The Journal of Systems and Software 161 (2020) 110480

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

# Performance evaluation of web service response time probability distribution models for business process cycle time simulation

Raghu Ramakrishnan[a,b,*], Arvinder Kaur[b]

[a] Tata Consultancy Services, Noida 201305, India
[b] USICT, Guru Gobind Singh Indraprastha University, Delhi 110076, India

ARTICLE INFO

*Article history:*
Received 5 May 2019
Revised 12 October 2019
Accepted 21 November 2019
Available online 22 November 2019

*Keywords:*
Simulation input modeling
Parametric distributions
Non-parametric distributions
Performance evaluation
Web service response time
Cycle time

ABSTRACT

*Context:* The adoption of Business Process Management (BPM) is enabling companies to improve the pace of building new capabilities, enhancing existing ones, and measuring process performance to identify bottlenecks. It is essential to compute the cycle time of the process to assess the performance of a business process. The cycle time typically forms part of service level agreements (SLAs) and is a crucial contributor to the overall user experience and productivity. The simulation technique is versatile and has broad applicability for determining realistic cycle time using historical data of web service response time. BPM tools offer inadequate support for modeling input data used in simulation in the form of descriptive statistics or standard probability distributions like normal, lognormal, which results in inaccurate simulation results.

*Objective:* We evaluate the effectiveness of different parametric and non-parametric probability distributions for modeling data of web service response time. We further assess how the choice of probability distribution impacts the accuracy of the simulated cycle time of a business process. The work is the first of such a study using real-world data for encouraging Business Process Simulation Specification (BPSim) standard setters and BPM tools to enhance their support for such distributions in their simulation engine.

*Method:* We consider several parametric and non-parametric distributions and explore how well these distributions fit web service response time from extensive public and a real-world dataset. The cycle time of the business process of a real-world system is simulated using the identified distributions to model the underlying web service data.

*Results:* Our results show that kernel distribution is the most suitable choice, followed by Burr. Kernel outperforms Burr by 86.63% for the public and 84.21% for the real-world dataset. The choice of distribution affects the percentile ranks like 90 and above than the median. The use of single-point values underestimates cycle time values at higher percentiles.

*Conclusion:* Based on our empirical results, we recommend the addition of kernel and Burr to the current list of distributions supported by BPSim and BPM tools.



## 1. Introduction

An increasing number of organizations are adopting intelligent business processes to offer enhanced value and experience to their customers. Consequently, Intelligent Business Process Management Suites (iBPMS) have become a key accelerator in their digital transformation journey. iBPMS in commercial space from IBM (IBM Business Process Manager Developer Center), Oracle (Oracle BPM), RedHat (Red Hat Process Automation Manager), Signavio (Signavio Business Transformation Suite), Bizagi, TIBCO, Pegasystems (Pega Digital Transformation Suite), SoftwareAG (ARIS), Appian, and jBPM in open source have matured over time and established their place in organizations. iBPMS help in analysis, design, low-code implementation, execution, monitoring, and optimization of business processes. A business process is a collection of tasks or activities performed in a specific order to accomplish a business objective or goal. It is essential to measure the time for a task, set of tasks, and the complete process to assess the performance of a business process. The Business Process Model and Notation 2.0 (BPMN2) provides a graphical notation for depicting the steps in a business process (BPM, 2014).

* Corresponding author.
*E-mail address:* raghuramakrishnan71@gmail.com (R. Ramakrishnan).

https://doi.org/10.1016/j.jss.2019.110480
0164-1212/

Tasks in a business process need user involvement or interaction with software systems. The user involvement, usually through a user interface, can be mechanical (e.g., scanning a document) or require thinking and making choices (e.g., approval and rejection decisions). The interaction with the system involves invoking web services or APIs from user interface actions. BPMN2 categorizes the task as a user task or service task. The activity performed by a person with the assistance of a software application is called a user task, and the activity involving system interaction is a service task. In this paper, we refer to the time spent in user task as user task time (UTT) and time taken by service task as service task time (STT).

The time between the start and completion of a business process is useful for assessing its operational efficiency. This time referred to as the cycle time, is a crucial contributor to the overall user experience, productivity, organizational revenue, and brand image. In the services sector, long cycle times can frustrate customers who have to wait to get serviced. For example, long cycle time in a customer service center may result in losing customers or negative publicity. Longer cycle time may mean opportunities for improving the efficiency of an existing process in terms of cost and time savings and enhancing customer satisfaction. As a consequence, cycle time is predominantly used as a performance target and made part of service level agreements (SLA) for confirming the quality of service delivery to customers. In practice, the cycle time SLA is set using intuition, experience, and ad-hoc approaches. Such SLA values are unrealistic and difficult to achieve most of the time resulting in poor user experience and penalties being incurred by the service provider during the operations phase.

It is essential to carry out quantitative business process analysis for understanding the cycle time characteristics to determine a realistic value of cycle time. The three techniques for process analysis are quantitative flow analysis, queuing theory, and process simulation (Abello and Oscar, 2013; Dumas et al., 2018). The flow analysis technique accounts for the sequential, alternative, parallel, and cyclic paths but does not consider waiting times arising due to resource contention. The use of queuing theory is not appropriate for processes with parallel or cyclic paths. Also, the results of this theory are valid only for long-term analysis because a steady-state system behavior is assumed. In comparison, the simulation technique is versatile, has broader applicability, and addresses the limitations of flow analysis and queuing theory. Although we can monitor the cycle time characteristics periodically or in real-time, simulation offers the benefit of identifying design issues in processes that could affect performance before actual implementation. Further, it also helps to identify bottlenecks and optimize existing processes and evaluate what would be the impact if we made specific modifications to a process. Thus, simulation is useful for both as-is analysis and what-if analysis. Considering the benefits of business process simulation, iBPMS vendors now provide default support for the same (IBM Business Process Manager Developer Center, Oracle BPM, Red Hat Process Automation Manager, Signavio Business Transformation Suite, Visual Paradigm, Bizagi, ARIS, Jansen-Vullers and Netjes). Also, simulation software MATLAB (Simulink and JSim) provide generic process modeling and simulation functionality. Gartner asserts the ability to simulate process behavior as one of the critical capabilities for differentiating products in the iBPMS space (Dunie et al., 2018). The modeling of each input variable using an appropriate probability distribution is an essential consideration in simulation experiments (Law, 2013) because an input variable exhibits randomness, and not modeling them properly can undermine the success of such experiments. iBPMS provide support for modeling business processes at varying levels of granularity. The Business Process Simulation Specification (BPSim) Gagne (2016) includes a list of parametric probability distributions that business process simulators are expected to support, to assist in modeling of the input variables. The list contains parametric distributions like beta, binomial, Erlang, gamma, lognormal, negative exponential, normal, Poisson, truncated normal, uniform, and Weibull and non-parametric distributions like triangular and user distribution (which is a list of values and the probability of its occurrence). In practice, most iBPMS simulation engines support a limited number of probability distributions, which can be used to generate values of input variables. The distributions usually supported are uniform (IBM BPM; OracleBPM; Red Hat JBoss BPM Suite; Signavio), normal (IBM BPM; OracleBPM; Red Hat JBoss BPM Suite; Signavio; Tibco), constant (IBM BPM; OracleBPM; Signavio; Visual Paradigm; Tibco), empirical (Tibco, 2016), exponential (OracleBPM), and Poisson (Red Hat JBoss BPM Suite). There are only a few simulation engines that support all the distributions given in the BPSim specification (Bizagi 11.1, ARIS Simulation, L-Sim). Due to the inadequate support offered by the tools, input modeling remains a neglected part of business process simulation studies carried out by practitioners in the industry.

The motivation for our work is to identify a set of distributions that need to be made part of the BPSim specification for better modeling of input variables. Through this research work, we intend to draw the attention of iBPMS vendors for enhancing the existing input modeling capabilities of their simulation engine by supporting more distributions. Such improvements will encourage industry practitioners to adopt simulation techniques for quantitative business process analysis and set realistic SLA values.

This research is validated using real-world data of STT, one of the significant input variables used in the simulation of the business process cycle time. STT depends on the response time of the underlying web services or web API. So to model STT, we need to model the response time of the web services or web API. The historical data is available from the production or testing environment. The second variable UTT is similar in concept to think time used in performance testing, where think time is the time spent by the user between system tasks (Wilson, 2011). We have leveraged the work done by researchers to model think time using probability distributions like the uniform (Barber, 2003; Subraya, 2006), normal (Barber, 2003; Subraya, 2006), lognormal (Wilson, 2011), exponential (Barber, 2003; Brady, 2012; Mansharamani et al., 2010; Subraya, 2006), hyper-exponential (Mansharamani et al., 2010), or double hump normal distribution (Barber, 2003).

Researchers have attempted to use single-point estimates like mean, minimum, and maximum for representing web service response time. Such estimates are easy to calculate and use, but they do not represent the randomness observed in real-world response time. The use of a single point or constant estimate like mean in simulation input modeling can undermine the accuracy of simulation experiment results because it is the variability in input that causes contention in queuing systems (Law, 2013). The use of probability distributions in modeling web service response time is limited to standard distributions like empirical, normal, t location-scale, and lognormal. The standard distributions do not model response time accurately because the observed data may be multimodal or skewed due to the presence of outliers, which appears to be a common occurrence (Dilley, 1996; Loosley, 2000; Dunne et al., 2012).

