Predicting Hospital Length of Stay *(PHLOS)* : A Multi-Tiered Data Mining Approach

Ali Azari*, Vandana P. Janeja*, Alex Mohseni†
*Department of Information Systems
University of Maryland, Baltimore County (UMBC), Baltimore, MD 21250
Email: {azari2,vjaneja}@umbc.edu
†Emergency Medicine Associates, Germantown, MD 20874
Email: alexmohseni@gmail.com

***Abstract*—A model to predict the Length of Stay (LOS) for hospitalized patients can be an effective tool for healthcare providers. Such a model will enable early interventions to prevent complications and prolonged LOS and also enable more efficient utilization of manpower and facilities in hospitals. In this paper, we propose an approach for Predicting Hospital Length of Stay (PHLOS) using a multi-tiered data mining approach. In this paper we propose a methodology that employs clustering to create the training sets to train different classification algorithms. We compared the performance of different classifiers along several different performance measures and consistently found that using clustering as a precursor to form the training set gives better prediction results as compared to non-clustering based training sets. We have also found the accuracies to be consistently higher than some reported in the current literature for predicting individual patient LOS. The classification techniques used in this study are interpretable, enabling us to examine the details of the classification rules learned from the data. As a result, this study provides insight into the underlying factors that influence hospital length of stay. We also examine our results with domain expert insights.**

***Keywords*-Length of Stay; Predictive Models; Classification;**

I. Introduction

A model that helps to predict a patient's Length of Stay (LOS) during a single visit, the time from hospital admission until discharge, can be an effective tool in hands of healthcare providers to (a) plan for preventive interventions, (b) to improve health services, and (c) to manage the hospital resources more efficiently. The productivity of hospitals drop significantly in two situations: First, if the hospital is in short supply for required resources such as manpower and facilities. Second, if the hospital is over equipped and the supply is more than the demand. Both of these situations occur due to significant fluctuations in hospital occupancy, which seriously restricts the efficient scheduling for resource allocation and management. With an accurate estimation of how long patients will stay, the hospital can plan for a better bed management and more efficient resource utilization [1]. Predicting the possible discharge dates can lead to better estimation of available bed hours, which finally results to higher average occupancy and less waste of hospital resources [2], [3]. On the other hand, hospitals are continuously being expected to do more with ever diminishing resources. As Medicare reimbursement trends towards 'pay for performance', tying payments to efficiency, hospitals stand to lose a lot of money if they cannot predict and prevent excessively long LOS. Therefore, predicting the patients who need the most aggressive early intervention, and those who require a moderate amount of intervention to prevent prolonged LOS seems to be critical. Just as hospitals have created rapid response teams of clinicians to treat patients with decompensated disease, we believe hospitals could create rapid response care teams to intervene on any patient predicted to have a prolonged LOS. It remains to be seen, from a domain perspective, how much of an impact early focused intervention can have on preventing complications and prolonged LOS. In this paper we present a Multi-Tiered Data mining approach for Predicting Hospital Length of Stay (PHLOS), to reduce the uncertainty associated with the length of stay for hospitalized patients. Specifically, we make the following contributions:

- We identify groups of similar hospital claims using clustering, where the number of clusters is determined based on the disease conditions identified in the literature [4] or by using the Charlson index [5], which provides the general categories of the diseases.
- We utilize these groups to predict, with high accuracy, the LOS. Our accuracy exceeds that of several other models for predicting LOS of hospitalized patients, in the current literature [1], [6] and [7]. In addition we provide a method to rank multiple classifiers for different levels of clustering.
- Through clearly defined LOS classes, we provide a method to identify which patients need the most aggressive early interventions, and which patients require a moderate amount of interventions to prevent complications and long LOS.
- We validate our findings with a domain expert in the area of Emergency Medicine who is one of the co-authors of this paper.

In our approach, we first create clustering (using K-means clustering) of preprocessed data. We use different numbers of K, for example, corresponding to the conditions or corresponding to Charlson index [5]. Then we use these cluster assignments to identify the training data for the classifier. Additionally, we extract the test set, which is non-overlapping from the training sets. We run different

978-0-7695-4925-5/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDMW.2012.69

classifiers such as Bayesnet, SVM, JRIP, J48, Bagging to name a few, to predict LOS. We extract several performance measures for the classifiers such as Accuracy, Kappa Statistic, Precision, Recall and Area under the Curve (AUC). However the classifiers may perform differently in each of these and it can be confusing to identify which is the best for which level of clustering. Therefore, we rank these using the Friedman test [8] to identify, which classifier has the best outcome, for which level of clustering. Our results indicate that using clustering as a precursor to form the training set provides better results as compared to non-clustering based training sets. The results also demonstrate that Bayesian Network (Bnet), Support Vector Machine (SMO), JRip, Bagging, and J48 have better overall performance compared to other classifiers.

The rest of the paper is organized as follows. In section 2 we discuss related work, in section 3 we outline our approach, in section 4 we discuss the results and finally in section 5 we present conclusions and future directions.

II. Related Work

The advantages of knowing how long patients will stay in a hospital have been studied extensively. There are several studies since the 1960s trying to address this problem by building prediction models. In this section, we briefly introduce some of the studies that are directly related to the prediction of LOS.

