Cloud-based Predictive Modeling System and its Application to Asthma Readmission Prediction

Robert Chen, BS[1], Hang Su, BS[1], Mohammed Khalilia, PhD[1], Sizhe Lin[1], Yue Peng, BS[1], Tod Davis, BA[2], Daniel A. Hirsh, MD[3], Elizabeth Searles, RN, BSN, MBA[2], Javier Tejedor-Sojo, MD[2], Michael Thompson, BS, MBA[2], Jimeng Sun, PhD[1]

[1]School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA, USA

[2]Children's Healthcare of Atlanta, Atlanta, GA, USA

[3]Pediatric Emergency Medicine Associates, LLC, Atlanta, GA, USA

Abstract

The predictive modeling process is time consuming and requires clinical researchers to handle complex electronic health record (EHR) data in restricted computational environments. To address this problem, we implemented a cloud-based predictive modeling system via a hybrid setup combining a secure private server with the Amazon Web Services (AWS) Elastic MapReduce platform.

EHR data is preprocessed on a private server and the resulting de-identified event sequences are hosted on AWS. Based on user-specified modeling configurations, an on-demand web service launches a cluster of Elastic Compute 2 (EC2) instances on AWS to perform feature selection and classification algorithms in a distributed fashion. Afterwards, the secure private server aggregates results and displays them via interactive visualization.

We tested the system on a pediatric asthma readmission task on a de-identified EHR dataset of 2,967 patients. We conduct a larger scale experiment on the CMS Linkable 2008-2010 Medicare Data Entrepreneurs' Synthetic Public Use File dataset of 2 million patients, which achieves over 25-fold speedup compared to sequential execution.

INTRODUCTION

High healthcare costs have placed a burden on the federal budget in the United States. Over 90% of Medicare expenditure can be attributed to the management of chronic diseases.[1] However, the quality of care is still far from optimal for many patients, especially those with chronic conditions such as asthma.[2] While there has long been an interest in lowering asthma readmission rates, most predictive modeling studies for asthma have applied a small number of models and may be limited by small datasets. Fortunately, the rapid adoption of electronic health records (EHRs) in healthcare systems provides an exciting opportunity for researchers to leverage this data for secondary uses such as predictive modeling.

While predictive modeling approaches can aid in the detection of readmissions, the predictive modeling process is tedious and time consuming. Researchers often evaluate many models and compare performance metrics between them. Each model may involve different cohort selection criteria, or different features used in predictive modeling tasks. Furthermore, researchers may elect to evaluate several different algorithms in order to choose the best method for predicting a particular target outcome. These iterative predictive modeling efforts will accumulate and lead to large differences in performance metrics attained when comparing the outcomes of different models. Furthermore, with the tsunami of EHR data we need a more scalable computing infrastructure. Taking the aforementioned drawbacks together, we argue that the traditional predictive modeling pipeline is in need of a major overhaul.

With the rapid adoption of EHR systems in hospitals, predictive modeling will be of major interest in the clinical setting. A number of studies have performed predictive modeling for applications such as asthma readmission prediction in hospitals.[3–6] However, most of these studies were done using either standalone software products for statistical analysis, or computer code written independently by researchers. Such approaches are often conducted entirely on the researchers' local computers, and are not scalable with large datasets that are made available as EHR adoption grows rapidly.

Meanwhile, there is evidence that cloud computing can be leveraged in order to support big data analytics on large datasets over a large number of machines in a distributed setting.[7,8] To date, there does not exist a cloud based web service that supports predictive modeling on large healthcare datasets using distributed computing. There have been some implementations of predictive modeling software. For example, McAulley et al. built a standalone application for clinical data exploration and machine learning.[9] However, the tool was run on local machines and was not deployed on the cloud for easy use by others. The lack of development of health analytics systems on the cloud may also partially be due to the concern of privacy and security of patient data on the cloud.

In addition to the problem with large datasets, researchers often run many iterations of predictive modeling studies before arriving at a desired result. Each iteration may involve changes in the study cohort, features used, and specific machine learning algorithms run. Constantly toggling these parts of the process is tedious and may result in errors. Ng et al. developed the PARAMO system, a predictive modeling platform which constructs a large number of pipelines in parallel with MapReduce/Hadoop.[10] However, PARAMO is built on the user's own cluster, which is not always available in every clinical institution, and also lacks scalability when faced with large datasets beyond the capacity of their existing cluster. In addition, most pipelines such as PARAMO are difficult to deploy in a clinical setting due to the large expenses required to maintain servers. Therefore, these systems make little to no impact on clinical decision-making.

To help address the limitations of current predictive modeling pipelines, we developed and deployed a hybrid system that combines a secure private server with the cloud-based Amazon Web Services (AWS) Elastic MapReduce platform. The system consists of a web service that runs on a private server in a secure environment for preprocess patient data into feature matrices, and an on-demand AWS web service to perform predictive modeling computations. Note that such a hybrid setup enables security of the patient data and at the same time leverages the scalable computing infrastructure on the cloud. Our system is highly customizable to support various predictive model configurations. Furthermore, the system is highly scalable, as the number of cloud-based machines launched can increase with the size of input data. Finally, the system is cost effective because the AWS Elastic MapReduce cluster is only launched when predictive modeling jobs need to be run.

