Forecasting Hospital Bed Availability
Using Simulation and Neural Networks

Matthew J. Daniels
Michael E. Kuhl
Industrial & Systems Engineering Department
Rochester Institute of Technology
Rochester, NY 14623

Elisabeth Hager
Hager Consulting
Pittsford, NY 14534

Abstract

The availability of beds is a critical factor for decision-making in hospitals. Bed availability (or alternatively the bed occupancy level) affects required staffing levels, scheduling of surgeries, patient acceptance decisions in emergency departments, and many other important hospital decisions. To better enable a hospital to make these decisions, this paper presents a forecasting methodology for predicting hospital bed availability based on the current status of the hospital. A two-phased approach is used. The first phase involves building a computer simulation model of a hospital, to determine the significance of various factors on bed availability. In the second phase, an artificial neural network (NN) is developed, using input variables and training data from the validated simulation, to create bed availability forecasts. The NN features the use of time as an explicit input, to model the significant changes in the system over time such as time dependent arrival and departure rates. An associated tolerance interval for the forecasts is calculated to establish a measure of the quality of forecasted value. This method is applied to scenarios of varying forecasting time horizons illustrating the general effectiveness of the method for generating forecasts of hospital bed availability.

Keywords: Health care, Forecasting, Simulation, Neural Networks

1. Introduction

Health care is one of the largest industries in the United States. National health care expenditures exceed $1 trillion annually [3], and nearly 11% of US employment is accounted for by the health care industry [2]. Cost increases on the order of 9% per year [3] have lead researchers to investigate opportunities for improving the efficiency of health care systems.

One way to improve health care efficiency is through the optimal use of resources. Through interaction with the health care industry, we have identified the problem of bed availability as one of the fundamental issues in the management of hospital operations. The effective utilization of resources, in particular, the utilization of the hospital beds, will be a fundamental factor in day-to-day operations such as staff scheduling and will be strongly correlated with the bottom line of the financial statement. Currently, there are no established scientific methods for predicting bed availability in hospitals. In many cases, a hospital employs a person who will rely on their own experience to schedule and assign beds to patients. The main objective of this research is to develop a method for modeling and forecasting hospital bed availability, in order to optimize hospital bed utilization.

Although the problem of bed availability is easy to define, the solution is the result of many interdependent variables and stationary and nonstationary stochastic processes. The current operations research methods available in the literature do not address the requirements for a solution to this problem. The solution methodology that we present is a two step process in which we first build a simulation model of the system to identify important factors and their interactions. Although a simulation model could be used to forecast the distribution of bed availability over a time horizon, the computational overhead and time involved in running the large number of replications of the model that would be required to give an accurate solution will be prohibitive. Therefore, we build on a proven operations
research tool, neural networks, to design a forecasting system that will be both computationally efficient and be able to adapt to changes in hospital behavior.

2. Methodology
To address the problem stated above – to create accurate short-term forecasts of hospital bed availability – a two-phased method is developed in this work. The first phase involves building a computer simulation model of a hospital, to determine the significance of various factors on bed availability. In the second phase, an artificial neural network (NN) is developed, using input variables and training data from the validated simulation, to create bed availability forecasts. The methods used in each of these phases are discussed in this section.

2.1 Phase 1: Computer Simulation
The function of the model is to facilitate understanding of factors affecting hospital bed availability. By experimenting with the simulation model, significant factors are identified and are used as input to the neural network constructed in Phase 2. The simulation model is also used to generate training data for the neural network.

2.1.1 The Hospital Simulation Model
The simulation model is composed of patients, which are modeled as entities, and hospital beds, which are treated as resources. The beds are allocated within various hospital units (see Figure 1 below). Patients arrive at one of two entrance points, are treated in a succession of units within the hospital, and then are discharged back into the outside world. A single patient entity is used in the model, and various attributes are used to create characteristics of patients with different illnesses and severities. Each patient’s attributes are diagnosis, severity, bed type, sequence number, and unit length of stay (LOS). “Diagnosis” is the patient’s illness, which determines the patient’s path through the hospital (using different sequences for different patient types). It also determines the average amount of time spent by a patient in each unit – that is, the patient’s unit LOS. “Severity” determines the patient’s priority in seizing a bed in any particular unit (bed type). By using discrete probability distributions to randomly assign diagnosis and severity, the mix of patient types arriving at each entrance point can be adjusted.

