REPORTS OF
THE MACHINE LEARNING AND INFERENCE LABORATORY

INTEGRATING COMPLEX HEALTH DATA FOR ANALYTICS

JANUSZ WOJTUSIAK, EMAN ELASHKAR, AND REYHANEH MOGHARAB NIA

MLI 16-1
OCTOBER 19, 2016

RESEARCH AND EDUCATION IN MACHINE LEARNING

Integrating Complex Health Data for Analytics

Janusz Wojtusiak, Eman Elashkar, and Reyhaneh Mogharab Nia

Machine Learning and Inference Laboratory

Department of Health Administration and Policy

George Mason University

Abstract

This report describes the yearlong project aimed at exploring a novel approach to integrate complex health data from different sources without the need of building a data warehouse. The approach allows preprocessing data, creating an analytic file and applying analytic algorithms while data are still distributed in their respective data sources. It relies on using data semantic mapping into a common ontology needed for data to be queried on conceptual level, rather than requiring users to know physical location and coding of the data.

To test the method, it has been applied to creation of computational model for predicting 30-day post hospital discharge mortality. The Computational Length of stay, Acuity, Comorbidities and Emergency visits (C-LACE) is an attempt to improve accuracy of popular LACE model frequently used in hospital setting. The model has been constructed and tested using MIMIC III data. The model accuracy (AUC) on testing data is 0.74. The model is available in the form of online calculator which is available in two versions: user-oriented based on 20-most important mortality indicators, and API-based which uses over 300 patient characteristics.

Acknowledgements

This project has been supported by the LMI-Academic partnership program grant. The authors thank Donna Norfleet, Chris Bistline, Brent Auble and all those who participated in our monthly meetings for their support and feedback that helped improve the project. The authors also thank Haile Moges for his participation in the early stages of the project.

1. Introduction

We live in a post-EHR era. Electronic health records (EHRs), personal health records (PHRs) and consumer-generated data are facts, and in many cases are required by laws and regulations. This is true for the Department of Defense (DOD), the Veteran Health Administration (VHA), other government-run healthcare systems, as well as for private healthcare. EHRs and health information exchange are past the research phase and are being implemented (slowly) by the industry and used in daily practice.

The real research questions arise once all the systems are in place and data are collected. *How can the data be used to improve health?* This question can be looked at from multiple angles. A common approach is to relate the improvement of quality and cost of care provided to the patients. On the broader spectrum health, which is not the same as health care, includes everything from nutrition, exercise, lifestyle and work all the way to the actual medical conditions and care. Improving health involves all of these areas.

Health data is extremely complex. In fact, health data is more complex than any other data because of its multimodality, large number of standards and coding systems and most importantly extremely complicated application area. This complexity creates a gap between even the most sophisticated analytical methods (including those used in machine learning and data mining), which work off flat and pre-processed data files that use uniform coding of data elements, and are organized by the unit of analysis (i.e., patient). Surprisingly, virtually all available methods "blindly" analyze data without any semantic knowledge about data, relationships between data elements, and "meaning" of what is being analyzed. This means that all the richness of what comes along with data is ignored (in fact only the NLP community stresses semantics of data while the ML and DM communities ignore it).

Electronic data exchange, mapping between different terminologies and data elements, as well as related privacy/security issues are widely addressed in the context of a single patient's data. For example, the meaningful use of electronic health records requires systems to have the ability to electronically transmit data of a patient. However, the existing standards, including HL7's CCA, CCD and FHIR standards, are not designed to handle datasets comprising thousands of patients' records whose copious data is needed for analytics. This makes traditional health information exchanges not adequate for large scale data integration needed for analytics.

The goal of the project was to investigate possibility of integration of health data without need for building a data warehouse, loading data to distributed file system, or building federated database. This breaks with standard approach in health data analytics: (1) integrate all data; (2) preprocess data; (3) create "analytic file"; (4) apply analytic algorithms (standard statistics, or ML/DM); and (5) repeat 2-4 if more data or additional variables are needed. In the investigated approach, there is no need for step 1 which is very time consuming and expensive (typically also impossible for a single project or when integrating data across multiple institutions). Sections 2 and 3 describe design and implementation of the method.

The hypothesis is that there is no need for constructing a data warehouse to integrate data from multiple distributed sources assuming that these sources can map their data to a common ontology (i.e., the Unified Medical Language System (UMLS) in case on health data). This hypothesis is tested by constructing a working prototype system that is capable of integrating data based on the semantic relationships. Conclusion at the end of this report outlines some limitations and discusses cases in which the presented approach may be better solution than traditional data integration.

The second goal of the project is to test the developed method by applying it to construction of computational models for predicting 30-day post-hospital discharge mortality for ICU patients. Section 3 of this report describes construction and testing of the model. The model is also made available online as a web calculator for researchers who want to apply it to own data.

Section 2 of this report presents background and design of the constructed data integration approach. Section 3 discusses prototype implementation. Section 4 presents application of the method to analysis of clinical data and construction of models for predicting patient mortality. Finally, Section 5 summarizes the project efforts and describes several directions of future work.

2. Distributed Online Data Integration

The project aims at creating a novel way of integrating data and its semantics during the analysis phase. This is in contrast to the currently used approaches in which data first needs to be transformed into a data warehouse (Bache et al., 2015), loaded into Big-Data repositories such as the Hadoop File System (HDFS), or integrated within a federated system (Ghaleb and Farag, 2015). Instead, the new approach is based on communication between independent systems, each responsible for its own data and communication using well-defined sets of queries. Such an integration provides not only a novel approach of communication between distributed data sources, but also changes the way machine learning and analytics software works. While several other methods for model learning from distributed data exist, they do not follow semantic integration principles pursued here (Meeker at al., 2015).