We applied our platform to a predictive modeling task of identifying patients at risk for asthma readmission using a cohort of patients from the Children's Healthcare of Atlanta EHR system. As one of the most common chronic illnesses in children, Asthma costs over $56 billion each year, placing a financial burden on the healthcare system.[11] Asthma affects 10.5 million children in the United States annually, and leads to a total of 10.5 million missed school days each year.[12] However, a child whose asthma is properly controlled with education, medication and lifestyle has a better chance of avoiding emergency department (ED) visits as an inpatient. In addressing these issues, care managers want to understand the trends and patterns in the entire patient population. Currently, that is done by grouping patients using diagnosis categories such as Clinical Risk Groups (CRGs)[13] or by risk stratification using risk scores such as clinical risk scores (CRS) for asthma. However, neither approach provides homogeneous patient clusters for purposes of determining targeted treatment protocols. As an alternative, we propose to use our predictive modeling system based on a machine learning strategy to identify patients at high risk for readmission from context-specific information.

In addition to the asthma readmission prediction task, we showed that our system is scalable to large datasets by successfully running a prediction task on the publicly available CMS Linkable 2008-2010 Medicare Data Entrepreneurs' Synthetic Public Use File (DE-SynPUF) dataset, which includes over 2 million patients.

METHODS

In this section we describe the design and implementation of our predictive modeling system.

General Overview

The system is composed of two main elements, a persistent web service in a private environment such as a hospital's internal server and an on-demand web service on a cloud-computing environment such as AWS. The persistent web service constantly runs on a dedicated server, houses raw EHR data, and performs preprocessing of data. Processed data in the form of event sequence files are sent to the predictive modeling module, which is powered by an on-demand web service such as an AWS Elastic MapReduce cluster. Once the predictive modeling module is finished, the results are aggregated and displayed to the user in the performance analysis module running on the persistent web service. Figure 1 illustrates the key components of our system.

Preprocessing

Data from the EHR are first converted into event sequence files to be used as input to the predictive modeling module. The event sequence files are in the form of text files where each line represents one distinct event from the database. Each record is represented by a tuple in the format **(patient, event, timestamp, value)**, where patient, event, timestamp and value represent the patient ID number, event name, date and time of the record, and a value for the event, respectively. In the case of binary events, such as medication and diagnostic events, the value for a tuple is set equal to 1. In the case of events that are normally associated with numerical values, such as lab values, the value for the tuple is set to the numerical value that is present in the record. In the case of categorical events, such as gender recorded in an admission, the value for the tuple is set to the alphabetical value (i.e. ‘F’ and ‘M’ for gender) from the record. The final event sequence files are used as inputs to the predictive modeling module.

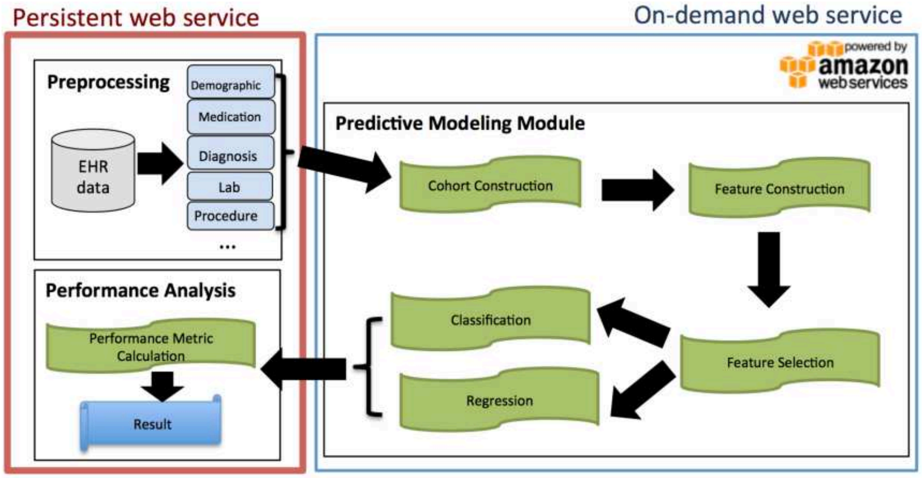


Figure 1. An overview of our predictive modeling system. We extract EHR information from the database from the hospital and store the information in these event sequence files through persistent web services running on a private dedicated server. These event sequence files are uploaded to the web service on the cloud.

Instead of using the raw information such as actual patient IDs and ICD-9 codes, the data in event sequence files are further transformed into internal coded values on the persistent web service before being uploaded to the on-demand AWS web service. In particular, patient IDs and event names can be hashed into internal IDs. Thus, the raw patient information would not be used in the predictive modeling processes running on the cloud for security considerations. After the predictive modeling module is finished, specific patient ID numbers and event names may be decoded on the persistent web server before running the performance analysis module on the dedicated server.