![Figure 1: Generic hospital layout.](image)

Patients arrive to an Admissions unit or as emergency patients to an Emergency Department. The arrival of patients at the entrance points varies throughout each day and between days. Further, the arrivals to each entrance point are independent of the other and vary considerably from each other. The arrival patterns of patients are determined through available data and through interviews with experts, and the patterns are modeled using time-varying arrival schedules in the model. LOS in each unit is determined using a triangular distribution around that patient type’s average unit LOS. The patient is ready for discharge from the hospital once it has finished its particular sequence; however, since hospitals typically do not discharge patients overnight, the model has rules to reflect these discharge procedures. The outputs of interest from the model are the hourly counts of each type of patient and the count of available beds each hour. Output files capture this information, facilitating its use in the second phase of the general forecasting method. The utilization levels of each bed type, the length of queues for each bed type, and average LOS (averaged over all patients) are automatically tracked as well.
2.1.2 Verification and Validation of the Simulation Model
Once built, model verification is done to ensure that the simulation functions as intended. Model validation is accomplished through comparison of a simulation model with the real system it is modeling. A number of comparisons between model output and expected outputs are performed—specifically, comparisons based on numbers of patients entering the hospital, average LOS for all patients, and patient mix. The validation results show the degree to which the simulation is able to reproduce the results of the real system.

2.1.3 Identification of Significant Factors for NN
Once a verified, validated model is created, it is used to determine the factors of interest when building the NN in phase two. The factors to be considered from the model are current bed availability, current time, and mix of patient types. A significant amount of the bed availability one time step ahead is explained by bed availability at the current time, so it is included in the NN. Two facets of the current time variable make it significant. First, the discharge policy prevents any patients from leaving at night, making bed availability necessarily lower at night than during the day. Second, arrival patterns result in many more arrivals at some times than at others, suggesting that time of day and time of week have an effect on bed availability. Third, the effect of patient mix is tested through a two-sample T-test for equal means, using a Base model and an Alternative model. The Base model is the validated simulation model described above. In the Alternative model, all patient types are equally likely to occur. 100 weeks of bed availability data are captured, and T-test results indicate whether patient mix is a significant factor.

2.2 Phase 2: Neural Network
The second phase of the methodology is to construct a NN to be used for forecasting the hospital bed availability.

2.2.1 Constructing the Neural Network
Since neural networks can be built in literally any arrangement, some experimentation is required to arrive at a network that works best for a particular problem. Some factors to be considered in constructing a NN are network inputs, number of layers, transfer functions, number of nodes (or “neurons”), and weight initializations, as discussed below. A representative neural network structure is shown in Figure 2.

![Figure 2: A representative neural network.](image)

In Figure 2, the network inputs are at the left. Each of the $m$ inputs are connected to each of the $n$ neurons in Layer 1, by way of a matrix of weights of size $n \times m$. Therefore, each neuron in Layer 1 receives $m$ weighted inputs. The symbol in the Layer 1 box depicts a linear transfer of the input signal to each neuron—that is, the output of a neuron $j$ in Layer 1 is computed as

$$ Y_j = \sum_{i=1}^{m} w_{ij} X_i $$

for $j = 1, \ldots, n$ \hspace{1cm} (1)

Each neuron’s output $Y_j$ constitutes an input to Layer 2, and in this case, each of the $n$ neurons in Layer 1 connects to the single neuron in Layer 2 via a weighted connection. The symbol in the Layer 2 box, again, depicts a linear transfer. Then, the output of the network can be written as

$$ Y_{\text{network}} = \sum_{j=1}^{n} Y_j w_{2j} $$

(2)
The inputs to the NN for this general method are current time, current number of beds available, and if significant current patient mix. Though time dependency is accounted for in some of the research reviewed, this was done primarily through the addition of model terms for seasonal effects [5] or through de-seasonalizing the data. The network input structure in this research, in contrast, uses an indicator node for each hour of the week. By choosing this structure, the neurons for each hour of the week have separate weights to adjust, allowing availability trends to be modeled. The current number of available beds is modeled as a percentage: the current beds available divided by the total number of beds. This percentage allows the input to be scaled to a range of [0,1], consistent with the other inputs used. Third, the current patient mix can be included as the number of each type of patient currently in the system. Each input value is also scaled to restrain the values to the [0,1] range. The resulting vector of inputs creates a picture of the hospital’s current status that can be used to generate forecasts of bed availability.

