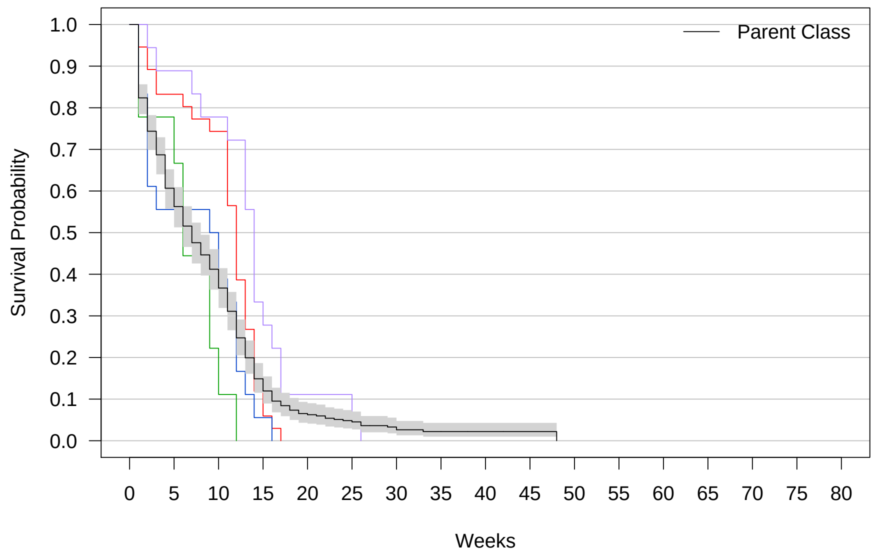


Figure 5: Survival functions of four tier-2 classes and their parent class for Java events.

Table 1: Performance for Forecasting Demand from Planned Events

Event Class Selection Method	Survival Analysis Models No-Tier	One-Tier	Two-Tier	Hybrid Models Trusted Mgr.	Regression	Both
Sequential Selection	15.42	16.00	16.60	19.80	22.84	23.82
Forward Selection	15.42	16.16	17.42	20.59	23.47	24.38

Note: Results are based on 10-fold cross validation.

applying the Cox estimator independently to each tier and preparation stage yields an out-of-sample Q-score of 15.59% as compared to 16.60% achieved by the Kaplan-Meier estimator. This suggests that the potential benefit of the Cox estimator is not sufficiently fulfilled to compensate for its shortcomings. It remains to be seen if the outcome can be improved by further stratification of each tier to maximize the benefit of the proportional hazards constraint.

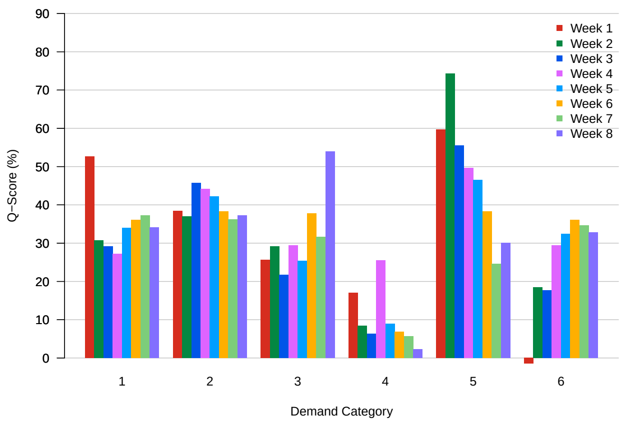
To evaluate the performance of the hybrid methods that incorporate the scheduled start date with the survival-based forecasts, Table 1 contains the out-of-sample Q-score for three variations of the methods: one which replies only on the trusted event managers, one which employs the regression model, and one which uses both methods. The trusted event managers are those with a history of at least 5 event-weeks in the training data and 80% or higher on-time rate for launching their events. All variations are built upon the forecasts of the two-tiered survival model. The results show that the regression method is more effective between the two, but applying both methods together achieves the highest accuracy.

For comparison, an hierarchy of event classes is also obtained using the forward selection procedure. It comprises 17 tier-1 classes for stage-1 events and 5 for stage-2 events; it also comprises 51 tier-2 classes for stage-1 events and 1 tier-2 class for stage-2 events. Table 1 shows the resulting Q-scores. As compared with the sequential selection procedure, the forward selection procedure offers higher accuracy with a smaller number of classes in the hierarchy, an indication of greater efficiency. However, this is achieved at a much higher computational cost (e.g., days instead hours). The sequential selection procedure suffers slightly in accuracy, but it requires much less time to compute and therefore qualifies as a practical choice.

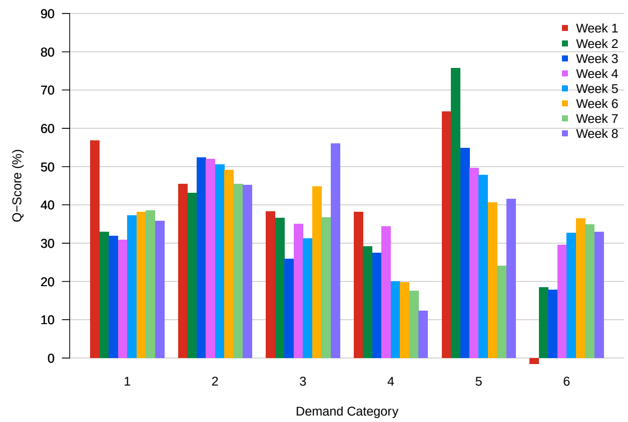
Finally, let us examine the effect of the survival analysis method more closely at the granularity of demand categories and forecasting horizons. Figure 6 depicts the Q-scores of four models at eight forecasting horizons for six demand categories. The results are based on in-sample RMSE's with the survival functions and the hybrid models trained on the entire dataset.

As can be seen from Figure 6(a), the one-tiered survival model is able to improve the baseline method in all but one cases, the only exception being week 1 of category 6 (Cognos). The Q-score varies from -1% to 74%, depending on the demand category and the forecasting horizon. For some demand categories, such as category 5 (SAP), the forecast in shorter terms tends to benefit more from the survival analysis method than the forecast in longer terms; the opposite is true for some other categories, such as category 6 (Cognos). Figure 6(b) shows the corresponding Q-scores of the two-tiered survival model that employs personalized event classes. Better results are obtained in 45 out of the 48 cases. The most noticeable ones include category 2 (Infosphere DataStage) and category 4 (SQL) at all horizons.

