Early Deterioration Warning for Hospitalized Patients by Mining Clinical Data

Yi Mao[1,2], Yixin Chen[1], Gregory Hackmann [1], Minmin Chen [1], Chenyang Lu [1], Marin Kollef [3], Thomas C. Bailey [3]

[1] Department of Computer Science and Engineering, Washington University in St. Louis, Saint Louis, USA

[2] School of Electromechanical Engineering, Xidian University, Xi'an, China

[3] Department of Medicine, Washington University School of Medicine, Saint Louis, USA

Abstract

Data mining on medical data has great potential to improve the treatment quality of hospitals and increase the survival rate of patients. Every year, 4--17%of patients undergo cardiopulmonary or respiratory arrest while in hospitals. Early prediction techniques have become an apparent need in many clinical areas. Clinical study has found early detection and intervention to be essential for preventing clinical deterioration in patients at general hospital units. In this paper, based on data mining technology, we propose an early warning system (EWS) designed to identify the signs of clinical deterioration and provide early warning for serious clinical events.

Our EWS is designed to provide reliable early alarms for patients at the general hospital wards (GHWs). The main task of EWS is a challenging classification problem on high-dimensional stream data with irregular, multi-scale data gaps, measurement errors, outliers, and class imbalance. In this paper, we propose a novel data mining framework for analyzing such medical data streams. The framework addresses the above challenges and represents a practical approach to early prediction and prevention based on data that would realistically be available at GHWs.

We assess the feasibility of the proposed EWS approach through retrospective study that includes data from 41,503 visits at a major hospital. Finally, we apply our system in a clinical trial at a major hospital and obtain promising results. This project is an example of multidisciplinary cyber-physical systems involving researchers in clinical science, data mining, and nursing staff. Our early warning algorithm shows promising result: the transfer of patients to ICU was predicted with sensitivity of 0.4127 and specificity of 0.950 in the real time system.

Keyword: Early Warning System, Logistic Regression, Bootstrap Aggregating, Exploratory undersampling, EMA (exponential moving average)

Ⅰ. Introduction

Within the medical community, there has been significant research into preventing clinical deterioration among hospital patients. Data mining on electronic medical records has attracted a lot of attention but is still at an early stage in practice. Clinical study has found that 4--17% of patients undergo cardiopulmonary or respiratory arrest while in the hospital (Commission, 2009). Early detection and intervention are essential to preventing these serious, often life-threatening events. Indeed, early detection and treatment of patients with sepsis has already shown promising results, resulting in significantly lower mortality rates (Jones, 08).

In this paper, we consider the feasibility of an Early Warning System (EWS) designed to identify at-risk patients from existing electronic medical records. Specifically, we analyzed a historical data set provided by a database from a major hospital, which cataloged 41,503 hospital visits between July 2007 and July 2011. For each visitor, the dataset contains a rich set of electronic various indicators, including demographics, vital signs (pulse, shock index, mean arterial blood pressure, temperature, and respiratory rate), and laboratory tests (albumin, bilirubin, BUN, creatinine, sodium, potassium, glucose, hemoglobin, white cell count, INR, and other routine chemistry and hematology results). All data contained in this dataset was taken from historical EMR databases and reflects the kinds of data that would realistically be available at the clinical warning system in hospitals.

Our EWS is designed to provide reliable early alarms for patients at the general hospital wards (GHWs). Unlike patients at the expensive intensive care units (ICUs), GHW patients are not under extensive electronic monitoring and nurse care. Sudden deteriorations (e.g. septic shock, cardiopulmonary or respiratory arrest) of GHW patients can often be severe and life threatening. EWS aims at automatically identifying patients at risk of clinical deterioration based on their

existing electronic medical record, so that early prevention can be performed. The main task of EWS is a challenging classification problem on high-dimensional stream data with irregular, multi-scale data gaps, measurement errors, outliers, and class imbalance.

To address such challenges, in this paper, we first develop a novel framework to analyze the data stream from each patient, assigning scores to reflect the probability of intensive care unit (ICU) transfer to each patient. The framework uses a *bucketing* technique to handle the irregularity and multi-scaleness of measuring gaps and limit the size of feature space. Popular classification algorithms, such as logistic regression and SVM, are supported in this framework. We then introduce a novel bootstrap aggregating scheme to improve model precision and address over-fitting. Furthermore, we employ a smoothing scheme to deal with the outliers and volatility of data streams in real-time prediction.

Based on the proposed approach, our EWS predicts the patients' outcomes (specifically, whether or not they would be transferred to the ICU) from real-time data streams. This study serves as a proof-of-concept for our vision of using data mining to identify at-risk patients and (ultimately) to perform real-time event detection.

II. Related Work

Medical data mining is one of key issues to get useful clinical knowledge from medical databases. These algorithms either rely on medical knowledge or general data mining techniques. A number of scoring systems that already exist use medical knowledge for various medical conditions. For example, the effectiveness of Several Community-Acquire Pneumonia (SCAP) and Pneumonia Severity Index (PSI) in predicting outcomes in patients with pneumonia is evaluated in (Yandiola, 2009). Similarly, outcomes in patients with renal failures may be predicted using the Acute Physiology Score (12 physiologic variables), Chronic Health Score

(organ dysfunction), and APACHE score (Knaus, 1985). However, these algorithms are best for specialized hospital units for specific visits. In contrast, the detection of clinical deterioration on general hospital units requires more general algorithms. For example, the Modified Early Warning Score (MEWS) (ko, 2010) uses systolic blood pressure, pulse rate, temperature, respiratory rate, age and BMI to predict clinical deterioration. These physiological and demographic parameters may be collected at bedside, making MEWS suitable for a general hospital.

An alternative to algorithms that rely on medical knowledge is adapting standard machine learning techniques. This approach has two important advantages over traditional rule-based algorithms. First, it allows us to consider a large number of parameters during prediction of patients' outcomes. Second, since they do not use a small set of rules to predict outcomes, it is possible to improve accuracy. Machine learning techniques such as decision trees (Thiel, 2010), neural networks (Zernikow, 1998), and logistic regression (Griffin, 2001, Gregory, 2010) have been used to identify clinical deterioration. In (Ye, 2008), integrate heterogeneous data (neuron-images, demographic, and genetic measures) is used for Alzheimer's disease (AD) prediction based on a kernel method. A support vector machine (SVM) classifier with radial basis kernels and an ensemble of templates are used to localize the tumor position in (Cui, 2008). Also, in (Hwang, 2008), a hyper-graph based learning algorithm is proposed to integrate micro array gene expressions and protein-protein interactions for cancer outcome prediction and bio-marker identification.

There are a few distinguishing features of our approach comparing to previous work. Most previous work uses a snapshot method that takes all the features at a given time as input to a model, discarding the temporal evolving of data. There are some existing time-series

classification method, such as Bayes Decision Tree (Rajan, 1993), Conditional Random Fields (CRF) (Sha, 2003) and Gussian Mixture Model (Johnson, 2004). However, these methods assume a regular, constant gap between data records (e.g. one record every second). Our medical data, on the contrary, contains irregular gaps due to factors such as the workload of nurses. Also, different measures have different gaps. For example, the heart rate can be measured about every 10 to 20 minutes, while the temperature is measured hourly. Existing work cannot handle such high-dimensional data with irregular, multi-scale gaps across different features. Yet another challenge is class imbalance: the data is severely skewed as there are much more normal patients than those with deterioration.

To overcome the above difficulty, we propose a *bucketing* method that allows us to exploit the temporal structure of stream data, even though the measuring gaps are irregular. Moreover, our method is novel in that it combines a novel *bucket bagging* idea to enhance model precision and address overfitting. Further, we incorporate an *exploratory undersampling* approach to address class imbalance. Finally, we develop a smoothing scheme to smooth out the output from the prediction algorithm, in order to handle reading errors and outliers.

