

Estimating Hospital Admissions with a Randomized Regression Approach

Kleber A. Garcia
Electronic Arts
Orlando, FL 32810
kecho_garcia@hotmail.com

Philip K. Chan
Department of Computer Sciences
Florida Institute of Technology
Melbourne, FL 32901
pkc@cs.fit.edu

Abstract—Boarding or holding in the Emergency Department (ED) reduces capacity of the ED and delays patients from receiving specialized care. Estimating accurately the number of admissions from the ED can help determine appropriate level of staffing to reduce holding. We propose a randomized non-linear regression algorithm, RT-KGERS, to estimate the number of admissions a week in advance. We also devise features based on cyclical patterns found with a Fast Fourier Transform analysis on the hospital admission data. We evaluate the accuracy and efficiency of RT-KGERS and three existing algorithms in a dataset provided by a local hospital. We then compare our features with related features. Initial experimental results from RT-KGERS encouraged the hospital and us to conduct a live trial study which yielded similar levels of accuracy using RT-KGERS and the six features we devised.

Keywords: *Medical informatics; regression trees; randomized algorithms*

I. INTRODUCTION

Many emergency departments (ED) in the United States are overcrowded. One reason for overcrowding is boarding or holding patients who have been admitted in the ED. Boarding not only reduces the capacity of the ED, but also delays the patients from receiving specialized care that they need. The problem is severe enough that the American College of Emergency Physicians (ACEP) created a task force on boarding and published a report on possible solutions [1].

The main reason for boarding is the lack of inpatient beds. As we learned from a local hospital, Holmes Regional Medical Center (HRMC), the underlying problem is not the lack of "physical" beds, as they can create beds in hallways or even conference rooms, but inadequate staffing. Beds need to be staffed at a certain level, for example, a nurse for 6 regular beds. However, due to the high variability of patient volume from the ED, the hospital cannot accurately estimate patient volume a few days in advance to adjust the staffing schedule. Many hospitals are non-profit; overstaffing is expensive, so understaffing sometimes happens which leads to boarding.

Currently, HRMC schedules staff based on their collective memory and experience of admissions in the past and does

not have an automated system to help them estimate the number of inpatients from the ED. To help adjust the staffing schedule, estimates from an automated system need to be at least a few days in advance—staff members cannot be expected to show up on short notice. Also, preferably, the estimates are for intervals no longer than 8-hour shifts. Lastly, the automated system needs to be reasonably efficient to process new data that arrive periodically (without significantly affecting other automated systems). Therefore, our study has four objectives:

1. design an algorithm that can accurately and efficiently estimate the number of admissions from the ED in 4-hour intervals a week in advance,
2. devise informative features to characterize the data,
3. evaluate our algorithms based on accuracy and efficiency on historical data on admissions from the ED, and
4. evaluate our algorithm in a live trial study on estimating admissions from the ED.

Our overall approach is to use machine learning algorithms, particularly non-linear regression methods, to build models from historical data and use the models to estimate the number of admissions a week in advance. Features that represent the data are based on a FFT (Fast Fourier Transform) analysis on cyclical/seasonal characteristics.

II. ESTIMATING HOSPITAL ADMISSIONS

The data used in this study are from Holmes Regional Medical Center (HRMC) in Melbourne, Florida. The dataset contains number of admissions from the ED in four-hour intervals—the average is about 9.8 admissions. We use the first two years of the data (training set) to train the regression models and the last year (test set) for evaluating the models. A summary of the data is in Table I.

TABLE I. EDINPT DATA FROM HRMC

Data set	Training Set	Test Set
Admissions from ED (EDInpt)	April 2008 – March 2010	April 2010 – March 2011

A. Features

The data we were provided can be considered as a time series of number of admissions. To identify features for predicting admissions, we first analyze the possibility of underlying cycles in the time series using Fast Fourier Transform (FFT). Our FFT analysis identifies potential underlying daily, yearly and weekly cycles (in order of strength) in admissions from the ED. The presence of daily cycles suggests that the number of admissions is similar at the same hour on different days, but dissimilar from other hours in the same day. We can make corresponding interpretation on yearly and weekly cycles. Based on this analysis, we identify the following 6 features.