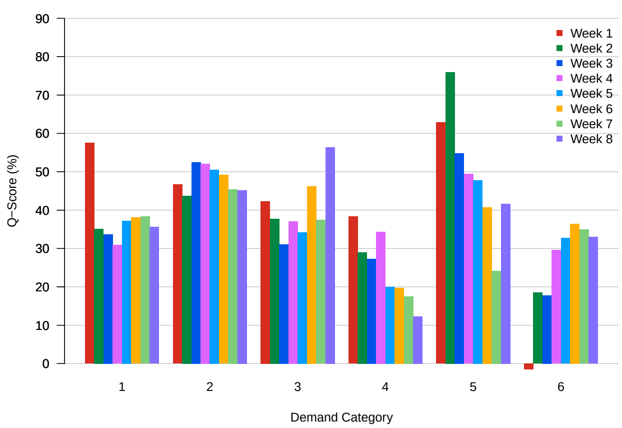
Figure 6(c) depicts the Q-scores of the hybrid method which uses the trusted event managers to revise the forecasts of the two-tiered survival model. In 30 out of the 48 cases, the hybrid model produces better results than the original two-tiered survival model. An example is week 1 of category 3 (Lotus Notes and Domino). Finally, Figure 6(d) shows the Q-scores of the hybrid method that employs both trusted event



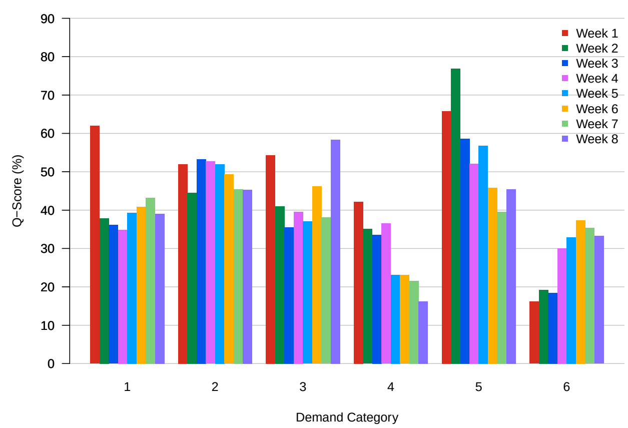
(a)



(b)



(c)



(d)

Figure 6: The Q-scores of four models for forecasting demand from planned events in six categories at eight horizons. (a) One-tiered survival model. (b) Two-tiered survival model. (c) Hybrid model using trusted event managers. (d) Hybrid model using trusted event managers and regression. Demand categories are: (1) Java, (2) Infosphere DataStage, (3) Lotus Notes and Domino, (4) SQL, (5) SAP, and (6) Cognos.

managers and regression. This method produces the best results, with the Q-scores ranging from 16% to 77%. It is also the only model that performs better than the baseline method for week 1 of category 6 (Cognos).

Table 2 shows the detailed regression models for two demand categories: Java and Infosphere DataStage. The models are presented in a reparameterized form rather than the original form (7) in order to perform the T-test for the statistical significance of the additional contributions from the baseline forecast and the interactions with the forecasting horizon. The last column contains the p-values of the T-test under the i.i.d. Gaussian assumption.

As the main effect, the significance of the survival-based forecast is evident in both models. The additional contribution from the baseline forecast is quite significant in the Java model but much less so in the Infosphere DataStage model. The forecasting horizon plays an important role in the Infosphere DataStage model through the interaction with the survival-based forecast. This result lends support to the horizon-dependent modeling approach in (7).

3 Demand from Unplanned Events

Unplanned events are the future events that arrive at the pipeline after the time of forecasting and before the targeted forecasting horizon. As shown in Figure 7, the unplanned events can contribute significantly to the total demand, and their share increases with the forecasting horizon.

3.1 Time Series Method

By dropping the demand category in notation for simplicity, let $D(t+h|t)$ denote the demand in week $t+h$ which is generated from the planned events at the end of week t. Then, the total demand in week $t+h$,

Table 2. Regression Models for Two Demand Categories

	Java			
Parameter	Estimate	Std. Error	T-Value	Pr($>$ $\|T\|$)
Survival Forecast	0.633	0.102	6.224	0.000
Survival Forecast:Horizon2	-0.021	0.172	-0.125	0.901
Survival Forecast:Horizon3	0.003	0.183	0.014	0.989
Survival Forecast:Horizon4	-0.025	0.172	-0.143	0.886
Survival Forecast:Horizon5	-0.001	0.186	-0.004	0.997
Survival Forecast:Horizon6	-0.075	0.198	-0.377	0.706
Survival Forecast:Horizon7	-0.126	0.191	-0.659	0.510
Survival Forecast:Horizon8	-0.090	0.212	-0.425	0.671
Baseline Forecast:Horizon1	0.102	0.057	1.789	0.074
Baseline Forecast:Horizon2	0.241	0.074	3.267	0.001
Baseline Forecast:Horizon3	0.229	0.081	2.816	0.005
Baseline Forecast:Horizon4	0.261	0.073	3.579	0.000
Baseline Forecast:Horizon5	0.196	0.078	2.511	0.012
Baseline Forecast:Horizon6	0.221	0.080	2.755	0.006
Baseline Forecast:Horizon7	0.258	0.074	3.499	0.000
Baseline Forecast:Horizon8	0.242	0.086	2.810	0.005
	Infosphere DataStage			
Parameter	Estimate	Std. Error	T-Value	Pr($>$ $\|T\|$)
Survival Forecast	0.629	0.078	8.037	0.000
Survival Forecast:Horizon2	0.220	0.112	1.958	0.051
Survival Forecast:Horizon3	0.399	0.112	3.551	0.000
Survival Forecast:Horizon4	0.443	0.111	4.004	0.000
Survival Forecast:Horizon5	0.482	0.114	4.217	0.000
Survival Forecast:Horizon6	0.344	0.120	2.857	0.004
Survival Forecast:Horizon7	0.351	0.128	2.741	0.006
Survival Forecast:Horizon8	0.337	0.132	2.557	0.011
Baseline Forecast:Horizon1	0.190	0.051	3.735	0.000
Baseline Forecast:Horizon2	0.076	0.056	1.359	0.175
Baseline Forecast:Horizon3	-0.089	0.055	-1.605	0.109
Baseline Forecast:Horizon4	-0.095	0.057	-1.674	0.095
Baseline Forecast:Horizon5	-0.145	0.062	-2.328	0.020
Baseline Forecast:Horizon6	-0.028	0.072	-0.389	0.697
Baseline Forecast:Horizon7	0.001	0.080	0.015	0.988
Baseline Forecast:Horizon8	-0.026	0.085	-0.307	0.759