III. Data Situation and Challenge

In the general hospital wards (GHWs), a collection of features of a patient are repeatedly measured at the bed-side. Such continuous measuring generates a high-dimensional data stream for each patient. In our datasets, each patient is measured for 34 indicators, including demographics, vital signs (pulse, shock index, mean arterial blood pressure, temperature, and respiratory rate), and lab tests (albumin, bilirubin, BUN, creatinine, sodium, potassium, glucose, hemoglobin, white cell count, INR, and other routine chemistry and hematology results).

Most indicators are typically collected manually by a nurse, at a granularity of only a handful of readings per day. Errors are frequent due to reading or input mistakes. The values of some features are recorded only once within an hour. Other features are recorded only for a subset of the patients. Hence, the overall high dimensional data space is sparse. This makes our dataset a very irregular time-series. Figure 1 and 2 plots the diversification of some vital signs for two randomly picked visits. It is obvious that they have multi-scale gaps: different vital signs have different time gaps. Even more, a vital sign may not have the same reading gap. To make things worse, we have extreme skewed data. Out of 41,503 patient visits, only 1983 (less than 0.5%) are transferred to ICU. In summary, the dataset contains skewed, noisy, high dimensional time series with irregular and multi-scale gaps.

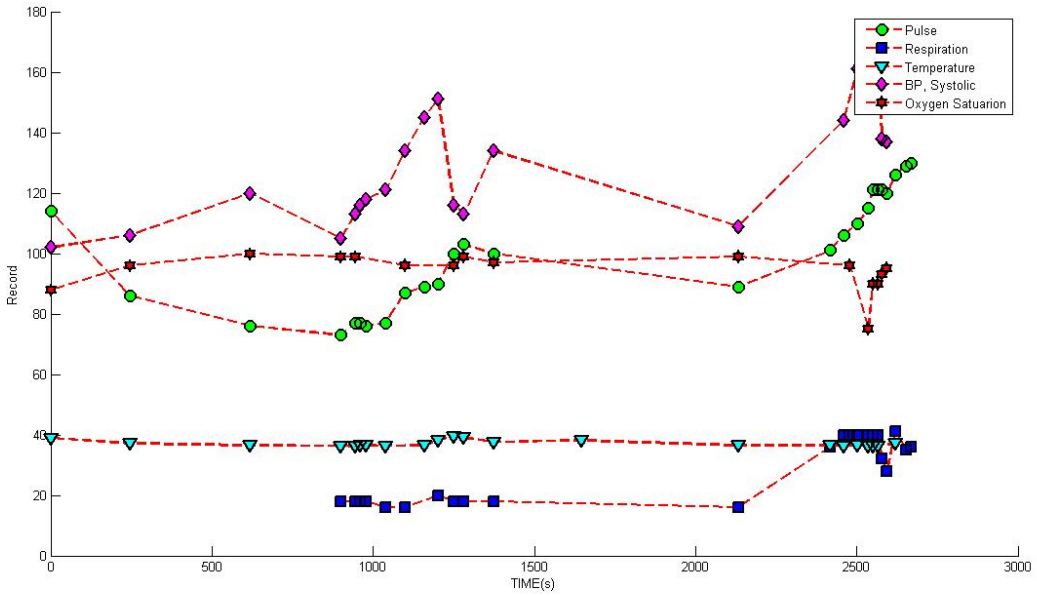

Figure 1: Mapping of a patient's vital signs.

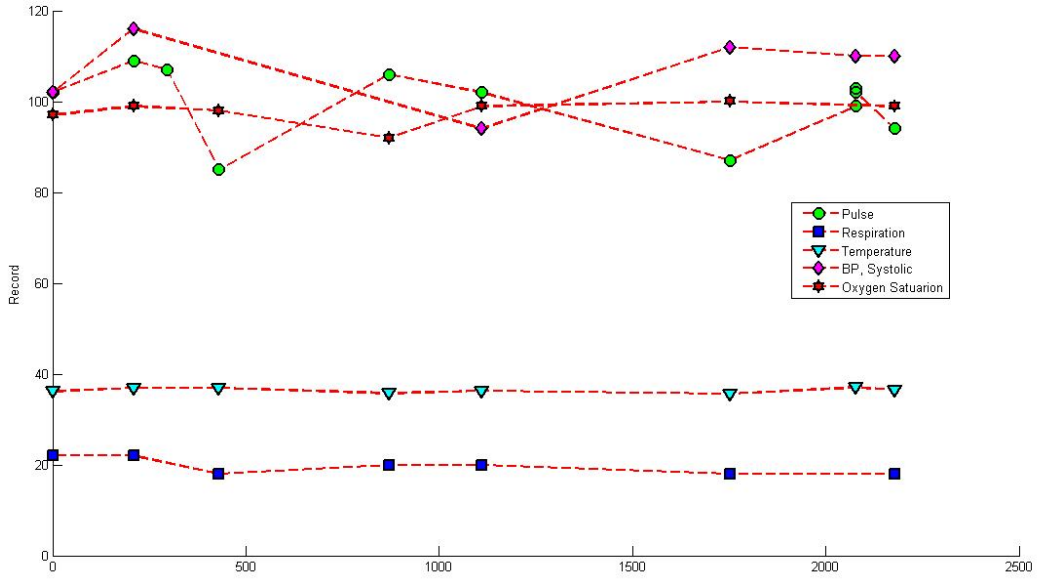

Figure 2: Mapping of a patient's vital signs.

Ⅳ. Algorithm Details

In this section, we discuss the main features of the proposed early warning system (EWS).

A. Workflow of the EWS system

We proposed a few new techniques in the EWS system, the details of which will be described later. Firstly, let us overview the workflow of our system.

As we shown in Figure 3, the system consists of a model building phase which builds a prediction model from a training set, and a deployment phase which applies the prediction model to monitored patients in real-time.

In the model building phase:

- Step 1: Data preprocessing: remove the outliers, fill the missing data, and normalize the data.
- Step 2: Bucketing: for each patient, generate the 24-hour window and divide the window into 6 buckets, and compute the features in each bucket.

