Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

A COMBINED SIMULATION AND MACHINE LEARNING APPROACH FOR REAL-TIME DELAY PREDICTION FOR WAITLISTED NEUROSURGERY CANDIDATES

Vaibhav Baldwa
Siddharth Sehgal
Varun Ramamohan

Department of Mechanical Engineering
Indian Institute of Technology Delhi
Hauz Khas
New Delhi, Delhi 110016, India

Vivek Tandon

Department of Neurosurgery
All India Institute of Medical Sciences
Ansari Nagar, Aurobindo Marg
New Delhi, Delhi 110029, India

ABSTRACT

In this study, we present a method to predict whether a patient seeking admission to the neurosurgery ward of a large public tertiary care hospital in north India receives admission within a prespecified duration. The prediction needs to be made at the time the patient is seeking admission at the ward, so that they can then decide whether to wait for admission into the neurosurgery ward or seek care elsewhere. We accomplish this by simulating the admission and patient stay processes at the neurosurgery ward, and use the simulation to generate data to train machine learning algorithms to predict whether the patient is admitted as a function of the state of the simulation at the time the patient is seeking admission at the ward. With ensemble tree classifiers, we achieve generalization area under the curve scores of 95% for all patients taken together and between 80-95% depending upon patient subtype.

1 INTRODUCTION

Large urban public tertiary care hospitals in India typically face substantially more demand than their available capacity. This occurs because of the perception of inconsistent quality of care at other public facilities, especially in semi-urban and rural India, leading patients to seek care at public hospitals in large urban (metropolitan) areas. For example, at large public hospitals in New Delhi (the Indian capital), the wait times for elective surgeries (e.g., tumour resections) range from six months to multiple years (Hindustan Times 2017; Times of India 2017). Given these wait times, and the lack of affordability of more expensive private hospitals for a majority of Indian patients (Sriram 2018), patients often elect to wait for admission to these public hospitals. Therefore, a first step towards alleviating this situation would involve providing these patients, at the time they present seeking admission, with information regarding whether they will be admitted within a certain time duration (e.g., 60 days) so that they can make an informed decision regarding whether to wait or seek care elsewhere. Further, providing real-time information about waiting times appears to improve customer experience (Hui and Tse 1996; Ibrahim 2018; Hu et al. 2018). In this paper, we consider estimation of admissions outcomes for patients seeking admission at the neurosurgery ward (for conduct of their surgeries) of a prominent public hospital in India.

The aim of this paper is to present a method, using a combination of discrete-event simulation and machine learning, to predict the following: (a) whether a patient will be admitted into the neurosurgery ward within a prespecified duration, and (b) for patients predicted to receive admission within this prespecified duration, the delay experienced by the patient before admission. These predictions are made at the time a patient presents at the neurosurgery ward seeking admission, as a function of the state of the ward at this time. Such real-time delay predictions at the point at which the patient arrives in the system are typically done

using analytical delay predictors for mathematically tractable queueing systems, or by adoping data-driven approaches wherein data logs for the queueing system are used with machine learning methods to generate the delay predictions (Ibrahim 2018). However, in the case of the neurosurgery ward under consideration, the admissions process and hence the queueing discipline is complex, and the admissions data maintained by the hospital administration does not capture sufficient information (e.g., state of the ward at the time of arrival of each patient seeking admission) to generate accurate delay predictions. Hence, we develop the approach below for generating these delay predictions:

1 Develop a discrete-event simulation (DES) of the admission and patient stay processes at the neurosurgery ward of a prominent Indian public hospital.
2 The DES is then used to record for each of, say, N patients in steady state (e.g., the j^{th} patient) the following:
   2.1 The state of the simulation (our definition of the system state is provided in section 3), and hence that of the neurosurgery ward, at the time they arrive in the system (denoted by S_j, $S_j \in \mathbb{R}^d$),
   2.2 Whether said patient is admitted within a prespecified duration T or not as a binary variable (i.e., $L_j = 1$ if patient j is admitted and 0 otherwise), and for admitted patients, the wait time before admission ($W_{j(ad)} \in \mathbb{R}$).
3 Construct training sets (S, L) using data recorded for all N patients (2.1 above) and (S_{ad}, W_{ad}) using data recorded for admitted patients (2.2 above);
4 Train (and validate) machine learning methods f and f_{ad} on (S, L) and (S_{ad}, W_{ad}), respectively;
5 Using the f and f_{ad} to generate predictions of L_j and $W_{j(ad)}$ respectively for newly arriving patients. For example, for a new patient (patient k) presenting at the neurosurgery ward seeking admission at time t, the state of the neurosurgery ward $S_{k(t)}$ at time t can be recorded, and predictions for whether this patient is admitted within T time units are generated as $\hat{L}_k = f(S_{k(t)})$. If $\hat{L}_k = 1$, then the wait time for this patient before admission can be estimated as $\hat{W}_{k(ad)} = f_{ad}(S_{k(t)})$.

Note that for most machine learning algorithms (e.g., decision tree based classifiers), $Pr(\hat{L}_j = 1)$ can be estimated easily by accessing classifier specific methods in their implementations.