Gustafson [1] proposed five methodologies for predicting hospital length of stay, two of them generate point estimates based on subjective judgement of surgeons and surgical residents, and the other three are distribution estimators that use the Bayesian theorem, which produce estimations based on empirical data and subjective judgements. The data is collected from eight inguinal herniotomy patients, and includes symptomatic and demographic data. Beside the fact that performance of the prediction models is not impressive, the low number of samples (8 patients) is another limitation of this study, causing doubt as to whether the results can be generalized. Woods et al [6] evaluated mortality and LOS prediction equations called Acute Physiology and Chronic Health Evaluation III (APACHE III). APACHEE III is a disease severity ranking classification, which is applied to the patients within 24 hours of admission to ICUs. A score between 0 and 299 is assigned to the patients based on some demographics and many symptomatic variables such as blood pressure, body temperature, heart rate, etc. [9]. Woods et al. compared the LOS predictions by APACHEE III with the actual length of stay in 22 Scottish intensive care units (ICU). The comparison results showed a weak correlation between the actual and predicted LOS. They concluded that there is a significant need for more accurate predictions in order to improve intensive care cost and quality. Liu et al. [10] built linear regression models to predict LOS based on a data set of 205,177 hospitalizations from 17 hospitals in Northern California. They also used logistic regression to predict the outlier LOS, longer than a week. In more sophisticated logistic regression models they added the Laboratory Acute Physiology Score (LAPS) and Comorbidity Point Score (COPS). These scores are calculated using different measures when a patient is hospitalized, and are proven to have excellent predictive value for a variety of clinical outcomes [10]. The results in all regression models showed that adding LAPS and COPS data to the models improved LOS predictions [10]. Kulinskaya et al. [11] used UK NHS data for the years 1997–1998 with 629,362 records. They compared many linear regression-based and maximum likelihood-based models, and found truncated maximum likelihood (TML) to have the best fitness value. However, the accuracy of the predictions that this model produces compared to actual length of stays is not discussed in the paper. In another study, David et al. [7] used Piecewise Exponential Model (PEM) to predict the LOS or mortality based on a data set of 2,646 seriously ill or injured patients in two trauma centers. The models calculated a risk factor (a probability score) for the chance of discharge or mortality in the first day of hospitalization and over each of the consecutive 8 days. Comparing the actual LOS with the LOS predicted by the model, they reported that their model showed an accuracy of 69% for cases who are younger than 55 years old, 13% for those between 55 and 70, and 17% for those older than 70 [7].

III. Multi-Tiered Data Mining

The overall approach is shown in Figure 1. We use hospital claims data for our approach to predict LOS. Generally such data has many missing values. We first perform data preprocessing on the original set of data points $D = \{d_1, \ldots, d_n\}$ where each d_i has m attributes such that $A = \{a_{i1}, \ldots, a_{im}\}$. Then relying on what we learned from literature as well as the judgement of our co-author who is an expert in the field, we select a subset of attributes $R^A = \{a_{iq}, \ldots, a_{iz}\}$ such that $R^A \subset A$ and $z < m$. We use the clustered data to create different training sets and a uniform test set. Then, we perform classification on these using different classifiers. We use a ranking method to identify which classifier and clustering level for training the data is the best in terms of multiple performance measures. Finally, we validate the results. As a test bed for our approach we have used the Heritage Health prize data [12]. We are not participating in the competition and are simply using this data as a testbed for our approach as it has individual claims for patient's visits to a hospital and also has the LOS. We next describe our approach in detail.

A. Creating Training and Test Sets

Classification involves two steps of training and testing [13]. First, a classifier is built by using a pre-existing set of data with labelled classes. In the learning step (training phase) a classification algorithm, builds a mapping (classifier) by learning the model from the training set. For example, the mapping is in the form of rules that identify a given hospitalization case to have a particular length of stay. In other words, these mappings use a set of attributes

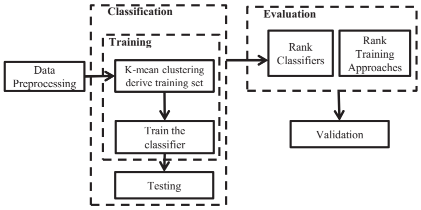

Fig. 1. PHLOS Approach

such as disease conditions, procedures to be performed, etc. to categorize the future claims to different LOS. The training tuples have a direct impact on the performance of the classifiers. Therefore, it is critical to identify the right training tuples to attain a high accuracy. In order to train the classification algorithms in this research, we designed four approaches. In the first approach (No-Cluster), we do not perform clustering and simply select the training tuples in a random manner from the original data set we used. In the three clustering-based training approaches we applied K-means as a precursor to form training sets where K is determined based on different criteria such as the number of disease conditions. We discuss each of the K-means variations below:

1) Clustering based training sets

In order to select representative training tuples, we apply k-means clustering on raw data. This will result in clusters, where they are similar to one another within the cluster, but dissimilar to the objects in other clusters [13]. We select K based on the following criteria for generating different training sets (TR):