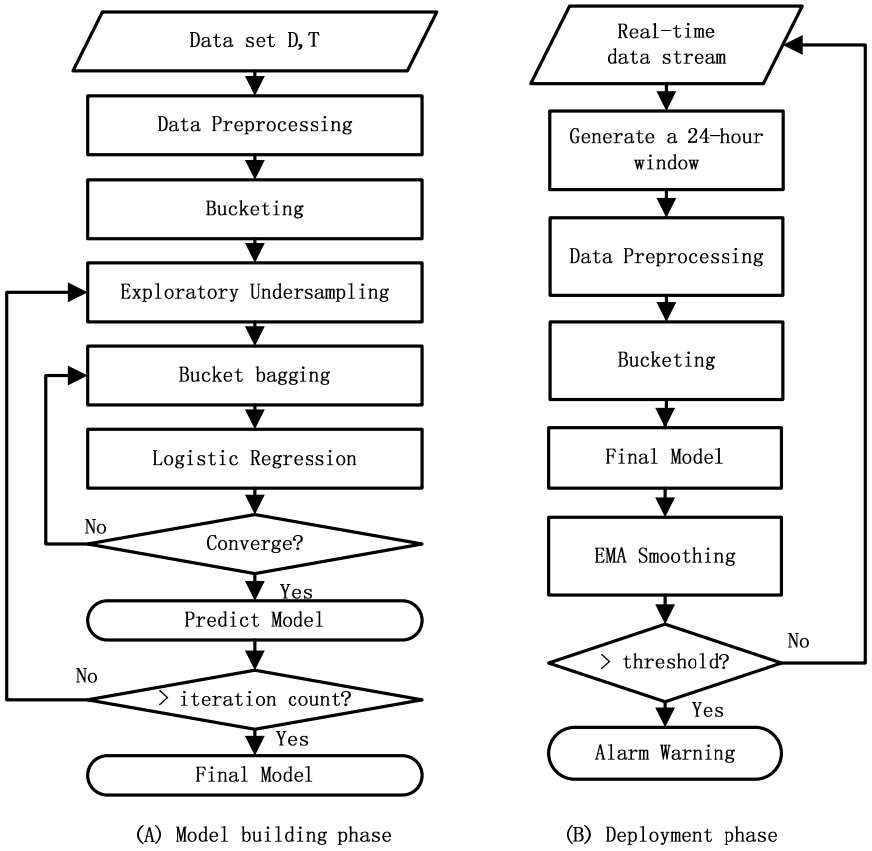

Figure 3: The flow chart of our system

- Step 3: Exploratory undersampling: a technique we employ to address class imbalance.
- Step 4: Bucket bagging: a bagging technique to improve model quality and address overfitting.
- Step 5: Apply the prediction algorithms to the data samples selected by Steps 3 and 4.
- Step 6: Repeat from Step 3 until the model converges.

In the deployment phase, for each target patient:

- Step 1: Generate the sliding 24-hour window of the stream data
- Step 2: Apply data preprocessing.
- Step 3: Apply bucketing and collect features.

- Step 4: Feed the features into the prediction model.
- Step 5: Apply EMA smoothing to the output of the prediction model.
- Step 6: Make an alert decision based on the smoothed output.

There are six major technical components in our EWS system, including data preprocessing, bucketing, prediction algorithm, bucket bagging, exploratory undersampling, and EMA smoothing. We now describe them one by one.

B. Data Preprocessing

Before building our model, several preprocessing steps are applied to eliminate outliers and find an appropriate representation of patient's states.

First, as most vital signs are measured by nurses, there are inevitably errors introduced by manual operations, including reading and input errors.

So we perform a sanity check of the data. For each of the 34 vital signs, we list its acceptable ranges based on the domain knowledge of the medical experts in our team. Some signs are bounded by both min and max, some are bounded from one side, and some are not bounded. For any value that is outside of the range, we replace it by the mean value of that patient (if available) or the mean of the entire dataset. This filters out the outliers in the dataset.

Second, a complication of using clinical data is that not all patients will have values for all signs. Many types of clinical features involved in lab tests are not routinely performed on all patients. This problem is compounded by the bucketing technique we will introduce later. In bucketing, since we divide the time into segments, even when a patient has had a particular lab test, it will only provide a data point for one bucket. To deal with missing data points, we calculate the mean value of a sign over the entire historical dataset. When a patient does not have data in a particular bucket, the corresponding mean value is used.

Finally, real valued parameters in each bucket for every vital sign are scaled so that all measurements lie in the interval [0, 1]. They are normalized by the min and max of the signal:

$$value' = \frac{value - min_j}{max_j - min_j}$$

While *value* is the raw value, $value'$ is the normalized value, min_j is the minimum of the j^{th} signal, and max_j is the maximum of the j^{th} signal in the dataset. Such normalization is helpful for many prediction algorithms such as logistic regression and support vector machine (SVM).

C. Bucketing and prediction algorithms

Many classification algorithms do not by themselves operate on stream data. That is, each variable input to the algorithm (e.g. logistic regression or SVM) corresponds to exactly one data point: e.g., a blood pressure feature would consist of a single blood pressure reading. In a clinical application, however, it is important to capture unusual changes in vital signs over time. Such changes may precede clinical deterioration by hours (Buist, 1999), providing a chance to intervene if detected early enough.

In addition, not all readings in stream data should be treated equally: the value of some data may change depending on its age. For example, a patient's condition may be better reflected by a blood-oxygenation reading collected one hour ago than a reading collected 12 hours ago. Although there are a few algorithms, such as a conditional random field (CRF), that can be adapted to classify stream data, they require regular and equal gaps of data and cannot be directly applied here. To capture the temporal effects in our data, we use a bucketing technique. We retain a sliding window of all the collected data points within the last 24 hours. We divide this data into *n* equally sized buckets. In our current system, we divide the 24-hour window into 6

sequential buckets of 4 hours each. In order to capture variations within a bucket, we compute several feature values for each bucket, including the minimum, maximum, and mean.

As a result, there are 34 $\times 6$ $\times 3$ features, since there are 34 vital signs, 6 buckets, and 3 features (min, max, mean) each bucket. In principle, our features from bucketing can be used as input to any classification algorithms. In our current study, we have tested two prediction algorithms, logistic regression and SVM.

For both logistic regression and SVM, in our proposed algorithm, we use *max*, *min*, and *mean* of each bucket and each vital sign as features. We comment that the changes between these features are also considered in our approach. For two features f_1 and f_2 in our current model, we consider another model that has an additional feature $\delta = f_1 - f_2$. Suppose the weights for f_1, f_2, and δ are w_1, w_2 and w_3, respectively. Since we know $w_1*f_1 + w_2* f_2 + w_3* f_3 = (w_1+ w_3)* f_1 + (w_2 - w_3)* f_2$, our model can be equivalent to the new one by setting the weights for f_1 and f_2 to $w_1+ w_3$ and $w_2 - w_3$, respectively. This shows that changes in vital signs are also captured by our approach. For example, f_1 and f_2 can be the mean heart rates in bucket 1 (first to fourth hours) and bucket 2 (fifth to eighth hours), respectively. Note that since all features are already normalized into the range of [0, 1], the difference directly reflects the percentage change. Therefore, our model can take into account the temporal changes of vital signs.

D. Bucket bagging

Bootstrap aggregating (bagging) is a meta algorithm to improve the quality of classification and regression models in terms of stability and classification accuracy. It also reduces variance and helps to avoid over-fitting. It does this by fitting simple models to localized subsets of the data to build up a function that describes the deterministic part of the variation in the data.

The standard bagging procedure is as follows. Given a training set *D* of size *n*, bagging generates *m* new training sets D_i, $i = 1 \ldots m$, each of size $n' \leqslant n$, by sampling examples from D uniformly and with replacement. By sampling with replacement, it is likely that some examples will repeat in each D_i. This kind of sample is known as bootstrap samples. The *m* models are fitted using the *m* bootstrap samples and combined by averaging the output (for regression) or voting (for classification).

We have tried the standard bagging method on our dataset, but the results did not show much improvement. We argue that it is due to the frequent occurrence of dirty data and outliers in the real datasets, which add variability in the estimates.

It is well known that the benefit of bagging diminishes quickly as the presence of outliers increases in a dataset (P. Hall &B.A. Turlach, 1997). Here, we propose a new bagging method named *biased bucket bagging* (BBB). The main differences between our method and the typical bagging are: first, instead of sampling from raw data, we sample from the buckets each time to generate a bootstrap sample; second, we employ a bias in the sampling which always keeps the 6th bucket in each vital sign and randomly sample 3 other buckets from the remaining 5 buckets. Since there are $C_5^3 = 10$ choices, we built 10 bootstrap samples and 10 models. Taking the average of the 10 models, we got the prediction model for the dataset.

We explain why bucket bagging works. First, from Table 2 which lists the features with the highest weights in a trained logistic regression model using all buckets, we found that the weights related to features in bucket 6 are significant. This reflects the importance of the most recent vital signs since bucket 6 contains the medical records that are collected in the most recent four hours. That is the reason why we always keep the 6th bucket. Second, the total expected error of a classifier is made up of the sum of bias and variance. In bagging, combining multiple

Table 1:

Comparison of the BBB method with different numbers of buckets in bootstrap samples.