### 1.1. Motivating example

In this section, we describe a real-world case study from the public sector, which motivates our research work. Government departments are delivering a wide variety of public services like driving license, passport, visa, and social benefits to the public in an increasingly digital mode. The public may need to interact with the public-facing interface of the department or its authorized service provider for availing the services (Scott et al., 2015). Efficient

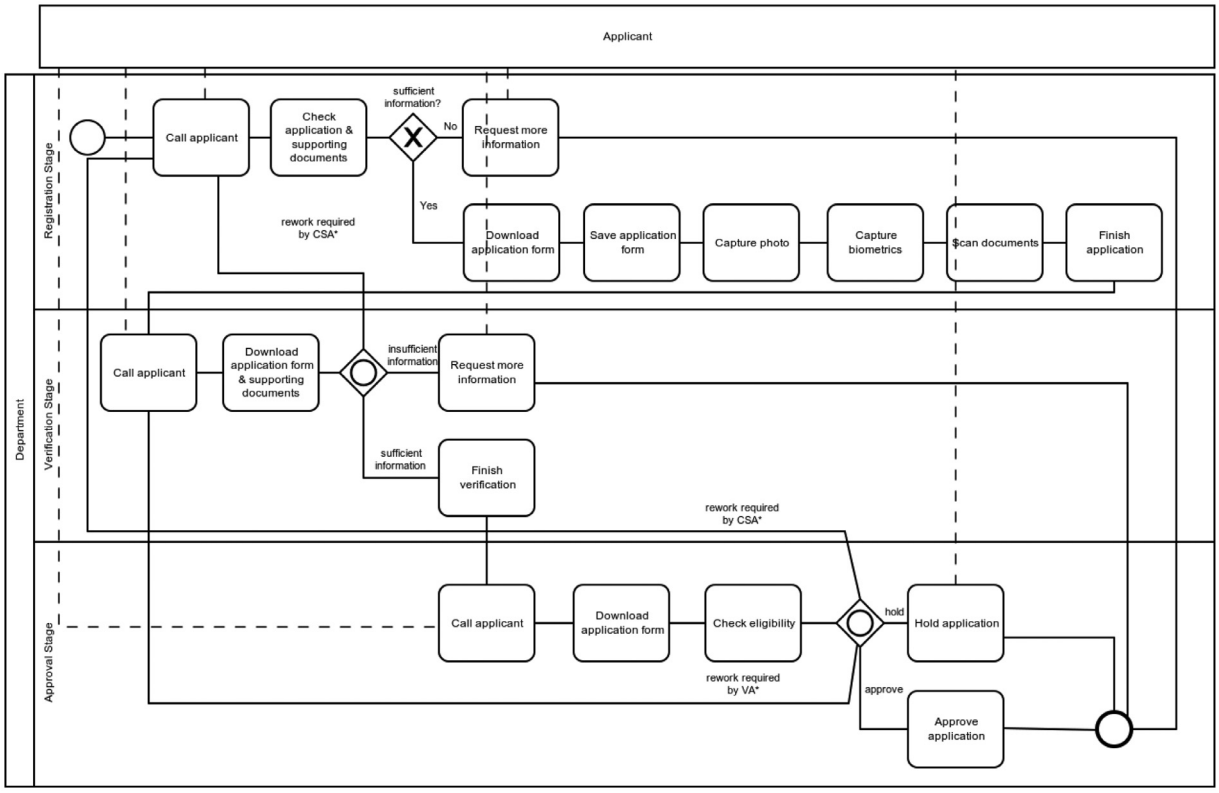



**Fig. 1.** Business process definition for the motivating example.

public service delivery is key to building trust and improving the perception of the department. The cycle time is the most relevant parameter for measuring the operational efficiency of services delivered by the department.

### 1.1.1. Business view

The scenario consists of a workflow initiated by an applicant for availing a service provided by a government department. The workflow is processed by the department staff using a software application with an underlying iBPMS. Fig. 1 shows the business process definition (BPD) developed in BPMN2. BPD is a process model or diagram which defines what is the same in all runtime instances of the process model.

The workflow is created on the applicant filling up an online application form available on the department website, paying the requisite processing fee electronically, and scheduling an appointment for an in-person interview at one of the facilitation centers of the department. A token is issued after a quick initial inspection of the documents when the applicant arrives at the facilitation centre on the appointment date and time. The workflow at the centre moves through the following three stages.

**Registration or CSA stage**: The applicant is serviced by the next available Customer Service Agent (CSA), who checks the application form and supporting documents for completeness. The CSA makes corrections in the application form if needed, captures the photograph and biometric data of the applicant, scans the supporting documents. Finally, the CSA then uploads the final application form, photo, biometric data, and the scanned documents to the server, initiating the next stage of the workflow.

**Verification or VA stage**: The applicant gets called by the next available Verifying Authority (VA). The VA checks the identity of the applicant, authenticity of documents, matches the details in the documents with the information in the application form, and compliance to prescribed rules. The VA marks the application as verified, initiating the next stage of the workflow.

**Approval or AA stage**: Finally, the applicant is called by the next available Approving Authority (AA), who checks the eligibility of the applicant and takes a decision. The applicant is notified of the decision by email.

In some cases, the AA may put a form on hold for an additional review by a senior officer. If some information is found to be insufficient at any stage, the applicant is requested to arrange for the same (shown by dashed lines terminating at the applicant). The applicant will visit the centre at a future date, after taking an appointment and the process resumes from where it ended during the previous visit. The VA and AA may also send back the applicant to any of the earlier stages for rectifying incorrect entries (marked with a *). The integrated electronic displays present in the facilitation center guide applicants to the respective service stations. The applicant may have to wait for sometime at each stage before getting serviced because the number of service stations is limited.

### 1.1.2. Implementation view

Fig. 2 shows an implementation of the Scan documents activity from the Registration stage. The activity comprises of a user task and two service tasks. The CSA performs this activity with a user interface having Scan, Upload, and Finish actions, with the user task representing the user interface. The Scan action initiates the scanning on the local scanner, the Upload action saves the scanned images on the server, and the Finish action denotes the completion of the scanning step. The Upload and Finish actions map to service tasks 'Save document' and 'Finish scanning' respectively, which internally invoke the saveDocumentApi and finishScanningApi. The implementation of other sub-processes is similar.

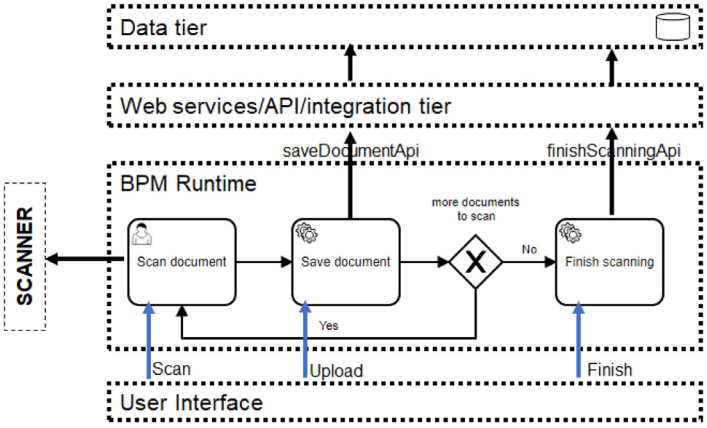



**Fig. 2.** Implementation view of Scan documents sub-process.

*1.1.3. As-is and what-if cycle time analysis*

For the department, the cycle time of the entire business process and its part (referred to as a stage in the example) is a critical SLA measurement as it influences business throughput, number of full-time employees (FTE), and number of counters at facilitation centers. Besides, cycle time is used by the department for business process re-engineering activities aimed at improving customer experience and reducing operational costs. The cycle time comprising of processing time and waiting time depends on various stochastic as well as deterministic attributes. The stochastic characteristics are the arrival pattern of the customer (e.g., uniformly distributed), UTT, and STT. The deterministic attributes are queue discipline (e.g., FIFO, priority), queue capacity (e.g., the waiting area accommodates a fixed number of customers), number of service stages, and number of service entities or stations (e.g., human resources) per stage. The department wants to do an as-is and what-if analysis using historical data before its next operational review aimed at removing bottlenecks and reducing operating costs.

- Q1: What is the realistic cycle time of a given stage based on the historical data?
- Q2: Will the cycle time significantly increase if we reduce the counters in a given stage by a specified number (cost reduction by resource optimization)?
- Q3: Will the cycle time significantly reduce if we increase the counters in a given stage by a specified number (user experience enhancement)?
- Q4: What will be the reduction in the cycle time of a given stage if the process was modified to make one of counter provided activities as part of the online application submission process (business process re-engineering)?

To find answers to the above questions, the department can carry out the modifications, deploy them in production, and measure the actual cycle time. The approach carries a significant risk of things going wrong, resulting in poor customer experience, bad publicity, loss of productivity, and SLA penalties. The modifications may also be difficult to rollback. The alternative to the trial and error approach is to carry out business process simulation experiments, evaluate the outcomes with stakeholders, and achieve consensus before making changes to production. The preference for simulation technique over quantitative flow analysis and queuing theory is because the described scenario has cyclic paths for rework and waiting time at each stage. The above real-world example is easy to generalize and apply in domains like financial services, insurance, health care, and the public sector with minor variations. The above scenario applies even if the implementation and operations are outsourced to a service provider, partially or wholly. The service provider needs to comply with the agreed SLA in such a case.

### 1.2. Research objectives

The research objectives of our work are as follows.

- O1: Compare and evaluate the effectiveness of different parametric and non-parametric probability distributions for modeling response time data of web services.
- O2: Evaluate how the choice of probability distribution impacts the accuracy of the simulated cycle time of a business process.

### 1.3. Organization of this paper

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 details our modeling approach. Section 4 presents the results of using the proposed approach on real-world web service response data. Section 5 provides a discussion on the findings. Section 6 addresses the internal and external validity of our work. Section 7 concludes this paper and provides directions for future work.

## 2. Related work

In this section, we explore the different approaches used by researchers for modeling web service response time and computing the response time of web service compositions. Response time data of web services has been analyzed extensively in the literature using descriptive statistics, probability distributions, or both.

**Descriptive statistics**: Dilley et al. summarize the historical response time data of web service using deterministic fixed-point calculations like average, median, or mode values, whereas Zheng et al. use only average (Dilley, 1996; Zeng et al., 2004). Loosely et al. use visual methods like histograms to describe the distribution of response time data (Loosley, 2000). Zheng et al. describe a method for calculating the QoS of a composite web service when the QoS of individual web services are available as constant values (Zheng et al., 2013).

**Probability distributions**: Hwang et al. consider the response time of atomic or composite services, as a discrete random variable, and model the same using a probability mass function (PMF) (Hwang et al., 2007). The approach involves partitioning the range of response time into discrete smaller intervals and then estimating the probability for each interval. For instance, if the probability of a web service completing in subintervals [0,1], (1,5] and (5, 8] is 0.3, 0.3 and 0.4 respectively, the PMF of response time is expressed as fx(0) = 0, fx(1) = 0:3, fx(5) = 0:3 and fx(8) = 0:4. Rosario

et al. model response time QoS data of six web services using the t location-scale distribution. The work also compares the SLAs calculated using two kinds of simulations in which response time values are generated using bootstrapping or obtaining random numbers from a tlocation-scale distribution with parameters estimated from the data (Rosario et al., 2008). Klein et al. model response time belonging to different periods and day of usage independently using the normal distribution, and then combine all such models (Klein et al., 2009). Zheng et al. use the non-parametric kernel distribution for modeling response time data of two web services and generating QoS of web service compositions for different invocation patterns like sequential, conditional, parallel, and loop using simulation (Zheng et al., 2010a).

**Descriptive statistics and probability distributions**: Cardoso et al. group modeling methods for web service QoS dimensions like time, cost, and reliability under two classes: basic and distributional (Cardoso et al., 2004). The basic class consists of single summary values like minimum, mean, and maximum values of the given QoS dimension for a given workflow task. The distributional class uses a constant value or a probability distribution like exponential, normal, Weibull, and uniform to describe the behavior of a task. The response time datasets from three software systems, namely Lotus Quickr Domino, SmartCloud Engage, and Lotus Domino, have been modeled using descriptive statistics like mean, median, standard deviation, interquartile range and probability distributions like lognormal and log-logistic (Dunne et al., 2012). The response time distributions of all three software systems were found to be positively skewed. The work also provides a method for modeling a multi-modal dataset, in which the data is partitioned based on mode, and then fitting is done on each partition separately. Hughes et al. have developed QoS Explorer, a tool for predicting QoS of a workflow using the QoS of its constituent services (Hughes and Hillman, 2006). The tool allows specifying the QoS of the constituent services as a range, expected value, or parameters of a probability distribution (like mean and variation of the best normal approximation). Geebelen et al. use Petri nets based simulation to generate a time series of the web service composition using QoS data of services making up the composition and predict future QoS values using a kernel-based quantile estimation (Geebelen et al., 2014). Ammannejad et al. introduce a tool to predict the percentiles of web service response time using historical data of response times corresponding to the transaction type and system load combination (Amannejad et al., 2016).