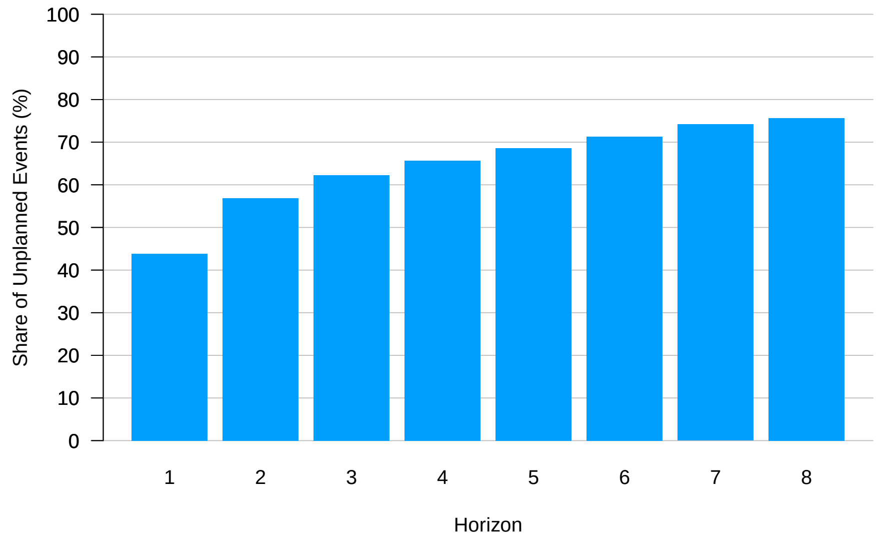


Figure 7: Average proportion of unplanned events in weekly total demand at different forecasting horizons for the dataset discussed in Section 1.

denoted by $\gamma(t+h)$, can be written as

$$\gamma(t+h) = D(t+h|t) + \delta_0(t+h) + \delta_1(t+h) + \cdots + \delta_{h-1}(t+h). \tag{11}$$

In this expression, the $\delta_k(t+h)$ $(k = 0, 1, \ldots, h-1)$ denote the number of events that are created in the future week $t+h-k$ and launched in week $t+h$ at age k. We call these terms the demand from unplanned events. Note that the events launched in week $t+h$ at age h or older are accounted for in $D(t+h|t)$ as the demand from planned events.

To predict the demand from unplanned events, one can work directly with the time series

$$\Delta_h(t) := \sum_{k=0}^{h-1} \delta_k(t) \qquad (t = h, h+1, \ldots), \tag{12}$$

which represents the weekly totals of events launched at an age less than h. Let $\Delta_h(t+h|t)$ denote the h-step ahead forecast of $\Delta_h(t+h)$ at time t by a suitable time series model based on the observed values $\Delta_h(t), \Delta_h(t-1), \ldots, \Delta_h(h)$. Let $\pi(t+h|t)$ denote the forecast of $D(t+h|t)$ by the survival analysis method.

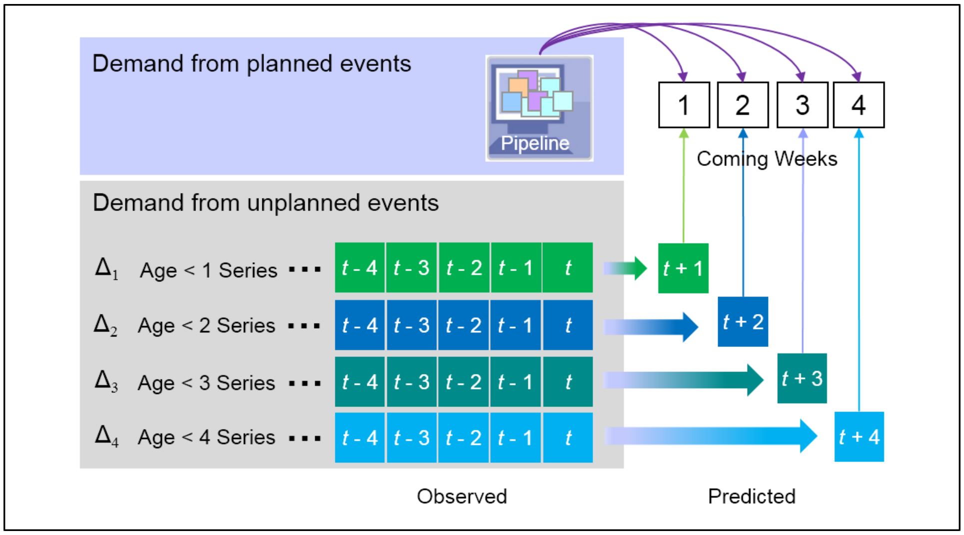


Figure 8: Composition of total demand and its prediction by planned and unplanned events.

Then, by virtue of (11) and (12), the total demand $\gamma(t+h)$ can be predicted by

$$\gamma(t+h|t) := \pi(t+h|t) + \Delta_h(t+h|t). \tag{13}$$

The composition of the total demand and its prediction is illustrated by Figure 8.

As shown in Figure 8, the forecasting of total demand at multiple horizons entails a unique requirement for the prediction of multiple time series: In order to obtain the forecasts of total demand at $h = 1, \ldots, h_0$ with some predetermined maximum horizon $h_0 > 1$, one is required to predict each of the time series $\{\Delta_1(t)\}, \ldots, \{\Delta_{h_0}(t)\}$ at a single but different horizon. Specifically, an h-step ahead prediction is needed for series $\{\Delta_h(t)\}$ $(h = 1, \ldots, h_0)$, rather than multiple predictions at all horizons for all series.

3.2 Time Series Modeling

Linear prediction is a widely-used technique for time series forecasting (Box, Jenkins, and Reinsel 2008). A simple method is autoregression, where the future value of a time series is predicted by a linear combination of past observations. More specifically, the k-step ahead prediction of $\Delta_h(t+k)$ at time t is given recursively by

$$\Delta_h(t+k|t) = a_0 + \sum_{i=1}^{p} a_i \Delta_h(t+k-i|t) \qquad (k = 1, 2 \ldots, h), \tag{14}$$

where $\Delta_h(t+k-i|t) := \Delta_h(t+k-i)$ for $i \geq k$. The MMSE optimality of the predictor in (14) requires the assumption of an autoregressive (AR) model of the form

$$\Delta_h(t+1) = a_0 + \sum_{i=1}^{p} a_i \Delta_h(t+1-i) + \varepsilon(t), \tag{15}$$

where $\{\varepsilon(t)\}$ is Gaussian white noise with mean zero and some variance σ^2. The coefficients $a_0, a_1, \ldots, a_p$ can be derived from historical data by several methods, including the Yule-Walker method and the least-squares method. The order p can be determined by data-driven criteria such as Akaike's information criterion (AIC) or the Bayesian information criterion (BIC).