Bucket 6 is always kept.

Methods	$E[\phi^2(D_i, y_i)]$	$E^2[\phi(D_i, y_i)]$	$SD[\phi(D_i, y_i)]$	AUC	Sensitivity	PPV	NPV	Accuracy
2 buckets	103989.0352	90524.9358	0.67178	0.89668	0.55572	0.35654	0.97722	0.93128
3 buckets	142662.3642	125106.354	0.64977	0.91415	0.59213	0.37123	0.97904	0.93301
4 buckets	**173562.7307**	**155526.458**	**0.72977**	**0.9218**	**0.60087**	**0.37466**	**0.97948**	**0.93342**
5 buckets	157595.163	14813.379	0.52066	0.89934	0.46103	0.31493	0.97249	0.92678

classifiers decreases the expected error by reducing the variance.

Let each (D_i, y_i), $1 \le i \le m$ be the bucket sample that is independently drawn from (D, y), $\Phi(D_i, y_i)$ is the predictor. The aggregated predictor is:

$$\Phi_A(D, y) = E\{\Phi(D_i, y_i)\}$$

The average prediction error e' in $\Phi(D_i, y_i)$ is:

$$e' = E[(y_i - \Phi(D_i, y_i))^2]$$

The error in the aggregated predictor is:

$$e = [E(y - \Phi_A(D, y))^2]$$

Using the inequality $(EZ)^2 \leqslant EZ^2$ gives us $e \leqslant e'$. We see that the aggregated predictor has lower mean-squared prediction error. How much lower depends on how large the difference $EZ^2 - (EZ)^2$ is. Hence, a critical factor deciding how much bagging will improve accuracy is the variance of these bootstrap models. Table 1 shows such statistics for the BBB method with different numbers of buckets in the bootstrap samples. We see that BBB with 4 buckets has the largest difference between $(EZ)^2$ and EZ^2 and the highest standard deviations.

Correspondingly, it gives the best prediction performance. That is why we choose to have 4 buckets in BBB.

E. Exploratory undersampling

Looking through the records, we have a skewed dataset. Among 41,503 records, 1983 are from visits that are transferred to ICU. Undersampling is a popular method in dealing with the class-imbalance problem. The idea is to combine the minority class with only a subset of the majority class each time to generate a sampling set, and take the ensemble of multiple sampled models. We have tried undersampling on our data but obtained very modest improvements. In our EWS system, we used a novel method called exploratory undersampling (Liu, 2006), which makes better use of the majority class than simple undersampling. The idea is to remove those samples that can be correctly classified by a large margin to the class boundary by the existing model.

Specifically, we fix the number of the ICU patients, and then randomly choose the same amount of non-ICU patients to build the training dataset at each iteration. The main difference to simple undersampling is that, at each iteration, it removes 5% in both the majority class and the minority class with the maximum classification margin. For logistic regression, we remove those ICU patients that are closest to 1 (the class label of ICU) and those non-ICU patients that are closest to 0. For SVM, we remove correctly classified patients with the maximum distance to the boundary.

F. Exponential Moving Average (EMA)

The smoothing technique is specific for using the logistic regression model in the deployment phase. At any time *t*, for a patient, features from a 24-hour moving window are fed into the model. The model then outputs a numerical output Y_t .

From the training data, we also choose a threshold δ so that the model achieves a specificity of 95% (i.e., a 5% false-positive rate).We always compare the model output Y_t with δ. An alarm will be triggered whenever $Y_t > \delta$. Observing the predicted value, we found there is often high volatility in Y_t, which will cause a lot of false alarms. Here, we imported exponential moving average (EMA), a smoothing scheme to the output values before we apply the threshold to do the classification.

EMA is a type of infinite impulse response filter that applies weighting factors which decrease exponentially. The weighting for each older data point decreases exponentially, never reaching zero. The formula for calculating the EMA at time periods $t > 2$ is (NIST):

$$S_t = \alpha \times Y_t + (1-\alpha) \times S_{t-1}$$

Where:

- The coefficient α is a smoothing factor between 0 and 1.
- Y_t is the model output at time *t*.
- S_t is the EMA value at *t*.

Using EMA smoothing, the alarm would be triggered if and only if $S_t > \delta$.

V. Results and Discussions

A. Evaluation Criteria

In the proposed early warning system, the accuracy is estimated by the following parameters: AUC (Area Under receives operating characteristic (ROC) Curve), PPV (Positive Predictive Value), NPV (Negative Predictive Value), Sensitivity, Specificity and Accuracy.

		Condition (as determined by "Gold standard")		
		Positive	***Negative***	
Test outcome	***Positive***	True Positive	False Positive (Type I error)	→ **Positive predictive value** $= \frac{\Sigma \text{ True Positive}}{\Sigma \text{ Test outcome Positive}}$
	Negative	False Negative (Type II error)	True Negative	→ Negative predictive value $= \frac{\Sigma \text{ True Negative}}{\Sigma \text{ Test outcome Negative}}$
		↓ Sensitivity $= \frac{\Sigma \text{ True Positive}}{\Sigma \text{ Condition Positive}}$	↓ Specificity $= \frac{\Sigma \text{ True Negative}}{\Sigma \text{ Condition Negative}}$	

Figure 4: The relationship between PPV, NPV, Sensitivity, and Specificity (Wikipedia).

Figure 4 illustrates how the measures are related. In clinical area, the positive predictive value means among the patients who really are transferred to ICU how many of them can be correctly classified. A high PPV means a low false alarm rate, while a high NPV means that the algorithm only rarely misclassifies a sick person as being healthy. Sensitivity measures the proportion of sick people who are correctly identified as having the condition. Specificity represents the percentage of healthy people who are correctly identified as not having the condition. For practical deployment in hospitals, a high specificity (e.g. >95%) is needed.

For any test, there is usually a trade-off between the different measures. This tradeoff can be represented using a ROC curve, which is a plot of sensitivity or true positive rate, versus false positive rate (1-specificity). In our evaluation system, we plot the curve using sensitivity and false negative rate. AUC represents the probability that a randomly chosen positive example is correctly rated with greater suspicion than a randomly chosen negative example (Bradley: 97). Finally, accuracy is the proportion of true results (both true positives and true negatives) in the whole dataset.

B. Results on real historical data

1) Study design.

Table 2:

The 10 highest-weighted variables of simple logistic regression on the history system.

Variable	
Oxygen Saturation, pulse oximetry (bucket 6 min)	-3.5209
Respirations (bucket 6 max)	3.0998
Shock Index (bucket 6 max)	2.6986
Respirations (bucket 6 min)	2.3891
coagulation modifiers (bucket 1)	2.3756
BP, Diastolic (bucket 6 min)	-2.2429
Pulse (bucket 6 max)	1.8251
Oxygen Saturation, pulse oximetry (bucket 6 mean)	-1.7756
Respirations (bucket 6 min)	1.7352
hormones/hormone modifiers (bucket 1)	1.7278

As shown in figure 3, our EWS system consists of two major sub-systems: history system and real-time system. Both systems contain data from tens of thousands of real patients from a large hospital. In the history system, we applied all the proposed techniques to get the prediction model. It gives us the coefficients and thresholds in the prediction model which would be needed in the real-time system. For all results, we perform a four-fold cross validation in which different parts of the dataset are used as training and testing data. In the real-time system, we use the prediction models learned from the history system and apply it to a 24-hour moving window, along with Exponential Moving Average (EMA) smoothing.

2) Performance of simple logistic regression

After implementing the logistic regression algorithm in MATLAB, we evaluated its accuracy in the history system. We first show the results from simple logistic regression with bucketing. Bucket bagging and exploratory undersampling are not used here.