- TR1: K= Number of Conditions [4]
- TR2: K= Number of Charlson index codes [5]
- TR3: K= ideal K determined using variation in Sum of Squared Errors (SSE)

We perform clustering on D which results in $C = \{c_1, \ldots, c_k\}$ where k is varied. We select the training tuples $TR_i = \{tr_{i1}, \ldots, tr_{in}\}$ from $\{c_1, \ldots, c_k\}$ such that $\{tr_{i1}, \ldots, tr_{in}\}$ are tuples that are as close as possible to the centroid of each cluster $\{c_1, \ldots, c_k\}$. Thus, we would have training tuples that are very similar to the other tuples within the same cluster and very dissimilar from the tuples belonging to other clusters. This will lead to selecting a more distinctive set of training rules and potentially will lead to better classifier performance. Based on this method and the insight we gained from the in-depth interview with the medical expert, we created three different training sets, called TR1, TR2, and TR3 as described next. In addition to the training sets, we select a set of random test tuples to test the accuracy of the classifiers created with each of these training sets. However, the selection of the test set tuples is done in such a way that test tuples are non overlapping with training sets. We select the test data $T = \{t_1, \ldots, t_x\}$ such that each t_i is randomly selected from $D = \{d_1, \ldots, d_n\}$ and $T \bigcap TR_1, TR_2, TR_3 = \Phi$. We next discuss each training set in detail.

2) Training Set 1 (TR1:K=number of conditions)

One of the most important factors that determine the LOS for each record in the data is the condition for which the patient is hospitalized. A patient who has a heart condition and is hospitalized for open heart surgery procedure is expected to stay for a longer time compared to a patient who is hospitalized for pregnancy. Thus, we consider that the data can be partitioned into groups corresponding to the number of medical conditions in the data. In emergency medicine when a patient is seen by a physician they have to enter a diagnosis for the patient. This diagnosis corresponds to an ICD code which is the International Classification of Diseases (ICD) code provided by World Health Organization [14]. There are 14,400 diagnosis codes in ICD-10 divided into several categories. For example, in Heritage Health Data [12], these ICD codes are categorized into 44 groups of general conditions based on a research [4], each general primary condition corresponds to a group of the ICD codes Therefore, this training data is derived after we perform K-means with K= number of conditions in the data. For each cluster $\{c_1, \ldots, c_k\}$, We select n training tuples such that $\{tr_{11}, \ldots, tr_{1n}\}$ are tuples that are as close as possible to μ_x where μ_x is the centroid of the cluster c_x.

3) Training Set 2 (TR2:K=Charlson index)

Patients presenting a heart condition may be impacted by other diseases, referred to as comorbidity, such as diabetes, obesity, etc. Charlson et al, (1987) [5] proposed a formal generalization of the diagnosis codes in the form of a categorized comorbidity score, called Charlson Index. These are 4 distinct values based on the risk associated with the comorbidity of the disease for which the patient is being admitted. We partition the data using this index with K=4. We use the clusters from this to identify the training dataset TR2.

4) Training Set 3 (TR3: IdealK)

In the third method we look for the ideal k, which partitions the data in such a way that the cluster errors are minimized as much as possible. For this we start with $K = 1$ and vary K to a sufficiently large number resulting in an SSE curve in the form of a decaying function. Given dataset $D = \{d_1, \ldots, d_n\}$ each d_i has m attributes such that $A_i = \{a_{i1}, \ldots, a_{im}\}$, the data is partitioned into clusters $C = \{c_1, \ldots, c_k\}$ where each $c_x = \{d_{x1}, \ldots, d_{xl}\}$ and μ_x is the centroid of cluster c_x, then sum of square error (SSE) is sum of squared differences between each data point in the cluster and μ_x. Thus $sse_x = \sum_{c_x=1}^{k} \sum_{1}^{l} (d_{xi} - \mu_x)$. So essentially SSE is very large when all data points are in one cluster, but ≈ 0 as individual points are in each cluster of their own. We vary K=z where $z = \{l, \ldots, z\}$, z is a large number such that $z \approx n$ and $SSE = \{sse_l, \ldots, sse_z\}$ such that $sse_z \approx 0$, and sse_l is a large number. Therefore, sse values are generally defined by a decay function such that

$sse_y \approx sse_{y-1}e^{-\lambda y}$, where λ is a constant for decay. Heuristically the ideal value of K is the mid point in the curve of the SSE values plotted against z. Thus, $I^k \approx sse_{\{(z-l)/2\}}$ where the SSE values start to stabilize. We call this mid point as the IdealK, and we use it in K-means clustering such that $K = I^k$ where $0 < K << n$. We use the clusters derived using IdealK to generate another training data set.