The number of network layers is a second characteristic that must be determined. In order to minimize the complexity of the NN, experiments are first tried for a single hidden layer of neurons, and then more layers are added as needed. The transfer functions to be used in the network are decided in a similar fashion as the number of layers. That is, the NN uses simple linear transfer functions, and the transfers are changed if needed. The correct number of hidden-layer neurons is shown in many sources to be difficult to determine optimally. The correct number depends on a number of factors, including numbers of inputs and outputs, the variability in the data, the complexity of the input-output relationship, and the hidden unit activation chosen [4]. The overall goal is to provide adequate hidden units to allow the network to learn the function of interest, while avoiding over-fitting of the training data that will be used. Lastly, the initial conditions of the NN for training and testing must be determined.

2.2.2 Training and Testing the Neural Network

The output of an untrained network will be “random” with respect to the desired behavior, since the initial settings of the weights are usually either zeroes or random values. During the training process, these weights are adjusted to make the network’s behavior approach the desired behavior. While there are many training algorithms from which to choose, the error back-propagation method has achieved widespread use and is suitable for many NN applications. In this algorithm, the NN sees an input and produces an output using its current weights; then, based on the error committed in that prediction, weight changes are propagated backwards through the network to reduce the error. In order to train the NN sufficiently, while avoiding over-fitting, the training data set is run through the neural network training algorithm until the maximum absolute error committed by the network is some predetermined number of beds. If a network is over-trained, it will not be able to generalize well to new data. A rule of thumb for the size of a training data set is to divide the number of weights in the network by the expected approximate level of error [6]. However, for an expected error of 0.05, for example, this rule dictates a need for 20 training inputs per network weight. A smaller number of data points may need to be chosen as a reasonable set, based on network size and the computational abilities of the machine on which the NN is trained.

Once it is trained, the NN is tested using a new data set to measure its performance. A data set is run through the NN without any weight changes, and the NN makes forecasts on the test set of data. It is important to provide, in addition to a forecast value, an indication of the precision of that estimate. Here, a tolerance interval for bed availability is used. It is calculated during network testing and is applied to the point estimate provided by the NN. The tolerance interval half-width (for a normally distributed sample) can be calculated as

$$\text{Half - width} = Z_a \cdot \sqrt{\text{MSE}}$$  \hspace{1cm} (3)$$

Here, square root of the mean squared error, \(\sqrt{\text{MSE}}\), is an estimate for the sample standard deviation and can be calculated from the forecast errors using equation (4), with \(T\) being the number of historical data points [1]:

$$\text{MSE} = \frac{\sum_{t=1}^{T} (e_t - \bar{e})^2}{T - 1}$$  \hspace{1cm} (4)$$

3. Example

To illustrate this methodology, we present an example that is based on a 200-bed hospital. The characteristics for the hospital are based on interviews with experts and on a Rochester, New York-area hospital of average size.
3.1 The Simulation Phase
The generic hospital conceived for modeling in this research is a 200-bed hospital providing an average range of services. Patients are assigned one of twelve diagnoses and one of five severity levels. In terms of arrivals, the volume is based on average weekly arrivals to a 200-bed hospital. Non-emergency patients tend to arrive in a cluster in the early morning since their procedures are scheduled during the day. In contrast, emergency patients arrive more randomly, usually during the evening and night hours. There are also week-level trends for patient arrivals: scheduled patients will tend to arrive early in the week, so that they can recuperate and be discharged before the weekend. ED patients, in contrast, are more likely to arrive on the weekend.