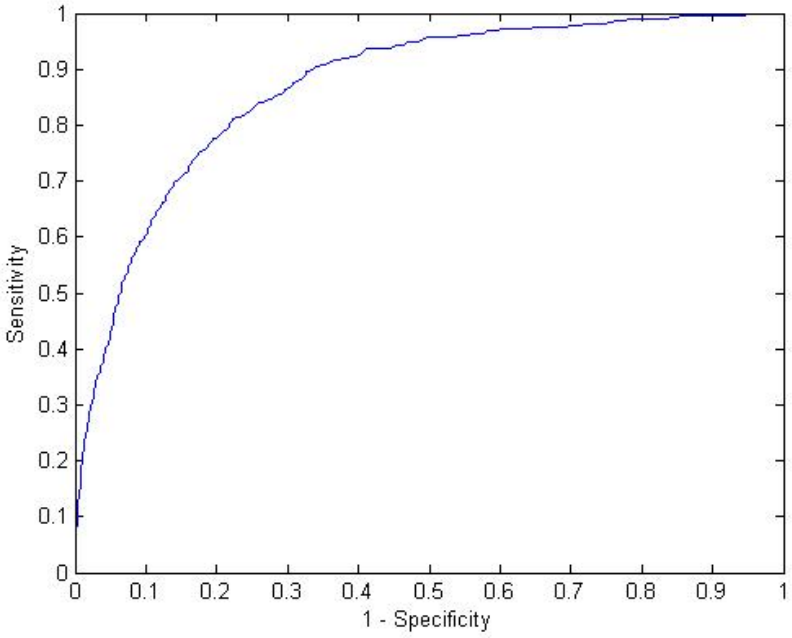

Figure 5: ROC curve of simple logistic model's predictive performance.

Table 2 provides a sample of the output from the training process, listing the 10 highest-weighted variables and their coefficients in the logistic regression model. A few observations can be made. First, bucket 6 (the most recent bucket) makes up 8 of the 10 highest-weighted variables, confirming the importance of keeping bucket 6 in bagging. Nevertheless, even the older data can have high values: for example, bucket 1 of the coagulation modifier drug class was the 5th highest-weighted variable. Also, minimum and maximum values make up 7 of the top 10 variables, reflecting the importance of extreme in vital sign data.

Figure 5 plots the ROC curve of the model's performance under a range of thresholds. The X axis plots 1 - the specificity, or the false negative rate, for a given threshold. The *Y* axis plots the corresponding sensitivity, or the true positive rate, at the same threshold. Both performance metrics are important from a clinical perspective. Higher *X* values correspond to a larger number of false alarms; if alarms are too frequent, clinicians may be more inclined to ignore them. Higher *Y* values correspond to more patients correctly being identified for early intervention.

Table 3:

The predictive performance of simple logistic regression at a chosen cutpoint.

Area under curve	0.86809
Specificity	0.94998
Sensitivity	0.44753
Positive predictive value	0.29562
Negative predictive value	0.97345
Accuracy	0.92747

For the purposes of further analysis, we select a target threshold of $y = 0.9338$. This threshold was chosen to achieve specificity close to 95% (i.e., a 5% false-positive rate). This specificity value was in turn chosen to generate only a handful of false alarms per hospital floor per day. Even at this relatively high specificity, the logistic regression approach achieves a sensitivity of 44.75%. Table 3 summarizes the performance at this cut-point for the other statistical metrics. In the following, we will use the same threshold to keep the 95% specificity, while we improve some other prediction indices.

3) Performance with bucket bagging and exploratory undersampling

We now show improved methods using the proposed bucket bagging and exploratory undersampling techniques. We compare the performance of 5 different methods as follows.

- Method 1 (Bucketing + logistic regression): This is the simple logistic regression model we discussed in the last subsection.
- Method 2 (Method 1 + standard bagging): We augment Method 1 with standard bagging. For each bootstrap sample, we randomly sample from raw data.

Table 4:

Comparison of various methods on the history system

Method	AUC	Specificity	Sensitivity	PPV	NPV	Accuracy
Method 1	0.79242	0.94996	0.33434	0.25076	0.96609	0.92058
Method 2	0.9025	0.94996	0.55572	0.35802	0.977707	0.93176
Method 3	0.90327	0.94996	0.58094	0.36829	0.97835	0.93237
Method 3'	0.90052	0.94996	0.56883	0.3632	0.97771	0.93176
Method 4	0.9086	0.94996	0.58339	0.36777	0.9786	0.93259

- Method 3 (Method 1 + biased bucket bagging): We augment Method 1 by biased bucket bagging. For each bootstrap sample, we keep all the features in the 6th bucket and randomly choose 3 buckets from the other buckets.
- Method 3' (Method 1 + bucket bagging): It is the same as Method 3 except that we do not always keep bucket 6 when we select bootstrap samples.
- Method 4 (Method 3 + exploratory undersampling): We augment Method 3 by using exploratory undersampling to address class imbalance.

Table 4 shows the comparison of all the methods. Through analyzing it, we got the following conclusions. First, the results show that all the other methods attain better result than Method 1, which indicate that using bagging improved the performance no matter which sampling method we employ. Second, Method 3 gives better outcome than Method 2, which means bucket bagging outperforms standard bagging. Third, exploratory undersampling in Method 4 is useful to improve the performance further. Method 4 combing these techniques together achieves significantly better results than the simple logistic regression in Method 1. Looking at the two most important measures in practice, PPV (positive predictive value) is improved from 0.25076 to 0.36777, and Sensitivity is improved from 0.33434 to 0.58339.

4) Comparison with SVM and decision tree

We also compared the performance of Method 4 with Support Vector Machine (SVM) and Recursive Partitioning And Regression Tree (RPART) analysis. In RPART analysis, a large tree that contains splits for all input variables is generated initially (Warren, 2009) Then a pruning process is applied with the goal of finding the "sub-tree" that is most predictive of the outcome of interest (Steven, 2010). The resulting classification tree (shown in Figure 6) was then used as a prediction algorithm and applied in a prospective fashion to the test data set.

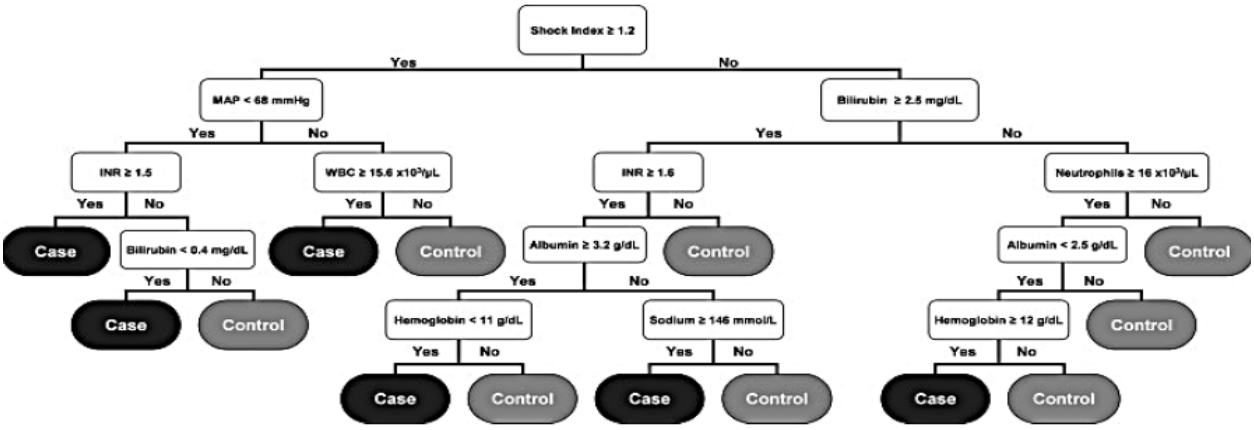

Figure 6: Classification tree for the clinical data. For each branch, to the left indicates that the patients meet the condition, and to the right either the patient does not meet the condition or the data are missing (Steven, 2010).

