

Clinical Predictive Modeling Development and Deployment through FHIR Web Services

Mohammed Khalilia, Myung Choi, Amelia Henderson, Sneha Iyengar, Mark Braunstein, Jimeng Sun
Georgia Institute of Technology, Atlanta, Georgia

Abstract

Clinical predictive modeling involves two challenging tasks: model development and model deployment. In this paper we demonstrate a software architecture for developing and deploying clinical predictive models using web services via the Health Level 7 (HL7) Fast Healthcare Interoperability Resources (FHIR) standard. The services enable model development using electronic health records (EHRs) stored in OMOP CDM databases and model deployment for scoring individual patients through FHIR resources. The MIMIC2 ICU dataset and a synthetic outpatient dataset were transformed into OMOP CDM databases for predictive model development. The resulting predictive models are deployed as FHIR resources, which receive requests of patient information, perform prediction against the deployed predictive model and respond with prediction scores. To assess the practicality of this approach we evaluated the response and prediction time of the FHIR modeling web services. We found the system to be reasonably fast with one second total response time per patient prediction.

Introduction

Clinical predictive modeling research has increased because of the increasing adoption of electronic health records¹⁻³. Nevertheless, the dissemination and translation of predictive modeling research findings into healthcare delivery is often challenging. Reasons for this include political, social, economic and organizational factors⁴⁻⁶. Other barriers include the lack of computer programming skills by the target end users (i.e. physicians) and difficulty of integration with the highly fragmented existing health informatics infrastructure⁷. Additionally, in many cases the evaluation of the feasibility of predictive modeling marks the end of the project with no attempt to deploy those models into real practice⁸. To achieve real impact, researchers should be concerned about the deployment and dissemination of their algorithms and tools into day-to-day decision support and some researchers have developed approaches to doing this. For example, Soto *et al.* developed EPOCH and ePRISM⁷, a unified web portal and associated services for deploying clinical risk models and decision support tools. ePRISM is a general regression model framework for prediction and encompasses various prognostic models⁹. However, ePRISM does not provide an interface allowing for integration with existing EHR data. It requires users to input model parameters, which can be time consuming and a particular challenge for researchers unfamiliar with the nuances of clinical terminology and the underlying algorithms. A suite of decision support web services for chronic obstructive pulmonary disease detection and diagnosis was developed by Velickovski *et al.*¹⁰, where the integration into providers' workflow is supported through the use of a service-oriented architecture. However, despite these few efforts and many calls for researchers to be more involved in the practical dissemination of their systems, little has been done and much less has been accomplished to utilize predictive modeling algorithms at the point-of-care.

An important missing aspect that retards bringing research into practice is the lack of simple, yet powerful standards that could facilitate integration with the existing healthcare infrastructure. Currently, one major impediment to the use of existing standards is their complexity¹¹. The emerging Health Level 7 (HL7) Fast Healthcare Interoperability Resources (FHIR) standard provides a simplified data model represented as some 100-150 JSON or XML objects (the FHIR Resources). Each resource consists of a number of logically related data elements that will be 80% defined through the HL7 specification and 20% through customized extensions¹². Additionally, FHIR supports other web standards such as XML, HTTP and OAuth. Furthermore, since FHIR supports a RESTful architecture for information and message exchange it becomes suitable for use in a variety of settings such as mobile applications and cloud computing. Recently, all the four major health enterprise software vendors (Cerner, Epic, McKesson and MEDITECH) along with major providers including Intermountain Healthcare, Mayo Clinic and Partners Healthcare have joined the Argonaut Project to further extend FHIR to encompass clinical documents constructed from FHIR resources¹³. Epic, the largest enterprise healthcare software vendor, has a publicly available FHIR server for testing that supports a subset of FHIR resource types including Patient, Adverse Reaction, Medication Prescription, Condition and Observation¹⁴ and support of these resources is reportedly included in their June 30, 2015 release of Version 15 of their software. SMART on FHIR has been developed by Harvard Boston Children's Hospital as a

universal app platform to seamlessly integrate medical applications into diverse EHR systems at the point-of-care. Cerner and four other EHR systems demonstrated their ability to run the same third party developed FHIR app at HIMSS 2014. Of particular importance to our work is the demonstrated ability to provide SMART on FHIR app services within the context and workflow of Cerner’s PowerChart EHR¹⁵.

As a result of these efforts, FHIR can both facilitate integration with existing EHRs and form a common communication protocol using RESTful web services between healthcare organizations. This provides a clear path for the widespread dissemination and deployment of research findings such as predictive modeling in clinical practice. However, despite its popularity, FHIR currently is not suitable to directly support predictive model development where a large volume of EHR data needs to be processed in order to train an accurate model. To streamline predictive model development it is important to adopt a common data model (CDM) for storing EHR data. The Observational Medical Outcomes Partnership (OMOP) was developed to transform data in disparate databases into a common format and to map EHR data into a standardized vocabulary¹⁶. The OMOP CDM has been used in various settings including drug safety and surveillance¹⁷, stroke prediction¹⁸, and prediction of adverse drug events¹⁹. We utilize a database in the OMOP CDM to support predictive model development. The resulting predictive model is then deployed as FHIR resources for scoring individual patients.