Let x_i be a feature and Y be the target value, which is the number of admissions at time t . Though time t is a four-hour period, to simplify our discussion, we refer the period as an "hour." Since daily cycles are the strongest, all the features are from the same hour as time t . To capture the yearly cycle, x_1 is the number of admissions 52 weeks before t (same day of the week). Since a single value could fluctuate a lot, x_2 is a "smoothed" value of x_1 by averaging x_1 with values 1 and 2 weeks before t and 1 and 2 weeks after t (i.e. an average of 5 values in a 4-week window). Feature x_3 is the number of admissions 365 days ago (notice this is different from 52 weeks, since it could be a different day of the week). Holidays such as 4th of July are strictly the same date, but might fall in a different day of the week. To incorporate weekly cycles, x_4 represents the average value of the past 3 weeks during the same day of the week and hour of the target value. Finally, to capture possible shift in the population of the community, we calculate the rate of change in value ("slope" in other words). Feature x_5 represents the change in last week from two weeks beforehand, while x_6 is the change in last week from the same week a year ago. The features are summarized in Table II.

TABLE II. FEATURES FOR EDINPT DATA

Feature	Feature Description
x_1	52 Weeks ago
x_2	Average of 52 weeks ago, 2 previous weeks, and 2 following weeks
x_3	365 days ago
x_4	Average of past 3 weeks
x_5	Weekly rate of change
x_6	Yearly rate of change

B. Algorithms

Let X be a vector of feature values x_1 through x_6 together with the target value Y . These form a tuple (X, Y) . By varying time t , a set of tuples can be generated. The goal is to find a function f such that

$$f(X_i) \approx Y_i,$$

where $i = 1, 2, \dots, n$ and n is the number of data tuples. That is, f is a model that can estimate the number of admissions at time t (Y) based on features values measured prior to time t (X).

We desire to use machine learning algorithms to construct function f from a set of data tuples. To accommodate the potential non-linearity of f and relatively fast computation, we choose to construct a piecewise linear function using a regression tree approach. This approach recursively splits the feature space into subspaces until a stopping criterion is met, at which point a regression algorithm is used to find a linear function for the data tuples in the subspace. Figure 1 shows the generic regression tree algorithm with three subprocedures (splitting criterion, stopping criterion and regression).

Our splitting criterion (line 5) is a threshold of a feature value—data tuples with values greater than the threshold is in one subset, otherwise in the other subset. To choose the criterion, we prefer features and threshold that result in subsets that are closer to linear, which is determined by the error incurred by a linear regression algorithm called KGERS (K–Gaussian Elimination from Randomized Subsets). Details of KGERS are discussed in the next section. Our stopping criterion (line 2) is statistically insignificant (t-test 95% confidence) change in error incurred from linear regression before and after a potential split. That is, splitting a dataset into two subsets does not significantly improve linearity in the data set. At this point, we use KGERS as the regression algorithm (line 3) to find a multidimensional linear function (hyper plane) for the leaf node. We refer to this algorithm as Regression Tree with KGERS or RT-KGERS.

```

RegressionTree
-inputs: [X, Y]
-outputs: RegressionTree
1: Begin
2: if STOPPING CRITERION is true
3:   perform REGRESSION in current [X,Y]
4: else
5:   perform SPLITTING CRITERION and split
   [X,Y] into two subsets (left and right)
6:   Recursively call RegressionTree on
   left and right subsets
7: End