**Limitations in existing work**: Most of the work described above covers modeling QoS of web services using descriptive statistics, empirical, or standard probability distributions. Further, the data used for analysis in existing works is limited to a few web services or synthetically generated. Thus, at the current state, we are far from the understanding which probability distributions are accurate as well as generalizable for modeling the response time of web services. To address this gap, we use real-world data and compare several parametric and non-parametric distributions for their ability to model web service response time correctly. We verify how well the distributions fit the given data using the statistical goodness of fit tests and other criteria. For estimating cycle time of business processes or workflows, existing approaches use hard composition logic such as summing or taking the maximum response time value of the services constituting the process. The composition logic does not consider the response time probability distributions of the constituents. As pointed out by Rosario et al., the values derived using such approaches tend to be pessimistic (Rosario et al., 2008). In real-world projects, the majority of practitioners rely only on descriptive statistics because they are intuitive, well understood, and easy to calculate in comparison to more precise probability distributions.

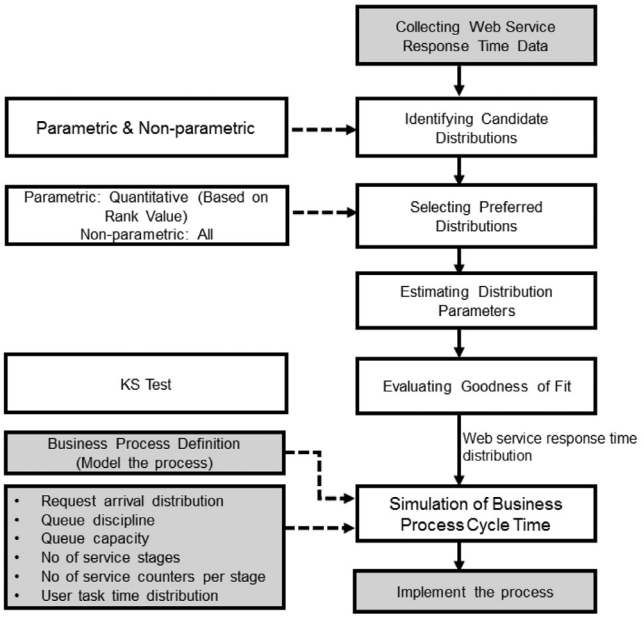



**Fig. 3.** Approach for choosing the right distribution and determining business process cycle time.

## 3. Methodology

In this section, we describe our approach for choosing suitable probability distribution models and the simulation model for determining the cycle time of the business process introduced in the motivating example. Fig. 3 presents the approach.

### 3.1. Choosing the right probability distributions

The term probability in probability distribution means that the future value of an attribute has an element of chance and is not predictable in a deterministic manner. Distribution fitting is the process of modeling a series of data represented as a random variable, using probability distributions. We obtain the data by repeated measurement of the variable phenomenon. The equations used to describe probability distributions are called probability density function (PDF) and cumulative distribution function (CDF). The distributions are categorized as parametric and non-parametric. A set of fixed parameters is used to describe a parametric distribution, whereas a set of parameters and the fitted data is used to describe a non-parametric distribution. We need to know only two parameters - mu (mean) and sigma (standard deviation) to find the probability of a normally distributed random variable taking on a given range of values. In comparison, we require to know the fitted data, the smoothing function, and the bandwidth parameter to find the probability of a random variable described using kernel distribution, taking on a given range of values.

The distribution fitting process consists of data preparation, candidate distribution selection, parameter estimation for selected distributions, and goodness of fit testing (Biller and Gunes, 2010; Law, 2013; Choi and Kang, 2013; Leemis, 2003). The choice of candidates is usually made using intuition, experience, or those supported by a tool. The standard goodness of fit tests does not provide a relative ranking of fitting done by different distributions. In this study, we present an enhanced fitting process that has an additional step for narrowing the set of parametric candidate distributions by calculating weighted rank of the models generated by each distribution.

**Identifying candidate distributions**: We select the list of statistical distributions by observing the characteristics of given data using histograms, measures of location (mean, median, mode, percentiles), measures of variability (range, variance, standard deviation), and measures of shape (skewness, kurtosis). Histograms help detect the presence of skewness and more than one mode (bimodal or multimodal), the modes appearing as distinct peaks. Parametric distributions that cover a broad set of skewness and kurtosis values appear to be more appropriate for modeling heavy-tailed and asymmetric data because skewness measures the lack of symmetry in the given data and kurtosis measures whether the given data is heavy-tailed. Only continuous distributions are considered as candidates because response time can take any value. Such distributions include Burr (Burr, 1942; Tadikamalla, 1980), lognormal, log-logistic (Dey and Kundu, 2009), Weibull, Pareto, generalized extreme value (Markose and Alentorn, 2005), t location-scale, inverse Gaussian and Birnbaum Saunders. The generalized Pareto distribution is useful in modeling the tail of a distribution containing fewer data points (Markose and Alentorn, 2005). The normal and t-distribution are suitable for modeling symmetric data. As an additional validation to check if any other distribution fits our data, we add distributions like beta, exponential, extreme value, gamma, logistic, Nakagami, Rayleigh and Rician to our list of candidates. Although we expected the distributions not being able to model our data, we start our analysis without bias towards any distribution. The total number of parametric distributions used in our work is 18.

The non-parametric distributions are the empirical, kernel (Wand and Jones, 1995; Scheid, 2004), piecewise linear, and triangular. The kernel distribution allows the use of Epanechnikov, normal, triangular, and uniform smoothing functions. The median or mean is used as the peak location parameter in triangular distribution. The non-parametric distributions are suitable for modeling data like bimodal or multimodal, which cannot be described by a parametric distribution (Dunne et al., 2012; Hughes and Hillman, 2006)

**Selecting preferred distributions using weighted rank**: We used Bayesian Information Criterion (BIC) Schwarz (1978), Akaike Information Criterion (AIC) Akaike (1974), and Negative Log-Likelihood (NlogL) Neath and Cavanaugh (2012) as criteria for narrowing down the initial list to the list of preferred parametric distributions. AIC and BIC both penalize likelihood value with BIC imposing more penalty on models with a higher number of parameters. The model with the minimum AIC, BIC, or NlogL is preferred over other models because it is the one with the highest plausibility of producing the observed values. The function *allfitdist* Sheppard is used to rank the fit of parametric distributions for the given data and criteria. The distribution having the lowest criteria value is assigned rank 1, the one having the next lowest value is assigned rank 2, and so on. We use all three measures to rank each distribution independently and then calculate the weighted rank (WR) for all distributions using Eq. (1). The distributions with higher WR are preferred over others.

$$WR_C = \sum_{j=1}^{T} \left( W_j * \frac{C_j}{M} \right) \tag{1}$$

- c is the criterion being considered
- T is the total number of ranks i.e. count of parametric distributions being considered
- $W_j$ is the weight assigned to the $j^{th}$ rank. The weights are assigned in the reverse order of the ranks. The rank 1 has weight T, rank 2 has weight T-1 and rank T has weight 1.
- $C_j$ is the number of times the distribution has been given the $j^{th}$ rank by the criterion being considered.
- M is the total number of datasets

For example, in 4676 web services, Burr distribution is ranked first 3449 times and second 1227 times when NlogL criterion is used. Rank 1 and 2 are assigned weights 2 and 1 respectively. By substituting the values in Eq. (1), the WR for burr distribution is calculated as 1.73 using the following expression.

$$WR_{NLogL} = 2 * \frac{3449}{(3449 + 1227)} + 1 * \frac{1227}{(3449 + 1227)}$$

We consider all non-parametric distributions as preferred because measures like BIC, AIC, and NlogL used for model selection do not apply in case of non-parametric distributions. We now are left with a smaller number of distributions for use in subsequent analysis. We also include normal distribution in our preferred list because researchers and practitioners commonly use the normal distribution. The inclusion allows us to compare the modeling capability of the preferred distributions with that of the normal distribution.

**Estimating distribution parameters**: We estimate the parameters of the preferred distributions for the given data using the maximum likelihood estimation method (MLE). MLE finds the parameter values in such a way that the likelihood of generating the observations with the estimated parameters, is maximized (Spinner et al., 2015).

**Evaluating goodness of fit**: To assess how well the selected distribution models the data, we use the Kolmogorov Smirnov (KS) test (Chakravarti et al., 1967). The KS test checks if the values generated from the specified distribution with estimated parameters and the given data come from the same distribution. The test statistic is the maximum difference between quantiles in data and quantiles determined by the proposed distribution for all values of the data. The null hypothesis that the preferred distribution models the given data cannot be rejected if the test statistic is less than the critical value or the p-value is greater than the level of significance.