Predictive Modeling Module

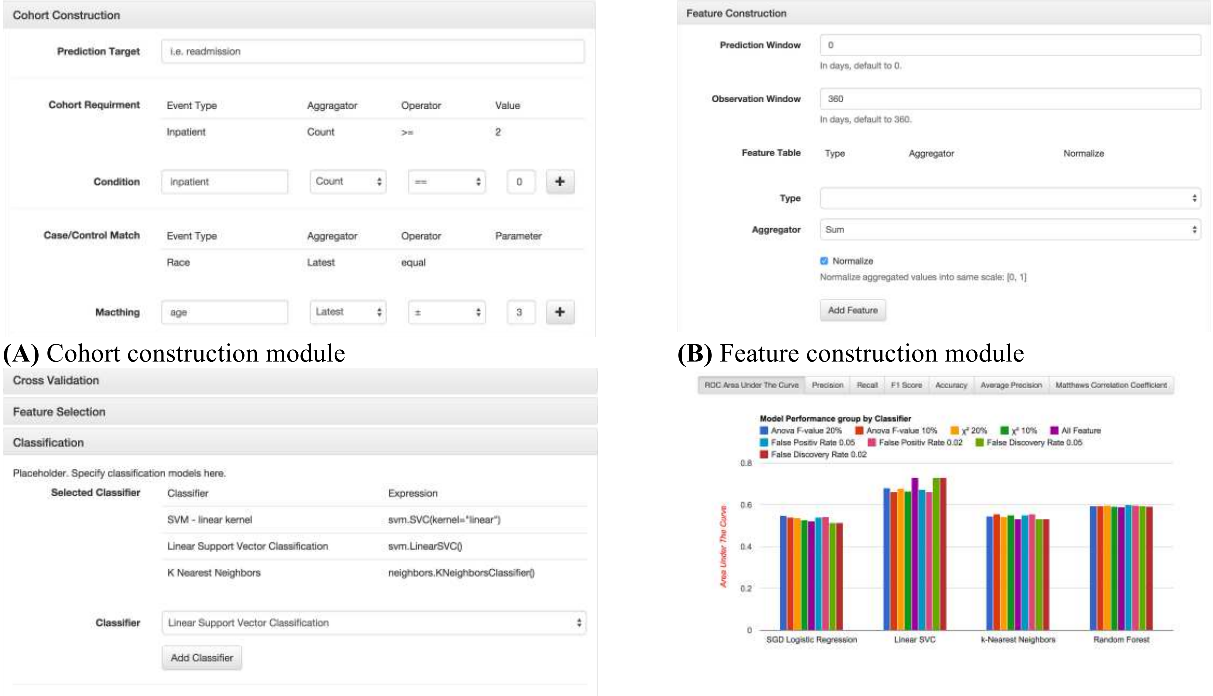
The predictive modeling module consists of several stages. First the cohort construction and feature construction stages are conducted. Next, cross validation stages comprised of feature selection followed by either classification or regression are run. We call a concrete step in one predictive modeling process a *task*. Examples of tasks include constructing diagnoses features, or building a logistic regression classifier on a specific training set. We organize all those tasks into different computation *stages*: cohort construction, feature construction, cross validation data splitting, feature selection, classifier training and classifier testing. Note that each stage corresponds to one or many computational tasks. All of these tasks occur on the on-demand web service, which is implemented with AWS Elastic MapReduce.

The predictive modeling module launches a new AWS Elastic MapReduce cluster consisting of multiple EC2 instances for each analysis workload. The predictive modeling module aggregates all computational tasks from all stages of predictive modeling process and schedules them to run in parallel on the provisioned AWS cluster. The user is allowed to choose the number EC2 virtual machine instances and the types[i] of machines to use in the MapReduce cluster. Next we introduce those computation stages in more details.

Cohort construction: Once event sequence files are uploaded to the on-demand web service, the user can specify a set of filtering criteria that can be used for constructing cohorts (Figure 2A). A user can define the patient selection criteria for cases and controls. For example, for an asthma readmission prediction study, the user may define the case patients by inputting *readmission* as the target event to predict. Patients who have the target event will be regarded as cases and others as controls.

After user defines the target event to predict, the user can further narrow down the cohort to study by specifying conditions. For example, user may require all patients in cohort should have at least 3 *inpatient* events. After the user selects case cohort inclusion conditions, the user may want to balance the number of cases and

[i] See http://aws.amazon.com/ec2/instance-types/ for more details



Cohort Construction

Prediction Target i.e. readmission

Cohort Requirment

Event Type	Aggragator	Operator	Value
Inpatient	Count	>=	2

Condition inpatient | Count | == | 0 | +

Case/Control Match

Event Type	Aggregator	Operator	Parameter
Race	Latest	equal	

Macthing age | Latest | ± | 3 | +

(A) Cohort construction module

Feature Construction

Prediction Window 0

In days, default to 0.

Observation Window 360

In days, default to 360.