```

Figure 1. Regression Tree Algorithm

C. KGERS Linear Regression

KGERS [2] is a randomized algorithm for linear regression. The algorithm is motivated by the following observations. Consider a 2D space with x_1 (only one feature) and Y , if the data tuples are perfectly linear, any pair of tuples can dictate the line. If the data tuples are roughly linear, the majority of the pairs will form lines close to the optimal one (that has

the least error). Therefore, instead of using all data tuples for regression, we can use far fewer tuples when they are roughly linear. Moreover, if a line is closer to the optimal one, it should have less error on the rest of the data tuples. Hence, the error of each line can help determine the final line.

```

K GERS
-inputs: X, Y, K (M is number of features)
-outputs: W (coefficients of final hyperplane)
1: Begin
2: For k=1 to K
3: | Let S be randomly selected M+1 rows in
   | [X,Y]
4: | Let V be another randomly selected
   | M + 1 rows in [X,Y]
5: | H[k] = GenerateHyperplane(S)
6: | R[k] = CalculateHyperplaneError(H[k], v)
7: W = GenerateFinalHyperplane(H, R)
8: Return W
9: END

```

Figure 2. KGERS Algorithm

Figure 2 has the KGERS algorithm for data with M features. KGERS repeats the following K times: select a subset (S) of $M + 1$ data tuples for training (line 3), select another subset (V) of $M + 1$ data tuples for validation (line 4), find a hyperplane using Gaussian Elimination (line 5), and measure error (R) on the validation subset (V) (line 6). KGERS generates the coefficients of the final hyperplane (W) by summing the K hyperplanes inversely weighted by their errors (line 7). K is a parameter to KGERS and is much less than N .

KGERS contains an outer loop that performs K sample trials in step 2. For each one of these trials, we generate a hyper plane using Gaussian elimination which has a time complexity of $O(M^3)$ which is in step 5. After this we calculate the hyperplane weight which has $O(M^2)$ time complexity in step 6. The final step 7, aggregates all the hyperplanes which has a time complexity of $O(KM)$. Overall, the time complexity of KGERS is $O(K(M^3 + M^2 + M))$, which can be simplified to $O(M^3)$ as K is a constant. Note that the complexity does not depend on the number of data tuples (N).

III. EXPERIMENTAL EVALUATION

A. Evaluation criteria and Experimental Procedures

The main evaluation criterion used for measuring the quality of the models is Root Mean Squared Error (RMSE), which amplifies larger errors (1).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i)^2} \quad (1)$$

For reporting to the hospital staff during our live trial period, we use Mean Average Error (MAE) since larger

errors are not amplified and the results are easier to interpret (2) — an MAE of 3.4 means an average error of ± 3.4 patients.

$$MAE = \frac{1}{n} \sum_{i=1}^n |f(X_i) - Y_i| \quad (2)$$

To measure efficiency of the algorithms in building the models, we use number of CPU clock ticks on an Intel Quad Core I-7© machine, running Microsoft Windows 7 64 bit operating system.

B. Non-linear regression algorithms

The algorithm devised in this study is RT-KGERS (Regression Tree KGERS). This is a non-linear regression algorithm capable of constructing a regression tree using KGERS in the leaves. The only parameter used in this algorithm is K , which is the number of candidate hyper planes used at the leaves.

A second algorithm is Regression Tree Mean (or RT-MEAN). Instead of using linear regression at the leaves, the mean Y values are used. This algorithm does not use any parameters.

A third algorithm is Regression Tree with Gradient Descent (RT-GD). Gradient descent is a common algorithm to regress hyper planes and is used to devise the leaf nodes. Gradient descent uses number of iterations as its main parameter.

The last algorithm is a Feed Forward Artificial Neural Network (ANN) [3], which is a commonly used non-linear regression algorithm. Our ANN has three layers (input, hidden, and output) of units with linear functions and non-linear thresholds to form a non-linear model. Similar to Gradient Descent, ANN uses the number of iterations as one of its main parameters. Other parameters of ANN include the number of neurons in the hidden layer.