The above method can be particularly useful in situations where a DES of the system may be developed or already available for the separate purpose of analysis of the system operations regardless of the delay prediction objective. Note that the DES itself can be used to determine the distribution of waiting times for patients (say, $F(t)$), from which the probability of the waiting time exceeding a certain threshold can be estimated. However, $F(t)$ does not provide delay information specific to a patient arriving at a given time, and non-specific information regarding waiting times provided to a patient who may have to wait longer than the announced delay time has been shown to significantly worsen both patient and provider experiences (Moriah et al. 2011). Further, it has also been shown in previous work that delay predictors that use system state information, such as queue length based predictors, typically outperform predictors utilizing only queue history based predictors (Whitt 1999b; Armony and Maglaras 2004).

Previous work on real-time delay prediction has either involved approaches grounded in queuing theory, or more recently in empirical approaches using actual delay data from service systems (Ibrahim 2018). Our work also adopts an empirical machine learning approach towards delay prediction; however, the data generated for the machine learning approach is generated by a (simulated) representation of the neurosurgery queueing system. Our literature survey (section 2) did not yield a study that utilized a DES of the service system to generate the system state data required for a machine learning algorithm to predict delays. This can be used when the service system data does not capture all information required for adequately accurate delay prediction; in particular, for systems with a large number of servers with general service times, complex queueing disciplines (similar to the neurosurgery ward modelled in this study), it is unlikely that the system state at patient arrival (or at other times) will be completely captured by operations data. Thus, in such situations, our study provides a demonstration of the method of using simulations in conjunction

with machine learning algorithms to predict whether a job will receive service or not and also the amount of time the job would have to wait prior to service.

2 Literature Review

Discrete-event simulations have been applied extensively to model healthcare delivery operations, and we refer readers to a comprehensive review summarizing work in this field (Zhang 2018). Here we focus on the literature in the field of delay prediction. We present key articles for delay prediction here; for a comprehensive discussion, we refer readers to a relatively recent review (Ibrahim 2018).

Approaches for the estimation of delays on a real-time basis typically are of two types: those grounded in queueing theory and models, or data-driven statistical learning approaches that make use of historical queue data for the system under consideration (Carmeli, Nitzan and Mandelbaum, Avishai and Yom-Tov, Galit 2017; Ibrahim 2018). The use of data-driven approaches is relatively recent (Balakrishna et al. 2008; Senderovich et al. 2015; Simaiakis and Balakrishnan 2016; Ang et al. 2016), with earlier approaches based on queueing models and analysis (Whitt 1999a; Nakibly 2002; Ibrahim and Whitt 2009a; Ibrahim and Whitt 2011a; Ibrahim et al. 2017). A few studies have attempted to develop a combined framework using both queueing-based estimators informed by analysis of real-world queue data. We briefly discuss both types of studies below.

Queueing theory based approaches have been used to estimate delays for a wide variety of systems, from the $M/M/s/r$ systems (Whitt 1999b) to $M(t)/GI/s(t)+GI$ systems (Ibrahim and Whitt 2011b), where the authors develop wait-time predictors for many-server systems with time-varying arrival rates and number of servers, and generally distributed abandonment times. Moderately loaded systems as well as systems operating under heavy traffic conditions (Ibrahim and Whitt 2009b) have been studied. A few studies have also focused on systems with multi-class customer arrivals (Nakibly 2002; Senderovich et al. 2015). Delay or wait-time predictors typically developed in these studies are versions of delay-history based predictors or queue-length based predictors. Delay-history based predictors, which use recent history of customer delays, include various versions of the wait-time elapsed for the customer at the head of the line (HOL), delay of the last customer to enter service (LES), etc. Queue-length based predictors use information regarding the system state, such as the number of customers in the queue at the time of arrival of the customer in question. As mentioned before, queue-length based predictors have been shown to outperform delay-history based predictors (Whitt 1999b; Armony and Maglaras 2004; Ibrahim and Whitt 2011b) as they utilize system state information in their predictions. Simulations have primarily been used in these studies to assess and verify the accuracies of analytical delay predictors.

The neurosurgery ward queueing system we simulate has 5 patient types and multiple (39) beds. Each patient type has exponential interarrival times, but generally distributed service times. We assume stationary interarrival and service times. We also assume that patients do not balk, but renege after a certain deterministic period of time that, for instance, can represent a significant deterioration in their health, depending upon the patient type. The patient types have priority ordering, and the admission algorithm we simulate is based upon the priority ordering as well as the state of the system. Finally, a subset of beds in the ward are reserved for a specific patient type. Thus the queueing system we simulate is not easily amenable to analytical modeling, and hence we develop a discrete-event simulation of the system.

We now discuss representative papers presenting data-driven approaches towards delay estimation. Balakrishna et al. (2008) develop a reinforcement learning model to predict airport takeoff delays. The model was trained using airport operational data. In a later paper, the same authors also develop a mathematical model calibrated using airport operational data for predicting queue delay (Simaiakis and Balakrishnan 2016). Senderovich et al. (2014) and Senderovich et al. (2015) mine queue log data to develop a customer transition system for predicting delay times, and enhance this approach by developing transition systems separately for system load levels identified via k-means clustering. The authors also use queue log mining to estimate both queueing history based as well as queue length based predictors of wait time, and thus their approach represents a hybrid of queueing theory as well as data-driven approaches.

Ang et al. (2016) develop an approach that combines statistical learning via lasso regression with queueing theory based fluid model estimators to predict hospital emergency department wait times. They train their statistical learning method on historical data from four hospitals. Bassamboo and Ibrahim (2020) develop a correlation-based approach to characterize the accuracy of delay announcements generated using different approaches, including a predictor trained across historical data from queueing systems.