Feature Table

Type	Aggregator	Normalize

Type

Aggregator Sum

☑ Normalize

Normalize aggregated values into same scale: [0, 1]

Add Feature

(B) Feature construction module

Cross Validation

Feature Selection

Classification

Placeholder. Specify classification models here.

Selected Classifier

Classifier	Expression
SVM - linear kernel	svm.SVC(kernel="linear")
Linear Support Vector Classification	svm.LinearSVC()
K Nearest Neighbors	neighbors.KNeighborsClassifier()

Classifier Linear Support Vector Classification

Add Classifier

(C) Predictive modeling module

(D) Performance analysis module

Figure 2. Screenshots of the cohort construction module (A), feature construction module (B), predictive modeling module (C) and the performance analysis module (D). The cohort construction module allows users to specify criteria for selection of cases and for identifying matching controls. In the predictive modeling module, the user may specify parameters for particular feature selection and classification algorithms. The performance analysis module allows users to visualize key performance metrics as well as top predictive features selected in feature selection.

controls via matching (Figure 2A). In this situation, the user may elect to select a limited number of matched controls using a matching algorithm based upon patient similarity metrics or propensity scores. Currently, a simple case-control matching algorithm is implemented, which selects control patients that have identical or numerically similar values for the matching criteria as the case patients.

Feature construction: Features from the event sequence input data may be aggregated depending on user preferences and study design. The feature construction module allows the user to specify a method of aggregation for each event type with respect to its values for each patient. The aggregation methods include *mean, sum, count* and *latest*. *Mean* and *sum* are the mean and sum of the values across all certain events for a given patient. *Count* is the number of times the feature occurs as an event. *Latest* is the value for the feature at the most recent occurrence. Figure 2B shows a screenshot of the feature construction module.

During the feature construction phase, besides the feature matrix, other entities will also be computed and stored for later usage. For example, feature value statistics will be used for the performance analysis module on the persistent web service. Feature value statistics we collect include percentage of cases/controls who has certain events and distributions of feature values within cases and controls.

Feature selection: The predictive modeling module runs a combination of feature selection[14] and classification tasks. Since EHR data is often high dimensional in nature, some features may have a large amount of missing or noisy information. Feature selection is used to filter out these features, which should not be considered for predictive modeling.[14]

The feature selection algorithms implemented include raw features, the Chi-square feature selection[15], analysis of variance (ANOVA) F-value based feature selection[16], and false discovery rate based feature selection.[17] The raw features method uses all constructed features from the data. The Chi-square method uses a Chi-squared test to evaluate p-values for each feature with respect to the target labels. A smaller p-value indicates a more predictive feature. For example, user may choose to select features with p-values less than 0.05. Likewise, the analysis of variance (ANOVA) F-score method applies an ANOVA test to each feature with respect to the target labels. The false discovery rate method selects features for being correlated with the target labels using the Benjamini-Hochberg algorithm.[17]

Classification and regression: We included all classification algorithms from the Python scikit-learn[18] package, including logistic regression (LR), support vector machine (SVM), k nearest neighbors (KNN), random forest (RF) and beyond. The user is able to specify parameters for each classifier, for example the type of kernel for SVM, the number of trees for RF, the number of neighbors and the distance metric for KNN, and the method of regularization and type of optimization in LR. Similarly, all regression algorithms from the scikit-learn package are also included.

Cross validation: In the case of classification or regression problems, cross validation is run for each possible pairing of feature selection method followed by classification or regression methods. The user is able to set the number of folds in cross validation and number of iterations to run cross validation (Figure 2B). For each fold of cross validation, feature selection is first run on the training folds of the dataset. The selected features are used in the classification algorithm to be run. The system will figure out the set of tasks to run and their dependencies. For example, on a given fold of cross-validation, classifier *C*'s input depends on the output of feature selector *S* of the same fold, thus the system runs *C* after *S*. For other tasks independent from *C* or *S*, the system could schedule them running in parallel with *C* and *S*.

Parallelization: As described previously, our system runs on an on-demand AWS Elastic MapReduce cluster. The system achieves speed and scalability from two levels of parallelization. The first level is within-task parallelization, which means a predictive modeling task itself is implemented using big data parallel processing techniques. For example, cohort construction and feature construction are implemented using Spark[19]. Thus, different features will be constructed in parallel. Another level is between-task parallelization. All tasks derived from the predictive model configuration form a dependency graph, and the system schedules tasks in topological order. A task will be ready to run whenever its dependencies finish. For example, testing of a classifier could be scheduled to run when the training of the given classifier finishes. All tasks whose dependencies satisfied could run in parallel. For example, if both classifier *A* and *B* depend on feature selector *C*, *A* and *B* can start running simultaneously when *C* finishes. The degree of parallelization will be determined by the capacity of the cluster, which is usually measured by total number of CPU cores and total memory available. All the scheduling and execution are done using Hadoop and Cascading running on AWS MapReduce clusters.

Once all tasks are finished, the system collects and aggregates the results from all clusters' distributed file systems. The results are sent to the performance analysis module on the persistent web service.