**Implementation**: We now present the algorithm *FindLongestSubset* (Figure Algorithm 1 ) for automating the distribution fitting and its evaluation. The algorithm also finds the longest subset in the given dataset X, which can be successfully modeled using the specified distribution. Such automation helps us to model a large

**Algorithm 1:** FindLongestSubset.

```
Input  : Dataset: X
Input  : Distribution: D
Input  : Discard percentage: r

Sort the values in X
n = length(X);
start = n;
end = round(r * n);
alpha = 0.05;

for a ← start to end by −1 do
    cutoff = X[a];
    Add all X_a's which are ≤ to cutoff in Xtrunc
    pd = fitdist(Xtrunc, D);
    pv = kstest(Xtrunc, pd);
    if pv > alpha then
        subset = length(Xtrunc) / n;
        Report good fit, subset
        return
    end
end
Report bad fit
```

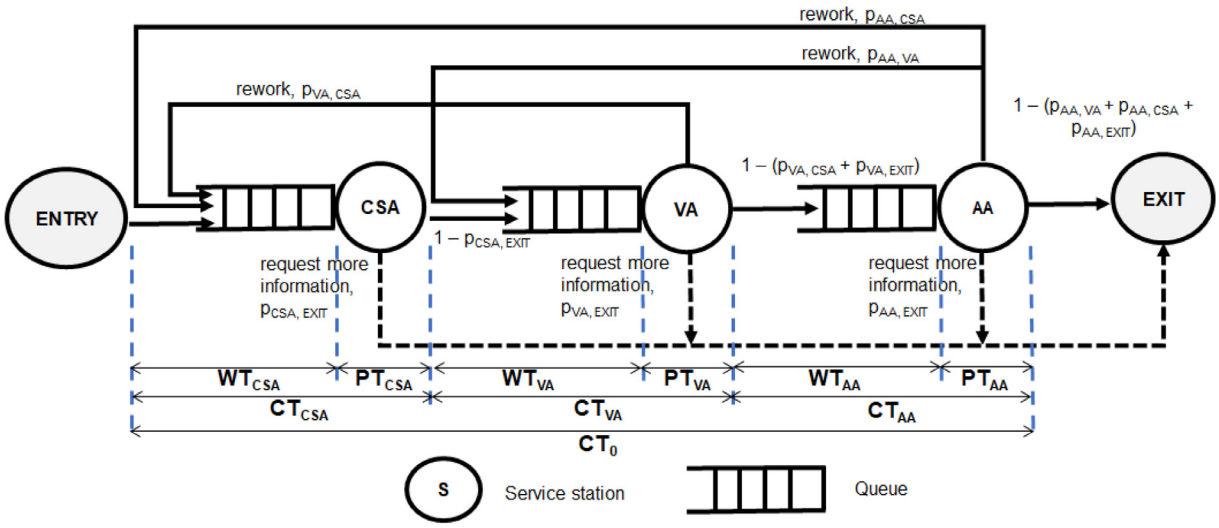



**Fig. 4.** FindProcessCycleTime simulation model.

number of web services using multiple distributions with minimal manual effort. The data is initially sorted in ascending order so that $X_1$ has the fastest time, and $X_n$ has the slowest time. The loop runs through our dataset in the reverse order starting from $X_n$. In each iteration, the value of X is set as the cutoff value. The dataset is truncated to include only those values which are less than equal to the cutoff value. The given distribution is fitted on this subset, and the quality of fitment is evaluated using the KS test. The iteration stops when the KS test returns a p-value greater than the level of significance $\alpha$ (set as 0.05 in our work) or the loop counter becomes equal to r percentage of n rounded off to the nearest units place. The discard percentage parameter r is useful for controlling the maximum percentage of data that is to be discarded during the distribution fitting. The parameter also assists us in getting insights about the volume of data corresponding to a good fit. The value of r is passed as 0.9 to exclude the top 10% values in the dataset. The value of r is given as 1 to include all data in the dataset. The functions *fitdist, makedist* and *kstest* in the algorithm are part of the MATLAB Statistics and Machine Learning toolbox (MathWorks Statistics and Machine Learning Toolbox). The *fitdist* and *makedist* functions fit specified distributions to given data and return a probability distribution object containing the estimated parameter values (Fitdist). For kernel and triangular distribution, the call to *fitdist* requires additional parameters whereas for piecewise linear distribution *fitdist* is replaced by *makedist*. The MATLAB scripts used in our work are available via GitHub.

While the quality of the fit is essential, detecting the absence of statistical significance (i.e., a p-value less than the 0.05) is not the only objective of our study. It is important to note that a statistically significant result does not always mean that the result is relevant in a real-world scenario. Further analysis, like checking the visual fit of CDFs, actual KS statistics may be required to conclude if the results have practical importance.

### 3.2. Simulation model for determining business process cycle time

Fig. 4 presents the *FindProcessCycleTime* simulation model for finding the cycle time of the business process introduced in the motivating example. We model each stage i.e., Registration, Verification, and Approval as a combination of a queue and a service station. The entities wait in the queue after arrival and advance to the next available service counter. The service counter renders service at a non-deterministic rate. The applicants, represented by entities, arrive at a specified rate at the Registration stage. The applicants depart from the system after receiving service from the AA or being asked to return at a later date with more information. The applicant may also move to any of the earlier stages for rework or corrections. The implementation was done using MATLAB but is easily portable to other iBPMS and generic simulation engines. The simulation model requires the following input variables.

- $N_s$ is number of service station in stage s
- $Q_s$ is queue discipline in stage s
- $C_s$ is capacity of queue in stage s
- $U_s$ is number of user tasks in stage s
- $S_s$ is number of service tasks in stage s
- $p_{i,j}$ is probability of moving from stage i to j.
- R - arrival rate of applicants in the system
- $UTT_{i,s}$ is time spent on the $i^{th}$ user task in stage s (in secs).
- $STT_{i,s}$ is time spent on the $i^{th}$ service task in stage s (in secs)

The simulation model requires the following additional notations for its working.

- $PT_s$ is time actually spent on processing the application in stage s (in secs) calculated using Eq. (2).
- $WT_s$ is time spent by an applicant waiting for service in stage s (in secs).
- $CT_s$ is cycle time for an applicant in stage s (in secs) calculated using Eq. (3).

$$PT_s = \sum_{i=1}^{U_s} UTT_{i,s} + \sum_{j=1}^{S_s} STT_{j,s} \tag{2}$$

$$CT_s = WT_s + PT_s \tag{3}$$

The cycle time of the complete business process is the sum of cycle times of its constituent stages in the process and is denoted by $CT_0$. The SLA value pertaining to processing time is normally stated in one or more of the following forms (Jin et al., 2002; Keller and Ludwig, 2003; Hwang et al., 2007; Rosario et al., 2008; Satyanarayana, 2012; Amannejad et al., 2016).

- The average cycle time of Y% instances of a business process to be less than the given SLA target.
- The maximum cycle time of Y% instances of a business process to be within the given SLA target. In real-world projects, there can be separate SLA targets corresponding to different percentages. For instance, the SLA target can be up to 10 minutes in 90% of the cases and 12 minutes in 99% of the cases.

**Table 1**
Distribution of skewness and kurtosis values for web services in WS-DREAM.

| Skewness | | | | Kurtosis | | |
|---|---|---|---|---|---|---|
| <0 | [0, 1) | [1, 5) | ≥5 | [1, 3) | [3, 10) | ≥10 |
| 33 | 39 | 3523 | 1798 | 18 | 541 | 4834 |

**Table 2**
Weighted ranks of parametric distributions for WS-DREAM, sorted by BIC value.

| Distribution | BIC | AIC | NlogL |
|---|---|---|---|
| burr | 16.6 | 16.86 | 17.11 |
| generalized extreme value | 15.88 | 16.1 | 16.23 |
| log-logistic | 15.59 | 15.32 | 15.07 |
| lognormal | 14.3 | 14.16 | 14.08 |
| inverse Gaussian | 13.4 | 13.28 | 13.22 |
| generalized Pareto | 13.29 | 13.47 | 13.56 |
| BirnbaumSaunders | 12.12 | 12.06 | 12.03 |
| tlocationscale | 10.92 | 11.04 | 11.11 |
| gamma | 10.43 | 10.46 | 10.53 |
| Weibull | 9.98 | 10.03 | 10.13 |
| exponential | 8.13 | 7.91 | 7.67 |
| Nakagami | 7.28 | 7.29 | 7.31 |
| logistic | 6.62 | 6.61 | 6.62 |
| Rayleigh | 4.6 | 4.54 | 4.48 |
| normal | 4.3 | 4.31 | 4.3 |
| Rician | 3.53 | 3.53 | 3.53 |
| extreme value | 2.44 | 2.44 | 2.45 |
| beta | 0.61 | 0.6 | 0.6 |

## 4. Results

### 4.1. Response time modeling

In this section, we present the results of our modeling exercise on data of web service response time from WS-DREAM and WS-REAL. The results will help address Research Objective O1 stated in Section 1.2.

#### 4.1.1. WS-DREAM data

WS-DREAM web service QoS dataset is a collection of 4 public datasets collected by Zheng et al. and made available to the service computing community to facilitate research (Zheng et al., 2010b; 2014). The datasets have been cited more than 200 times by researchers in papers Ws-dream. In our study, we have used the dataset referred to as the '339 * 5825 Web service QoS dataset'. It contains 339 response time measurements each for 5825 publicly accessible real-world web services of different applications hosted in more than 80 countries. Each web service is accessed by 339 virtual users positioned on a distributed network of computers across 30 countries. The dataset description does not mention the duration of data collection or include the timestamp associated with each observation that can be used to infer the duration.

Table 1 gives the distribution of skewness and kurtosis values for these 5393 web services. Most of the web services exhibit right skewness and heavy tails, indicating the presence of large values. For modeling, we have considered the web services for which 95% or more of its response time measurements are available. Only 5393 out of 5825 web services were found to meet the above requirement.

**Selecting preferred distributions**: For the selection of preferred parametric distributions from the list of candidate distributions, we calculate the weighted rank of each of distributions using Eq. (1). Table 2 shows the weighted rank of each distribution for all BIC, AIC, and NlogN measures. For further analysis, the top six parametric distributions are picked up by weighted rank. In this case, the selection of distribution remains the same irrespective of the measures. The selected distributions are Burr, generalized extreme value, log-logistic, lognormal, generalized Pareto, and inverse Gaussian. As shown in Table 3, 92.68% of web services can be modeled by one of these six distributions. All non-parametric distributions referred to in Section 3.1 viz. kernel, piecewise linear, triangle, and empirical form part of the preferred list.

**Table 3**
Percentage contribution by count of top six parametric distributions as per weighted ranks for WS-DREAM.

| BIC | AIC | NlogL | Mean |
|---|---|---|---|
| 92.68 (4887) | 91.79 (4950) | 90.62 (4998) | 91.70 (4982) |

**Estimating distribution parameters/evaluating goodness of fit**: The algorithm *FindLongestSubset* described in Section 3.1 is used to model response time data of each of the 5393 web services independently using the preferred parametric and non-parametric distributions. The given distribution is fitted on the complete data and three different subsets by setting the discard percentage r in the algorithm to 0.1, 0.2, and 0.3. Table 4 summarizes the result obtained from modeling web services of WS-DREAM. The value in each cell is the count and percentage (in parenthesis) of web services in which statistical significance is absent i.e., the p-value of KS test greater than 0.05.

**Non-parametric distributions**: In 99.5% (5366) web services, statistical significance was absent when piecewise linear distribution with a group size of two was used to model the data. Increasing the group size to 10, we could fit 99.2% (5350) of web services. With kernel distribution, the absence of statistical significance was observed in 66.75% (3600) of web services when using normal smoothing function. Replacing smoothing function with triangle, 63.68% (3434) of web services passed the goodness of fit test. The percentage of web services corresponding to normal and triangle smoothing function, passing the test improved by 5.6% (302) and 5.75% (310) respectively on discarding up to top 10% of the values.

**Parametric distributions**: The absence of statistical significance was found in 35.77% (1929) of web services when using burr distribution to model the data. The percentage of web services in which statistical significance was absent dropped to 30.08% (1622) and 22.81% (1230) on using log-logistic and generalized extreme value distribution. If we remove up to top 10% of the values, the percentage of web services corresponding to Burr, log-logistic, and generalized extreme value increased by 20.01% (1079), 22.96% (1238) and 27.52% (1484) respectively. The distributions like lognormal and normal could model only 3.91% (211) and 0.06% (3) of web services.