The model was verified and validated with respect to the proportion of each type of patient in the model, the expected proportions from the real system is compared against the actual value from the simulation. The average absolute difference between expected and actual values for these patient proportions is 0.33%, and the maximum absolute error is only 0.75%. Table 1 summarizes the performance of the simulation model. In summary, the simulation model built for this example is able to reproduce the conditions of the hospital on which it is based.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Expected Value</th>
<th>Actual Value</th>
<th>% Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of patients entering the hospital (per day)</td>
<td>36.71</td>
<td>34.48</td>
<td>6.07%</td>
</tr>
<tr>
<td>Average LOS (all patients)</td>
<td>173.99</td>
<td>168.24</td>
<td>3.30%</td>
</tr>
</tbody>
</table>

The model is used to test the effect of patient mix on average bed availability. As described in the Methodology section above, 100 weeks of data are taken from the Base model, and 100 weeks are taken from an Alternative model where the only difference is the patient mix (all patient types equally likely to arrive). The result of the simulation runs is that average bed availability in the Base Model is 25.13%, while it is 17.83% in the Alternative model. This difference of 7.3% is shown through the T-test to be significant (P-value less than 0.001).

3.2 Neural Network Phase
The inputs to the NN are current hour of the week and current number of available beds (as described in the Methodology above), and current patient mix (confirmed through the experiment above). The network only has a single hidden layer. Comparisons against NNs with more layers show that the extra layers do not significantly improve performance. The number of neurons in that layer is initially set at 168, which is the same size as the vector for the current time input. This NN is compared against two smaller networks: one is half the size of the original, and the third alternative network is of an intermediate size. As seen in Table 2 below, the smaller networks are not as good in training performance, but they are better than the original network in test performance.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Original Network</th>
<th>Half-size Network</th>
<th>Third Alternative Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE (training error)</td>
<td>6.39 beds(^2)</td>
<td>10.67 beds(^2)</td>
<td>8.43 beds(^2)</td>
</tr>
<tr>
<td>Maximum Absolute training error</td>
<td>12.38 beds</td>
<td>13.67 beds</td>
<td>13.89 beds</td>
</tr>
<tr>
<td>MSE (test error)</td>
<td>463.13 beds(^2)</td>
<td>204.32 beds(^2)</td>
<td>164.92 beds(^2)</td>
</tr>
<tr>
<td>Maximum Absolute test error</td>
<td>41.56 beds</td>
<td>38.87 beds</td>
<td>43.52 beds</td>
</tr>
</tbody>
</table>

Linear transfers provide good performance on this problem and so avoid the need for more complex functions. In terms of weight initialization, initializing the network weights to zero does not appear to work well for this specific problem. However, initialization to weights in the range (0,0.1) works well. The training data are obtained from the computer simulation model. Five-hundred weeks worth of data are used for training, and each data point is passed through the network until the maximum error committed is ten beds difference from the target. During this process, the data set is passed through the network 400 times for training, resulting in 3.36 million input patterns being presented to the network.

Following training, the network is tested on a new data set of 500 time steps. NN weights are not changed during this test run. The results are used to create a tolerance interval and to plot the network’s performance. The outputs of the NN from this test are indicated as “Network Output” in Figure 3. As the Figure shows, the NN consistently overestimates the bed availability on this single test run. Since the training data come from a much longer horizon, there are periods of high availability and low availability. The NN ends up averaging these periods in its learning. Thus, a
possible cause of the over-estimation is that the data are from a period of lower-than-average availability. Further, this test is conducted over 500 time steps, while the weights are not updated during the test. The use of dynamic updates to the NN’s weights would help minimize errors. Note, finally, that the network is consistently able to follow the complex time-dependent pattern of changing bed availability, even during extreme low periods (such as around hour 200).

Figure 3: Neural network test results from 500 time steps of one test, with tolerance interval applied.

4. Conclusions
The general method presented here is a viable approach to the problem of short-term hospital bed availability forecasting. A simulation model has been built to model the non-stationary arrivals and overall complexity present in hospitals. That model has been used to determine factors affecting bed availability. A neural network has been created to forecast the availability of hospital beds, using current hospital status information. A forecasting tolerance interval has been built from the MSE of the network’s forecast errors, to provide a more robust forecast result. An example application of this method has demonstrated its ability to create accurate forecasts.

Many advances and enhancements to this research are possible and encouraged including integrating neural network forecasting system into a hospital information system, so that current data can be used to continuously update the neural network’s learning. This method can be expanded into other applications in a variety of ways, including use in other industries, specialization to single hospital units, and through expansion into a community-wide model of shared hospital capacity management.

References