Performance Analysis Module

The final module of the system is the performance analysis module on the dedicated server. The results from all classifiers run in the predictive modeling module's MapReduce service are collected, aggregated and stored in a MongoDB database running on the persistent web service. The performance metrics calculated include area under the receiver operating curve (AUC), positive predictive value (PPV), sensitivity, F1 score, accuracy and Matthews correlation coefficient. The web service retrieves the performance metric data from the MongoDB database and displays results to the user in an intuitive interactive interface. Figure 2C shows a screenshot of the webpage displayed to users.

RESULTS

Next we describe the results from the application of our predictive modeling platform to an asthma readmission task.

Asthma Prediction Task Experiment Setup

The study involved a cohort of 2,967 inpatient pediatric asthma patients from the Children's Hospital of Atlanta (CHOA). There were 1,493 patients who had at least one readmission for asthma treatment, and 1,474 patients who did not have any readmissions. Data for inpatient events representing emergency department initial visits and readmission visits were used. Table 1 showcases general patient characteristics of the study cohort.

To run the preprocessing and performance analysis modules, we hosted the persistent web service on dedicated server running the Ubuntu 14.04.1 LTS operating system, with 24 Intel Xeon 2.6GHz processors (six cores each) and 256 GB of RAM. An on-demand web service was launched in order to run the predictive modeling module. The on-demand web service consisted of an AWS Elastic MapReduce cluster consisting of 1 master m3.medium EC2 instance and 20 slave c3.xlarge EC2 instances.

Feature Construction: We preprocessed the data set and obtain 5,728 unique patient visits. Each visit has a readmission label showing that whether or not the visit has led to one or more visits within the next 12 months.

We constructed six groups of features: demographic features, diagnosis features, medication features, procedure features, and visit features. The demographic, diagnosis, medication and procedure features are categorical features while the lab and visit features are numeric features. Table 1 shows descriptions and example values for features in each group. We converted the categorical features to 1-over-K binary code representations. For example, the feature "race" has five distinct categories: *White*, *Black or African American*, *Asian*, *American Indian or Alaska Native*, and *Others*. If a patient belongs to the category *white*, his or her "race" will be represented by a 5-dimensional feature vector [1,0,0,0,0]. The diagnosis and medication features were binary features, where a value of 1 indicates that an event occurred for that feature and a value of 0 indicates that an event did not occur. Furthermore, we convert the numeric features to z-scores. Taking a 40-month-old patient for example, the z-score of feature "age" will be -0.91 given that the average and standard deviation of "age" in our cohort are 100.10 months and 66.13, respectively.

Feature Group	Features	Type	Example Name	Aggregation
	Race	Categorical	White	N/A
Demographic	Sex	Categorical	Female	N/A
	Age	Numeric	40 months	Latest
Diagnosis	ICD9 Code	Categorical	33.9	Count
Medication	Medication Name	Categorical	Albuterol	Count
Procedure	Procedure ID	Categorical	404082	Count
Lab	Lab Name	Numeric	Glucose	Mean
Administration	Length of Stay	Numeric	5 hours	Mean

Table 1: A summary of features constructed in the experiment on prediction of asthma readmission. The aggregation method used during the predictive modeling module is also reported.

***Cohort Construction**:* We obtained a cohort of 1,493 unique patients with asthma readmission within one year after being discharged and 1,474 unique control patients without readmission matched on age in month and gender[ii]. We use a 1919-dimensional feature vector to represent each patient. We summarize the statistics of demographics of the cohort in Table 2.

	All Patients	**Readmission**	**No Readmission**
n=	2967	1493	1474
Age, years (mean)	5.0	5.2	4.9
Gender (% male)	59.9%	59.9%	60.0%
Race (% white)	36.1%	36.0%	36.2%
Race (% black)	52.6%	52.4%	52.7%

Table 2: General patient characteristics of the study cohort. Demographic features are shown for all patients, as well as for patients with at least 1 readmission event, and patients without any readmission events.

Feature selection: We performed feature selection using four separate methods: raw features, ANOVA F-score feature selection, Chi-square feature selection, and false discovery rate (FDR) feature selection.

Classification: We formulated the asthma readmission prediction as a binary classification problem where the two target labels are defined as follows:

1: at least one readmission within 12 months of any inpatient visit
0: otherwise

We applied four commonly used classifiers: logistic regression (LR),[20] linear support vector machine (linear SVM),[21] K-nearest neighbor (KNN), and random forest (RF)[22]. We used stochastic gradient descent with L2 regularization for the logistic regression, set K=1 and use Euclidean distance for KNN, used a linear kernel with c=1 for SVM, and used 100 trees for RF.

Performance analysis: We partitioned the patients into training and testing cohorts in a 3 times 5-fold cross validation process, meaning cross validation was run for 3 iterations. For each fold, we first performed feature selection and then trained the model on the training set (80% of the entire data) using the selected features. Afterwards, we evaluated the model performance on the testing set (20% of the entire data). We used the following evaluation metrics: a) area under the receiver operating characteristic curve (AUC); b) positive predictive value

[ii] There is a larger number of cases than controls because multiple cases can match to the same control patient.