#### 4.1.2. WS-REAL data

WS-REAL has 294,620 response time values of 13 web services from the Registration stage of the business process described in the motivating example. The business process is built using an iBPMS. A browser-based client application running on desktops connected to an Intranet invokes the web services. The web services contain the complete functionality and do not internally call other web services. The client application is instrumented to record the time taken to serve a web service request and periodically upload the captured data containing a timestamp, a web service identifier, and execution time to the server. The response time of each web service corresponding to a business hour on every working day of the week is stored as a separate dataset. The measurements are taken at the same hour every day to eliminate any time-related influence on our findings. Each web service has five sets of response time data corresponding to five consecutive days of the week. Thus, the total number of datasets in WS-REAL is 65 (13 x 5). We model response time contained in each of the 65 datasets independently using probability distributions. Table 5 shows the skewness and

**Table 4**
Count (percentage) of web services of WS-DREAM in which $p > 0.05$ considering X% of data.

| Distribution | Type | 100% | [90%, 100%) | [80%, 90%) | [70%, 80%) | Total |
|---|---|---|---|---|---|---|
| empirical | Non-parametric | 5393 (100) | 0 | 0 | 0 | 5393 (100) |
| piecewiselinear (group size 2) | Non-parametric | 5366 (99.5) | 0 | 0 | 0 | 5366 (99.5) |
| piecewiselinear (group size 5) | Non-parametric | 5363 (99.44) | 0 | 0 | 0 | 5363 (99.44) |
| piecewiselinear (group size 10) | Non-parametric | 5350 (99.2) | 0 | 0 | 1 (0.02) | 5351 (99.22) |
| kernel - normal | Non-parametric | 3600 (66.75) | 302 (5.6) | 187 (3.47) | 170 (3.15) | 4259 (78.97) |
| kernel - triangle | Non-parametric | 3434 (63.68) | 310 (5.75) | 211 (3.91) | 176 (3.26) | 4131 (76.6) |
| kernel - Epanechnikov | Non-parametric | 3194 (59.22) | 354 (6.56) | 209 (3.88) | 212 (3.93) | 3969 (73.59) |
| kernel - box | Non-parametric | 2975 (55.16) | 337 (6.25) | 222 (4.12) | 237 (4.39) | 3771 (69.92) |
| Burr | Parametric | 1929 (35.77) | 1079 (20.01) | 92 (1.71) | 30 (0.56) | 3130 (58.05) |
| log-logistic | Parametric | 1622 (30.08) | 1238 (22.96) | 137 (2.54) | 49 (0.91) | 3046 (56.49) |
| generalized extreme value | Parametric | 1230 (22.81) | 1484 (27.52) | 211 (3.91) | 72 (1.34) | 2997 (55.58) |
| lognormal | Parametric | 211 (3.91) | 1949 (36.14) | 298 (5.53) | 71 (1.32) | 2529 (46.9) |
| inverse Gaussian | Parametric | 120 (2.23) | 1343 (24.9) | 254 (4.71) | 56 (1.04) | 1773 (32.88) |
| generalized Pareto | Parametric | 29 (0.54) | 106 (1.97) | 93 (1.72) | 9 (0.17) | 237 (4.4) |
| normal | Parametric | 3 (0.06) | 606 (11.24) | 1100 (20.4) | 571 (10.59) | 2280 (42.29) |
| triangular - peak as mean | Non-parametric | 0 (0) | 46 (0.85) | 189 (3.5) | 185 (3.43) | 420 (7.78) |
| triangular - peak as median | Non-parametric | 0 (0) | 56 (1.04) | 195 (3.62) | 194 (3.6) | 445 (8.26) |

**Table 5**
Distribution of skewness and kurtosis values for web services in WS-REAL.

| Skewness | | | | Kurtosis | | |
|---|---|---|---|---|---|---|
| <0 | [0, 1) | [1, 5) | ≥5 | [1, 3) | [3, 10) | ≥10 |
| 0 | 0 | 64 | 1 | 0 | 6 | 59 |

**Table 6**
Weighted ranks of parametric distributions for WS-REAL, sorted by BIC value.

| Distribution | BIC | AIC | NlogL |
|---|---|---|---|
| generalized extreme value | 17.08 | 17.08 | 17.08 |
| burr | 16.78 | 16.82 | 16.86 |
| inverse Gaussian | 15.2 | 15.2 | 15.2 |
| log-logistic | 14.66 | 14.63 | 14.58 |
| lognormal | 14.55 | 14.55 | 14.55 |
| BirnbaumSaunders | 13.46 | 13.45 | 13.45 |
| generalized pareto | 11.12 | 11.14 | 11.14 |
| gamma | 10.97 | 10.98 | 11.02 |
| tlocationscale | 10.42 | 10.43 | 10.43 |
| weibull | 9.51 | 9.52 | 9.57 |
| Nakagami | 7.52 | 7.63 | 7.74 |
| logistic | 6.51 | 6.54 | 6.55 |
| exponential | 5.91 | 5.89 | 5.83 |
| Rayleigh | 5.88 | 5.71 | 5.57 |
| Rician | 4.57 | 4.57 | 4.57 |
| normal | 3.74 | 3.74 | 3.74 |
| extreme value | 2.12 | 2.12 | 2.12 |
| beta | 0 | 0 | 0 |

**Table 7**
Percentage contribution by count of top three parametric distributions as per weighted ranks for WS-REAL.

| BIC | AIC | NlogL | Mean |
|---|---|---|---|
| 95.38 (62) | 95.38 (62) | 95.38 (62) | 95.38 (62) |

kurtosis values for the response time of the web services. As observed in the case of WS-DREAM, most of the response time exhibit heavy tails and right skewness, indicating the presence of large values.

**Selecting preferred distributions**: Table 6 shows the weighted rank of each distribution for all BIC, AIC, and NlogN measures. Based on the weighted rank, the top three parametric distributions are Burr, generalized extreme value, and inverse Gaussian. As shown in Table 7, 95.38% of web services can be modeled by one of these three distributions.

**Estimating distribution parameters/evaluating goodness of fit**: The response time data of 13 web services is modeled independently for each day using the preferred distributions viz. Burr, generalized extreme value and inverse Gaussian, normal, kernel, piecewise linear, triangle, and empirical. The given distribution is fitted on the complete data and three different subsets by setting the discard percentage r in the algorithm to 0.1, 0.2, and 0.3. Table 8 summarizes the result obtained from modeling web services of WS-REAL.

**Non-parametric distributions**: In 100% (65) web services, statistical significance was absent when piecewise linear distribution with a group size of two was used to model the data. Increasing the group size to 10, we could fit 96.92% (63) of web services. With kernel distribution, the absence of statistical significance was observed in 53.85% (35) of web services when using normal smoothing function. Replacing smoothing function with triangle, 52.31% (34) of web services passed the goodness of fit test. The percentage of web services corresponding to normal and triangle smoothing function, passing the test improved by 15.38% (10) and 12.31% (8) respectively on discarding up to top 10% of the values.

**Parametric distributions**: The absence of statistical significance was found in 29.30% (19) of web services when using burr distribution to model the data. The percentage of web services in which statistical significance was absent dropped to 26.15% (17), and 9.23% (6) on using generalized extreme value and log-logistic distribution. If we remove up to top 10% of the values, the percentage of web services corresponding to burr, generalized extreme value, and log-logistic increased by 9.23% (6), 7.69% (5) and 6.15% (4) respectively. The distributions like lognormal and normal could model only 1.54% (1) and 0% (0) of web services.

The above analysis is made use of in the subsequent section to demonstrate the application of the probability distribution models in establishing the cycle time SLA.

### 4.2. Simulation model runs

To find answers to questions Q1 to Q4 of Section 1.1 and Research Objective O2 specified in Section 1.2, we carry out runs on the *FindProcessCycleTime* model described in Section 3.2. The runs cover the Registration stage because the data in WS-REAL used as input for modeling STT comprises of web services only from that stage. The response time data of each web service is independent of the weekday because of the near-constant daily load on the system. The applicants have to take an appointment before visiting, resulting in no variability in the daily load. The same was further confirmed by looking at the overlap of the CDFs of a web service

**Table 8**
Count (percentage) of datasets of WS-REAL in which $p > 0.05$ considering X% of data.

| Distribution | Type | 100% | [90%, 100%) | [80%, 90%) | [70%, 80%) | Total |
|---|---|---|---|---|---|---|
| empirical | Non-parametric | 65 (100) | 0 | 0 | 0 | 65 (100) |
| piecewiselinear (group size 2) | Non-parametric | 65 (100) | 0 | 0 | 0 | 65 (100) |
| piecewiselinear (group size 5) | Non-parametric | 64 (98.46) | 0 | 0 | 0 | 64 (98.46) |
| piecewiselinear (group size 10) | Non-parametric | 63 (96.92) | 0 | 0 | 0 | 63 (96.92) |
| kernel - normal | Non-parametric | 35 (53.85) | 10 (15.38) | 10 (15.38) | 5 (7.69) | 60 (92.3) |
| kernel - triangle | Non-parametric | 34 (52.31) | 8 (12.31) | 12 (18.46) | 6 (9.23) | 60 (92.31) |
| kernel - Epanechnikov | Non-parametric | 33 (50.77) | 7 (10.77) | 13 (20) | 7 (10.77) | 60 (92.31) |
| kernel - box | Non-parametric | 32 (49.23) | 7 (10.77) | 14 (21.54) | 6 (9.23) | 59 (90.77) |
| Burr | Parametric | 19 (29.23) | 6 (9.23) | 6 (9.23) | 5 (7.69) | 36 (38.46) |
| generalized extreme value | Parametric | 17 (26.15) | 5 (7.69) | 7 (10.77) | 6 (9.23) | 35 (53.84) |
| log-logistic | Parametric | 6 (9.23) | 4 (6.15) | 2 (3.08) | 5 (7.69) | 17 (26.15) |
| lognormal | Parametric | 1 (1.54) | 13 (20) | 4 (6.15) | 8 (12.31) | 26 (40) |
| inverse Gaussian | Parametric | 0 (0) | 14 (21.54) | 0 (0) | 8 (12.31) | 22 (33.85) |
| triangular - peak as mean | Non-parametric | 0 (0) | 0 (0) | 1 (1.54) | 0 (0) | 1 (1.54) |
| triangular - peak as median | Non-parametric | 0 (0) | 0 (0) | 2 (3.08) | 0 (0) | 2 (3.08) |
| normal | Parametric | 0 (0) | 0 (0) | 5 (7.69) | 3 (4.62) | 8 (12.31) |