To understand the tradeoff between efficiency (CPU clock ticks) and accuracy (RMSE) in each algorithm, we vary the parameter values, which can affect either/both evaluation criteria, in each algorithm. For RT-KGERS, we vary K , the number of hyperplanes, in KGERS. For RT-GD, we vary the number of iterations, in Gradient Descent. For ANN, we vary the number of iterations. Other parameters for GD in RT-GD and ANN are kept constant and were determined by initial experiments. RT-MEAN does not have a parameter to be varied.

IV. EXPERIMENTAL RESULTS

A. Comparing Algorithms

Figures 3 plots results from our experiments on the EDInpt. To visualize the tradeoff between computational overhead and accuracy, the X axis represents CPU clock ticks while

the Y axis represents RMSE. The lower left corner of a plot has lower error (RMSE) as well as computational overhead (CPU), which is more desirable. RT-KGERS, RT-GD, and ANN have multiple points in each plot because we vary the parameter values in each of those algorithms. RT-MEAN has only one point in each plot because it does not have a parameter to be varied.

Figure 3 shows our results from the EDInpt dataset. In general, not always, RT-KGERS achieves a lower RMSE with lower computational overhead than the other three algorithms. For RT-KGERS, we also observe that a higher concentration of points with lower RMSE, indicating RT-KGERS tends to have lower RMSE on average. However, due to its randomized algorithm, sometimes RMSE could be larger when the K data subsets were poorly selected to generate the K hyperplanes. We also observe from Figure 3 that there are vertical bands of points for RT-KGERS, indicating different RMSE with a similar amount of computation. This artifact is probably due to changes in the number of nodes in the regression tree. Because of the relative large range of different K values used in the experiment, a large number of points can cluster in one vertical band corresponding to the same number of tree nodes with similar amount of computation.

As expected, RT-MEAN is faster than the other algorithms because calculating the mean at the leaves is much faster than KGERS or Gradient Descent; however, it does not have the lowest RMSE. As expected, RMSE of ANN decreases as number of iterations (clock ticks) increases, however, the computational overhead is generally higher to achieve a certain level of RMSE. RT-GD achieves relatively low RMSE, however, the computational overhead is higher than RT-KGERS.

Though RT-MEAN is the most efficient, it is usually not the most accurate. As expected, ANN reduces RMSE with more iterations and incurs the most computational overhead. We expected RT-GD to reduce RMSE with more iterations in gradient descent; however, the expectation was not observed. One possible reason is the interaction between the regression tree algorithm and the gradient descent algorithm. Since the splitting criterion in the regression tree algorithm uses the gradient descent algorithm, more iterations in gradient descent could get a more accurate split and reduce the size of the regression tree, which reduces computation. Similarly, we expected a larger K (more computation) for KGERS would improve RMSE for RT-KGERS; however, a larger K could get a more accurate split and reduce the size of the regression tree, which reduces computation.

B. Comparing Features

Batal et al. [4] propose using linear regression with calendar features to estimate patient admissions on a daily basis. The features include 7 days of the week (e.g. Monday is true/false), 12 months of the year (e.g. January is true/false),

holiday (true/false), and day before holiday (true/false). That is a total of 21 features. Since our data are sampled in 4-hour intervals, for a fair comparison, we added six features for the six 4-hour intervals. Hence, we have a total of 27 calendar features.