(PPV); c) sensitivity; d) F1 score; e) accuracy. To calculate the final value for each performance metric, we find the mean of the means of each metric across all iterations.

Asthma Experiment Results

Feature selection: Of all of the feature selection methods, the false discovery rate (FDR) method achieved the best overall performance with AUC 0.69, PPV 0.69, sensitivity 0.46, F1 score 0.55, and accuracy 0.77. Table 3 shows the top predictive features selected by the FDR feature selection method in all 10 folds. Six out of the 10 features were verified by pediatric clinicians to be possible indicators for asthma readmission (highlighted in Table 3). Two of the features, the medication *fluticasone-salmeterol* and the lab *total immunoglobulin E (IgE)*, are known to be strong indicators for asthma readmission. The *fluticasone-salmeterol* feature is present in 15% of all cases while present in only 8% of all controls. This result is clinically meaningful because fluticasone-salmeterol is commonly prescribed in more severe asthmatic patients. The *total immunoglobulin E (IgE)* lab value is 565 IU/mL in cases and 258 IU/mL in controls. This result is clinically meaningful as well, since more severe asthmatic patients tend to have higher values for IgE, a marker indicating sensitivity to allergens.[23]

Type	Description	Percent (case)	Percent (control)
Medication	**Montelukast**	**30.3**	**19**
Medication	Fluzone	2.6	1
Diagnosis	**Extrinsic asthma with status asthmaticus**	**26**	**14.1**
Diagnosis	**Other pulmonary insufficiency, not elsewhere classified**	**8.7**	**3.8**
Diagnosis	Contact dermatitis and other eczema, unspecified cause	22	13.1
Medication	**Fluticasone-salmeterol**	**15.2**	**8.1**
Medication	0.9% sodium chloride (PF)	14.4	8.4
Medication	D5-1/2NS	23.2	16.7
Lab	**Point of care hemoglobin test**	**6.9**	**4.7**
Lab	**Total Immunoglobulin E (IgE)**	**1.6**	**0.7**

Table 3: The top 10 most predictive features selected by univariate feature selection based upon ANOVA F-value. Features verified by clinicians to be possible indicators for asthma readmission are shown in bold print.

Classification: We performed cross-validation to choose the appropriate number of features that gives the best performance. Cross validation was performed on each possible combination of feature selection algorithm and classification algorithm. For each feature selection method, we collected all features that met the feature selection criteria. These features were used as predictive features in the classification tasks. Table 4 shows the performance of the linear SVM classifier while using different feature selection methods and raw features. The feature selection method with the highest average of all performance metrics was determined to be the one with the best performance.

	AUC	PPV	Sensitivity	F1	Accuracy
ANOVA	0.63 (0.01)	0.70 (0.01)	0.33 (0.01)	0.45 (0.01)	0.75 (0.004)
Chi-square	0.63 (0.01)	0.69 (0.03)	0.32 (0.01)	0.44 (0.01)	0.75 (0.01)
FDR	0.69 (0.01)	0.69 (0.02)	0.46 (0.01)	0.55 (0.01)	0.77 (0.01)
All features	0.69 (0.01)	0.69 (0.02)	0.46 (0.01)	0.55 (0.01)	0.77 (0.01)

Table 4: Performance of linear SVM with different feature selection algorithms. Feature selection algorithms used include: ANOVA F-value, Chi-square, false discovery rate, false positive rate, and all features. Values shown are mean (standard deviation) across all iterations and folds of cross validation.

The results of the four different classifiers using the feature selected by the FDR feature selection method are shown in table 5. There was variability in performance of the classifiers. Linear SVM achieved the highest AUC (0.69). Logistic regression achieved the highest sensitivity (0.99), while random forest achieved the highest PPV (0.89), F1 score (0.74), and accuracy (0.86).

It is important to consider these results in the context of the particular application. For the asthma readmission prediction problem, the SVM, logistic regression, or random forest methods may all be considered effective models based upon different use cases. In cases where sensitivity may be important (e.g., detecting high risk patients who may need urgent care), logistic regression may be the best model. In cases where positive predictive value may be important (e.g., when treatment for positively predicted patients is expensive, and financial resource allocation is important), then random forest may be the best model.

	Logistic Regression	Linear SVM	KNN	Random Forest
AUC	0.55 (0.004)	0.69 (0.01)	0.52 (0.01)	0.56 (0.01)
PPV	0.31 (0.002)	0.69 (0.02)	0.54 (0.03)	0.89 (0.01)
Sensitivity	0.99 (0.001)	0.46 (0.01)	0.40 (0.01)	0.63 (0.02)
F1	0.47 (0.002)	0.55 (0.01)	0.46 (0.02)	0.74 (0.01)
Accuracy	0.32 (0.004)	0.77 (0.01)	0.71 (0.01)	0.86 (0.002)

Table 5: Performance metrics for four classification algorithms implemented on features selected using the false discovery rate univariate feature selection method. Metrics reported include area under the receiver operating curve characteristic (AUC), positive predictive value (PPV), sensitivity, F1 score, accuracy and Matthews correlation coefficient. Values shown are the mean (standard deviation) across all iterations and folds of cross validation.