In essence, the autoregressive model in (15) assumes that the conditional distribution of the random variable $\Delta_h(t+1)$, given historical information up to time t, which we denote by $\mathcal{F}_t$, is independent Gaussian with mean $a_0 + \sum_{i=1}^{p} a_i \Delta_h(t+1-i)$ and variance σ^2. Because the time series in our problem consists of nonnegative integer-valued counts, the Gaussian assumption is not satisfied. In particular, the symmetrical property of the Gaussian distribution about the mean is an ill fit to the time series of counts with positively skewed distributions. As a consequence, the predictor (14) may yield negative values that need to be replaced by zero in order to conform with the nonnegative property of the data. To overcome this deficiency, we consider some alternative methods which are developed by substituting the Gaussian distribution with an integer-valued probability model (e.g., Zeger and Qaqish 1988; Christou and Fokianos 2015).

The most popular example is the Poisson model where it is assumed that $\Delta_h(t+1)$, conditional on $\mathcal{F}_t$, has an independent Poisson distribution with certain mean $\lambda(\mathcal{F}_t)$, i.e.,

$$\Pr\{\Delta_h(t+1) = d|\mathcal{F}_t\} = \frac{\lambda(\mathcal{F}_t)^d}{d!}\exp\{-\lambda(\mathcal{F}_t)\} \qquad (d = 0, 1, \ldots).$$

As in the Gaussian case, the mean $\lambda(\mathcal{F}_t)$ can be modeled as a linear combination of past observations of the time series such that

$$\lambda(\mathcal{F}_t) := a_0 + \sum_{i=1}^{p} a_i \Delta_h(t+1-i). \tag{16}$$

The coefficients can be derived from historical data under the framework of generalized linear models (GLM). However, unlike the Gaussian case, the mean $\lambda(\mathcal{F}_t)$ must be positive in the Poisson model. It is easy to see that the positivity constraint requires the coefficients to satisfy $a_0 > 0$ and $a_i \geq 0$ for $i = 1, \ldots, p$. Furthermore, it is desirable that the model should lead to a stable time series with constant mean $\mu > 0$. By virtue of (16), μ must be the solution to the equation $\mu = a_0 + \sum_{i=1}^{p} a_i\mu$, so the stability constraint requires $\sum_{i=1}^{p} a_i < 1$. This condition also ensures the second-order stationarity of the time series (Ferland, Latour, and Oraichi 2006). Because of these constraints, the computational complexity of training the Poisson model is much higher than the Gaussian model.

An alternative way of specifying the mean $\lambda(\mathcal{F}_t)$ is through a log transform such that

$$\log\{\lambda(\mathcal{F}_t)\} := a_0 + \sum_{i=1}^{p} a_i \log(\Delta_k(t+1-i)+1). \tag{17}$$

Unlike the linear model in (16) where the past observations have an additive effect on the mean, the log-linear model in (17) assumes an multiplicative effect,

$$\lambda(\mathcal{F}_t) = \exp(a_0)\prod_{i=1}^{p}(\Delta_k(t+1-i)+1)^{a_i}.$$

An advantage of the latter is that the coefficients can be derived under the GLM framework without the need of additional positivity constraint. However, a certain stability constraint is still desirable, although the exact form of such constraint remains unknown except for the special case of $p = 1$ which requires $|a_1| < 1$ (Fokianos and Tjøstheim 2011).

For these time series models, the one-step ahead prediction $\Delta_h(t+1|t) := E\{\Delta_h(t+1)|\mathcal{F}_t\}$ can be easily obtained because it is equal to $\lambda(\mathcal{F}_t)$ by definition. Under the linear model (16), the recursive algorithm in (14) remains valid for computing the k-step ahead prediction $\Delta_h(t+k|t) := E\{\Delta_h(t+k)|\mathcal{F}_t\}$ with $k \geq 2$. For example, by the law of iterated expectations, we have

$$\begin{aligned}
\Delta_h(t+2|t) &:= E\{\Delta_h(t+2)|\mathcal{F}_t\} \\
&= E\{E[\Delta_h(t+2)|\mathcal{F}_{t+1}]|\mathcal{F}_t\} \\
&= E\{\lambda(\mathcal{F}_{t+1})|\mathcal{F}_t\} \\
&= E\left\{a_0 + \sum_{i=1}^{p} a_i \Delta_h(t+2-i) \middle| \mathcal{F}_t\right\} \\
&= a_0 + \sum_{i=1}^{p} a_i \Delta_h(t+2-i|t).
\end{aligned}$$

In this algorithm, future values of the time series are simply substituted by the predicted values as the recursion progresses from $t+1$ to $t+2$ based on (16). In the Gaussian case, all negative predictions are set to zero at the end of the recursion.

For the log-linear model (17), the substitution method becomes invalid in the recursive algorithm. For example, it is easy to show that

$$\begin{aligned}
\Delta_h(t+2|t) &= E\left\{ \exp\left[a_0 + \sum_{i=1}^{p} a_i \log(\Delta_h(t+2-i)+1)\right] \middle| \mathcal{F}_t \right\} \\
&\neq \exp\left[a_0 + \sum_{i=1}^{p} a_i \log(\Delta_h(t+2-i|t)+1)\right].
\end{aligned}$$

A practical way of computing the predicted value $\Delta_h(t+k|t)$ for $k \geq 2$ is Monte Carlo simulation. More specifically, the predicted value $\Delta_h(t+k|t)$ is approximated by the average of m independent samples (for suitably large m). The ℓ-th sample, denoted by $\Delta_h^{(\ell)}(t+k|t)$, is drawn randomly from the Poisson distribution with mean

$$\exp\left\{a_0 + \sum_{i=1}^{p} a_i \log(\Delta_h^{(\ell)}(t+k-i|t)+1)\right\} \qquad (\ell = 1,\ldots,m), \tag{18}$$

and the initial values are given by $\Delta_h^{(\ell)}(t+k-i|t) := \Delta_h(t+k-i)$ if $i \geq k$. Note that the mean in (18) employs simulated samples from previous recursions where actual values are not available.

Finally, let us discuss a different kind of time series models which are tailored to the special structure of the demand from unplanned events. According to (13), it is only the h-step ahead prediction that is needed for series $\{\Delta_h(t)\}$ in order to obtain the forecast of the total demand at horizon h. The time series models discussed before produce the h-step ahead prediction through a recursion which generates intermediate predictions from step 1 to step $h-1$ before arriving at the final prediction in step h. We call them recursive models of prediction. An alternative approach is to employ a linear function of past observations in the form of (16) to model directly the h-step ahead conditional mean rather than the one-step ahead conditional mean. In other words, it is assumed that $\Delta_h(t+h)$, rather than $\Delta_h(t+1)$, has a conditionally independent Gaussian or Poisson distribution with mean

$$\lambda(\mathcal{F}_t) := E\{\Delta_h(t+h)|\mathcal{F}_t\},$$

which takes the linear form (16). In the Poisson case, the log-linear form (17) remains applicable. They are equivalent to the subset autoregressive models of order $h+p-1$ in which the coefficients of the first $h-1$ lagged variables are set to zero.