**Table 9**
UTT and STT statistics (in secs) for registration stage.

| Web service | | $UTT_{i,CSA}$ | $STT_{i,CSA}$ using empirical distribution | | | | $STT_{i,CSA}$ using kernel distribution | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| No | Count | | Mean | Median | 90th percentile | 99th percentile | Mean | Median | 90th percentile | 99th percentile |
| 1 | 1 | 15 | 0.595 | 0.432 | 1 | 2.217 | 0.595 | 0.438 | 1.001 | 2.214 |
| 2 | 1 | 45 | 0.299 | 0.088 | 1.353 | 2.189 | 0.299 | 0.089 | 1.353 | 2.188 |
| 3 | 1 | 15 | 0.515 | 0.216 | 1.631 | 3.256 | 0.515 | 0.217 | 1.632 | 3.253 |
| 4 | 3 | 45 | 3.574 | 3.043 | 5.826 | 11.564 | 3.574 | 3.063 | 5.836 | 11.612 |
| 5 | 1 | 30 | 0.923 | 0.472 | 2.016 | 3.851 | 0.923 | 0.479 | 2.024 | 3.854 |
| 6 | 1 | 15 | 0.929 | 0.764 | 1.521 | 3.698 | 0.929 | 0.765 | 1.523 | 3.675 |
| 7 | 1 | 30 | 2.613 | 2.188 | 4.224 | 8.933 | 2.613 | 2.194 | 4.222 | 8.968 |
| 8 | 1 | 30 | 0.47 | 0.241 | 1.519 | 3.002 | 0.47 | 0.242 | 1.518 | 3.003 |
| 9 | 1 | 45 | 0.647 | 0.517 | 1.254 | 3.237 | 0.647 | 0.515 | 1.263 | 3.233 |
| 10 | 1 | 30 | 2.565 | 2.077 | 4.157 | 9.487 | 2.565 | 2.093 | 4.164 | 9.504 |
| 11 | 3 | 30 | 2.539 | 2.024 | 4.473 | 9.997 | 2.539 | 2.03 | 4.485 | 9.983 |
| 12 | 4 | 45 | 3.532 | 2.927 | 5.879 | 12.75 | 3.532 | 2.937 | 5.903 | 12.742 |
| 13 | 4 | 30 | 2.298 | 1.737 | 4.407 | 10.021 | 2.298 | 1.754 | 4.412 | 10.086 |

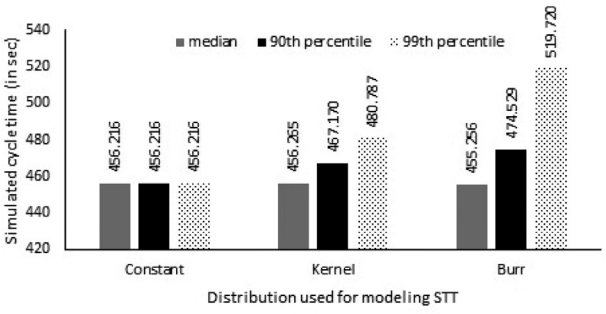
corresponding to different working days. To have a large set of data for modeling each web service, we consolidate the response time of each web service for the different working days into a single dataset. In this process, no value gets discarded as an outlier. Response time models are derived for each of the 13 web services of WS-REAL using the non-parametric kernel probability distributions, found to be the most suitable as part of the analysis described in Section 4.1.2. On modeling with kernel distribution using the normal smoothing function, four datasets out of 13 showed an absence of statistical significance with the KS test. For the remaining nine datasets, we assessed the quality of fit by plotting the empirical CDFs of the data and the corresponding distribution. In all the cases, the empirical CDF closely overlaps the CDF of the kernel distribution. The difference in the KS statistic and critical value ranges from 0.001 to 0.040. The mean, median, 90th and 99th percentile value calculated for the empirical and kernel distribution of each web service is shown in Table 9. The mean, median, and 90th and 99th percentile values of the kernel distribution is very close to that of the empirical distribution confirming our assessment on the quality of fit. The inputs used in the simulation are taken from the actual environment and summarized below.

- $N_{CSA}$ : 22, 20, 18, 16, 15 or 14 service counters
- $Q_{CSA}$ : FIFO
- $C_{CSA}$ : unlimited, as applicants visit is controlled by an appointment system and there is no variable traffic (*)
- $U_{CSA}$ : 13 user tasks, as per 3rd column of Table 9
- $S_{CSA}$ : 23 service tasks, as per 2nd column of Table 9
- $p$ : 0, as an insignificant number of applicants are returned from CSA, VA, AA stage or require rework
- R : one applicant every 30 s
- $UTT_{i,s}$ : as per $UTT_{i,CSA}$ column of Table 9
- $STT_{i,s}$ : as per $STT_{i,CSA}$ column of Table 9

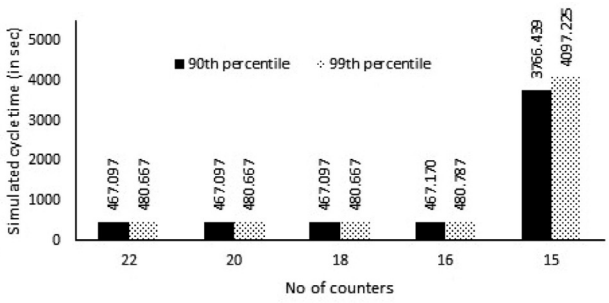
The average of each human task time is the only statistics available with us. The simulation is run for 75 hours, corresponding to 10 working days, with each day comprising of 7.5 business hours. For each applicant, the simulation program generates STT using constant, Burr, and kernel distributions and UTT using constant distribution. The STT and UTT are used in Eq. (2) to calculate the cycle time of the Registration stage.

Fig. 5 (a) shows descriptive statistics for cycle time with constant, kernel, and Burr distribution used for modeling STT in a 16 service counter setup. The median cycle time remains similar for all the three distributions (455.256 to 456.265 s), but for 90th and 99th percentile, using a single point STT like mean underestimates cycle time (456.216 s) and using STT produced from Burr overestimates cycle time (474.529 and 519.720 s) in comparison to using STT generated from kernel distribution (467.170 and 480.787 s). The result demonstrates that the choice of probability distribution impacts the accuracy of the simulated cycle time, thus addressing research objective O2. We state the realistic cycle time of the Registration stage based on the available historical data as “the maximum Registration stage cycle time in 90%, or more cases will be within 468 s, and in 99% or more cases it will be within 481 s”. We use the ceiling function on the simulated cycle time statistics to ensure that the final values are integers. The above answers question Q1.

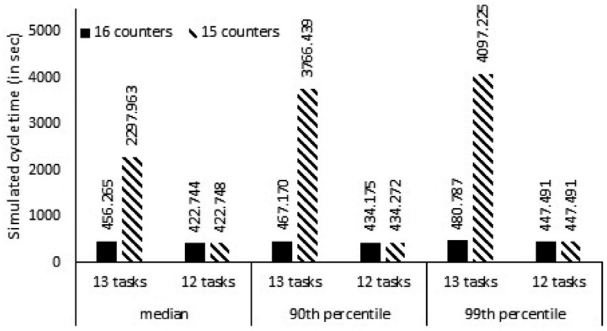
Fig. 5 (b) shows the 90th and 99th percentile cycle time with kernel distribution used for modeling STT in a 15, 16, 18, 20 and 22 service counter setup. The cycle time does not get influenced if the number of service counters is reduced from 22 to 16, with




(a) Using different distributions for modeling STT in 16 counter setup




(b) Using kernel distribution for modeling STT in 15 to 22 counter setup




(c) Using kernel distribution for modeling STT for re-engineered Registration stage

**Fig. 5.** Simulated cycle time statistics for Registration stage.

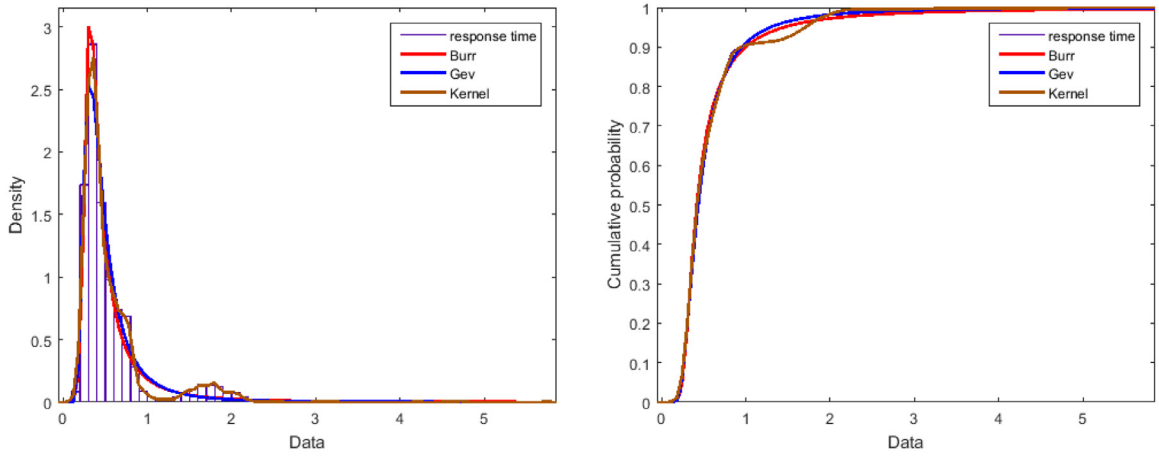
the 90th and 99th percentile remaining unchanged at 468 s and 481 s respectively. There is a steep increase in cycle time when the number of counters is reduced further to 15. Thus, the minimum number of counters required to adhere to the Registration stage cycle time is 16. This finding will result in cost optimization for the department without compromising the applicant experience at the facilitation center. The above findings answer questions Q2 and Q3.

To improve the cycle time further and optimize operational costs, the department carries out a detailed business process re-engineering exercise. One of the recommendations of the exercise is to provide user task 10 as part of the online application submission process, reducing the number of user and service tasks in the Registration stage to 12 and 20, respectively. Fig. 5(c) shows the 90th and 99th percentile cycle time with kernel distribution used for modeling STT in a 15 and 16 service counter setup. The 90th percentile reduces from 467.170 s to 434.175 s, and the 99th percentile decreases from 480.787 s to 447.491 s). This reduction is on account of the elimination of the user and service task time for task 10. The number of counters can be further optimized to 15, keeping the cycle time unchanged. The finding answers Q4. The MATLAB model and scripts used in this section are available via GitHub.