For SVM, the most two important parts for our experiment are the cost factors and the kernel faction. A problem with imbalanced data is that the class boundary (hyper plane) learned by SVMs can be too close to the positive examples so that the recall suffers. To overcome this problem, we propose to use Cost-weighted SVMs (cwSVMs), which incurs no additional

Table 5：

Comparison of the methods

Method	AUC	Specificity	Sensitivity	PPV	NPV	Accuracy
Method 4	0.9086	0.94996	0.58339	0.36777	0.9786	0.93259
RPART	-	0.93	0.55	0.287	0.977	0.912
SVM(RBF kernel)	0.67157	0.9522	0.3909	0.2908	0.9689	0.9194

Table 6:

SVM & SVM voting

Method	AUC	Specificity	Sensitivity	PPV	NPV	Accuracy
SVM (24-hour window)	0.67157	0.9522	0.3909	0.2908	0.9689	0.9194
SVM (last 4-hour window)	0.74709	0.84065	0.65354	0.17044	0.97977	0.83172
SVM (last data)	0.79034	0.90729	0.67879	0.26837	0.98257	0.89639
SVM voting	0.76759	0.94125	0.59394	0.33619	0.97884	0.92468

overhead. The value of the ratio between cost factors is crucial for balancing the tradeoff between precision and recall. Morik showed that the setting $\frac{C_+}{C_-} = \frac{\text{number of negative examples}}{\text{number of positive exapmles}}$ is an effective heuristic (Morik, 99). Hence, here we set $\frac{C_+}{C_-} = 21$ as there are 1983 ICU transfers out of 41,503 hospital visits. The area under the ROC curve (AUC) has been shown to exhibit a number of desirable properties as a classification performance measure when compared to overall accuracy (Bradley, 97).

We know that the SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a margin that is as large as possible. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the

training data points of any class, since in general the larger the margin the lower the generalization error of the classifier will be. Due to the sparseness of our dataset, when we use the same bagging strategy as we did for logistic regression, we did not get the expected improvement. In this paper, we change the traditional bagging strategy which samples data by time to sampling by variables or buckets. Besides this, we adopt the *voting strategy* to implement the classifier combination. The voting strategy works like this: for a specific item, an integrated decision is made by taking the majority of different classifiers that use different sampling methods. As such, the predictor function is:

$$\Phi_A(D, y) = \text{sgn}\{\Phi(D_i, y_i)\}, i = 1 \cdots m$$

Specifically, in this paper, we use three SVMs, trained over the data in the last 24-hour window, last 4-hour window, and the last measurements, respectively. We then use a simple majority voting strategy to make the final decision.

From Table 5, we found that Method 4 has much better result than SVM and RPART in terms of AUC. Note that unlike logistic regression, it is not flexible to adjust the Specificity/Sensitivity tradeoff in SVMs and decision trees. Hence, AUC provides a fair comparison for these methods. We also see that, comparing to SVMs and decision tree, Method 4 achieves much higher sensitivity and AUC with other metrics being comparable. In Table 6, we found that using the voting strategy (shown as *SVM voting*) can give significant improvement to the single SVM using only the 24-hour window data and achieve a better balance between specificity and positive predictive value.

C. Results on the Real-time System

In this real-time system, for each patient, first we generate a 24-hour window once new record came, then feed it into our logistic regression model and output a predicted value.

EMA smoothing (with $\alpha = 0.06$) can be used on the output. At last, the smoothed values were compared with the threshold to convert these values into binary outcomes. An alarm will be triggered once the smoothed value meets the criteria. For comparison, we also show the

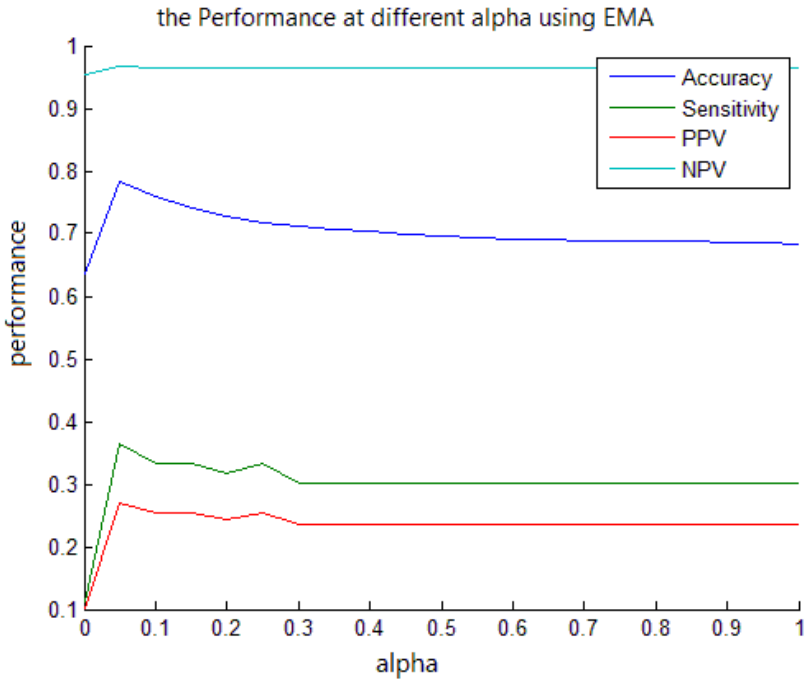

Figure 7: The performance of different α.

prediction performance that only considers the record in the past 24 hours without using EMA. Further, in our experiments with EMA, we evaluate the performance by varying α. When α is increasing, the influence of the historical record is decreasing. From the result we get in Table 7, we found that we can get a better or equal performance every time we use EMA, for every method. From Figure 7, we see that the optimal value for α is around 0.06. We also show performance of 4-fold cross-validation in Figure 8. We can see that all cases attain best performance when α is around 0.06, showing that the choice of α is robust. This small optimal α value shows that historical records play an important role for prediction.

Table 7: *Performance results on the real-time system.*

Method	AUC	Specificity	Sensitivity	PPV	NPV	Accuracy
Method 1	0.68346	0.94998	0.30159	0.23457	0.96342	0.9128
Method 1 with EMA	0.78203	0.94998	0.36508	0.27059	0.96664	0.9128
Method 2	0.74359	0.94996	0.30159	0.23457	0.96342	0.9293
Method 2 with EMA	0.777737	0.94996	0.38095	0.27907	0.96342	0.92134
Method 3	0.77689	0.94996	0.38095	0.27907	0.96745	0.9336
Method 3 with EMA	0.81411	0.94996	0.39683	0.28736	0.96825	0.92212
Method 4	0.79902	0.94996	0.4127	0.29545	0.96096	0.9229
Method 4 with EMA	0.79902	0.94996	0.4127	0.29545	0.96096	0.9229