B. Predictive Models

We design our approach in a way that uses a diverse set of classifiers in order to evaluate which one is more appropriate for our task. For this purpose, algorithms that create interpretable models such as J48 as well as algorithms that create more complicated and less interpretable models such as ensemble methods are included. We use several classification algorithms including, J48 [15], jRIP [16] classification rules, K-Nearest Neighbours [17], Logistic regression [18], Naive Bayes [19], Support Vector Machines (SVM) [20], Bayesian Networks [21], Bagging [22], Random Forests [23] and Boosting [24]. We perform classification using different classifiers and different training datasets discussed above. Thus, various predictive models $M = \{m_1, \ldots, m_i\}$ are formed and each model generates a set of performance measures $pm_i = \{p_{i1}, \ldots, p_{iz}\}$. For example , m_1 could be J48 applied to TR1. We evaluate each model with five performance measures. We use traditional performance measures for classification that are based on the four values of the confusion table: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). We use these values to compute accuracy (A = (TP+TN)/ (TP+TN+FP+FN)), precision (P = TP/ (TP+FP)) and recall (R = TP/ (TP+FN)). In addition, we also use the Kappa statistic [13], which measures the degree of consistency between the predicted and observed values. The Kappa statistic evaluates the model to see if the result is indeed a true outcome or occurring by chance. We also use Area Under the Curve (AUC) [13]. The AUC represents the area under the Relative Operating Characteristic(ROC) curve. ROC gives the trade-off between true positive rate and false positive rate for a given model. In classification, the closer AUC is to 0.5, the less accurate the corresponding model is. A model with perfect accuracy will have an area of 1.0 [13].

C. Rank Classifiers and Training levels

In this study, various models $M = \{m_1, \ldots, m_i\}$ are formed, each model m_i generates a set of performance measures $pm_{ij} = \{p_{i1}, \ldots, p_{ij}\}$. For our approach j=5,including accuracy, recall, precision, Kappa statistic and AUC. Having a variety of performance measures, it is difficult to simply answer the question, which classification algorithm performs better across the different measures? Thus, we use a ranking method to identify, which classifier and clustering level for training the data is the best for the classification task. $R(M, pm)$ is a ranking function, which produces an ordered set M^o. The algorithm or the training approach with the best overall performance gets the rank of 1, the second best gets the rank of 2 and so on. We use the Friedman test [8] to perform this ranking. Friedman Test is a non-parametric test used to check whether differences between columns (test cases) across multiple rows are statistically significant.

TABLE I
HERITAGE HEALTH DATA ATTRIBUTES AND ABBREVIATIONS

Table name	Attribute name	Attribute description
Claim	Specialty	The specialty of the service provider e.g. internal, laboratory, and etc.
Claim	LengthOfStay	The number of days of stay in hospital for the claim
Claim	DSFS	days since first service that year
Claim	Primary Condition Group	a generalization of the primary diagnosis codes
Claim	CharlsonIndex	a generalization of the diagnosis codes in the form of a categorized comorbidity score

D. Validation

We consider two ways to validate our results using (1) a Uniform test set to test across all the models, and (2) domain expert input. We randomly select tuples from the original data to create a test set. We use the same test set to test all the training models to have a uniform testing strategy. Secondly, we validate our findings with our co-author who is a domain expert in this area. For this we select the tuples that we have classified and pick certain conditions, for example, Renal failure. We compare our LOS prediction with the estimate of the domain expert, to see how we did qualitatively in terms of the opinion of the domain expert. Although this is not a quantitative measure it helps us to measure the benefit of this approach from the perspective of domain expert.

IV. EXPERIMENTAL RESULTS

We designed the study in a way that uses a diverse set of classifiers in order to evaluate which one is more appropriate for our task. For this purpose, algorithms that create interpretable models such as J48 as well as algorithms that create more complicated and less interpretable models such as ensemble methods are included. We use the WEKA Data Mining package for the clustering and classification task [25].

A. Data Description

As a testbed for our approach we use the Heritage Health Prize data [12]. We did not participate in the competition but have purely used this data as a testbed for evaluating our approach. The data set released in three phases and six tables. The data set includes 1,048,576 records of claims in 3 years. We focus on certain hospital claims related attributes as shown in Table I.

B. Data Preprocessing

The primary problem with the data set was the extremely high number of missing values. Since the purpose of this study is to classify LOS classes, and about 97 % of the claim records were missing a value for the LOS, we decided to eliminate all the claim records (rows) with a missing LOS value. There is no explanation about why the rate of records with missing LOS is so high in Heritage Health Prize's documentation, but one reason could be that there are multiple claim records for a single hospitalization case and only the main claim that is usually related to 'Internal' speciality has an LOS to be recorded, and all other claims such as those related to 'pathology', 'Anaesthesiology', 'Laboratory', 'Diagnostic Imaging', and 'General practice' are missing a value for the LOS column. for all other attributes with few missing values, we used a global constant, 'unknown', to fill in for missing values. As a result, the remaining number of records of claims after elimination adds up to 28,362 records. Selecting appropriate attributes for the classification task was another important issue that directly influences the success of the prediction models. Since we believe that selecting attributes to predict LOS requires a good knowledge of the domain, we conducted an interview with an emergency medicine specialist in order to identify the attributes in the Heritage Health Prize dataset that are expected to contribute to our prediction models. As a result, we decided to use some attributes in the Heritage Health dataset as it is presented in Table I. We divided LOS into three different functional groups: First we merged LOS of 1 and 2 days into one bin representing those patients for whom minimal intervention is required to reduce LOS. Next, we merged all the records with more than 2 and less than 7 days of LOS and put them into the second bin, labeled '1 week or less.' These patients make up a large proportion of patients and could benefit from moderate interventions to reduce their LOS. Lastly, we put all the length of stays longer than a week into the third bin labeled 'more than a week.' These patients represent the greatest need for early aggressive intervention to prevent costly prolonged hospital admissions. This pragmatic approach works well because it directly translates into the type of action hospital staff need to take. Some other studies also consider length of stays more than a week as outliers because of high amount of needed interventions [10]