Based on these considerations, in this paper we propose to develop and deploy:

- A predictive modeling development platform using the OMOP CDM for data storage and standardization
- A suite of predictive modeling algorithms operating against data stored in the OMOP CDM
- FHIR web services, that use the resulting trained predictive models to perform prediction on new patients
- A pilot test using MIMIC2 ICU and ExactData chronic disease outpatient datasets for mortality prediction.

Methods

Overview and system architecture

Figure 1 shows how our architecture supports providing predictive modeling services to clinicians via their EHR. On the server side, the model development platform trains and compares multiple predictive models using EHR data stored in the OMOP CDM. After training, the best predictive models are deployed to a dedicated FHIR server as executable predictive models. On the client side, users can use existing systems such as desktop or mobile applications within their current workflows to query the predictive model specified in the FHIR resource. Such integrations are done by using FHIR web services. Client applications use FHIR resource to package patient health information and transport it using the FHIR RESTful Application Programming Interface (API). Once the FHIR server receives the information, it passes it on to the deployed predictive model for the risk assessment. The returned result from the predictive model will be sent to the client and also stored into a resource database that can be accessed by the client to read or search the Risk Assessment resources for later use.

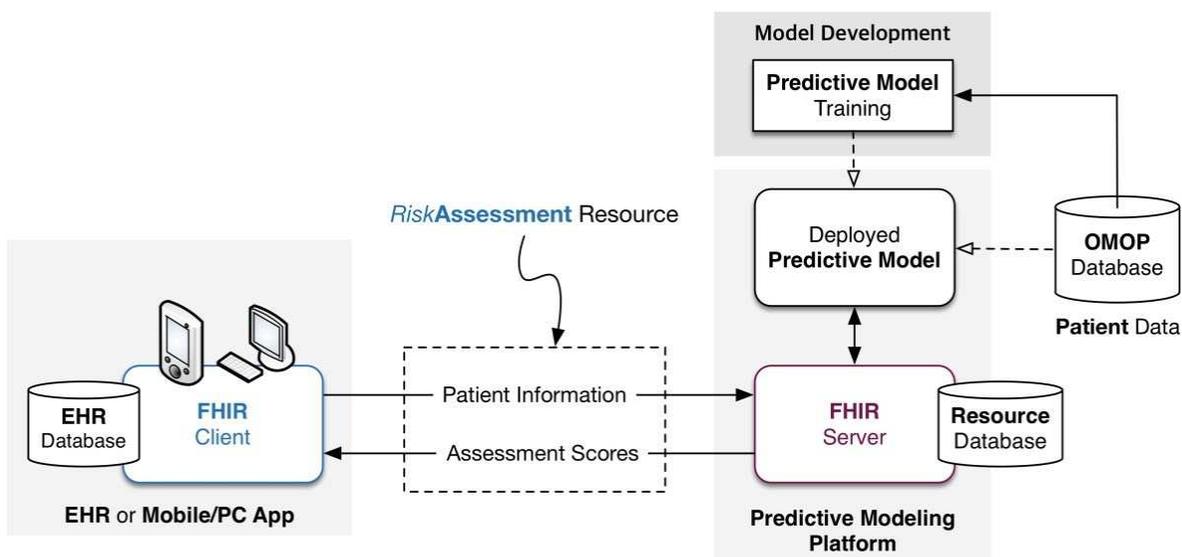


Figure 1. System Architecture

The common data model

Developing a reliable and reusable predictive model, requires a common data model into which diverse EHR data sets are transformed and stored. For the proposed system we used the OMOP CDM designed to facilitate research using some important design principles. First, data in the OMOP CDM is organized in a way that is optimal for data analysis and predictive modeling, rather than for the operational needs of healthcare providers and other administrative and financial personnel. Second, OMOP provides a data standard using existing vocabularies such as the Systematized Nomenclature of Medicine (SNOMED), RxNORM, the National Drug Code (NDC) and the Logical Observation Identifiers Names and Codes (LOINC). As a result predictive models built using data in an OMOP CDM identify standardized features, assuming the mapping to OMOP is reliable. The OMOP CDM is also technology neutral and can be implemented in any relational database such as Oracle, PostgreSQL, MySQL or MS SQL Server. Third, our system can directly benefit the existing OMOP CDM community to foster collaborations.

The OMOP CDM consists of 37 data tables divided into standardized clinical data, vocabulary, health system data and health economics. We only focused on a few of the CDM clinical tables including: person, condition occurrence, observations and drug exposure. As we enhance our Extract, Transform, Load (ETL) process and the predictive model, we can incorporate additional data sources as needed. Figure 2 shows a high level overview of the ETL process in which multiple raw EHR data are mapped to their corresponding CDM instances. In the transformation process, EHR source values such as lab names and results, diagnoses codes and medication names are mapped to OMOP concept identifiers. The standardized data can then be accessed to train the predictive models.