System Scalability

To demonstrate the scalability of our system, we ran our system on a much larger dataset, a set of 2.1 million patients from the CMS Linkable 2008-2010 Medicare Data Entrepreneurs' Synthetic Public Use File (DE-SynPUF), a publicly available synthetic dataset. It contains approximately 300,000 different kinds of events from the patients. Thus, there were about 2.1 million patients and 300,000 features in the input dataset. The raw event sequence input file size is 6.88GB. We created a predictive modeling workload with more than 3000 tasks. The feature selection and classifier settings are almost identical to those used in the asthma readmission prediction task. The on-demand web service is composed of 10 r3.2xlarge AWS EC2 virtual machines. Figure 3 shows the timeline of parts of the task run and the amounts of time spent. the CMS data. The entire running time of the pipeline workflow is about 3 hours. To serve as a baseline, we ran all the tasks sequentially on a single server of the same machine configuration to calculate the total running time of a sequential run. We find that our system achieves a 36-fold speedup over the baseline sequential running time. Note that the feature construction step is only conducted once, while the data splitting, feature selection, model training and model testing steps are done for each iteration of cross validation.

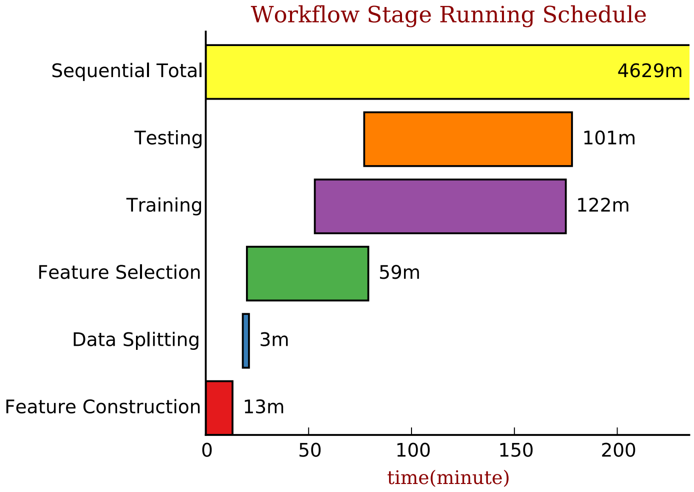


Figure 3: Timeline of modules run and elapsed time. The data-splitting, training and testing times refer to the run times for each respective step of cross validation. Times are shown in seconds (s).

DISCUSSION

We have developed a cloud based system for clinical predictive modeling. Our system is the first of its kind to date, and leverages Amazon Web Services' Elastic MapReduce technology to run distributed feature selection and classification jobs in a time efficient manner.

Challenges in Privacy and Security

While the usage of the cloud is relatively new, many users are already using the cloud for hosting personal health information (PHI).[1,7,24–26] In the case that the users are unwilling to store EHR information into the cloud, our system architecture can be used in such a way that preprocessing of data can be performed on the persistent web server, and PHI can be mapped to codes. For example, all information including patient ID numbers and medication, procedure, lab and diagnosis names may be hashed and mapped to different values such that the raw data were not uploaded to the cloud based system.

Our system mitigates potential concerns regarding privacy and security of healthcare data. However, we also recognize the heuristic nature of our approach, and in the future we plan to conduct more focused studies on privacy consideration using the cloud in an effort to provide a more theoretical guarantee of privacy.

CONCLUSION

We have proposed and implemented a hybrid version of predictive modeling system, which combines a private dedicated instance and public cloud computing services. In this system, raw EHR data are converted into standardized features written into event sequence data files via persistent web services on the private server. The de-identified event sequence files are uploaded to an on-demand web service via Amazon Web Services, which subsequently constructs cohorts and features and schedules a series of distributed predictive modeling tasks using big data systems such as Spark and Hadoop. The results of the predictive modeling tasks are collected and displayed to the user in a highly intuitive, interactive user interface on the private server.

We applied our system to a specific task of prediction for pediatric asthma readmission using a cohort of case patients with asthma readmission and matching control patients. The predictive modeling module was successful in the prediction task through a 5-fold cross validation scheme. The system predicted patients at risk for 12-month asthma readmission with an AUC of 0.69.

We plan to improve upon the system by expanding the suite of cohort construction strategies, feature selection algorithms and classification algorithms. Furthermore, we plan to add functionality for testing multi-class classification tasks (e.g., to be used for detecting multiple types of readmissions).

ACKNOWLEMENT

This work was supported by the National Science Foundation, award #1418511, Children's Healthcare of Atlanta, CDC i-SMILE project, Google Faculty Award, AWS Research Award, Microsoft Azure Research Award, and the National Institutes of Health award T32-GM105490, a Computational Biology and Predictive Health Genomics Training Program at the Georgia Institute of Technology.