C. K-Means Clustering to Derive Training and Test Sets

1) Training Set 1 (TR1:K=44)

Because we have 44 different conditions in our original dataset, in this first training set, we set K to be 44, WEKA runs the K-means in 29 iterations and resulting SSE was 1,376.6. Later on, we sorted the data in each cluster, and selected about 6 tuples as close as possible to the cluster centroids and formed the first training set (TR1) with 274 records.

2) Training Set 2 (TR2:K=4)

We next run K-means clustering with a K equals to 4 based on the 4 Charlson index values. It took WEKA 15 iteration to calculate an SSE of 5995.72 for the new clustering. We followed the same steps that we took to create TR1 in order to create training set 2 (TR2) with 275 tuples.

3) Training Set 3 (TR3:K=9)

As it is explained before (section 3.1), we see a dramatic drop in the value of SSE for K equal to 9, therefore we selected 9 as the ideal value of k. It took WEKA 25 iteration to compute an SSE of 3577.56. Then we created training set 3 (TR3) with 279 tuples as it was described earlier.

The corresponding SSE values for clustering tasks for k equal 4, 9, and 44 are relatively large values.Thus, we conjecture that the clusters are heterogeneous inside, and there is a high degree of diversity within clusters. Therefore, we kept this point in mind, and selected the training tuples from the points that are very close (or similar) to each cluster's centroids to have well defined training tuples.

4) Random training Set (No_cluster)

In order to figure out if creating training sets from clustered data has a significant impact on the performance of the predictive models, we make a fourth training set. We selected 275 training tuples from the original data set in a completely random manner, and compare the resulting performance measures with other training approaches e.g., TR1, TR2, and TR3.

5) Test Set

The test set is made up of 153 test tuples and their associated class labels; we selected these tuples randomly from the general dataset. The same test set is supplied to WEKA to test all the predictive models built in this study.

D. Classification

We mentioned that we designed 4 different training sets, to train the classification algorithms; we used these to train ten different classification algorithms that gave us 40 models for predicting the LOS. We set the algorithm parameters in our experiments to the default values. However, there are some exceptions: nearest neighbours is set to 4 for the KNN classifier (iBK). 4 tuples is set as the minimal number of tuples required for forming a leaf for tree algorithms. In the bagging algorithm, we use J48 as the base algorithm with four minimum tuples to form a leaf. The number of iterations is set to 10, and the size of each bag is 100%. Tree pruning is set to true for tree algorithms with the goal of improving classification accuracy on unseen data [13]. For Adaboost ensemble algorithm we set the resampling to true.

In this section we first analyze and compare different predictive models using different performance measures. We present the performance of each classification model in terms of their accuracy, Kappa statistic, precision, recall, and AUC. Table II shows the results grouped in frames (a) to (e), by performance measures. Each column contains the results of one algorithm. Each row shows the results of one training set approach.

Table II (a) shows the accuracy measures of different predictive models built in this study. The maximum accuracy score belongs to JRip (0.743), when we trained it with training set 1 (TR1), the lowest accuracy belongs to iBK (0.42) when we used NO_Cluster to train it. The highest average accuracy among four different training approaches belongs to TR1 where the average accuracy for all models is 0.648. To explore more about the accuracy of the predictions, we downloaded the test files with actual

TABLE II

Comparison of the performance measures for the predictive models built in this study