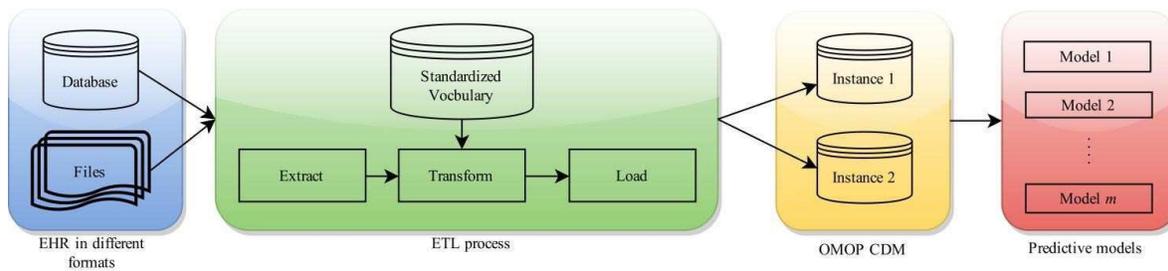


Figure 2. EHR data to OMOP CDM

Predictive model development

The CDM provides the foundation for predictive modeling. As various datasets are transformed and loaded into the OMOP CDM predictive modeling training can be simplified because the OMOP CDM instances all have the same structure. For instance, one can equally easily train a model for predicting mortality, future diseases or readmission using different datasets. Figure 3 provides an overview of predictive model training.

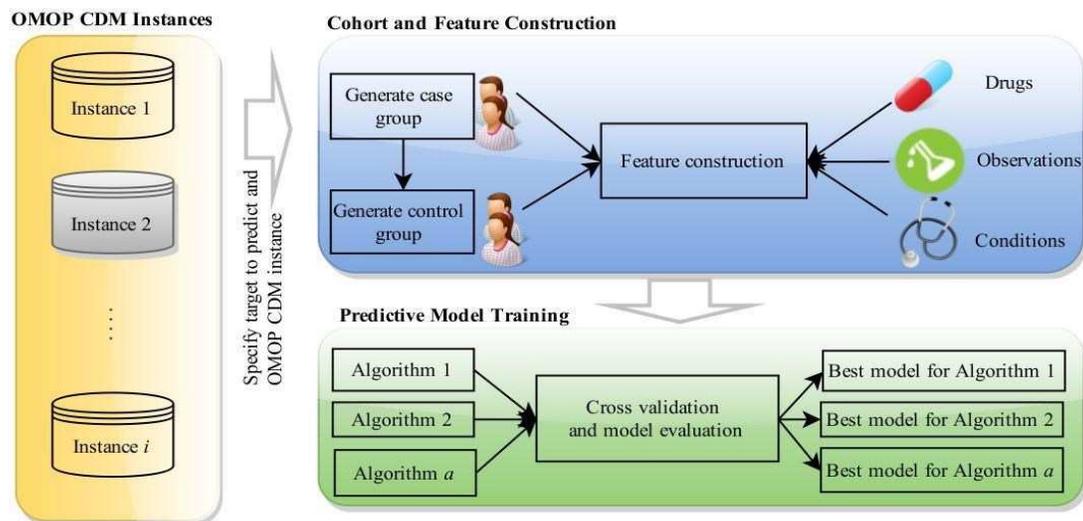


Figure 3. Training predictive models based on OMOP CDM

The predictive models are trained offline and can be re-trained as additional records are added to the database. Training consists of three modules: 1) **Cohort Construction**: this is the first step in the training phase. At this stage the user specifies the OMOP CDM instance, prediction target (i.e. mortality) and the cohort definition. Based on the specified configuration the module will generate the patient cohort. 2) **Feature Construction**: at this stage the user specifies which *data sources* (e.g. drugs, condition occurrence and observations) to include when constructing the features. Additional configurations can also be provided for each data source. The user can include the *observation window* (e.g. the prior year, to utilize only patient data recorded in the past 365 days). Other data source configurations include the *condition type concept identifier* to specify which types of conditions to include (i.e. primary, secondary), *observation type concept identifier* and *drug type concept identifier*. The final configuration is the feature value aggregation function. For example, lab result values can be computed using one of five aggregation functions: sum, max, mean, min, and most recent value. 3) **Model Training**: This module takes the feature vectors constructed for the cohort and trains multiple models using algorithms such as Random Forest, K-Nearest Neighbor (KNN) and Support Vector Machine (SVM). The parameters for each of these three algorithms are tuned using cross validation in order to select the best performing model. The best model will be deployed as a FHIR resource. In the next section, we will describe how to deploy such a predictive model for scoring future patient in real time.

FHIR web services for model deployment

Our approach is focused on API based predictive modeling services that can be easily implemented in thin client applications, especially in the mobile environment. FHIR defines resources represented as JSON or XML objects that can contain health concepts along with reference and searchable parameters. FHIR further defines RESTful API URL patterns for create, read, update, and delete (CRUD) operations. In this paper, we propose to use the *RiskAssessment* resource defined in the FHIR Draft Standard for Trial Use 2 (DSTU2) for our predictive analysis. Readers should note that this particular FHIR resource is still in the draft stage. The current approved DSTU1 version of FHIR does not have a *RiskAssessment* resource. However, our development version of FHIR is currently being balloted on by HL7 members and should be approved soon²⁰. Detailed information about the FHIR development version for *RiskAssessment* can be obtained from Ref. 20.