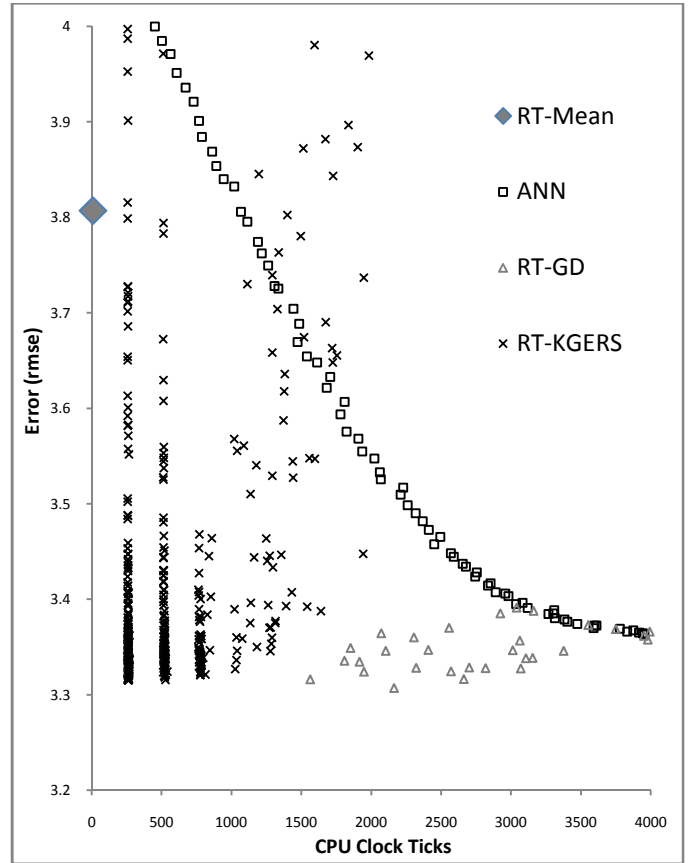


Figure 3. RMSE vs CPU for EDInpt

Table III summarizes RMSE for the four algorithms using three different sets of features for the EDInpt dataset. Our feature set is from Table II. The combined feature set has our features as well as the calendar features.

TABLE III. RMSE WITH DIFFERENT FEATURES

Algorithm	Our Features	Calendar Features	Combined Features
RT-KGERS	3.31	3.40	3.47
ANN	3.37	4.58	3.33
RT-MEAN	3.80	3.47	3.41
RT-GD	3.33	3.36	3.34

Overall, our results indicate that RT-KGERS with our feature set can achieve the lowest RMSE amongst the different combinations of algorithms and feature sets. Using either the calendar feature set or combined feature set does not seem to improve RMSE for RT-KGERS and RT-GD.

However, combining our features with calendar features seems to yield lower RMSE for ANN and RT-MEAN. When we compare the algorithms using the same feature sets, our results indicate that RT-KGERS is more accurate with our feature set or the calendar feature set, while ANN is more accurate with the combined feature set.

Note that our feature set has only 6 features, compared to 27 features in the calendar feature set and 33 features in the combined feature set. That is, the calendar feature set is more than 4 times larger than our feature set and the combined feature set is more than 5 times larger. Consequently, each algorithm would be at least 4 or 5 times slower.

Our features are “relative” values based on previously observed number of admissions (after a FFT analysis), while the calendar features are “absolute” values based on characteristics of the day/time. We thought the combined feature set would generally improve accuracy, but we did not observe that with RT-KGERS and RT-GD. This indicates that the 27 calendar features may not provide additional information about the data beyond the information captured by the 6 features in our feature set.

C. Live Trial Study with the Hospital

Currently, the hospital staff uses their collective memory to estimate patient volume and the appropriate staffing. After we showed our initial results from the EDInpt dataset to the hospital, they were eager to use our system. To evaluate how accurate our algorithms can estimate the number of ED inpatients in “real life,” we performed a “live” trial study with the hospital between February 25th and April 14th, 2011. Each week, we sent, via email, our estimates of the number of hospital admissions from the ED for the following week in 4-hour intervals. The estimates were based on models learned from two years of data prior to the week to be estimated. At the end of each week, the hospital provided the actual number of admissions, which we used to measure the accuracy of our estimates and added to the training set to build an updated model for estimating the number of admissions in the following week.

We used RT-KGERS with features in Table II. The results are summarized in Table IV. To help the hospital staff understand the results easier, we measure MAE (Mean Absolute Error) instead of RMSE. We also measure the % error, which is MAE divided by the average observed value. To help the hospital staff understand our estimates could have errors, we also provided them with a 90% confidence interval based on the range of errors incurred in the estimates.