	j48	iBk	NB	JRip	SMO	LogR	Adaboost	Bag48	RF	Bent
	(a)Accuracy									
No_Cluster	53.290	42.105	53.947	51.974	57.895	57.895	53.29	53.290	57.237	59.211
TR1_k44	62.50	59.211	69.737	74.342	71.711	61.184	55.263	63.816	59.868	70.395
TR2_k4	63.158	57.895	59.211	61.184	58.552	57.895	53.29	62.50	52.63	59.868
TR3_k9	72.368	59.211	59.211	71.053	72.368	60.526	55.263	69.079	61.184	61.842
	(b) Kappa statistic									
No_Cluster	0.220	0.000	0.243	0.214	0.318	0.343	0.220	0.220	0.308	0.343
TR1_k44	0.416	0.366	0.537	0.598	0.564	0.404	0.246	0.436	0.376	0.547
TR2_k4	0.444	0.372	0.395	0.401	0.384	0.373	0.220	0.438	0.293	0.403
TR3_k9	0.580	0.350	0.382	0.550	0.579	0.399	0.246	0.532	0.401	0.425
	(c) Precision									
No_Cluster	0.418	0.177	0.474	0.527	0.575	0.552	0.418	0.418	0.546	0.524
TR1_k44	0.634	0.603	0.691	0.749	0.707	0.602	0.513	0.644	0.592	0.698
TR2_k4	0.637	0.605	0.630	0.660	0.629	0.612	0.418	0.651	0.538	0.633
TR3_k9	0.724	0.563	0.585	0.709	0.723	0.606	0.513	0.693	0.601	0.624
	(d) Recall									
No_Cluster	0.533	0.421	0.539	0.520	0.579	0.579	0.533	0.533	0.572	0.592
TR1_k44	0.625	0.592	0.697	0.743	0.717	0.612	0.553	0.638	0.599	0.704
TR2_k4	0.632	0.579	0.592	0.612	0.586	0.579	0.533	0.625	0.526	0.599
TR3_k9	0.724	0.592	0.592	0.711	0.724	0.605	0.553	0.691	0.612	0.618
	(e) AUC									
No_Cluster	0.664	0.678	0.710	0.615	0.669	0.716	0.667	0.667	0.758	0.729
TR1_k44	0.766	0.759	0.804	0.813	0.792	0.777	0.704	0.806	0.741	0.811
TR2_k4	0.769	0.797	0.818	0.781	0.76	0.782	0.608	0.806	0.678	0.815
TR3_k9	0.813	0.768	0.799	0.812	0.813	0.757	0.704	0.835	0.750	0.803

and predicted LOS columns after the classification task was performed using Jrip algorithm trained by TR1. We calculated the accuracy scores for the test tuples related to each class label separately. We observe that the class label '3 days to 1 week' has lower accuracy 35% as compared to other labels (81% and 85%). In general, we noticed that the training tuples for the class label '3 days to 1 week' belong to the clusters with higher SSE values. We conjecture that high degree of heterogeneity within those clusters leads to a high degree of dissimilarity between the selected training tuples, which resulted in a lower accuracy of the predictions for '3 days to 1 week' LOS. In addition, within this class, we also found that false predictions are mostly among the tuples where 'Speciality' equals to 'Internal', but the primary conditions among the falsely predicted tuples varies. Table II (b) shows the Kappa statistics scores. The kappa statistic measures the agreement of prediction with the true class labels. A score value of 1.0 signifies complete agreement, and a value greater than 0 means that the classifier is doing better than pure random behavior. The results show that the maximum kappa score belongs to JRip (0.598) ensemble classifier when we used TR1 to train the algorithm. The highest average Kappa score among four different training approaches belongs to TR1 where the average Kappa score for all models is 0.449. Table II (c) shows the precision measures of different predictive models built in this study. The highest precision score again belongs to JRip when we used TR1 to train it. The highest average precision score among four different training approaches belongs to TR1 where the average precision score for all models is 0.643. Table II (d) shows the recall measures of different predictive models. The maximum recall score belongs to JRip when we used TR1 to train the algorithm (0.743), and the lowest recall score is for iBK when we used NO_Cluster to train the algorithm. The highest average recall score among four different training approaches belongs to TR1 where the average precision score for all models is 0.648. Table II (e) shows the AUC measures of different predictive models. The maximum score belongs to Bag48 when we used TR3 to train the algorithm (0.835), and the lowest score is for Adaboost (0.608) when we used TR2 to train the algorithm. The highest average AUC score belongs to TR3 where the average AUC score equals to 0.785.

E. Rank Classifiers

Because of the multiple training approaches, it is not easy to say which algorithm has better accuracy across all training approaches. For example JRip when trained by TR1 is the leading algorithm in terms of producing the most accurate predictions as discussed above. However, it is not the number one algorithm if it is trained by TR2, TR3 or No_Cluster training sets. We ran Friedman test using all the accuracy measures in frame (a) of Table II to see which algorithm produce the most accurate predictions overall. We use SPSS statistical package to run the Friedman test. The null is 'there is no significant difference between accuracy of predictions produced by different algorithms. Although JRip when trained by TR1 (see Table II (a)) has the highest value for the accuracy score, Table III shows SMO and Bnet produce predictions with highest accuracy overall. We ran several other Friedman tests for Kappa statistic, precision, recall, and AUC, using the measures in frame (b) to (e) in Table II respectively. The corresponding

TABLE III
RANKING OF CLASSIFICATION ALGORITHM ACCORDING TO EACH PERFORMANCE MEASURE

	Accuracy	Kappa statistic	Precision	Recall	AUC
SMO	1	2	2	1	5
Bnet	2	1	3	2	1
J48	3	3	4	3	8
Jrip	4	4	1	4	4
BagJ48	5	5	5	5	2
NB	6	6	7	6	3
LogR	7	7	6	7	6
RF	8	8	8	8	9
iBK	9	9	9	9	7
AdaBoost	10	10	10	10	10

TABLE IV
RANKING OF CLASSIFICATION ALGORITHMS

Algorithms	Ranks	Mean Rank
Bnet	1	7.68
SMO	2	7.53
Jrip	3	7.05
BagJ48	4	6.63
j48	5	6.53
NB	6	5.75
LogR	7	5.23
RF	8	4.05
iBK	9	2.80
AdaBoost	10	1.78

ranks are demonstrated in Table III. SMO gets the best rank for accuracy and recall, Bnet gets the best rank for Kappa statistic and AUC score, and Jrip gets the best rank for precision.