## 5. Discussions

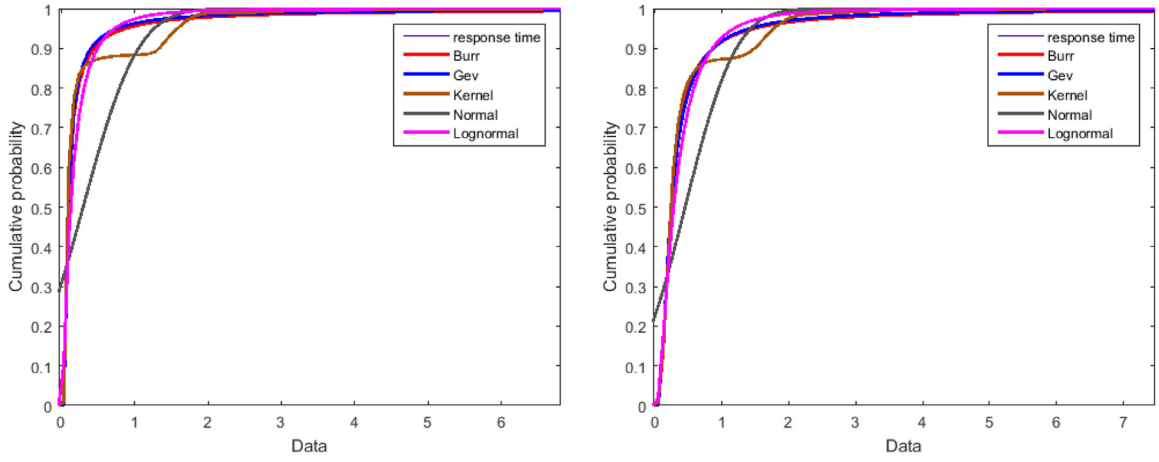
The empirical results presented in Section 4.1 establish that web service response time can be modeled reasonably well by parametric distributions like Burr, generalized extreme value, log-logistic, and non-parametric distribution like the kernel (Research Objective O1). The underlying rationale for the same is intuitive. The response time of software systems shows right side skewness and high kurtosis value because of the presence of a few high response time values. An attempt was made to understand the reasons for the occurrence of high response time values by examining the data of WS-REAL. For this, we considered the system metrics (e.g., CPU, disk), application, and verbose GC logs corresponding to the time interval when high values of web service times occurred. All the time intervals correlate with significant garbage collection by the java virtual machine of the application server or a varying filter clause in database queries (e.g., using an index field, range queries). High constant values like 120 s indicate that a time-out occurred, and the client application did not receive a response. In production systems running normally, timeouts are rare. In our data, we do not have any timeouts. The high values in web response time data are not chance occurrences or measurement errors but indicate the presence of heavy tails or a mix of probability distributions. They occur because of valid technical reasons and should not be discarded as outliers while developing distribution models. Burr distribution with its three parameters is flexible enough to cover a wide range of kurtosis and skewness values, which is necessary for modeling such data. The generalized extreme value and log-logistic distribution were also found suitable for modeling data having high values.

Further, the distribution of web service response time may not always be unimodal. Fig. 6 shows the histograms, PDF, and CDF for the data in dataset 26 from WS-REAL. The histogram in Fig. 6a has two distinct peaks or summits. The second peak appears as a kink in Fig. 6b. Such multi-modal distribution indicates the presence of different groups. The parametric distributions are unable to model such features in comparison to non-parametric distributions. Modeling such data using a mix of parametric or non-parametric distributions is possible but brings in additional complexity. The kernel distribution with normal or triangle smoothing function has the potential to model such complex web service response data very well. The reason for the same is that the kernel generates the




(a) Histogram of dataset 11 from WS-REAL (b) Cdf of dataset 11 from WS-REAL

**Fig. 6.** Parametric and non-parametric models for dataset 11 from WS-REAL.




(a) Dataset 16 from WS-REAL (b) Dataset 51 from WS-REAL

**Fig. 7.** Cdf of fitted kernel distribution for the data overlaid on its empirical distribution to show visual fit.

PDF by creating a distinct probability density curve for each data and then combines all such curves, resulting in a continuous and smooth PDF for the given data. Such PDF can model data features like the presence of multiple peaks or modes, which may not be possible with parametric distributions. Empirical cumulative distribution and its variant piecewise linear distribution at group sizes of 2, 5, and 10 also result in a perfect fit. However, we do not prefer its use in modeling the web service response time due to certain limitations (Law and Kelton, 1997). Empirical cumulative distribution based on small sample sizes may have irregularities. It cannot also generate a value outside of the lowest and highest observed value. Piecewise linear distributions are only a continuous variant of empirical cumulative distribution.

To validate the above discussion, we superimpose the PDFs of Burr, generalized extreme value, and kernel distributions on the histograms of Fig. 6a. Similarly, the CDFs of three distributions are overlaid on the empirical CDF of the data in Fig. 6b. The CDF of the kernel distribution closely overlaps the empirical CDF in Fig. 6b, showing kernel distribution is model complex data features found in web service response time. The KS test showed statistical significance when the data is modeled using generalized extreme value (p-value 0) and burr ($p$-value 0.026) distribution. Statistical significance is absent when the data is modeled using kernel distribution with normal smoothing function ($p$-value 0.089). The evaluation of the goodness of fit usually relies only on statistical tests like KS. However, it is essential to understand the notion of statistical significance and practical significance when we interpret the results of statistical tests. Statistical tests provide objective results but also have limitations. Statistical significance, present when $p$-value less than 0.05, is strongly influenced by sample size. With a small sample size, the KS test may not be sensitive enough, rejecting only the most abnormal behavior. With a large sample size, the KS test may be oversensitive, rejecting even minor deviations between the proposed distribution and data. In the latter case, the practical significance or effect of the magnitude of difference may be too small to have any meaningful influence in the real-world.

If the KS test reports statistical significance, we need to let practitioners decide if the results are meaningful in the given context. The decision can be made using heuristic techniques like density/histogram over plots, distribution function difference plot (Law and Kelton, 1997). The use of such techniques helps us to avoid oversensitivity. To illustrate this further, we studied the complete data and chose datasets 16 and 51 from WS-REAL. Fig. 7 shows the CDF of the Burr, generalized extreme value, kernel, normal, and lognormal distribution estimated from the data overlaid on the empirical CDF in both cases. The CDFs of lognormal and normal distributions are included to avoid any bias. The dataset 16 and 51 contain 3371 values and 13,119 values, respec-

tively. Both datasets show statistical significance in using kernel distribution with a normal smoothing function to model the data. The calculated KS statistic 0.0295 is more than the critical value of 0.0233 for dataset 16. Likewise, the calculated KS statistic 0.0131 is more than the critical value 0.0118 for dataset 51. The difference of just 0.0062 and 0.0013 between the respective KS statistic and the critical value is not very large. Further, the CDFs of kernel distribution completely overlap the data showing a practically acceptable fit. The existence of statistical significance can be rejected in both these cases. In such cases, minor deviations also tend to assume statistical significance with an increase in sample size. We also compared the percentage of web services in WS-DREAM and WS-REAL, which did not show statistical significance when modeled using piecewise linear, kernel, Burr, generalized extreme value, log-logistic, lognormal and lognormal distributions. The order of the distributions in terms of percentage remained the same in both cases except log-logistic, which was able to model more number of web services than generalized extreme value distribution in WS-DREAM.

Web service response time is an essential input parameter for calculation of cycle time SLA. For determining the realistic value of cycle time, it is required to carry out simulation using predictor variables, namely web service response time and human task time. Since these times have some degree of variability associated with them, they cannot be represented using single-point estimates like mean or median. The motivation behind the use of simple distributions like normal and lognormal is because of its simplicity and the fact that normally distributed data is a pre-requisite for statistical techniques like hypothesis testing or estimating a value. Although Central Limit Theorem assures normally distributed sample means if sample sizes are large, the sample size required for applying this theorem may be challenging to achieve (Chen et al., 2015). The probability distributions can describe the behavior of such uncertain predictor variables. The selection of appropriate probability distributions impacts the accuracy of simulation outcomes. As seen in Fig. 5(a), the cycle time statistics produced on using STT generated from kernel distribution is more realistic when compared to using a constant STT or STT from Burr distribution (Research Objective O2).

## 6. Threats to validity

### 6.1. Internal validity

This study required several statistical and data handling functions for modeling the response time. To reduce the internal validity threat, we used methods available in a standard platform like MATLAB.

### 6.2. External validity

We used one publicly available and widely cited dataset containing response time data of real-world web services and one real-world software system in the industry to reduce risk arising from generalizing the findings. We also have made the MATLAB scripts developed as part of this study publicly available so that researchers and practitioners can replicate the work on their work data.

## 7. Conclusion

In this work, we compared and evaluated the effectiveness of different parametric and non-parametric probability distributions for modeling data of web service response time. The work is a comprehensive empirical study in this area as there are minimal studies on such large publicly available datasets. The study used two sets of data - one is publicly available WS-DREAM QoS dataset widely used by researchers, and another is web service response time from a real-world software system used to service a large number of users. Statistical significance was absent in 66.75% of web services from WS-DREAM when response time was modeled using kernel distribution. Our results show non-parametric kernel distribution is the most suitable choice, with parametric Burr coming second. Kernel distribution outperforms Burr by 86.63%. Based on our empirical results, we strongly recommend the addition of kernel and Burr distributions to the current list of distributions supported as part of BPSim (Gagne, 2016).

In the real-world software system, statistical significance was absent in 53.85% of web services when response time was modeled using kernel distribution. In this case, kernel distribution again outperforms Burr by 84.21%. However, on using techniques like density/histogram over plots and distribution function different plot, 100% of web services could be modeled using kernel distribution. From this, we can infer that statistical significance by itself might not be practically relevant in the given context.

Our study indicates that the kernel distribution of non-parametric family has the potential for modeling web service response time with high accuracy. The work also establishes the importance of using an appropriate probability distribution to model the STT used as input in a cycle time simulation model. The suitability of probability distribution determines the accuracy of the simulated cycle time.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.jss.2019.110480.

## References


Abello, A., Oscar, R., 2013. Business Process Management Quantitative.

Akaike, H., 1974. A new look at the statistical model identification. IEEE Trans. Autom. Control 19 (6), 716–723. doi:10.1109/TAC.1974.1100705.

Amannejad, Y., Krishnamurthy, D., Far, B., 2016. Predicting web service response time percentiles. In: 2016 12th International Conference on Network and Service Management (CNSM), pp. 73–81. doi:10.1109/CNSM.2016.7818402.

Appian BPM Platform Suite, https://www.appian.com/platform/bpm-suite/.

ARIS Process Management and Transformation Platform, www.softwareag.com/bpa.

ARIS Simulation - Analyse and improve processes http://www.softwareag.com/corporate/images/SAG_ARIS_Simulation_FS_Feb16_web_tcm16-78560.pdf.

Barber, S., 2003. Designing performance tests to predict user experience. Online; accessed 24 September 2016.

Biller, B., Gunes, C., 2010. Introduction to simulation input modeling. In: Proceedings of the Winter Simulation Conference. Winter Simulation Conference, pp. 49–58.

Bizagi - The Digital Business Platform https://www.bizagi.com/en/products.

Bizagi 11.1 BPM Suite User Guide Bizagi 11.1 BPM Suite User Guide - A Digital Business Platform. http://help.bizagi.com/bpm-suite/en/index.html?simulation_levels.htm.

Brady, J.F., 2012. When load testing large user population web applications the devil is in the (virtual) user details. In: Int. CMG Conference.