All empirical approaches discussed above utilized historical queueing system data to train their delay predictors. On the other hand, our literature search did not identify a study that utilized a DES model of the complex queueing system under consideration to generate the system state data required to adequately train a statistical learning method for online delay prediction. We utilized this simulation approach because in addition to the neurosurgery ward being a complex queueing system, detailed data regarding system state was also not maintained by the ward. In such a situation, a DES model of the system, which may be developed to analyze the operations of the system regardless of the delay prediction objective, can be used to generate the data required for this objective. Our work provides a proof-of-concept of this approach.

3 Methodology

In section 3.1, we describe the development of the neurosurgery ward simulation, and in section 3.2, we describe the generation of system state data for training and using the machine learning methods.

3.1 Neurosurgery Ward Simulation Development

The neurosurgery (NS) department at our collaborating hospital broadly classifies patients into six types: glioma, cranial sick, spinal disease, patients to be treated as soon as their administrative processing is completed (abbreviated as ASAPC patients), routine care and day-care patients. These types are assigned in the descending order of priority assigned to them - that is, glioma patients have the highest priority for admission, followed by cranial sick patients, spinal disease patients, etc. Patients are admitted into the ward for neurological surgeries; that is, after admission, they are prepared for surgery over a certain time period, undergo the surgery, and are then discharged (or die) after a certain in-hospital recovery period.

The NS ward has approximately 40 beds. Four of these beds are usually reserved for glioma patients as the preferred operating theatre for these patients requires special equipment, and the number of glioma surgeries that are performed per week are such that a maximum of four glioma patients can be present in the ward at any point in time. Thus, as will be described later, the admission process for glioma patients forms a separate subsystem of its own - an $M/G/4+D$ system (i.e., with deterministic reneging times). We do not consider day-care patients in our analysis because as the name indicates, they are admitted and discharged in a single day, and hence day-care cases are considered to be of a significantly lower priority given the severity and high load for other case types. Thus the NS ward administration limits the number of day-care cases in a day to one, and hence we assume that one bed in the ward is reserved for these cases.

For the remaining 35 beds and four patient types, a priority-based admission algorithm is followed. Before we describe this algorithm, we note that due to clinical and administrative exigencies, this algorithm is not followed at the NS ward in an exact manner, and hence it represents an overarching admission policy than an algorithm implemented without deviation. Further, this algorithm has not been derived to be mathematically optimal in terms of admitting the maximum number of patients available, or maximizing provision of care to the most sick patients, or maximizing bed utilization. The algorithm has evolved such that a reasonable proportion of patients, as determined by the governing clinical staff, are admitted given the varying arrival rates, lengths of stay, and reneging thresholds of these patient types.

For these 35 beds, if the number of empty beds at any point in time exceeds 4, beds are allocated on a first-come first-served basis from the waitlist until the number of empty beds equals 4. At this point, the beds are allocated using the priority order: the first empty bed is allocated to a cranial sick patient, the next to a spinal disease patient and so on until a routine care patient is allocated a bed, after which the patient type to be allocated the next bed is reset to the top priority patient: a cranial sick patient. Note that

at steady state, because the arrival rate of patients is much higher than the rate at which patients leave the ward, encountering a system state with 4 empty beds is very unlikely. In such cases, the patient type to whom the bed was last allocated is recorded, and the next empty bed is allocated to the patient type with the next (lower) priority level. Further, we assume that if an empty bed is available, but the appropriate patient type is not present in the waitlist, then the empty bed is held unoccupied until said patient type arrives. While this may seem unrealistic, we make this assumption because in practice this algorithm may not be applied for all beds, and hence it is likely that at all times, a small number beds may be unavailable for admitting waitlisted patients via this algorithm. The admission algorithm is depicted in Figure 1 below.

Let priority level of patient type currently being considered for allocation = p. Let patient type with top priority level be 0, and bottom-most priority level be P.

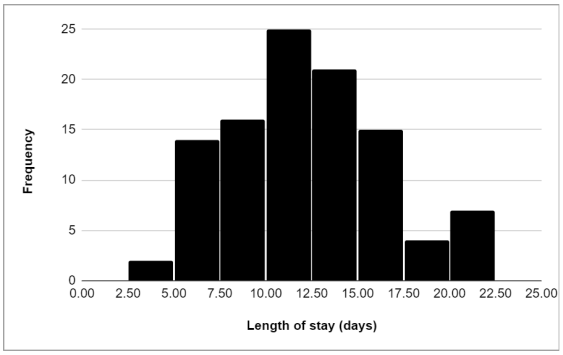
1. If N beds become empty:
    a. If $0 < N \leq 4$:
        i. While $N > 0$:
            1. Check whether waitlist is empty. If empty:
                a. Hold until arrival of next patient.
                b. Check whether priority of next arriving patient = p.
                    i. If yes:
                        1. Allocate bed. Set priority level = p + 1 (if p + 1 = P: set priority level = 0) and N = N – 1.
                        2. Go to **(1.a.i)**.
                    ii. If no:
                        1. Add patient to waitlist.
                        2. Go to **(1.a.i.2)**.
            2. If waitlist is not empty:
                a. Check whether patient type with priority p exists in waitlist.
                    i. If yes:
                        1. Allocate bed. Set priority level = p + 1 (if p + 1 = P: set priority level = 0) and N = N – 1.
                        2. Go to **(1.a.i).**
                    ii. If no:
                        1. Hold until arrival of next patient.
                        2. Repeat process **(1.a.i.1.b)**.
    b. If $N > 4$:
        i. If waitlist is empty:
            1. Allocate bed to next arriving patient. Set N = N – 1.
            2. Go to **(1)**.
        ii. If waitlist is not empty:
            1. Allocate bed to first patient at head of waitlist. Set N = N – 1.
            2. Go to **(1)**.
    c. If $N = 0$:
        i. Hold until $N > 0$.
        ii. If $N > 0$:
            1. Go to **(1)**.