A prediction request to the FHIR server starts by forming the CREATE operation, which is used to request for scoring specific patients in real-time using the deployed predictive model. This creates *RiskAssessment* resources at the server. Client applications then receive a status response with a resource identifier that refers to the newly created resource. Clients can use this resource identifier to read or search the resource database via a FHIR RESTful API. For this paper, we used a SEARCH operation as we need to retrieve more than one result. Groups of FHIR resources are called a bundle and there is a specified format for that. As all performed analyses are stored in the resource database, additional query types can be implemented in the future. The operation process is depicted in Figure 4.

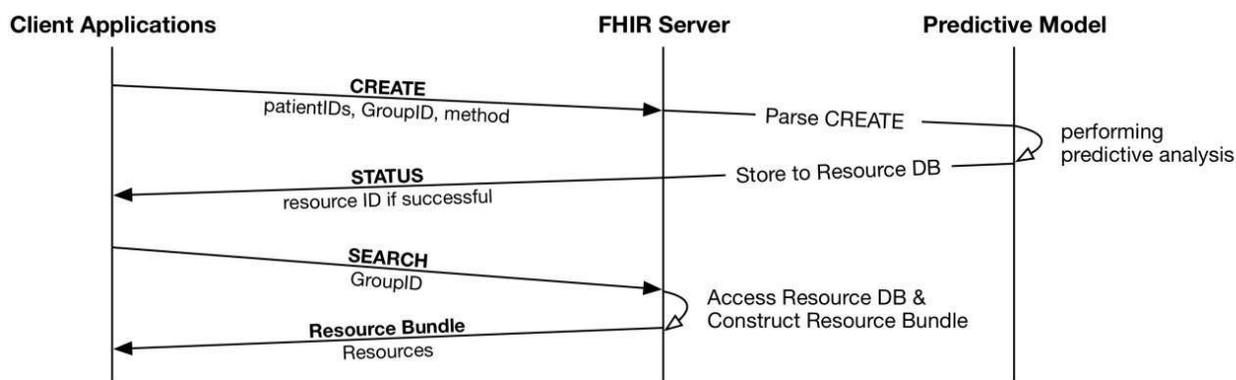


Figure 4. Operation Process

During the process, clients and server need to put appropriate information into elements available in the *RiskAssessment* resource. However, most of the data elements in the *RiskAssessment* resource are optional for predictive modeling which gives us flexibility in choosing the model output. For the CREATE operation, the *RiskAssessment* resource is constructed with *subject*, *basis*, and *method* elements as shown in Figure 5(a).

```

{"resourceType": "RiskAssessment",
 "basis": [
   { "reference": "Patient/patient 1 identifier" },
   { "reference": "Patient/patient 2 identifier" },
   ...
 ],
 "method": [ { "text": "predictive model identifier" } ],
 "subject": [ { "reference": "Group/group identifier" } ]
}

{"entry": [
 {"resource": "RiskAssessment",
  "id": "patient 1 identifier",
  "prediction": { "probabilityDecimal": patient 1 score },
  "method": "predictive model identifier",
  "subject": { "reference": "Group/group identifier" }
 },
 {"resource": "RiskAssessment",
  "id": "patient 2 identifier",
  "prediction": { "probabilityDecimal": patient 2 score },
  "method": "predictive model identifier",
  "subject": { "reference": "Group/group identifier" }
 },
 ...
 ]}

```

(a) Request

(b) Response

Figure 5. JSON RiskAssessment FHIR request and the corresponding response

Subject is used to define a *group identifier*, which we also refer to as the *resource identifier*. *Basis* can contain information used in assessment, which are the patient identifiers. All patients included in *basis* are bound to the group identifier specified in the *subject* element. Therefore, the results of assessment for patients will be grouped in the resource database by the group identifier at the FHIR server. *Subject or group identifier* provides a simple mechanism for users to specify a group of patients, whose prediction score can be retrieved from the resource database.

Predictive model scoring creates feature vectors (one for each patient), which are derived from the patient’s health information. The FHIR *Patient* resource can be referenced by either the patient identifier or by a collection of resources such as *MedicationPrescriptions*, *Conditions*, *Observations*, etc. In our initial implementation we used patient identifier. Using the patient identifier, the predictive model constructs feature vectors by pulling the appropriate information from various resources that contain each patient’s clinical data occurring within the specified observation window. In our implementation the clinical data is stored in the OMOP database. This all happens during the predictive analysis period in Figure 4. In the future, we will construct features directly from EHR databases through querying other FHIR resources in real-time.

Results from the predictive analysis are sent back to the FHIR server in JSON format and the FHIR server stores the information in a resource database. If the CREATE operation is successful, the server will send a 201-status created message. In case of any errors, an appropriate error status message will be sent back to the client with the *OperationOutcome* resource, if required¹².