An advantage of these so-called horizon-specific models is that the required h-step ahead prediction

$\Delta_h(t+h|t)$ is given directly by $\lambda(\mathcal{F}_t)$ without the need of recursion. Furthermore, the coefficients in these models can still be derived under the GLM framework. In the Gaussian case, it is a straightforward application of the least-squares method. In the Poisson case, there are two options for the constraint on the coefficients. One option, called universal constraint, is to enforce the positivity of the mean for all possible values of counts. It leads to the same constraint on the coefficients as in the recursive linear Poisson model. The other option, called conditional constraint, is to enforce the positivity of the mean only for its values on the training data. This option may result in negative predictions. But, because they are not the intermediate values to be used in a recursion, one can simply set them to zero as in the Gaussian case. Similarly, because no recursion is needed to produce the h-step ahead prediction, the stability constraint seems less critical in practice. Without imposing the stringent constraints, the horizon-specific models can be trained much more quickly than their recursive counterparts.

3.3 Case Study

For each given demand category, the weekly time series $\{\Delta_h(t)\}$ for $h = 1,\ldots,8$ are constructed from the snapshot data discussed in Section 1 based on the launch time of each event as well as its age and category at the launch time. In this study, we investigate the predictions of these time series using three recursive models and three horizon-specific models. The recursive models are built under the (linear) Gaussian, linear Poisson, and log-linear Poisson assumptions, respectively. So are the horizon-specific models.

The recursive Gaussian model is trained by the Yule-Walker method using the R function `ar.yw`. It serves as the benchmark for comparison of prediction accuracy. The recursive linear and log-linear Poisson models are obtained by using the function `tsglm` in the special R package `tscount` for time series of counts (Liboschik, Fokianos, and Fried 2015). This function enforces the universal positivity constraint and the stability constraints on the linear model (16) as discussed before. It also imposes a conjectured stability constraint on the log-linear model (17) such that $|a_i| < 1$ for $i = 1,\ldots,p$ and $|\sum_{i=1}^{p} a_i| < 1$.

The horizon-specific Gaussian model is trained by the least-squares method using the R function `glm`. The horizon-specific linear and log-linear Poisson models are trained by two methods. One method uses the function `glm` with identity and log links, respectively. It imposes only the conditional positivity constraint on the linear model, and no stability constraints for either linear or log-linear models. The other method employs the function `tsglm` which imposes the universal positivity constraint on the linear model and the stability constraint on both linear and log-linear models. The order p of each model is determined empirically by minimizing the BIC criterion of the form $-2\times$ log-likelihood $+$ number of parameters $\times \log(\text{length of time series})$. The number of parameters is set to $p+1$ for the Poisson models and $p+2$ for the Gaussian models. The maximum order is constrained to be 8.

Three practical remarks are in order. First, with the aim of automation in mind, the default control options of the fitting algorithms are used in all cases for all models. All failed attempts are excluded from consideration. Moreover, to alleviate numerical difficulties caused by data disparity, the time series models are applied only to the cases where 10 or more nonzero values exist in the training data, and the remaining cases are predicted by the sample mean. Finally, it is worth pointing out that training the Poisson models using `tsglm` is roughly 30 times slower than training them using `glm` due to the more stringent constraints imposed by `tsglm`. The Gaussian models are the most computationally efficient.

Predictions of the recursive log-linear Poisson model are computed by the Monte Carlo method using 200 random samples. Predictions of the other models are straightforward to compute. As a statistical metric of performance, the accuracy of each model is measured by the RMSE of predictions at each horizon h. It is calculated over 25 consecutive weeks (week 75 through week 99) and across all demand categories.

Using the recursive Gaussian model trained by the Yule-Walker method as benchmark, Table 3 shows the percentage improvement in RMSE by the other models at each horizon as well as the average percentage improvement across the horizons. Based on this result, the recursive linear Poisson model does not perform as well as the recursive Gaussian model, but the recursive log-linear Poisson model offers a slight edge

Table 3. Performance of Time Series Models for Forecasting Demand from Unplanned Events

Model	Horizon 1	2	3	4	5	6	7	8	Average
Poisson1-R(tsglm)	0.1	-4.3	-5.3	-7.9	-9.7	-7.1	-4.6	-5.4	-5.5
Poisson2-R(tsglm)	0.6	0.4	0.5	0.4	0.9	1.0	0.5	0.4	0.6
Gaussian-H(glm)	-0.6	-3.3	-3.3	-3.4	-3.2	-6.0	-7.7	-10.7	-4.8
Poisson1-H(glm)	-0.0	-1.7	-2.2	-1.5	-2.9	-3.9	-6.3	-10.5	-3.6
Poisson2-H(glm)	-0.6	-7.6	-28.3	-19.2	-22.9	-64.9	-86.4	-Inf	-Inf
Poisson1-H(tsglm)	0.1	-0.2	0.4	0.3	0.0	-0.4	0.0	-0.8	-0.1
Poisson2-H(tsglm)	0.6	0.1	0.6	0.5	0.4	-0.1	0.1	-5.3	-0.4
BIC-Selected	0.2	-3.2	-2.9	-4.6	-7.0	-2.9	-2.7	-1.4	-3.1
BIC-Weighted	0.4	-2.2	-2.3	-3.4	-5.7	-2.6	-1.9	-1.0	-2.3
Error-Weighted	1.0	0.6	1.5	2.3	2.4	0.9	2.3	0.9	1.5
Oracle	6.7	9.4	10.4	11.6	13.4	11.6	13.2	12.6	11.1

Note: The table shows the percentage improvement in RMSE over the recursive Gaussian model by `ar.yw`. Poisson1, linear Poisson model; Poisson2, log-linear Poisson model. R(tsglm), recursive model by `tsglm`; H(glm), horizon-specific model by `glm`; H(tsglm), horizon-specific model by `tsglm`.

over the benchmark across all horizons. The horizon-specific models trained by `glm` do not perform well, especially the log-linear Poisson model whose prediction errors become unbounded at horizon 8. With the stability constraint imposed, the horizon-specific models trained by `tsglm` become more competitive, though still somewhat inferior, in comparison with the benchmark and the best-performing recursive log-linear Poisson model.

Of course, better on average does not necessarily mean better in every case. As an example, Figure 9 shows the series $\{\Delta_4(t)\}$ and its 4-step ahead forecast at $t = 93$ using the three recursive models for two demand categories. In the first case, the Gaussian model gives the most accurate forecast and the linear Poisson model gives the least accurate forecast. However, the opposite is true in the second case. Therefore, it is natural to ask whether the predictions from different models can be utilized jointly to produce better results.