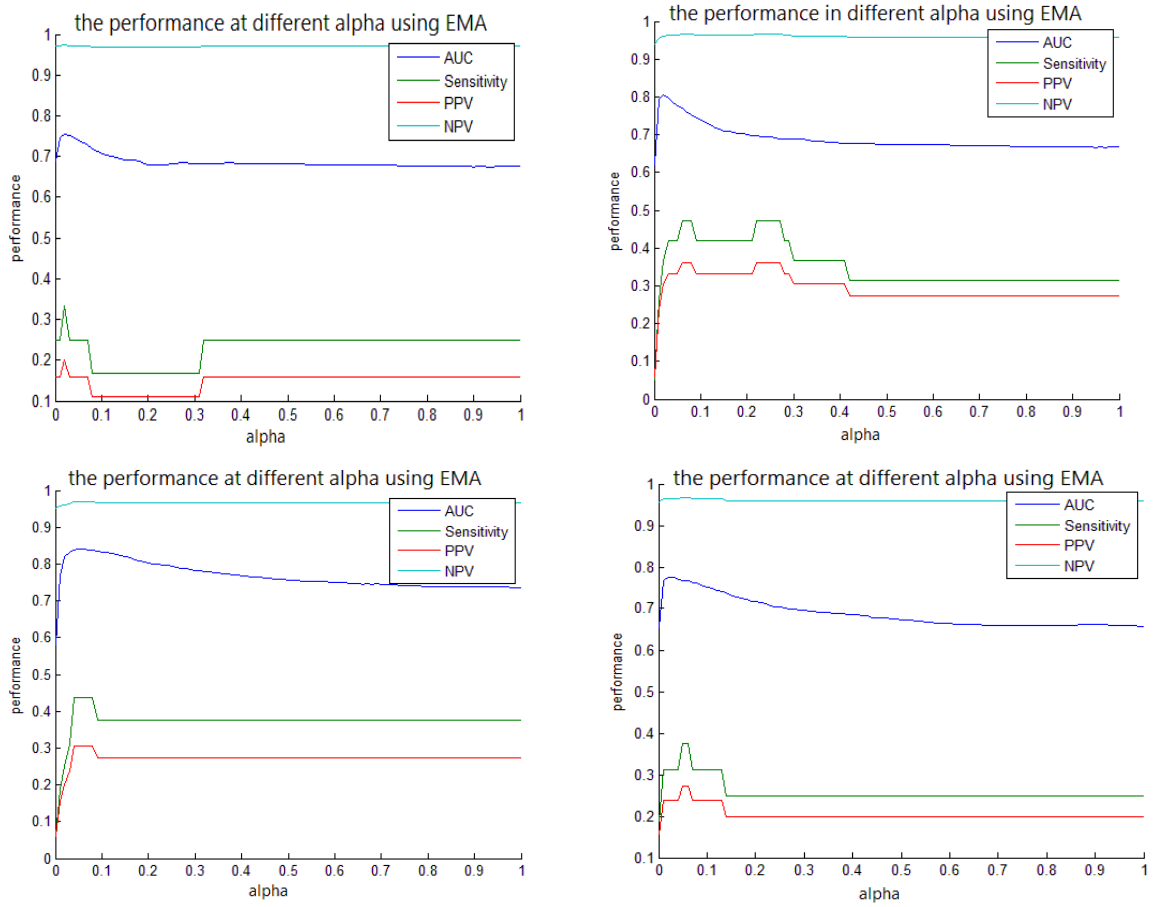

Figure 8： The cross-validation results of varying α .

D. Lead time analysis.

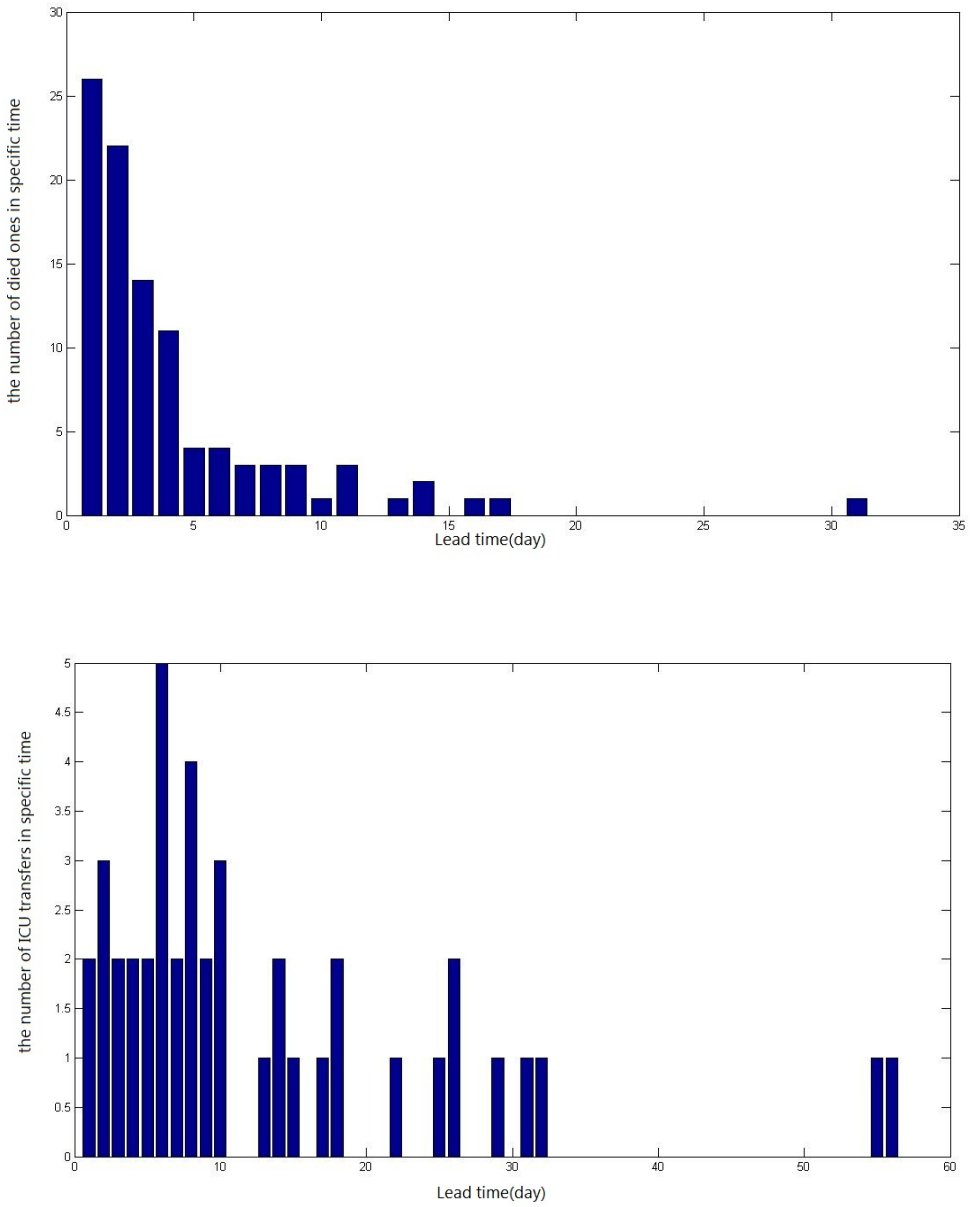

Figure 9: the lead time distribution for the ICU transfers (upper) and died patients (down).

In clinical analysis, lead time is the length of time between the detection of a disease (usually based on new, experimental criteria) and its usual clinical presentation and diagnosis (based on traditional criteria). Here we define the “lead time” as the length between the time we give alert and the ICU transfer/death date. Figure 9 shows the histogram distribution of the lead time of all

patients. For true ICU transfers, we can give the alert at least 4 hours before the transfer time. For deaths, we can give the alert at least 30 hours earlier. For the ICU lead time, we can see that most of the patients can get the alert in less than 24 hours before ICU transfers, which shows that our alert is highly related to the patients' actual situation.