Clients can query the prediction results using the *resource identifier* returned in the CREATE response (i.e. *group identifier* under *subject* in this case). This *resource identifier* can be used to query the stored prediction results using the SEARCH operation. Recall that the *resource identifier* is bounded to a list of patient identifiers. Once the SEARCH operation request is received with the *resource identifier*, the FHIR server constructs a response FHIR resource for each patient. In the *RiskAssessment* resource, the *prediction* element contains the risk score for a patient, as shown in Figure 5(b). When the SEARCH operation is completed, the *RiskAssessment* resources are packaged into a bundle and sent to the client. For example, the resource contains an *identifier* that represents the patient for whom the predictive analysis is performed. In our experiments, mortality prediction was performed and risk scores are returned. The mortality scores are populated in the *probabilityDecimal* sub element of the *prediction* element, as shown in Figure 5(b).

Experiments

We tested our implementation using two datasets: Multiparameter Intelligent Monitoring in Intensive Care (MIMIC2) and a dataset licensed from ExactData. MIMIC2 contains comprehensive clinical data for thousands of ICU patients collected between 2001 and 2008. ExactData is a custom, large realistic synthetic EHR dataset for

chronic disease patients. Using ExactData eliminates the cost, risk and legality of real EHR data. Table 1 presents key statistics about the two datasets.

Table 1. Key ExactData and MIMIC2 Dataset Statistics

	ExactData	MIMIC2
Number of patients	10,460	32,074
Number of condition occurrences	313,921	314,648
Number of drug exposures	82,258	1,736,898
Number of observations	772,189	19,272,453
Number of deceased patients	53	8,265

Two ETL processes were implemented to move the raw MIMIC2 and ExactData datasets into two OMOP CDM instances. From each instance a cohort was generated for mortality prediction. The cohort for ExactData is limited to the 53 patients with death records matched with 53 control patients to keep it balanced. The MIMIC2 cohort is somewhat larger with 500 case patients and 500 control patients. To generate the control group we performed a one-to-one matching with the case group based on age, gender and race. As illustrated in Figure 3, the feature vectors for each cohort are generated using these MIMIC2 and ExactData cohorts. The features include condition occurrence, observation and drug exposures. The observation window was set to 365 days for both cohorts and the feature values for observations were aggregated by taking the mean of the values.

For each cohort three mortality predictive models were trained offline using Random Forest, SVM and KNN. After performing cross validation and parameter tuning for each algorithm, the model with the highest Area Under the Receiver Operating Characteristic curve (AUC) was deployed. This results in three final models, one for each algorithm, allowing clients to specify which algorithm to use to predict new patients. The predictive models were trained using Python scikit-learn machine learning package²¹. The final model training runtime, AUC, accuracy and F1 score are reported in Table 2.

Table 2. Model evaluation using ExactData and MIMIC2 cohorts

	Algorithm	Runtime (seconds)	AUC	Accuracy	F1 score
Exact Data	KNN	2.5	0.87	0.75	0.76
	SVM	2.5	0.90	0.81	0.82
	Random Forest	2.52	0.95	0.84	0.84
MIMIC2	KNN	392	0.71	0.70	0.77
	SVM	387	0.80	0.63	0.77
	Random Forest	404	0.78	0.74	0.81

Our entire software platform was evaluated and tested on a small Linux server with 130GB hard drive, 16GB memory, and two Intel Xeon 2.4GHz processors with two cores each. FHIR web services were built on the Tomcat 8 application server using the HL7 API (HAPI) RESTful server library²². The FHIR web service expects a request from the client to perform patient scoring (in our case mortality prediction) with patient data or patient identifier contained in the request body. We evaluated the performance of the web service in two ways. The first evaluation method measures the *response time* from the client's request to the web service until a response is received back from the server. This is an important measure since a very slow response time makes the system unusable for practical applications including at the point-of-care where busy clinicians demand a prompt response. A single API request to score a patient is a composite of two different requests: 1) *CREATE request* made to the FHIR server that sends the data in the JSON format. The data comprises of resource type, a set of patient identifiers, classification algorithm to be employed (e.g. KNN, SVM or Random Forest) and the OMOP CDM instance to be used (e.g. MIMIC2 or ExactData). The response time of this request increases with the number of patients (as can be observed from Figures 6 and 7). *SEARCH request* is made by passing the patient group identifier as a parameter to obtain the prediction for the patients. Similar to the CREATE request, a larger number of patients increases the response time of the SEARCH request.

Figure 6 shows the FHIR web service response time using KNN and MIMIC2 datasets as we vary the number of patients to score in the request. For a single patient, a typical point-of-care scenario, the response time is around one second. The CREATE request consumes most of the time and the response time increases with the number of patients. This could become an issue in an application such as population health to identify those chronic disease patients most likely to be readmitted or screening an entire ICU to identify patients most in need of immediate medical attention. The SEARCH request has a relatively constant response time and does not change significantly as the number of patients increases.