Having different performance measure such as precision, accuracy, recall, Kappa, and AUC, it can get difficult to identify which classification algorithm performs better across all measures. Some models show high accuracy scores but low scores in other measures. For example, SMO is the leading algorithm in producing predictions with high accuracy (see Table III), but it is the fifth best algorithm in terms of AUC score, or J48 ranks third in terms of accuracy but it ranks eighth in terms of AUC score. In order to interpret all these performance measures and make a decision about which classification algorithm has overall better performance, we used Friedman test to create a composite rank of performance measures. Our Null hypothesis for this test is: there is no significant difference between performance of different classification algorithms in terms of accuracy, precision, recall, Kappa statistics and RUC score combined. The output of SPSS shows that the p-value of Friedman test is 0.00, which means Null hypothesis (H_0) is rejected. We can clearly see that there is a significant difference between performance of different classification algorithms. The respective ranks of each classification algorithms have been shown in Table IV. Comparing the results of Table III and Table IV, it is clear that although SMO has the best performance in terms of accuracy and recall, Bnet is the best ranked algorithm overall. According to the Friedman rankings, BNet, SMO and JRip generate relatively better LOS predictions.

TABLE V
RANKING OF TRAINING APPROACHES

	Ranks	Mean Rank
1	TR3_k9	3.27
2	TR1_k44	3.18
3	TR2_k4	2.33
4	No_Cluster	1.22

F. Rank Training Approaches

We made an assumption that clustering the data set, and creating training sets from tuples that are close to the centroid of clusters improves the performance of prediction models. We use the Friedman test to examine the validity of this assumption. Friedman ranks the performance of each training approach across all different classification algorithms and for different performance measures. Thus, the best performing training approach gets the rank of 1, the second best the rank of 2 and so on. Our Null hypothesis for this test is: there is no significant difference between performances of predictive models in terms of the way we trained the classifiers. According to the results of SPSS the p-value of Friedman test is 0.00, which means Null hypothesis (H_0) is rejected. We can clearly see that there is a significant difference between performance of different training approaches. The respective ranks of each training approaches are shown in Table V. According to the Friedman rankings, using TR3 and TR1 as the training set consistently leads to better predictive results. Therefore, it is relatively better if we cluster the dataset using the ideal k value, and then select the training tuples. Using TR3 still produces significantly better predictions compared to No_Cluster but not as good as TR1 and TR2. Thus, selecting the training tuples randomly from the data set produced the poorest prediction results. As a conclusion, the results of Friedman test validate our hypothesis that selecting the training tuples from the data points close to the cluster centroids contributes to a better prediction performance.

G. Validation

We validate our findings with our co-author who is a domain expert in this area. We mapped two randomly selected primary conditions, namely 'HEART4', 'RENAL2' to the corresponding ICD-9 codes based on an existing study [4], then we compared the predictions made by our models with domain expert's insight and experience. 'HEART4' is mostly consist of diagnoses codes related to conditions that often might take 1-2 days according to our expert's experience, for example: patient with diagnose code 451.19 (Deep vein thrombosis, other leg veins) commonly stay only 1 day, although rarely 3+ days, and those with diagnose code 455.4 (Haemorrhoids, external thromboses) commonly stays 1 day. In our models, the tuples with 'HEART4' primary condition are predicted to be of '1 or 2 days' and rarely '3 days to 1 week, which agrees with domain experience and knowledge. The second primary condition we examined is 'RENAL2'. For this condition, the most important ICD-9 code is probably 585 (chronic renal failure), as clarified by the expert. Dialysis patients tend to have significant comorbidity and high risk for medical complications. LOS of 'more than a week' for this group of patients is expected. However, a significant proportion of these dialysis patients, typically non-compliant dialysis patients who missed dialysis, will get admitted for just '1 or 2 days'. In our models, tuples with 'RENAL2' are all predicted to be 'more than a week' because most the tuples corresponding to this condition in the training sets have the same class label. Although the expert knowledge validates long LOS for majority of the cases, it causes false predictions for smaller but significant percentage of patients that are admitted only for '1 or 2 days'. We conclude that for some primary conditions, there are windows of anomalies that have unfortunate consequence for the majority of tuples when we made the training sets. These anomalous tuples have been hidden in the much larger class when we perform clustering and as a result are not part of the training dataset.

We propose that future research should address this problem. There can be a few ways to deal with this type of scenario. First we have used K-Means which does not address outliers or anomalous groups of outliers. Using another clustering method, such as DBSCAN [26], may lead to better outcome. Second, it may be possible that the accuracy varies by condition. Thus, we could separate the data before clustering. For example, if we separate the data by Charlson index and then apply the clustering we may get a more representative training sets. Third, we can potentially address anomaly detection as a preprocessing step and then perform our approach. We intend to explore these aspects in future research.

H. Repeatability

The dataset used in this research is publicly available from [12]. The PHLOS approach can be independently duplicated by other researchers by using the parameters specified for creating the training sets, classification and clustering techniques and ranking of the training approaches and classification algorithms in WEKA [25] as described earlier in this section. In addition we have posted a sample dataset with parameter settings to replicate the approach at [27].