Figure 1: Admission algorithm for non-glioma patients.

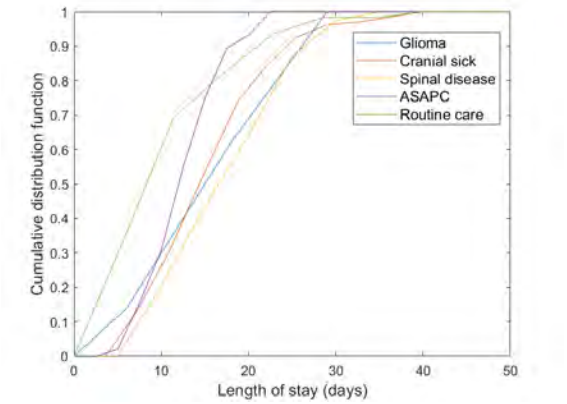
We now discuss the parameterization of the NS ward DES. The DES has two types of parameters: the rates of arrival of each patient type, and the lengths of stay. These parameters are estimated using non-identifiable admissions data provided by our collaborators from the neurosurgery department. The data provided consisted of the date at which patients arrived at the ward seeking admission, the dates at which they were admitted, their date of discharge and the patient type. However, for interarrival times, the data was not organized in a manner suitable to determining the best fit distribution to interarrival times using the dates of arrival. It appeared that data for patients was entered in batches - for example, in a month, patients appeared to arrive in batches on a few days (e.g., 6-7 days in a month), and hence it is unlikely that the data represents 'true' interarrival rates. Also, given that there was limited usable data (for each patient type, approximately 200-300 data points spread over 6-8 months), we were unable to fit discrete

distributions (e.g., Poisson) to the monthly arrival rates as well. Therefore, we assumed Poisson arrivals for each patient type, calculated the average monthly arrival rate for each patient type from the admissions data, and used the monthly arrival rates to calculate the mean interarrival times for the corresponding exponential distributions. Arrival rates for each patient type are provided in Table 1 along with the simulation outcomes (section 4).

Data regarding length of stay in the ward for each patient type was more amenable to statistical analysis, even if the size of the samples were limited. However, we found that lengths of stay varied widely and standard continuous distributions did not appear to fit the data well, evaluated by Kolmogorov-Smirnov goodness of fit tests. The histogram for the length of stay of ASAPC patients is presented below as an illustration (Figure 2a). Hence an empirical cumulative distribution function was fit to the length of stay data for each patient type, with lengths of stay sampled uniformly from the intervals corresponding to the histogram bins. The empirical CDFs for each patient type are depicted in Figure 2b, and the length of stay parameter estimates are listed in Table 1.



(a) Frequency distribution of length of stay for ASAPC patients.



(b) Cumulative distribution functions for length of stay by patient type.

Figure 2: Estimation of length of stay in the neurosurgery ward by patient type.

Table 1: Length of stay parameter estimates by patient type.

	Mean length of stay	Standard deviation
Glioma	15.12	7.37
Cranial	15.12	7.08
Spinal	16.86	6.91
ASAPC	12.24	4.27
Routine	9.95	7.46

Once the above sets of parameters are estimated, we then run the simulation to generate data for the classification process. We describe this process now.

3.2 Training Dataset Generation and Classification

The simulation was programmed in Python using the Salabim and the SimPy packages. The simulation is run for a period for 2500 days of ward operation for generation of training data for the classification process. The simulation was run on a computer with a quad-core Intel i5 1.8 GHZ processor and 8 GB RAM. The 2500 day simulation requires approximately 6 hours for recording and generating the system

state data. Based on our computational experiments, the first 10% of the simulation time was treated as the warm-up period, and to maintain a conservative approach, we also deleted data from the last 10% of simulation time to ensure we collected system state data only from the steady state operation of the NS ward. We describe the standard simulation outcomes (e.g., average time on the waitlist for those who are admitted, average proportion of patients admitted) in the following section. We now describe the dataset collected from the simulation.

For each patient entering the simulation, we record the following the information at their time of arrival into the system. The number of features corresponding to each system state information type is indicated in parentheses next to the type description.

- The type of the patient, coded from 0 to 4 in decreasing order of priority as described in the previous subsection (1);
- The number of empty beds among the 35 beds available for patient types other glioma (1);
- The number of beds occupied by each patient type at the current time (5);
- The patient type currently holding priority for allocation, per the admission algorithm in Figure 1 (1);
- The number of patients in the waitlist of each type at the current time (5); and
- For each of the 39 beds, the duration of time it has been occupied - the mean length of stay for that patient type (39).

The label for the classification component of our algorithm is whether the patient received a bed or not within the deterministic reneging threshold (a binary variable). In addition to the above, the simulation clock times at the point when the patient entered and exited the waitlist are also recorded, from which the time spent in the waitlist is calculated. This forms the label (the dependent variable) for the regression component. Note that we do not include the arrival rates in the above system state information set; however, the mean length of stay is included in the variable representing the occupation time of each bed. The reneging thresholds for each patient type are as follows: (a) glioma = 30 days, (b) cranial sick = 45 days, (c) spinal disease = 60 days, (d) ASAPC = 90 days and (e) routine care = 90 days. Given the preliminary nature of the work presented here, these thresholds are assumed. However, given the high severity of cases that typically present the NS ward in our collaborating hospital, these are likely to be reasonable.