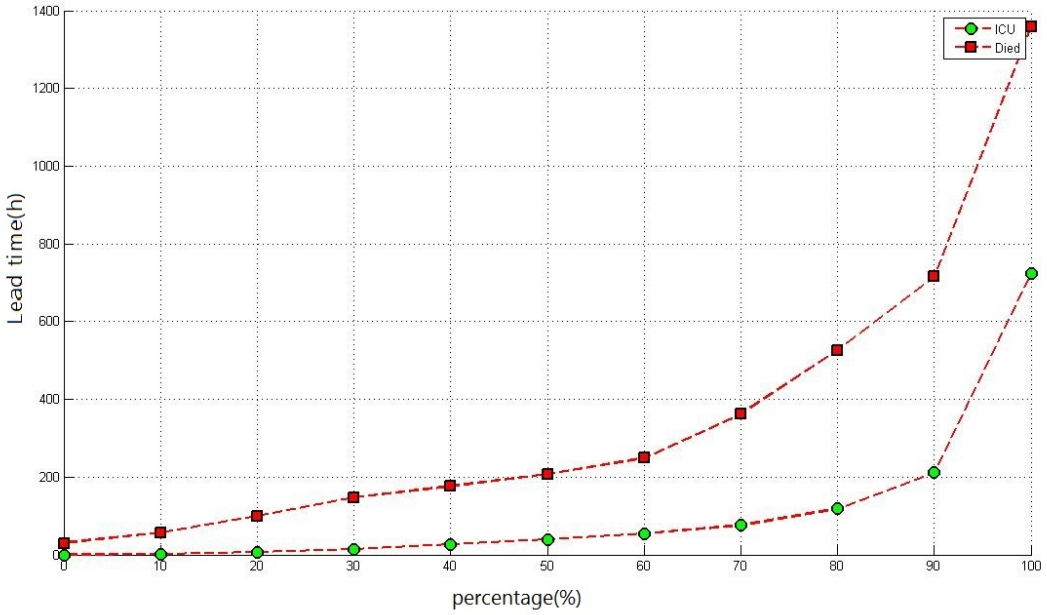

Figure 10: Lead time percentile.

Percentile represents the percentage of the patients under a certain lead time. In Figure 10, we can see that for the same percentage of patients, the lead time for death is always longer than that for ICU transfers, which also shows that our alert is indeed related to the actual situation of the patients.

VI Conclusions

In this paper, we have presented a predictive system for hospitalized patients that can provide early warning of clinical deterioration. This is an important advance, representing a significant opportunity to intervene prior to clinical deterioration. We introduced a bucketing technique to capture the changes in the vital signs. Meanwhile, we handled the missing data so that the visits

that do not have all the parameters can still be classified. We conducted a pilot feasibility study by using a combination of logistic regression, bucket bootstrap aggregating for addressing overfitting, and exploratory undersampling for addressing class imbalance. We showed that this combination can significantly improve the prediction accuracy for all performance metrics, over other major methods. Further, in the real-time system, we use EMA smoothing to tackle volatility of data inputs and model outputs.

Our performance is good enough to warrant an actual clinical trial at the Barnes-Jewish Hospital, one of the largest hospitals in the United States. During the period between 1/24/2011 and 5/4/2011, there were a total of 89 deaths among 4081 people in the study units. 49 out of 513 (9.2%) people with alerts died after alerts, and 40 out of 3550 (1.1%) without alerts died (Chi-square $p<0.0001$). Thus, our alerts are highly associated with patient death, and they identified 55% of patients who died during hospitalization. There were a total of 190 ICU transfers among 4081 (4.7%) people in the study units. 80 of 531 (15.1%) people with alerts were transferred to the ICU, and 110 of 3550 (3.1%) without alerts were transferred ($p<0.0001$). Thus, our alerts are highly associated with ICU transfers, and they identified 42% of patients who were transferred to ICU during hospitalization. Such results clearly show the feasibility and benefit of employing data mining technology in digitalized healthcare.

References

A. P. Bradley. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 1997.

A. M.T.Johnson and J.Ye. (2004). Time series classification using the gaussian mixture models of reconstructed phase spaces. *IEEE Transcations on Knowledge and Data Engineering.*

B. Zernikow, K. Holtmannspoetter, E. Michel, W. Pielemeier, F. Hornschuh, A. Westermann, and K. H. Hennecke. (1998). Artificialneural network for risk assessment in preterm neonates. *Archives of Disease in Childhood - Fetal and Neonatal Edition*.

F. F.Sha. (2003). Shallow parsing with conditional random fields. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, 1, 134－141.

G. G. Warren, J. Ewens. (2009). *Statistics for Biology and Health*.

J. Ye, K. Chen, T. Wu, J. Li, Z. Zhao, R. Patel, M. Bae, R. Janardan, H. Liu, G. Alexander, and E. Reiman. (2008). Heterogeneous data fusion for alzheimer’s disease study. *Proceeding of the 14th ACM SIGKDD*.

J. Ko, J. H. Lim, Y. Chen, R. Musvaloiu-E, A. Terzis, G. M. Masson, T. Gao, W. Destler, L. Selavo, and R. P. Dutton. (2010). Medisn: Medical emergency detection in sensor networks in *ACM Trans. Embed Computer System.*

K. Morik, P. Brockhausen, and T. Joachims. (1999). Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring in *ICML*.

M. P. Griffin and J. R. Moorman. (2001). Toward the early diagnosis of neonatal sepsis and sepsis-like illness using novel heart rate analysis. American Academy of Pediatrics, *107, 97 -104.*

M. C. Gregory Hackmann, O. Chipara, C. Lu, Y. Chen, T. C. Bailey, and Marin. (2010). Toward a two-tier clinical warning system for hospitalized patients. *American Medical Informatics Association Annual Symposium.*

M. Bloodgood and K. Vijay-Shanker. (2009). Taking into account the differences between actively and passively acquired data: the case of active learning with support vector machines for imbalanced datasets. The North American Chapter of the Association for Computational Linguistics (*NAACL*) short paper.

M. D. Jones, A. E.and Brown, S. Trzeciak, N. I. Shapiro, J. S. Garrett, A. C. Heffner, and J. A. Kline. (2008). The effect of a quantitative resuscitation strategy on mortality in patients with sepsis: a meta-analysis. *Crit. Care Med.*

N. I. of Standards of Technology. Single exponential smoothing.

P. P. E. Yandiola, A. Capelastegui, J. Quintana, R. Diez, I. Gorordo, A. Bilbao, R. Zalacain, R. Menendez, and A. Torres. (2009). Prospective comparison of severity scores for predicting clinically relevant outcomes for patients hospitalized with community-acquire pneumonia.

P. J.Rajan. (2002). Time series classification using the volterra connectionist model and bayes decison theory. *IEEE Internation Conference on Acoustics, Speech, and Signal Processing*.

S. W. Thiel, J. M. Rosini, W. Shannon, J. A. Doherty, S. T. Micek, and M. H. Kollef. (2010). Early prediction of septic shock in hospitalized patients. *Journal of Hospital Medicine* .

T. J. Commission. 2008 national patient safety goals.

T. Hwang, Z. Tian, R. Kuangy, and J.-P. Kocher. (2008). Learning on weighted hyper graphs to integrate protein interactions and gene expressions for cancer outcome prediction in

Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. Washington, DC, USA: IEEE Computer Society, 293 – 302.

W. A. Knaus, E. A. Draper, D. P. Wagner, and J. E. Zimmerman. (1985). Apache ii: a severity of disease classification system.” *Crit Care Med*, 13.

W. S. J. A. D. S. T. M. M. H. Steven W. Thiel, Jamie M. Rosini. (2010). Early prediction of septic shock in hospitalized patients. *Journal of Hospital Medicine*.

X. ying Liu, J. Wu, and Z. hua Zhou. (*2006)*. Exploratory undersampling for class-imbalance learning. *Proc. ICDM* .

Y. Cui, J. G. Dy, G. C. Sharp, B. M. Alexander, and S. B. Jiang. (2008). Learning methods for lung tumor markerless gating in image-guided radiotherapy in *Proceeding of the 14th ACM SIGKDD Conference*.