V. Conclusion

There is a strong demand to make more accurate and robust models to predict LOS. So far, the prediction models are built using statistical methods, which are mostly based on regression and correlation analysis. In this paper, we propose a novel multi-tiered data mining approach to predict the LOS using a wide range of clustering and classification techniques. The accuracy of predictions made by our approach exceeds that of several other methods for predicting the LOS. Implementing this model can enable efficient management of hospital resources and planning for preventive interventions for patients with intense conditions. As a result, this study provides better insight into the underlying factors that influence hospital length of stay.

References

[1] D. H. Gustafson, "Length of stay prediction and explanation," *Health Services Research*, vol. 37, no. 3, pp. 631–645, 2002.

[2] G. H. Robinson, L. E. Davis, and R. P. Leifer, "Prediction of hospital length of stay." *Health Services research*, vol. Winter, no. 1, p. 287, 1966.

[3] G. H. Robinson, L. E. Davis, and G. C. Johnson, "The physician as an estimator of hospital stay." *Human factors*, vol. 8, no. 3, pp. 201–208, 1966.

[4] G. J. Escobar, J. D. Greene, P. Scheirer, M. N. Gardner, D. Draper, and P. Kipnis, "Risk-adjusting hospital inpatient mortality using automated inpatient, outpatient, and laboratory databases," *Medical care*, vol. 46, no. 3, pp. 232–239, 2008.

[5] M. E. Charlson, P. Pompei, K. L. Ales, and C. MacKenzie, "A new method of classifying prognostic comorbidity in longitudinal studies: Development and validation," *Journal of Chronic Diseases*, vol. 40, no. 5, pp. 373 – 383, 1987.

[6] A. W. Woods, F. N. MacKirdy, B. M. Livingston, J. Norrie, and J. C. Howie, "Evaluation of predicted and actual length of stay in 22 scottish intensive care units using the apache iii system." *Anaesthesia*, vol. 55, no. 11, pp. 1058 – 1065, 2000.

[7] D. E. Clark and L. M. Ryan, "Concurrent prediction of hospital mortality and length of stay from risk factors on admission," *Health Services Research*, vol. 3, no. 1, pp. 12–34, 1968.

[8] E. L. Lehmann, *Nonparametrics: Statistical Methods Based on Ranks.*, ser. San Francisco. San Francisco: McGraw-Hill, 1985.

[9] W. A. Knaus, D. P. Wagner, E. A. Draper, J. E. Zimmerman, M. Bergner, P. G. Bastos, C. A. Sirio, D. J. Murphy, T. Lotring, , and A. Damiano, "The apache iii prognostic system. risk prediction of hospital mortality for critically ill hospitalized adults," *CHEST*, vol. 100, no. 6, pp. 1619 – 1636, 1991.

[10] V. Liu, P. Kipnis, M. K. Gould, and G. J. Escobar, "Length of stay predictions: Improvements through the use of automated laboratory and comorbidity variables," *Medical care*, vol. 48, no. 8, pp. 739–744, 2010.

[11] E. K. Kulinskaya and H. D. Gao, "Length of stay as a performance indicator: robust statistical methodology," *IMA JOURNAL OF MANAGEMENT MATHEMATICS*, vol. 16, no. 4, pp. 369–381, 2005.

[12] HHP, "Heritage health prize," http://www.heritagehealthprize.com.

[13] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, ser. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011.

[14] W. H. O. (WHO), "International classification of diseases (icd)," http://www.who.int/classifications/icd/en/.

[15] J. R. Quinlan, *C4.5: Programs for Machine Learning*, 1993.

[16] W. W. Cohen, "Fast effective rule induction," in *International Conference on Machine Learning*, 1995, pp. 115–123.

[17] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, 1991.

[18] D. Hosmer and S. Lemeshow, *Applied logistic regression*, ser. Wiley series in probability and statistics: Texts and references section. Wiley, 2000.

[19] J. GH and L. P, "Estimating continuous distributions in bayesian classifiers," in *11th conference on uncertainty in artificial intelligence*, 1995, p. 338345.

[20] N. Saravanan, V. N. S. K. Siddabattuni, and K. I. Ramachandran, "A comparative study on classification of features by svm and psvm extracted using morlet wavelet for fault diagnosis of spur bevel gear box," *Expert Systems With Applications*, vol. 35, pp. 1351–1366, 2008.

[21] R. R. Bouckaert, "Bayesian network classifiers in weka for version 3-5-5," 2006.

[22] L. Breiman, "Bagging predictors" machine learning," 1996.

[23] Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[24] Y. Freund and R. Schapire, "Some experiments with a new boosting algorithm," in *International Conference on Machine Learning*, 1996.

[25] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools And Techniques*, ser. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufman, 2005.

[26] M. Ester, H. peter Kriegel, J. S, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." AAAI Press, 1996, pp. 226–231.

[27] PHLOS, "Phlos reapitability for icdm2012," https://sites.google.com/site/phlosicdm2012/home.