The MAE ranges from 2.4 to 3.4 within the 7 weeks and the % error ranges from 19% to 30%. The error results from the live trial study are similar to our “offline” study discussed earlier. Generally, the live data have similar seasonal patterns as the historical data. The hospital had times that

were unusually busy or quiet (e.g. during April 1st to 7th), which caused large errors in some of the 4-hour intervals. Since the average number of patients in each interval is fewer than 10, one patient off is at least 10% in error.

TABLE IV. LIVE TRIAL STUDY WITH THE HOSPITAL

Week	Mean Absolute Error	90% confidence interval	% error
Feb 25 th – Mar 3 rd	2.8	-6.2 to 5.2	26%
Mar 4 th – Mar 10 th	2.7	-6.2 to 5.2	24%
Mar 11 th – Mar 17 th	2.9	-6.0 to 5.2	26%
Mar 18 th – Mar 24 th	3.2	-6.0 to 5.2	28%
Mar 25 th – Mar 31 st	2.5	-6.0 to 5.2	23%
Apr 1 st – Apr 7 th	3.4	-6.0 to 5.2	30%
Apr 8 th – Apr 14 th	2.4	-6.0 to 5.2	19%

The hospital staff thought our “live” estimates were useful for them. Though the live trial study was intended to evaluate how well our algorithms perform in the real world, one staff member said she considered using the “live” estimates to adjust staffing.

V. RELATED WORK

Much of the related work focuses on estimating presentations at the ED, not admissions from the ED. Batal et al. [4] propose using calendar and weather features to estimate daily ED presentations. They used linear regression and found that calendar features are more useful than weather features. Champion et al. [5] propose using ARIMA models to forecast monthly ED presentations. Jones et al. [6] investigated techniques, such as linear regression, SARIMA, exponential smoothing, time series regression and artificial neural network to forecast daily ED presentations. They found time series regression with calendar features seem to be more accurate. Hoot et al. [7] use discrete event simulation to forecast ED presentations 2 to 8 hours in advance. It is not clear how accurate their simulation can forecast one week in advance.

Fewer studies are on forecasting admissions. Jones et al. [8] use SARIMA to forecast daily number of occupied beds due to admissions from the ED. They found that forecasting occupancy is easier than admissions. Boyle et al. [9] propose using regression techniques to forecast monthly admissions from the ED. They found linear regression with calendar features yields the most accurate estimates. Leegon et al. [10] propose using age, gender, ICD-9 coded chief complaint, and other medical characteristics with artificial neural network to predict if the patient is going to be admitted. This can help predict the number of admissions in the next few hours, however, medical

characteristics of ED patients cannot be available a week in advance to adjust staffing.

For the related studies, features used to represent the data generally fall into three categories: previous admission numbers, calendar, and weather. Weather features are less applicable to our purposes because forecasting weather a week in advance itself is a very challenging task. Previous admission numbers and calendar features are more applicable and are investigated in this study. However, we use FFT to identify seasonal/cyclic behavior and select features accordingly. We also investigated “rate of change” (or “slope”) features to capture potential changes in the underlying population in the community. Rate of change features cannot be easily incorporated as calendar features.

Most of the studies estimate presentations or admissions in relatively long intervals—a day or a month. Estimates for these relatively long intervals are less useful to adjust staffing that is usually based on shifts shorter than a day (e.g. 8-hour). Also, we observe daily cycles in our data (more admissions during the day than at night), hence daily estimates do not help schedule staff during different times of the day. Finally, in general, daily measurements are easier to estimate than measurements for shorter (e.g. 4-hour) intervals, because daily measurements aggregate and smooth out the variability in shorter intervals. For example, consider 12-hour intervals versus daily intervals. If the system overestimates 10 patients in one 12-hour interval and underestimates 10 patients in the other, the system estimates perfectly in the 24-hour interval. Hence, the relatively lower reported error rates for daily (monthly) estimates in related work might not translate well to estimates for shorter intervals.