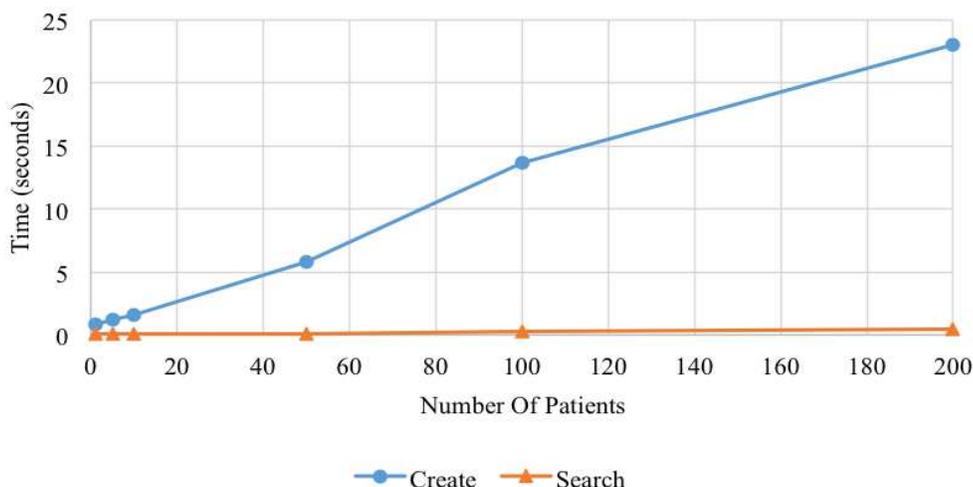


Figure 6. Response time for CREATE and SEARCH requests using KNN and MIMIC2 data vs. the number of patients

The actual scoring or prediction of the patient is performed in the CREATE request. There are two major tasks that take place in this CREATE request. First is the feature construction that is performed for each patient included in the request. This requires querying the OMOP CDM database, extracting and aggregating feature values, which can be an expensive operation when scoring a large number of patients. The second task is the actual prediction operation which takes the least amount of time. Figure 7 shows the response time (after subtracting the feature construction and prediction time), feature construction time, and prediction time as we increase the number of patients. As the number of patients increases the prediction time does not increase much compared to the response time or the feature construction time. For instance, a request for scoring 200 patients takes about 23 seconds, out of which 8 seconds were spent constructing features and only 16 milliseconds were spent on the actual predictions.

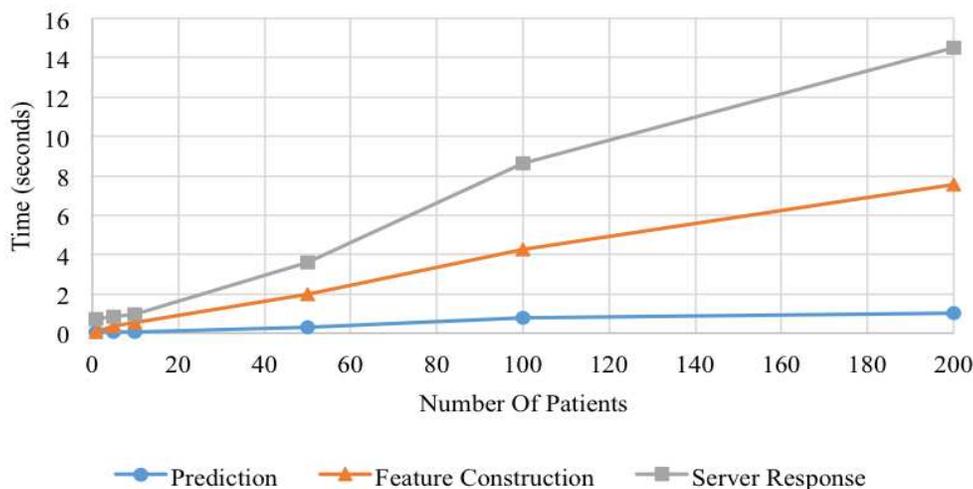


Figure 7. Response time, feature construction and prediction time vs. number of patients

The second web service performance evaluation measures the amount of load that the service can handle within some defined time interval. As this service handles multiple clients it is important to guarantee that it is available and able to respond in a timely manner to requests as the number of clients increases. However, keep in mind that our evaluation is only a proof of concept done on a moderate server. Figure 8 shows the average response time when 1000 clients send CREATE requests (one patient in each request) to the FHIR server within a minute. Figure 8 is generated by the Simple Cloud-based Load Testing tool²³ and has three axes: response time on y-axis, number of clients on secondary y-axis and time on x-axis, where 00:05 indicates the fifth second from the start of the test. The green curve (upper curve) shows the number of simultaneous clients sending the requests at some point in time. For instance, on the fifth second, about 50 clients were sending simultaneous requests. The blue line shows the average response time for those clients (on the fifth second, the average response time for 50 clients was about 1800 milliseconds). Overall, the average response time for the 1000 clients is 1506 millisecond (about 1.5 seconds). A 1.5 second response time is acceptable when the task is performed at the point-of-care.