Toward that end, two experiments are conducted: one that combines the predictions from all models by weighted average, one that selects the best prediction from the pool of predictions. While the latter involves a hard (binary) decision, the former can be reviewed as a soft-decision method. The key challenge is to

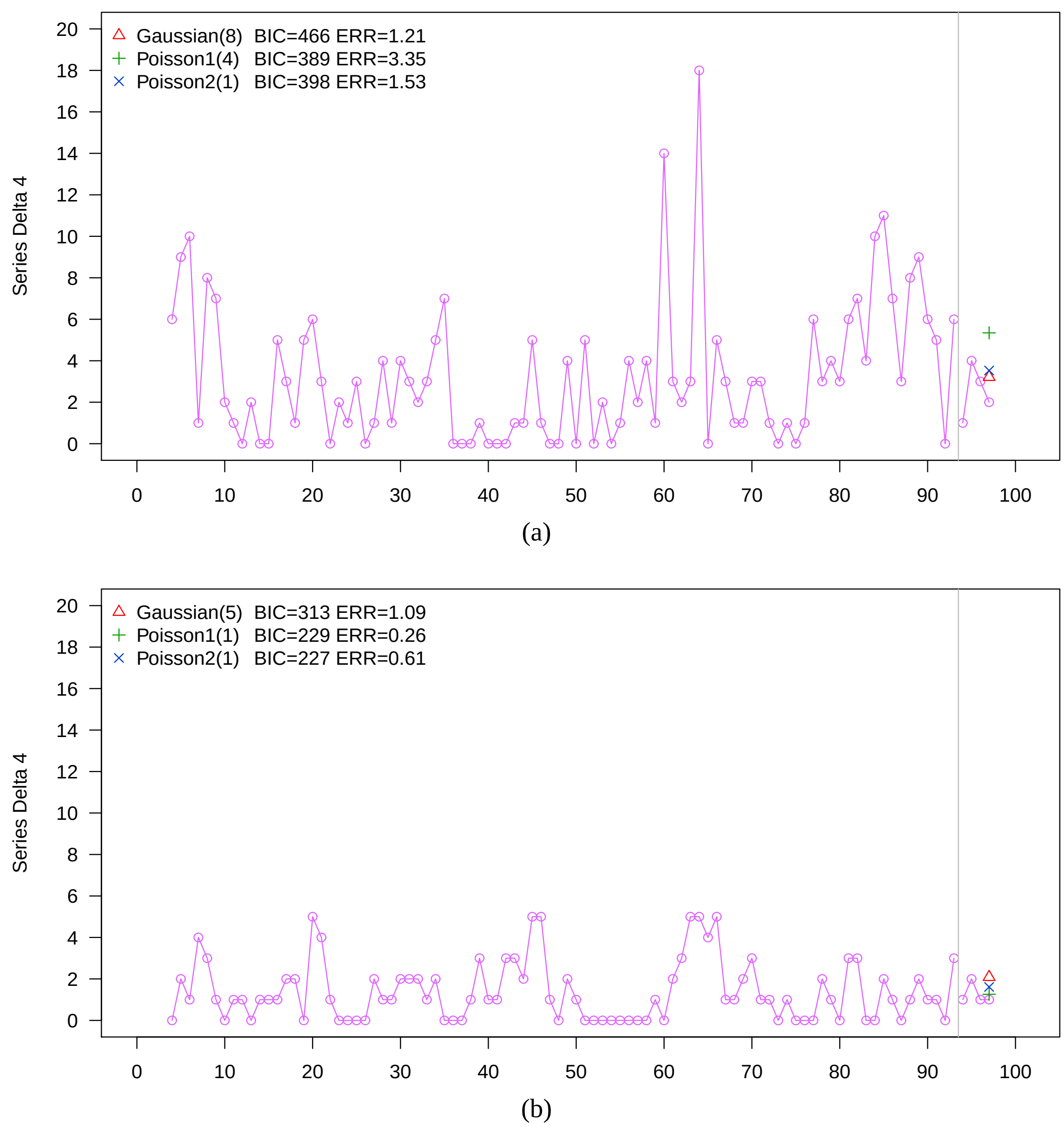


Figure 9: Forecast of series $\{\Delta_4(t)\}$ by recursive time series models for two demand categories: (a) J2EE and (b) JCL and PL/1. $\triangle$, forecast by the Gaussian model; $+$, forecast by the linear Poisson model; $\times$, forecast by the log-linear Poisson model; $\circ$, actual value. ERR is the absolute error of the forecast at horizon 4. The vertical line separates the training data from the testing data.

design a data-driven weighting mechanism for the soft-decision method and a data-driven selection criterion for the hard-decision method.

A simple choice of the selection criterion is the BIC. Because it is already calculated for determining the order of the time series models, no additional calculation or data storage is required for practical implementation. For the soft-decision method, we consider two designs for the weights with increasing complexity of implementation. The first design uses the available BIC values and makes the weights proportional to the exponential of $-\mathrm{BIC}/2$ which can be interpreted as the likelihood of the model discounted by the number of parameters. The second design employs a feedback mechanism to reflect the accuracy of the past predictions by each model. More specifically, let $w_k(t-1)$ denote the weight at time $t-1$ for the prediction of model k and let $\tilde{w}_k(t-1)$ be proportional to the exponential of the negative squared error of the prediction made at time $t-1$, which becomes available at time t. Then, the new weight at time t is given by

$$w_k(t) := \mu w_k(t-1) + (1-\mu)\tilde{w}_k(t-1),$$

where $\mu \in (0,1)$ is a tuning parameter that controls the rate of discounting past errors. Because it requires a feedback mechanism to collect the prediction error of each model, the second design has a higher complexity for practical implementation.

Table 3 shows the results of the experiments under the labels “BIC-Selected”, “BIC-Weighted”, and “Error-Weighted”. The horizon-specific log-linear Poisson model by `glm` is excluded from the experiments because of its poor individual performance. As can be seen, the model selection method by BIC performs better than some individual models but not as well as the benchmark and some other models. This result is not too surprising because a smaller BIC does not necessarily correspond to a smaller prediction error, as indicated by the examples in Figure 9. The BIC-weighted soft-decision method performs better than its hard-decision counterpart at all horizons, but remains generally inferior to the benchmark. By utilizing the past prediction errors directly, the error-weighted soft-decision method ($\mu = 0.5$) manages to offer a significant edge over all individual models across all horizons, thus proving the usefulness of the model pooling approach. The row labeled "Oracle" in Table 3 shows the potential performance that can be achieved if the best prediction can be determined correctly every time. The still large gap between this and the other rows signifies the room for improvement.