The training dataset for the classification component thus consists of 52 features representing the system state and a binary label. Training and testing of the classification technique was conducted by dividing the dataset into training and generalization sets in a 4:1 ratio. After removing data from the warm-up and the last 10% of simulation run-time (approximately 2,000 patients), the overall dataset contained 12,000 samples. In general, for the overall dataset as well as the datasets for each patient type number of samples with a label of 1 (patient is admitted before reneging threshold) were not approximately the same as that of the samples with a 0 label - i.e., the datasets were imbalanced, which affected classification accuracies. Therefore, we performed random undersampling - without replacement - from the majority class (samples with a greater number) to balance the dataset; that is, until the number of samples in each class is the same. The degrees of imbalance (the ratio of the number of samples in the minority class to that in the majority class) for all datasets are provided along with the classification results in Table 3.

While we applied a variety of classifiers on the dataset, ranging from feedforward neural networks, decision tree classifiers (e.g., gradient-boosted trees, ensemble bagged trees, random forests), to logistic regression, we report the results of the top four classifiers that achieved the best results. These were chosen based on cross-validation and generalization area under the receiver operating characteristic curve (AUC) scores. We primarily use the AUC metric because of the imbalanced nature of our datasets, even though balancing was done prior to training and testing.

We describe our simulation and classification results in the following section.

4 Simulation & Classification Results

We begin by presenting the outcomes from running a version of our simulation without recording and storing the system state information. These are presented in Table 2 below. We find that simulation runtimes are reduced by nearly two orders of magnitude when recording and storing the system state is not required. The results in Table 2 were generated from 20 replications, which required approximately one minute per replication.

Before we discuss the simulation results, we briefly discuss validation of the NS ward simulation. In an ideal situation, the outcomes of the simulation, such as the proportion of patients admitted within a specified time frame and the average waiting time for those admitted would be compared against admissions data. However, while the admissions data provided to us contained dates of admission for a certain proportion of patients, many of these dates of admission are a year or two years after the date when they present at the ward seeking admission. Thus it is unclear whether these patients have sought care elsewhere or not in this interim period, and hence these dates of admission may refer to a subsequent surgery the patient has undergone after the first surgery for which the patient presented at the ward seeking admisison. Hence in cases where they have indeed sought care elsewhere, it may be inappropriate to include these patients in the validation dataset. Further, in cases where no admission dates are provided, it is unclear whether these patients have reneged, or are still present in the waitlist. Thus we were unable to directly validate the outcomes of the simulation. Finally, the fact that the simulated admissions algorithm is applied more as an approximate guideline than a set of rules enforced without exception presents an additional hindrance to validation. Therefore, we discuss the outcomes in Table 2 only as a means to illustrate the impact of the simulated algorithm on admissions outcomes for each patient type *if* the algorithm were applied in an exact manner and all assumptions regarding ward operations and reneging are valid, and to illustrate the need for adjustments to the admissions algorithm, even if implemented as an approximate guideline, that may improve admissions outcomes.

Table 2: Neurosurgery ward simulation outcomes.

Patient type (reneging threshold in days)	Arrival rate: monthly, daily	Proportion of patients admitted (%): mean (SD)	Average wait time for admitted patients: mean (SD)	Average number of patients admitted per week: mean (SD)
Glioma (30)	25.44, 0.85	30.42 (0.74)	27.80 (0.074)	1.80 (0.040)
Cranial sick (45)	23.08, 0.77	35.97 (0.01)	41.62 (0.299)	1.95 (0.048)
Spinal disease (60)	8.10, 0.27	99.99 (0.001)	1.27 (0.128)	1.88 (0.046)
ASAPC (90)	50.84, 1.69	66.65 (0.009)	84.50 (0.474)	7.86 (0.101)
Routine care (90)	51.72, 1.72	51.75 (0.014)	83.55 (0.663)	6.22 (0.095)

It is evident from the above table that given the high priority of glioma patients, the current capacity for glioma beds is not sufficient. The current admissions algorithm, if applied in an exact fashion, also appears to be suboptimal for cranial sick patients. Further, substantially smaller reneging thresholds and the longer lengths of stay of these patient types exacerbates the situation. The admission proportions for ASAPC and routine care patients are higher likely because of their shorter lengths of stay, significantly higher reneging thresholds, and higher patient arrival rates. Their higher arrival rates become an important factor when beds become available in batches (in particular when batch size > 4, see Figure 1) across a short duration (e.g., within a few hours), when patients are allocated beds on a first-come first-served basis from the waitlist. In such a situation, when the waitlist contains larger proportions of routine and ASAPC patients when compared to other patient types, it is significantly more likely that one of these patient types will be closer to the head of the line. With regard to spinal disease patients, their substantially lower arrival rates and reasonably larger reneging thresholds result in their nearly 100% admission rate. We also see

that for admitted patients, their wait times are close to their reneging thresholds, and hence this represents an undesirable situation if the thresholds (which are assumed values here) are chosen to indeed represent wait times beyond which unacceptable deterioration in their condition occurs. This indicates that if this approach for delay prediction is indeed implemented in a clinical environment, it might be advisable for hospital administration to specify thresholds well below durations beyond which unacceptable levels of deterioration are likely to occur.