Burr, I.W., 1942. Cumulative frequency functions. Ann. Math. Stat. 13 (2), 215–232.

Business Process Model and Notation Specification Version 2.0, 2014. Technical Report formal/2013-12-09.

Cardoso, J., Sheth, A., Miller, J., Arnold, J., Kochut, K., 2004. Quality of service for workflows and web service processes. Web Semantics: Science, Services and Agents on the World Wide Web 1 (3).

Chakravarti, I.M., Laha, R.G., Roy, J., 1967. Handbook of Methods of Applied Statistics, Vol. I. John Wiley and Sons, USE.

Chen, T., Guo, Q., Temam, O., Wu, Y., Bao, Y., Xu, Z., Chen, Y., 2015. Statistical performance comparisons of computers. IEEE Trans. Comput. 64 (5), 1442–1455. doi:10.1109/TC.2014.2315614.

Choi, B.K., Kang, D., 2013. Modeling and Simulation of Discrete Event Systems, 1st ed Wiley Publishing.

Dey, A.K., Kundu, D., 2009. Discriminating between the log-normal and log-logistic distributions. Commun. Stat. - Theory Methods 39 (2), 280–292. doi:10.1080/03610920902737100.
Dilley, J.A., 1996. Web Server Workload Characterization. HP Labs Technical Reports HPL-96-160.
Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A., 2018. Quantitative Process Analysis. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 255–296. doi:10.1007/978-3-662-56509-4_7.
Dunie, R., Baker, V., Wong, J., Kerremans, M., 2018. Critical Capabilities for Intelligent Business Process Management Suites. Technical Report G00315744.
Dunne, J., Kristiansen, M., Darisa, A., Timmins, A. J., Saxena, P., 2012. Measuring the Distribution of Workload Transaction Response Times. http://public.dhe.ibm.com/software/dw/lotus/DistributionSkewWorkloadTransactionResponseTimesFinal.pdf.
Fit probability distribution object to data - fitdist. http://in.mathworks.com/help/stats/fitdist.html.
Gagne, D., 2016. Business Process Simulation Specification 2.0. Technical Report WFMC-BPSWG-2016-1.
Geebelen, D., Geebelen, K., Truyen, E., Michiels, S., Suykens, J.A.K., Vandewalle, J., Joosen, W., 2014. Qos prediction for web service compositions using kernel-based quantile estimation with online adaptation of the constant offset. Inf. Sci. 268, 397–424. doi:10.1016/j.ins.2013.12.063.
Hughes, C., Hillman, J., 2006. Qos explorer: a tool for exploring qos in composed services. In: 2006 IEEE International Conference on Web Services (ICWS'06), pp. 797–806. doi:10.1109/ICWS.2006.108.
Hwang, S.-Y., Wang, H., Tang, J., Srivastava, J., 2007. A probabilistic approach to modeling and estimating the qos of web-services-based workflows. Inf. Sci. 177 (23), 5484–5503. doi:10.1016/j.ins.2007.07.011.
IBM BPM - Setting up simulation profiles https://www.ibm.com/support/knowledgecenter/SSFTN5_8.5.6/com.ibm.wbpm.wle.admin.doc/topics/set_up_sim_profiles.html.
IBM Business Process Manager Developer Center https://developer.ibm.com/bpm/.
Jansen-Vullers, M., Netjes, M., 2006. Business process simulation - tool survey. In: Jensen, K. (Ed.), The Seventh Workshop on the Practical Use of Coloured Petri Nets and CPN Tools. University of Aarhus, pp. 77–96.
Jin, L.-j., Machiraju, V., Sahai, A., 2002. Analysis on service level agreement of web services. Technical Report HPL-2002-180.
JSim - Java based simulation system. https://www.physiome.org/jsim/.
Keller, A., Ludwig, H., 2003. The wsla framework: specifying and monitoring service level agreements for web services. J. Network Syst. Manage. 11 (1), 57–81. doi:10.1023/A:1022445108617.
Klein, A., Ishikawa, F., Bauer, B., 2009. A Probabilistic Approach to Service Selection with Conditional Contracts and Usage Patterns. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 253–268.
L-Sim BPMN simulation engine https://www.lanner.com/en-us/technology/l-sim-bpmn-simulation-engine.html.
Law, A.M., 2013. A tutorial on how to select simulation input probability distributions. In: 2013 Winter Simulations Conference (WSC), pp. 306–320. doi:10.1109/WSC.2013.6721429.
Law, A.M., Kelton, W.D., 1997. Simulation Modeling and Analysis, 2nd ed McGraw-Hill Higher Education.
Leemis, L., 2003. Input modeling. In: Proceedings of the 35th Conference on Winter Simulation: Driving Innovation. Winter Simulation Conference, pp. 14–24.
Loosley, C., 2000. e-commerce response time: A reference model. In: 26th International Computer Measurement Group Conference, December 10–15, 2000, Orlando, FL, USA, Proceedings.
Mansharamani, R., Khanapurkar, A., Mathew, B., Subramanyan, R., 2010. Performance testing: far from steady state. In: 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops, pp. 341–346. doi:10.1109/COMPSACW.2010.66.
Markose, S., Alentorn, A., 2005. Option pricing and the implied tail index with the generalized extreme value (GEV) distribution. Computing in Economics and Finance 2005 397. Society for Computational Economics.
MathWorks Statistics and Machine Learning Toolbox https://in.mathworks.com/help/stats/.
Matlab scripts used in this paper https://github.com/raghuramakrishnan71/wsdistributionfitting.
Neath, A.A., Cavanaugh, J.E., 2012. The bayesian information criterion: background, derivation, and applications. Wiley Interdisc. Rev. 4 (2), 199–203. doi:10.1002/wics.199.
Oracle BPM - Business Process Management. http://www.oracle.com/us/technologies/bpm.
Pega Digital Transformation Suite https://www.pega.com/products/pega-platform/case-management.
Red Hat JBoss BPM Suite - Distribution Types. https://access.redhat.com/documentation/en-us/red_hat_jboss_bpm_suite/6.0/html/user_guide/sect-distribution_types.
Red Hat Process Automation Manager https://www.redhat.com/en/technologies/jboss-middleware/process-automation-manager.
Rosario, S., Benveniste, A., Haar, S., Jard, C., 2008. Probabilistic qos and soft contracts for transaction-based web services orchestrations. IEEE Trans. Serv. Comput. 1 (4), 187–200. doi:10.1109/TSC.2008.17.
Running Simulations in Oracle BPM https://docs.oracle.com/middleware/1221/bpm/bpm-develop/GUID-71B99537-03B9-47F6-9EB6-DBFDCD691A99.htm#BPMPD1619.
Satyanarayana, J., 2012. Managing Transformation: Objectives to Outcomes. Eastern Economy Edition. PHI Learning.
Scheid, S., 2004. Introduction to Kernel Smoothing.
Schwarz, G., 1978. Estimating the dimension of a model. Ann. Stat. 6 (2), 461–464. doi:10.2307/2958889.
Scott, S., Hoffman, K., See, K., 2015. Public facing government services a business ecosystem perspective. In: 2015 Annual IEEE Systems Conference (SysCon) Proceedings, pp. 798–801. doi:10.1109/SYSCON.2015.7116848.
Sheppard, M., Fit all valid parametric probability distributions to data. https://www.mathworks.com/matlabcentral/fileexchange/34943-fit-all-valid-parametric-probability-distributions-to-data.
Signavio -Managing simulation scenarios. https://docs.signavio.com/userguide/editor/en/modeling_and_notations/bpmn/managing_bpmn_simulation.html.
Signavio Business Transformation Suite https://www.signavio.com/products/business-transformation-suite/.
Simulink - Simulation and Model'Based Design. https://www.physiome.org/jsim/.
Spinner, S., Casale, G., Brosig, F., Kounev, S., 2015. Evaluating approaches to resource demand estimation. Perform. Eval. 92, 51–71. doi:10.1016/j.peva.2015.07.005.
Subraya, B., 2006. Integrated Approach to Web Performance Testing: A Practitioner's Guide: A Practitioner's Guide. IGI Global research collection. IGI Global.
Tadikamalla, P.R., 1980. A look at the burr and related distributions. Int. Stat. Rev./Revue Int. Stat. 48 (3), 337–344.
Tibco business studio simulation guide, 2016. https://docs.tibco.com/pub/amx-bpm/4.1.0/doc/html/bpmhelp/GUID-175C7A82-E71A-44CB-9FC4-7D4F24CD804C.html.
TIBCO BPM, https://www.tibco.com/products/business-process-management.
Visual Paradigm, - How to Use Business Process Simulation? https://www.visual-paradigm.com/tutorials/process-simulation.jsp.
Wand, M.P., Jones, M.C., 1995. Kernel Smoothing. Monographs on Statistics and Applied Probability, Vol. 43. Chapman & Hall.
What is jBPM?, https://www.jbpm.org/.
Wilson, T., 2011. What were they thinking: Modeling think times for performance testing. CMG MeasureIT.
Ws-dream web service qos datasets, http://inpluslab.com/wsdream/.
Zeng, L., Benatallah, B., H.H. Ngu, A., Dumas, M., Kalagnanam, J., Chang, H., 2004. Qos-aware middleware for web services composition. IEEE Trans. Softw. Eng. 30 (5), 311–327. doi:10.1109/TSE.2004.11.
Zheng, H., Yang, J., Zhao, W., 2010a. Qos probability distribution estimation for web services and service compositions. In: 2010 IEEE International Conference on Service-Oriented Computing and Applications (SOCA), pp. 1–8. doi:10.1109/SOCA.2010.5707144.
Zheng, H., Zhao, W., Yang, J., Bouguettaya, A., 2013. Qos analysis for web service compositions with complex structures. IEEE Trans. Serv. Comput. 6 (3), 373–386. doi:10.1109/TSC.2012.7.
Zheng, Z., Zhang, Y., Lyu, M.R., 2010b. Distributed qos evaluation for real-world web services. In: 2010 IEEE International Conference on Web Services, pp. 83–90. doi:10.1109/ICWS.2010.10.
Zheng, Z., Zhang, Y., Lyu, M.R., 2014. Investigating qos of real-world web services. IEEE Trans. Serv. Comput. 7 (1), 32–39. doi:10.1109/TSC.2012.34.

**Raghu Ramakrishnan** is Chief Architect and Head, Technology in Public Sector Unit of Tata Consultancy Services. He has over 26 years of experience in designing and performance engineering business-critical software systems for government, airlines, and financial institutions. He is pursuing research at the University School of Information and Communication Technology, Guru Gobind Singh Indraprastha University, Delhi, India. He is a member of the IEEE, ACM, and Computer Society of India.

**Arvinder Kaur** is a Professor at the University School of Information and Communication Technology, Guru Gobind Singh Indraprastha University, Delhi, India. Her research interests include software engineering, software testing, software metrics, fault prediction, and project management. She has authored more than 80 research papers in International Journals and Conferences.