In practice, the accuracy has to be considered in conjunction with the complexity in order to arrive at an implementation plan. The single model approach remains attractive in this regard, especially the ordinary recursive Gaussian autoregressive model, which offers reasonable accuracy at the lowest computational cost. The recursive log-linear Poisson model cannot be ruled out based on the accuracy; but the longer training time and higher risk of numerical difficulties must be taken into account. The error-weighted pooling method offers the highest accuracy, but the computational cost multiplies because all models in the pool have to be trained and the past prediction errors of all models have to be tracked.

4 Total Demand

By definition, the total demand in a coming week comprises the demand from planned events and the demand from unplanned events. It can be predicted by simply summing up the predictions of these two components according to (13). While it is straightforward to obtain the point forecast, interval forecast needs additional assumptions that lead to different variations.

4.1 Point and Interval Forecast

The point forecast (13) can be regarded as the conditional mean of the total demand $\gamma(t+h)$ given the historical information up to time t which is denoted by $\mathcal{F}_t$. In this calculation, $\pi(t+h|t)$ and $\Delta_h(t+h|t)$ are the conditional mean of $D(t+h|t)$ and $\Delta_h(t+h)$, respectively. If the corresponding conditional variances are denoted by $\sigma_D^2(t+h|t)$ and $\sigma_\Delta^2(t+h|t)$, then, under the assumption that $D(t+h|t)$ and $\Delta_h(t+h)$ are conditionally uncorrelated, and by virtue of (11) and (12), the conditional variance of $\gamma(t+h)$ takes the form

$$\sigma^2(t+h|t) = \sigma_D^2(t+h|t) + \sigma_\Delta^2(t+h|t). \tag{19}$$

The second term $\sigma_\Delta^2(t+h|t)$ in (19) can be specified by the variance of the h-step ahead prediction error produced by the time series model of $\{\Delta_h(t)\}$. The first term $\sigma_D^2(t+h|t)$ in (19) depends on the choice of $\pi(t+h|t)$: if $\pi(t+h|t)$ is given by (4), then $\sigma_D^2(t+h|t)$ takes the form (5); if $\pi(t+h|t)$ is the revised forecast in (7), then $\sigma_D^2(t+h|t)$ is given by the variance of the prediction error from the linear regression model.

The conditional variance $\sigma^2(t+h|t)$ in (19), together with the conditional mean $\gamma(t+h|t)$ in (13), can be used to construct interval forecast under suitable assumptions about the conditional distribution. For example, under the Gaussian assumption, an interval forecast with coverage probability $(1-\alpha)\times 100\%$ for some $\alpha \in (0, 1/2)$ is defined by the $\alpha/2$ quantile and the $(1-\alpha/2)$ quantile of the Gaussian distribution with mean $\gamma(t+h|t)$ and variance $\sigma^2(t+h|t)$.

The Gaussian distribution may not be entirely suitable for the demand data which are inherently non-negative integer-valued. An alternative model is the negative binomial distribution of the form

$$\Pr\{\gamma(t+h) = d|\mathcal{F}_t\} := \frac{\Gamma(\theta+d)}{\Gamma(d+1)\,\Gamma(\theta)}\rho^\theta(1-\rho)^d \qquad (d = 0, 1, \ldots), \tag{20}$$

where the parameters ρ and θ are specified by the method of moments,

$$\rho := \frac{\gamma(t+h|t)}{\sigma^2(t+h|t)}, \quad \theta := \frac{\gamma^2(t+h|t)}{\sigma^2(t+h|t) - \gamma(t+h|t)}. \tag{21}$$

The negative binomial model is valid only if $\sigma^2(t+h|t) > \gamma(t+h|t)$, which is known as the over-dispersion condition. Otherwise, the Poisson distribution with mean $\gamma(t+h|t)$ can be used as a conservative choice for interval forecast.

4.2 Case Study

Consider the data discussed in Section 1. For each demand category and forecasting horizon, the demand from planned events is predicted by the two-tiered survival method combined with the trusted manager technique and the regression model (7). To forecast the demand from unplanned events, we use the linear prediction (recursive Gaussian) method defined by (14) for simplicity. Combining these components according to (13) gives the final forecast of the total demand.

Figure 1 shows the forecast together with the true weekly demand series for two demand categories. Also depicted in Figure 1 is the 90% interval forecast under the negative binomial model given by (20) and (21). The forecast is made at the end of week 75 for the next 8 weeks (week 76 through week 83). As can be seen, the forecast is able to generate the big swing in Figure 1(a) and the upward trend in Figure 1(b). Overall, the RMSE of the forecast across the horizons is equal to 1.7 in Figure 1(a) and 2.0 in Figure 1(b).

Figure 1 also shows the baseline forecast based solely on the planned events and their scheduled start date. It is not surprising that the baseline forecast tends to underestimate the demand, yielding a larger RMSE of 3.0 in Figure 1(a) and 4.8 in Figure 1(b).

Given the baseline forecast for demand category c, denoted by $\gamma_B(t+h|t,c)$, the so-called pickup model takes the form

$$\gamma_P(t+h|t,c) = a(h,c) + b(h,c)\gamma_B(t+h|t,c), \tag{22}$$

where $a(h,c)$ and $b(h,c)$ are horizon and category dependent coefficients. This method has been successfully used in reservation-based service industries such as airlines and hotels to predict bookings to come based on bookings on hand (L'Heureux 1986; Weatherford and Kimes 2003). The linear equation (22) is a generalization of the classical pickup models which only include $a(h,c)$ or $b(h,c)$, representing the addi-

Table 4. Performance for Total Demand Forecasting

Method	Horizon 1	2	3	4	5	6	7	8
Proposed	17.20	19.90	19.00	18.10	18.00	17.20	17.10	16.50
Pickup	8.70	13.40	12.70	12.20	12.20	13.70	14.00	13.80
ARIMA/ES	0.10	12.60	12.10	9.30	5.40	0.40	-5.00	-18.50

tional bookings to come relative to the bookings on hand. In the generalized form (22), these coefficients are derived by least-squares regression. Figure 1 shows that the pickup method yields better results than the baseline method but remains inferior to the proposed method. The RMSE is equal to 2.6 in Figure 1(a) and 3.0 in Figure 1(b).

Finally, consider the results of the ARIMA/ES method shown in Figure 1. These results are produced by the best of the ARIMA and ES models which are selected adaptively by SPSS Expert Modeler. The method yields a nearly flat forecast in Figure 1(a) and a downward trend in Figure 1(b), badly missing the swing and the upward trend, respectively. The resulting RMSE is equal to 3.1 in Figure 1(a) and 4.5 Figure 1(b).