VI. CONCLUSION

We proposed a randomized algorithm RT-KGERS for non-linear regression to estimate ED admissions one week in advance. We evaluated our algorithm against three other algorithms. Our experimental results indicate that RT-KGERS can build regression models faster and more accurate. We also proposed features based on a FFT analysis on seasonal characteristics. We evaluated calendar features using the EDInpt data set and found that adding calendar features do not help improve accuracy for RT-KGERS. That is, the 27 calendar features do not seem to provide additional information beyond the information contained in our 6 features. Furthermore, our results indicate that RT-KGERS with our features is more accurate than other combinations of algorithms and features. After the hospital saw our initial results, they were eager to use our system. We subsequently conducted a live trial study to

evaluate our algorithm in “real life” and found accuracy levels similar to those obtained from historical data.

The biggest limitation with KGERS is stability—a randomized algorithm usually generates different outputs in different runs. KGERS algorithm uses an EPS design (equal probability of selection). One issue with an EPS design with respect to the hospital data is that several points occur more often than others. For example, the likelihood of having 25 patients is lower than that of having only 13 patients a time slice in the EDInpt. Sampling theory states that a biased population requires a biased sampling. Amongst the existent techniques are stratified sampling and clustering sampling. Stratified sampling will extract candidate subsets from those feature vectors that share similar semantics, in terms of their Y value. Clustering sampling can use the feature vectors, cluster them in groups and extract samples of these subgroups with similar probability of proportion. This will bias the training data in terms of sampling, so it reflects the true population; can potentially give more honest results. Another possibility for future work involves looking at Box Jenkins models [11], such as ARIMA. These models also use information about the distribution of noise in the data, calling it random shock modeling.

REFERENCES

- [1] ACEP Boarding Task Force, Emergency Department Crowding: High-Impact Solution, *American College of Emergency Physicians*, 2008.
- [2] Garcia K, Using a Randomized Regression Approach to Estimate Hospital Admissions to Reduce Emergency Department Holding, *Florida Institute of Technology, M.Sc. Thesis*, 2011;Chp 3, 5.
- [3] Mitchell T. *Machine Learning*. McGraw-Hill, 1997;Chp 4.
- [4] Batal H, Tench J, McMillan S, Adams J, Mehler PS. Predicting patient visits to an urgent care clinic using calendar variables, *Acad Emerg Med*, 2001;1:48-53.
- [5] Champion R, Kinsman L, Lee G, Masman K, May E, Mills T, Taylor M, Thomas PR, Williams R. Forecasting emergency department presentations. *Australian Health Review*, 2007; 31(1):83-90.
- [6] Jones SS, Thomas A, Evans RS, Welch SJ, Haug PJ, Snow GL. Forecasting daily patient volumes in the emergency department. *Acad Emerg Med.*, 2008;15(2):159-70.
- [7] Hoot NR, Epstein SK, Allen TL, Jones SS, Bauml KM, Chawla N, Lee AT, Pines JM, Klair AK, Gordon BD, Flottesmesch TJ, LeBlanc LJ, Jones I, Levin SR, Zhou C, Gadd CS & Aronsky D. Forecasting emergency department crowding: an external, multicenter evaluation. *Ann Emerg Med*, 2009;54(4):514-522.
- [8] Jones S, Joy M, Pearson J. Forecasting Demand of Emergency Care. *Health Care Management Science*, 2002;5(4):297-305.
- [9] Boyle J, Wallis M, Jessup M, Crilly J, Lind J, Miller P, Fitzgerald G, Regression Forecasting of Patient Admission Data, *30th Annual International IEE EMBS Conference 2008*;3819-3822.
- [10] Leegon J, Jones I, Lanaghan K., Aronsky D, Predicting Hospital Admission in a Pediatric Emergency Department using an Artificial Neural Network, *AMIA Annu Symp Proc.*, 2006; 1004.
- [11] Loh WY, Classification and Regression Tree Methods, *Encyclopedia of Statistics in Quality and Reliability*, Wiley, 2008; 315-32.