Our clinical collaborator informed us that the above simulation outcomes broadly represented admissions trends observed in practice for each patient type. For example, they agreed that the majority of spinal disease patients might get admitted, and in very short timeframes, because in addition to their substantially lower arrival rate, almost all spinal disease patients who present at the ward at their hospital are severely ill and require urgent care, and hence in most cases are admitted in very short timeframes. They also agreed, given that in practice the algorithm in Figure 1 is applied along with continual adjustments based on clinical and administrative exigencies (for example, very sick patients admitted ahead of priority and/or less sick patients ahead of them on the waitlist), it is likely that the algorithm taken together with these adjustments yield to more favourable outcomes for patients. We are currently working with them to develop validation data, and once these are available, we anticipate that the algorithm in Figure 1 may also be modified to represent the admissions process more comprehensively as part of meeting validation targets.

We now discuss the results of the prediction of whether a patient gets admitted within their reneging thresholds, that is, the estimation of $\hat{L}_j = f(S_j)$. We present the results of applying the ensemble bagged decision trees classifier, the ensemble gradient boosted trees classifier, a feedforward neural network, and a single decision tree classifier on the dataset in Table 3 below. Note that we do not present the results for spinal disease patients, because they are almost certain to be admitted, and hence their dataset consists of only a few samples with a 0 label. We report the AUC, the precision and recall scores for each classifier. We used the scikit-learn machine learning package for the classification process. The ensemble bagged trees classifier implementation contained 10 estimator trees, the gradient boosting classifier implementation contained 100 boosting stages, and the feedforward neural network contained three hidden layers, each with 10 neurons and a rectified linear unit activation function. The ADAM solver was used for the neural network (Kingma and Ba 2014), with a learning rate of 0.001. These hyperparameters were chosen based on hyperparameter tuning results performed via cross-validation on the 80% training dataset.

Table 3: Waitlist outcome prediction accuracies - AUC, precision, recall.

Patient type (degree of imbalance)	Ensemble bagged trees	Gradient boosted trees	Neural network	Single decision tree
All patients (0.87)	0.947, 0.95, 0.95	0.951, 0.95, 0.95	0.919, 0.92, 0.92	0.95, 0.95, 0.95
Glioma (0.41)	0.841, 0.85, 0.85	0.83, 0.86, 0.84	0.711, 0.70, 0.69	0.86, 0.86, 0.86
Cranial (0.18)	0.932, 0.94, 0.93	0.96, 0.96, 0.96	0.944, 0.96, 0.95	0.97, 0.98, 0.98
ASAPC (0.94)	0.967, 0.97, 0.97	0.971, 0.97, 0.97	0.961, 0.96, 0.96	0.96, 0.96, 0.93
Routine (0.44)	0.95, 0.96, 0.95	0.952, 0.95, 0.95	0.941, 0.95, 0.94	0.94, 0.94, 0.94

Note that the single decision tree had more than 190 nodes with a maximum depth of 14 leaves. Overall, the reasonably high AUC scores are encouraging. The relatively lower AUC scores for glioma patients are likely because this patient type has the smallest sample size. This is supported by preliminary results from larger datasets constructed using alternative admissions algorithms that yield larger accuracies. Further, it appears that the tree-based classifiers perform better than the neural networks. The probability that a given patient will be admitted within their reneging threshold can easily be estimated by function methods associated with the scikit-learn implementations of the above classifiers.

We now briefly describe how the interpretability of tree-based classifiers can be leveraged to understand the key features that determine whether a patient is admitted or not. A visualization of the single decision

tree classifier was created, and using this the top branches were identified using their position in the tree and their Gini coefficients. The features corresponding to the top level branches (levels 1 and 2) were all based on the status of the beds: for example, a subset of the 39 features that indicate the amount of time a bed has been occupied by a given patient type, or the number of beds occupied by a patient type (e.g., routine patients). The next two levels consisted of a set of the same features as in the two levels above and additionally, waitlist-based features: for example, number on the cranial sick or ASAPC waitlist above patient j. This supports our contention that waitlist-based analytical delay estimators or delay history-based estimators alone will not yield accurate delay predictions for complex queueing systems such as the neurosurgery ward, and complete system state information consisting of both waitlist as well as resource status information will be required.

We also briefly discuss the results from applying regression methods to predict the waiting time to admission for admitted patients; that is, the estimation of $W_{j(ad)} = f_{ad}(S_{j(ad)})$. Note that for patients not admitted; their time spent in the waitlists will just be equal to the reneging threshold value corresponding to their patient type. For regression, given their universal approximation capabilities (LeCun et al. 2015), we utilize a simple feedforward neural network with three hidden layers. The three hidden layers consisted of 8, 12 and 20 neurons, respectively, with the *tanh* activation function. Once again, the ADAM solver and a learning rate of 0.001 were used. In addition, we also use an ensemble gradient boosting trees algorithm implemented as a regressor. Once again, we used 100 boosting stages.

We quantify regression accuracy using the mean absolute deviation (MAD), specified by: $MAD = \frac{1}{N}\sum_{i=1}^{N}|\hat{Y}_i - Y_{i(obs)}|$, where $\hat{Y}_i$ and $Y_{i(obs)}$ represent the predicted and observed waiting times (as generated by the simulation), respectively (we assume N test samples are available). We do not choose the mean absolute percentage error for reporting the regression accuracy, as several 0 waiting times were observed. We also report the root mean square error (RMSE), defined as $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_{i(obs)})^2}$. The results of our regression exercise are provided in Table 4.