A more comprehensive comparison of these methods is given in Table 4. It contains the average Q-scores calculated over 25 consecutive weeks (week 75 through week 99) and all demand categories at each horizon. In this experiment, the proposed method is able to outperform the others at all horizons. The pickup method comes in second. The advantage of the proposed method over the pickup method, especially at shorter horizons, can be attributed to its better handling of the noisy events in the pipeline and its incorporation of the serial correlation of unplanned events. Without utilizing the planned events, the ARIMA/ES method does not perform as well as the the proposed method and the pickup method.

5 Concluding Remarks

The main objective of this article is to propose a statistical framework of demand forecasting for capacity planning and management of resource-pool-based software development services. Under the framework, the predictive information in the pipeline is utilized though survival models, and the predictions based on the pipeline information is combined with special complementary time series models to produce weekly demand forecasts at multiple horizons for multiple demand categories. The empirical study based on a real dataset confirms the benefit of the proposed method in comparison with some alternatives, including the baseline method which takes the pipeline projections at their face value, the pickup method which incorporates the pipeline projections by linear equations, and the conventional time series method which exploits serial correlation while ignoring the pipeline information.

The specific statistical techniques demonstrated in this article under the proposed framework are not intended to be exhaustive. There are in fact numerous possibilities to explore additional techniques of survival analysis and time series analysis for better results. In the following, we mention just a few. A detailed examination of these techniques is beyond the scope of this article and will be reported elsewhere in the future.

In regard to survival function modeling, the deficiencies of the nonparametric Kaplan-Meier estimator can be mitigated by incorporating suitable parametric models (Kalbfleisch and Prentice 2002) either globally or locally in the right-hand tail of the survival function. Appropriate stratification of the Cox proportional hazards estimator (Cox and Oakes 1984) is another way of dealing with the problem. One may also apply the regularized Cox regression method (Tibshirani 1997; Gui and Li 2005) by treating the subpopulations as categorical covariates subject to variable selection. Nonparametric Bayesian estimators such as that proposed in Susarla and Van Ryzin (1976) offer additional possibilities for handling data disparity.

For time series modeling, it is observed that the time series of demand from unplanned events may contain more zeros than expected under the Poisson assumption. The excessive zeros can be accommodated by mixing the integer-valued distribution with a point mass at zero to obtain a zero-inflated model (Lambert 1992). One can also use the so-called hurdle models (Mullahy 1986) to boost the probability of zero. Additional models for time series of counts are discussed in Kedem and Fokianos (2002) and Cameron and Trivedi (2013).

References

Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (2008), *Time Series Analysis: Forecasting and Control* (4th ed.), New York: Wiley & Sons.

Cameron, A. C., and Trivedi, P. K. (2013), *Regression Analysis of Count Data* (2nd ed.), Cambridge, UK: Cambridge University Press.

Canals-Cerdá, J. J., and Kerr, S., "Forecasting credit card portfolio loses in the great recession: a study in model risk," Working Paper No. 14-10, Research Department, Federal Reserve Bank of Philadelphia.

Cox, D. R., and Oakes, D. (1984), *Analysis of Survival Data*, Boca Raton, FL: Chapman & Hall/CRC.

Christou, V., and Fokianos, K. (2015), "On count time series prediction," *Journal of Statistical Computation and Simulation*, 85, 357–373.

Ferland, R., Latour, A., and Oraichi, D. (2006), "Integer-valued GARCH processes," *Journal of Time Series Analysis*, 27, 923–942.

Fokianos, K., Tjøstheim, D. (2011). "Log-linear Poisson autoregression," *Journal of Multivariate Analysis*, 102, 563–578.

Gui, J., and Li, H. (2005), "Penalized Cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data," *Bioinformatics*, 21, 3001–3008.

Howe, J. (2008), *Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business*, New York: Three Rivers Press.

Kalbfleisch, J. D., and Prentice, R. L. (2002), *The Statistical Analysis of Failure Time Data* (2nd ed.), New York: Wiley & Sons.

Kaplan, E. L., and Meier, P. (1958), "Nonparametric estimation from incomplete observations," *Journal of the American Statistical Association*, 53, 457–481.

Kedem, B., and Fokianos, K. (2002), *Regression Models for Time Series Analysis*, New York: Wiley & Sons.

Lambert, D. (1992), "Zero-inflated Poisson regression, with an application to defects in manufacturing," *Technometrics*, 34, 1–14.

Lawless, A. F. (2002), *Statistical Models and Methods for Lifetime Data* (2nd ed.), New York: Wiley & Sons.

Lee, A. O. (1990), "Airline reservations forecasting: probabilistic and statistical models of the booking process," Flight Transportation Laboratory Report R 90-5, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

L'Heureux, E. (1986), "A new twist in forecasting short-term passenger pickup," in *Proceedings of the 26th Annual Symposium of the Airline Group of the International Federation of Operational Research Societies (AGIFORS)*, pp. 234–247.

Liboschik, T., Fokianos, K., and Fried, R. (2015),"tscount: an R package for analysis of count time series following generalized linear models," Retrieved from http://cran.r-project.org/web/packages/tscount/vignettes/tsglm.pdf.

Malm, A., Ljunggren, O., Bergstedt, O., Pettersson, T. J. R., and Morrison, G. M. (2012), "Replacement predictions for drinking water networks through historical data," *Water Research*, 46, 2149–2158.

Marubini, E., and Valsecchi, M. (1995), *Analysing Survival Data from Clinical Trials and Observational Studies*, New York: Wiley & Sons.

Mullahy, J. (1986), "Specification and testing in some modified count data models," *Journal of Econometrics*, 33, 341–365.

Peng, X., Ali Babar, M., and Ebert, C. (2014), "Collaborative software development platforms for crowdsourcing," *IEEE Software*, 31, 30–36.

Read, R. R. (1997), "The use of survival analysis in the prediction of attrition in large scale personnel flow models," Technical Report NPS-OR-97-006, Naval Postgraduate School, Monterey, CA.

Susarla, V., and Van Ryzin, J. (1976), "Nonparametric Bayesian estimation of survival curves from incomplete observations," *Journal of the American Statistical Association*, 61, 897–902.

Tibshirani, R. (1997), "The Lasso method for variable selection in the Cox model," *Statistics in Medicine*, 16, 385–395.

Vuković, M. (2009), "Crowdsourcing for enterprises," *Proceedings of 2009 World Conference on Services - I*, 686–692.

Weatherford, L., and Kimes, S. (2003), "A comparison of forecasting methods for hotel revenue management," *International Journal of Forecasting*, 19, 401–415.

Zeger, S., and Qaqish, B. (1988), "Markov regression models for time series: a quasi-likelihood approach," *Biometrics*, 44, 1019–1031.