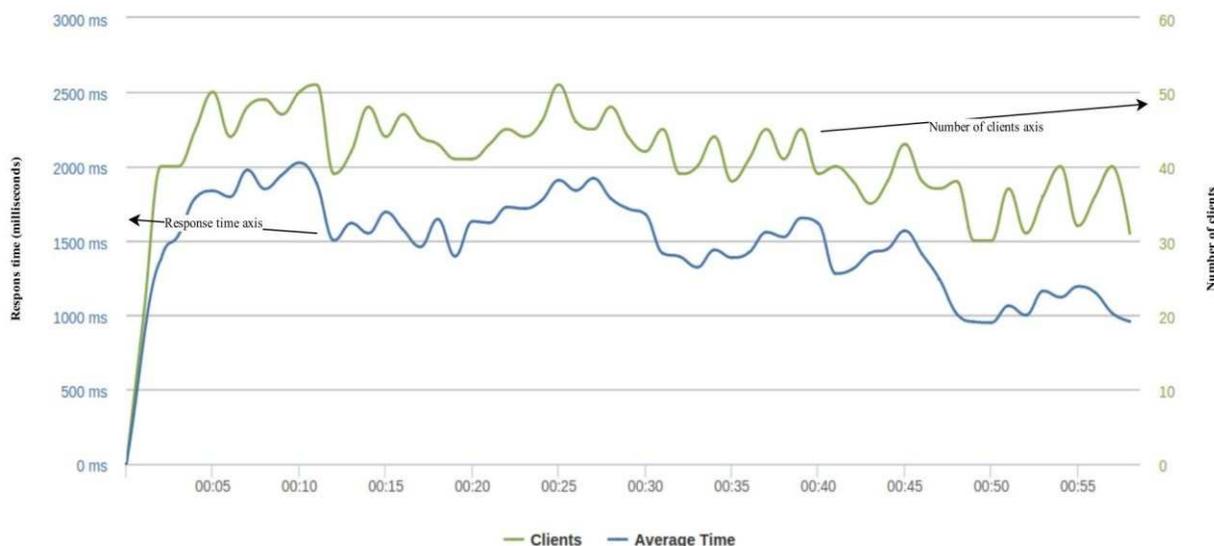


Figure 8. Average response time as 1,000 clients send CREATE request within one minute

Conclusion and Discussion

In this paper we presented a real-time predictive modeling development and deployment platform using the evolving HL7 FHIR standard and demonstrated the system using MIMIC2 ICU and ExactData chronic disease datasets for mortality prediction. The system consists of three core components: 1) **The OMOP CDM** which is more tailored for predictive modeling than healthcare operational needs and stores EHR data using standardized vocabularies. 2) **Predictive model development**, which consists of cohort construction, feature extraction, model training and model evaluation. This training phase is streamlined, meaning that it will work for any type of data stored in an OMOP CDM. 3) **FHIR web services for predictive model deployment** that we use to deliver the predictive modeling service to potential clients. The web service takes a prediction request containing patient identifiers whose features can be extracted from the OMOP database. The prediction or the scoring is not pre-computed but is performed in real-time. Our future work on the FHIR web services will enhance the feature extraction by accessing EHR data over FHIR Search operations. In this case, our predictive platform will become a client to the FHIR-enabled EHR. Recent FHIR ballot for DSTU2 includes a *Subscription* resource²⁴. This resource, if included in the standard, can be utilized in our platform to subscribe the patient's feature related data from their EHR. The *Subscription* resource uses Search string for its criteria, thus our future work will comply with this new resource with only a few modifications. Even if this resource couldn't be included, our future platform will use a pull mechanism to retrieve EHR data for feature extraction.

A total of six predictive models were trained on MIMIC2 ICU and ExactData chronic disease datasets from which we generated cohorts based on patients with a death event and a matched set of control patients. The FHIR web service routes the incoming requests to the desired predictive model, which is usually a client specified parameter. For a practical, real-time web service it is important to achieve a fast response time. The observed response time for scoring one patient was around one second, of which the actual prediction took only few milliseconds. The total response time increases with the number of patients in a single request, but the actual prediction time remains very

small, reaching 16 milliseconds when scoring 200 patients. However, scoring many patients in one request is not always desired or needed. For instance, in many cases providers are only interested in querying one patient at a time from the point-of-care. In such a direct care delivery scenario only a single request containing one patient's information would be sent to the server with an expected response time of one second. Additionally, this response time can be easily reduced by expanding the server hardware and optimizing the implementation.

We present a prototype with much work and many needed improvements yet to be done. First, additional predictive algorithms should be added. We demonstrated Random Forest, SVM and KNN, but others such as Logistic Regression and Decision Trees could be added. Second, the system should allow for updating the predictive model as new patients get scored. This requires using the patient EHR data passed into the web service for retraining the predictive model. For this to take place, the web service should allow the provider to give feedback on the patient scoring, which can then be used to improve future predictions. Third, the system needs to be scalable and should handle larger datasets to train the predictive models. This can be done by utilizing big data technologies such as Hadoop²⁵ or Apache Spark²⁶. Fourth, the response time needs to be improved especially for requests that contain large numbers of patients, which can be done by allocating more resources and improving the FHIR web service software. Additionally, the response time can be decreased if the client includes patient EHR data in the request, thus avoiding expensive database querying. Fifth, the web service must have privacy and security protocols implemented such as OAuth2, which has been already implemented in the SMART on FHIR app platform.

The proposed approach in this paper assumes that the client that is deploying our predictive model stores the patient data in an OMOP CDM model and the CREATE request contains the identifier for the patient for which the analysis is to be performed. Any ETL process for moving data from EHR to OMOP model is outside the scope of this paper. However, FHIR resources such as *MedicationPrescriptions*, *Conditions*, *Observations* can be included in the body of the CREATE request instead of the patient identifier. This allows for performing analysis for patients not included in the database. Additionally, two processes can be performed on the FHIR resources, if passed in the CREATE request: 1) perform prediction using the patient data (resources) in the request and 2) subscribe the patient's features related data from EHR to the predictive platform.