Table 4: Prediction (regression) accuracies for wait times of patients admitted to the neurosurgery ward.

Patient type	Neural network		Gradient boosting trees	
	MAD	RMSE	MAD	RMSE
Overall	0.73	4.35	0.58	2.85
Glioma	0.88	1.41	0.93	1.19
Cranial Sick	2.41	8.75	0.80	2.46
ASAFC	6.44	23.52	0.21	0.79
Routine	7.01	24.66	0.19	0.63

For the neural network regressor, the regression accuracies, while reasonable for all patients and glioma patients, are relatively lower for ASAPC and routine care patients in particular. However, in comparison to the average wait times for admitted patients (Table 2), the MAD values for these patient types appear reasonable. However, the gradient boosting trees regressor outperforms the neural network regressor by a substantial amount with a maximum RMSE of only 2.85 in comparison to a maximum RMSE of nearly 25 for the neural network. This is likely because the set of features that comprise the system state can all be considered as categorical variables, which tree-based methods can typically handle better than neural networks.

5 DISCUSSION & CONCLUSIONS

In this paper, we present a method for prediction of admission outcomes for patients presenting at a neurosurgery ward seeking admission at the time of their arrival to the ward. Our method is motivated by results in the literature indicating that using system state information at the time of arrival yields more accurate delay predictions (Whitt 1999a).

In order to implement our approach of delay prediction in practice, the hospital administration will need to track the system state variables as defined in section 3 at regular intervals (e.g., six times a day) - for example, the amount of time for which each bed has been occupied by a given patient type will need to be tracked. Note that such information is already likely to be tracked for billing purposes. Then, for each arriving patient, this information (S_j) will have to be collected, processed, and then input into the machine learning method f to generate $\hat{L}_j$. If $\hat{L}_j = 1$ (that is, the patient is predicted to be admitted), then their time to admission $\hat{W}_{j(ad)} = f_{ad}(S_j)$ is also estimated. Note that real-time simulation, an approach wherein the neurosurgery ward DES is initialized with the system state S_j each time a patient arrives at the ward and $\hat{L}_j$ and $\hat{W}_{j(ad)}$ are generated by allowing the DES to run from that point in time onwards for multiple replications, is another potential approach for generating $\hat{L}_j$ and $\hat{W}_{j(ad)}$. However, this would require capturing the system state, loading it into the simulation, running a number of replications, and then estimating the average proportion of cases (replications) in which the patient gets admitted and the corresponding time to admission. Our approach is a significantly less computationally expensive approach when considered on a real-time basis: once the trained and validated machine learning method is created, it outputs the predictions as a one-time function evaluation. This makes it easier to deploy as an online calculator for the system administrator, as compared to deploying and running a simulation each time a new patient arrives in the system seeking admission.

The effectiveness of our method is contingent upon having a validated DES model of the queueing model under consideration. In the specific case of the system we model, we have already discussed in section 4 the challenges in validating the outcomes of our simulation. Another limitation of our current analysis is our assumptions for the reneging thresholds. Note that these thresholds may be used in multiple ways - for one, they may be used to represent actual reneging times. Alternatively, given the lack of data regarding reneging times, one can assume no reneging, and generate system state data for arriving patients from such a system. The thresholds can then be used in assigning labels to the system state based features - for example, assign 1 to patients getting admitted within a threshold T and 0 otherwise. The disadvantage with such a method is that in heavy-traffic conditions such as our case, it is very unlikely that patients arriving after the waitlist size has grown beyond a threshold size will be admitted within any meaningful time duration. Further, the assumption of no reneging is unrealistic in itself. For these reasons, despite the lack of data regarding reneging behaviour for neurosurgery admissions, we chose an approach that involved assuming reneging thresholds.

Several avenues of future research suggest themselves. Some of these include: (a) more comprehensive validation of our DES model, and simulation of alternate versions of the admissions algorithm; (b) evaluate prediction accuracies when stochastic reneging times are used; (c) comparison of our prediction accuracies with standard history and queue-length based delay estimators; (d) evaluation of our proposed method for other complex queuing systems; and (e) finding optimal admissions policies for the NS ward under consideration.

Overall, we present this work as a demonstration of our method for prediction of delays in complex queueing systems as encountered in healthcare systems, and hope it is useful for other practitioners attempting to predict delays for complex queueing systems.

REFERENCES

Ang, E., S. Kwasnick, M. Bayati, E. L. Plambeck, and M. Aratow. 2016. "Accurate Emergency Department Wait Time Prediction". *Manufacturing & Service Operations Management* 18(1):141–156.

Armony, M., and C. Maglaras. 2004. "Contact Centers with a Call-back Option and Real-time Delay Information". *Operations Research* 52(4):527–545.

Balakrishna, P., R. Ganesan, L. Sherry, and B. S. Levy. 2008. "Estimating Taxi-out Times with a Reinforcement Learning Algorithm". In *Proceedings of the 2008 IEEE/AIAA 27th Digital Avionics Systems Conference*. October 26th-30th, St. Paul, Minnesota, USA, 3.D.3-1-3.D.3-12.

Bassamboo, A., and R. Ibrahim 2020. "A General Framework to Compare Announcement Accuracy: Static vs LES-based Announcement". http://www.roubaibrahim.com/Final_draft_may_18_2020_nonblind.pdf, accessed 23rd June 2020.