References

1. Braunstein, M. L. in *Health Informatics in the Cloud* 1–8 (Springer New York, 2013). at <http://link.springer.com/chapter/10.1007/978-1-4614-5629-2_1>
2. Bodenheimer, T. & Fernandez, A. High and Rising Health Care Costs. Part 4: Can Costs Be Controlled While Preserving Quality? *Ann. Intern. Med.* **143,** 26–31 (2005).
3. Gorelick, M., Scribano, P. V., Stevens, M. W., Schultz, T. & Shults, J. Predicting Need for Hospitalization in Acute Pediatric Asthma: *Pediatr. Emerg. Care* **24,** 735–744 (2008).
4. Reznik, M., Hailpern, S. M. & Ozuah, P. O. Predictors of Early Hospital Readmission for Asthma Among Inner-City Children. *J. Asthma* **43,** 37–40 (2006).
5. Feudtner, C. *et al.* How Well Can Hospital Readmission Be Predicted in a Cohort of Hospitalized Children? A Retrospective, Multicenter Study. *PEDIATRICS* **123,** 286–293 (2009).
6. Salamzadeh, J., Wong, I. C. K., Hosker, H. S. R. & Chrystyn, H. A Cox regression analysis of covariates for asthma hospital readmissions. *J. Asthma Off. J. Assoc. Care Asthma* **40,** 645–652 (2003).
7. Dillon, T., Wu, C. & Chang, E. Cloud Computing: Issues and Challenges. in *2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)* 27–33 (2010). doi:10.1109/AINA.2010.187
8. Kuo, A. M.-H. Opportunities and Challenges of Cloud Computing to Improve Health Care Services. *J. Med. Internet Res.* **13,** (2011).
9. McAullay, D. *et al.* A delivery framework for health data mining and analytics. in *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38* 381–387 (Australian Computer Society, Inc., 2005). at <http://dl.acm.org/citation.cfm?id=1082203>

10. Ng, K. *et al.* PARAMO: A PARAllel predictive MOdeling platform for healthcare analytic research using electronic health records. *J. Biomed. Inform.* doi:10.1016/j.jbi.2013.12.012
11. Barnett, S. B. L. & Nurmagambetov, T. A. Costs of asthma in the United States: 2002-2007. *J. Allergy Clin. Immunol.* **127,** 145–152 (2011).
12. Akinbami, O. J., Moorman, J. E. & Liu, X. *Asthma Prevalence, Health Care Use, and Mortality: United States, 2005–2009*. (US Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Health Statistics, 2011). at <http://www.cdc.gov/nchs/data/nhsr/nhsr032.pdf>
13. Clinical Risk Groups (CRGs): A Classification System for Ris... : Medical Care. *LWW* at <http://journals.lww.com/lww-medicalcare/Fulltext/2004/01000/Clinical_Risk_Groups__CRGs___A_Classification.11.aspx>
14. Guyon, I. & Elisseeff, A. An Introduction to Variable and Feature Selection. *J Mach Learn Res* **3,** 1157–1182 (2003).
15. Manning, C. D., Raghavan, P. & Schütze, H. *Introduction to information retrieval*. **1,** (Cambridge university press Cambridge, 2008).
16. Lomax, R. G. & Hahs-Vaughn, D. L. *Statistical Concepts: A Second Course*. (Routledge, 2013).
17. Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B Methodol.* 289–300 (1995).
18. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J Mach Learn Res* **12,** 2825–2830 (2011).
19. Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S. & Stoica, I. Spark: cluster computing with working sets. in *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing* 10–10 (2010). at <http://static.usenix.org/legacy/events/hotcloud10/tech/full_papers/Zaharia.pdf>
20. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. & Lin, C.-J. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.* **9,** 1871–1874 (2008).
21. Chang, C.-C. & Lin, C.-J. LIBSVM: a Library for Support Vector Machines. at <http://stocktrendresearch.googlecode.com/svn-history/r77/trunk/Paper/SVM_ANN/libsvm.pdf>
22. Breiman, L. Random Forests. *Mach. Learn.* **45,** 5–32 (2001).
23. Pollart, S. M., Chapman, M. D., Fiocco, G. P., Rose, G. & Platts-Mills, T. A. E. Epidemiology of acute asthma: IgE antibodies to common inhalant allergens as a risk factor for emergency room visits. *J. Allergy Clin. Immunol.* **83,** 875–882 (1989).
24. Löhr, H., Sadeghi, A.-R. & Winandy, M. Securing the e-Health Cloud. in *Proceedings of the 1st ACM International Health Informatics Symposium* 220–229 (ACM, 2010). doi:10.1145/1882992.1883024
25. Li, M., Yu, S., Ren, K. & Lou, W. in *Security and Privacy in Communication Networks* (eds. Jajodia, S. & Zhou, J.) 89–106 (Springer Berlin Heidelberg, 2010). at <http://link.springer.com/chapter/10.1007/978-3-642-16161-2_6>
26. Li, M., Yu, S., Zheng, Y., Ren, K. & Lou, W. Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption. *IEEE Trans. Parallel Distrib. Syst.* **24,** 131–143 (2013).