In conclusion, we have demonstrated the ease of developing and deploying a real-time predictive modeling web service that uses open source technologies and standards such as OMOP and FHIR. With this web service we can more easily and cost effectively bring research into clinical practice by allowing clients to tap into the web service using their existing EHRs or mobile applications.

Acknowledgment

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, and Microsoft Azure Research Award.

References

1. Brownson, R. C., Colditz, G. A. & Proctor, E. K. *Dissemination and implementation research in health: translating science to practice*. (Oxford University Press, 2012).
2. Dobbins, M., Ciliska, D., Cockerill, R., Barnsley, J. & DiCenso, A. A Framework for the Dissemination and Utilization of Research for Health-Care Policy and Practice. *Worldviews Evid.-Based Nurs. Presents Arch. Online J. Knowl. Synth. Nurs.* **E9**, 149–160 (2002).
3. Sun, J. *et al.* Predicting changes in hypertension control using electronic health records from a chronic disease management program. *J. Am. Med. Inform. Assoc. JAMIA* (2013). doi:10.1136/amiajnl-2013-002033
4. Glasgow, R. E. & Emmons, K. M. How Can We Increase Translation of Research into Practice? Types of Evidence Needed. *Annu. Rev. Public Health* **28**, 413–433 (2007).
5. Wehling, M. Translational medicine: science or wishful thinking? *J. Transl. Med.* **6**, 31 (2008).
6. Hörig, H., Marincola, E. & Marincola, F. M. Obstacles and opportunities in translational research. *Nat. Med.* **11**, 705–708 (2005).
7. Soto, G. E. & Spertus, J. A. EPOCH and ePRISM: A web-based translational framework for bridging outcomes research and clinical practice. in *Computers in Cardiology, 2007* 205–208 (2007). doi:10.1109/CIC.2007.4745457
8. Bellazzi, R. & Zupan, B. Predictive data mining in clinical medicine: Current issues and guidelines. *Int. J. Med. Inf.* **77**, 81–97 (2008).
9. Soto, G. E., Jones, P. & Spertus, J. A. PRISM trade;: a Web-based framework for deploying predictive clinical models. in *Computers in Cardiology, 2004* 193–196 (2004). doi:10.1109/CIC.2004.1442905

10. Velickovski, F. *et al.* Clinical Decision Support Systems (CDSS) for preventive management of COPD patients. *J. Transl. Med.* **12**, S9 (2014).
11. Barry, S. The Rise and Fall of HL7. (2011). at <<http://hl7-watch.blogspot.com/2011/03/rise-and-fall-of-hl7.html>>
12. Bender, D. & Sartipi, K. HL7 FHIR: An Agile and RESTful approach to healthcare information exchange. in *2013 IEEE 26th International Symposium on Computer-Based Medical Systems (CBMS)* 326–331 (2013). doi:10.1109/CBMS.2013.6627810
13. Miliard, M. Epic, Cerner, others join HL7 project. *Healthcare IT News* (2014). at <<http://www.healthcareitnews.com/news/epic-cerner-others-join-hl7-project>>
14. Open Epic. at <<https://open.epic.com/Interface/FHIR>>
15. Raths, D. SMART on FHIR a Smoking Hot Topic at AMIA Meeting. *Healthcare Informatics* (2014). at <<http://www.healthcare-informatics.com/article/smart-fhir-smoking-hot-topic-amia-meeting>>
16. Ryan, P. B., Griffin, D. & Reich, C. OMOP common data model (CDM) Specifications. (2009).
17. Zhou, X. *et al.* An Evaluation of the THIN Database in the OMOP Common Data Model for Active Drug Safety Surveillance. *Drug Saf.* **36**, 119–134 (2013).
18. Letham, B., Rudin, C., McCormick, T. H. & Madigan, D. *An Interpretable Stroke Prediction Model using Rules and Bayesian Analysis.* (2013). at <<http://dspace.mit.edu/handle/1721.1/82148>>
19. Duke, J., Zhang, Z. & Li, X. Characterizing an optimal predictive modeling framework for prediction of adverse drug events. *Stakehold. Symp.* (2014). at <<http://repository.academyhealth.org/symposia/june2014/panels/7>>
20. HL7 FHIR Development Version. at <<http://www.hl7.org/implement/standards/FHIR-Develop/>>
21. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J Mach Learn Res* **12**, 2825–2830 (2011).
22. HL7 Application Programming Interface. at <http://jamesagnew.github.io/hapi-fhir/doc_rest_client.html>
23. Simple Cloud-based Load Testing. at <<https://loader.io/>>
24. FHIR Resource Subscription. at <<http://hl7.org/fhir/2015May/subscription.html>>
25. Shvachko, K., Kuang, H., Radia, S. & Chansler, R. The Hadoop Distributed File System. in *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)* 1–10 (2010). doi:10.1109/MSST.2010.5496972
26. 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>