Carmeli, Nitzan and Mandelbaum, Avishai and Yom-Tov, Galit 2017. "Data-Based Resource-View of Service Networks: Performance Analysis, Delay Prediction and Asymptotics". http://iew3.technion.ac.il/serveng/References/PhD_Proposal_Nitzan_Carmeli.pdf, accessed 22nd April 2020.

Hindustan Times 2017. "Want a Surgery at AIIMS? Wait Could be Four Years". https://www.hindustantimes.com/delhi/want-a-surgery-at-aiims-wait-could-be-four-years/story-Ndl1iIL8wFEn4Vm5fNPoBM.html, accessed 23rd March 2020.

Hu, M., Y. Li, and J. Wang. 2018. "Efficient Ignorance: Information Heterogeneity in a Queue". *Management Science* 64(6):2650–2671.

Hui, M. K., and D. K. Tse. 1996. "What to Tell Consumers in Waits of Different Lengths: An Integrative Model of Service Evaluation". *Journal of Marketing* 60(2):81–90.

Ibrahim, R. 2018. "Sharing Delay Information in Service Systems: A Literature Survey". *Queueing Systems* 89(1-2):49–79.

Ibrahim, R., M. Armony, and A. Bassamboo. 2017. "Does the Past Predict the Future? The Case of Delay Announcements in Service Systems". *Management Science* 63(6):1762–1780.

Ibrahim, R., and W. Whitt. 2009a. "Real-time Delay Estimation Based on Delay History". *Manufacturing & Service Operations Management* 11(3):397–415.

Ibrahim, R., and W. Whitt. 2009b. "Real-time Delay Estimation in Overloaded Multiserver Queues with Abandonments". *Management Science* 55(10):1729–1742.

Ibrahim, R., and W. Whitt. 2011a. "Real-Time Delay Estimation Based on Delay History in Many-Server Service Systems with Time-Varying Arrivals". *Production and Operations Management* 20(5):654–667.

Ibrahim, R., and W. Whitt. 2011b. "Wait-time Predictors for Customer Service Systems with Time-varying Demand and Capacity". *Operations Research* 59(5):1106–1118.

Kingma, D. P., and J. Ba. 2014. "Adam: A Method for Stochastic Optimization". *arXiv preprint arXiv:1412.6980*.

LeCun, Y., Y. Bengio, and G. Hinton. 2015. "Deep Learning". *Nature* 521(7553):436–444.

Moriah, H., D. Efrat-Treister, A. Rafaeli, A. Cheshin, and S. Agasi. 2011. "Situational Antecedents of Customer Conflict and Aggression Toward Healthcare Professionals in the Hospital Setting". In *Proceedings of the 24th Annual Conference of the International Association for Conflict Management*. July 3rd-6th, Istanbul, Turkey.

Nakibly, E. 2002. *Predicting Waiting Times in Telephone Service Systems*. M.S. thesis, Department of Industrial Engineering, Technion - Israel Institute of Technology, Haifa, Israel. http://132.68.160.12/serveng/References/PredictingWaitingTime.pdf.

Senderovich, A., M. Weidlich, A. Gal, and A. Mandelbaum. 2014. "Queue Mining – Predicting Delays in Service Processes". In *Proceedings of the Conference on Advanced Information Systems Engineering*, edited by M. Jarke, J. Mylopoulos, C. Quix, C. Rolland, Y. Manolopoulos, H. Mouratidis, and J. Horkoff, 42–57. Cham, Switzerland: Springer.

Senderovich, A., M. Weidlich, A. Gal, and A. Mandelbaum. 2015. "Queue Mining for Delay Prediction in Multi-class Service Processes". *Information Systems* 53:278–295.

Simaiakis, I., and H. Balakrishnan. 2016. "A Queuing Model of the Airport Departure Process". *Transportation Science* 50(1):94–109.

Sriram, S. 2018. "Availability of Infrastructure and Manpower for Primary Health Centers in a District in Andhra Pradesh, India". *Journal of Family Medicine and Primary Care* 7(6):1256–1262.

Times of India 2017. "Lack of Hospital Beds Means Up To 2-year Wait for Surgery Patients". http://timesofindia.indiatimes.com/articleshow/61537514.cms?, accessed 23rd March 2020.

Whitt, W. 1999a. "Improving Service by Informing Customers about Anticipated Delays". *Management Science* 45(2):192–207.

Whitt, W. 1999b. "Predicting Queueing Delays". *Management Science* 45(6):870–888.

Zhang, X. 2018. "Application of Discrete Event Simulation in Health Care: A Systematic Review". *BMC Health Services Research* 18(1):1–11.

AUTHOR BIOGRAPHIES

VAIBHAV BALDWA is a junior undergraduate in the Department of Mechanical Engineering at the Indian Institute of Technology Delhi. His email address is me1170702@mech.iitd.ac.in.

SIDDHARTH SEHGAL is a junior undergraduate in the Department of Mechanical Engineering at the Indian Institute of Technology Delhi. His email address is me1170698@mech.iitd.ac.in.

VIVEK TANDON is an additional professor in the Department of Neurosurgery at the All India Institute of Medical Sciences, New Delhi, India. His email address is drtandonvivek@gmail.com.

VARUN RAMAMOHAN is an assistant professor in the Department of Mechanical Engineering at the Indian Institute of Technology Delhi. His email address is varunr@mech.iitd.ac.in.