Traditionally, data analytic methods are applied to pre-processed data extracted from operational systems, loaded to a warehouse and consequently analyzed (Figure 1 left). While clean and easy to use by analysts, this approach is not feasible when dealing with big data, nor should it be used when integrating and analyzing complex healthcare data. In domains where data needs to be secure and only selected information is shared, a different approach is required.

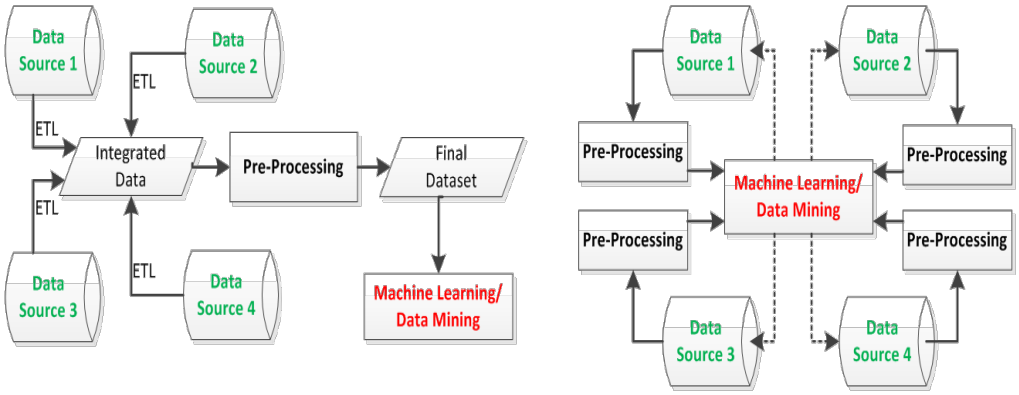

Figure 1: Traditional (left) and new (right) approaches to learning from integrated data. Solid lines represent data flow. Dashed lines in the right figure represent data request.

This project investigated the possibility of specifying analytic problems first, and then integrating data as needed (Figure 1 right). Distributed systems "understand" their own data and are best equipped to pre-process data, pre-analyze and map concepts and infer complex relationships. In fact, the distributed systems not only map existing data to standardized formats and concepts, but can run advanced analytic algorithms on their own subsets of data and transmit resulting models/relationships. Despite significant progress in the areas of Health Data Exchange, Interoperability, Data Warehousing, Big Data, and Health Analytics, to the best of the PIs knowledge no such framework currently exists.

An analytic module depicted in the center of Figure 1 (right) starts with defining the analytic problem and request initial list of concepts in order to build first "skeleton" dataset. The dataset is based on list of needed concepts, rather than specific fields in database. Based on the initial dataset the module can start application of analytic algorithm or request additional data from distributed sources, which are received in already pre-processed form that can be directly plugged-in the data so far. This is an iterative process as more data may be needed during the analysis. This is illustrated from the perspective of the analytic module in Figure 2a.

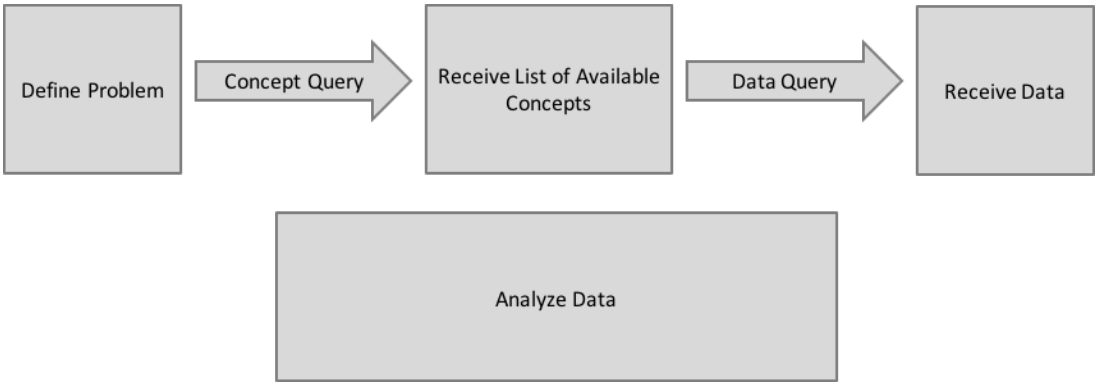

Figure 2a: Data request and retrieval from the perspective of the analytic module.

Key component of the process is the fact that the data are pre-processed by distributed data sources and no or minimum data manipulation is needed by the analytic module. It is reasonable to assume that data sources should understand own data, and be capable of mapping that data into desired terminology or concepts, as well as inferring concepts not explicitly available in the data. As illustrated in Figure 2b, once data request is received, the data source determines if it is based on existing data or needs to be inferred. The data request is further delegated to database engine for aggregation or encoding or to Machine Learning/Decision Support engine for needed extraction and inference of data that is not readily available. Once inferred, the data can be aggregated and encoded and returned as a response to the original request. One important aspect of the method is that the data is integrated on the level specified in request, which typically corresponds directly to the unit of analysis in the data mining problem.

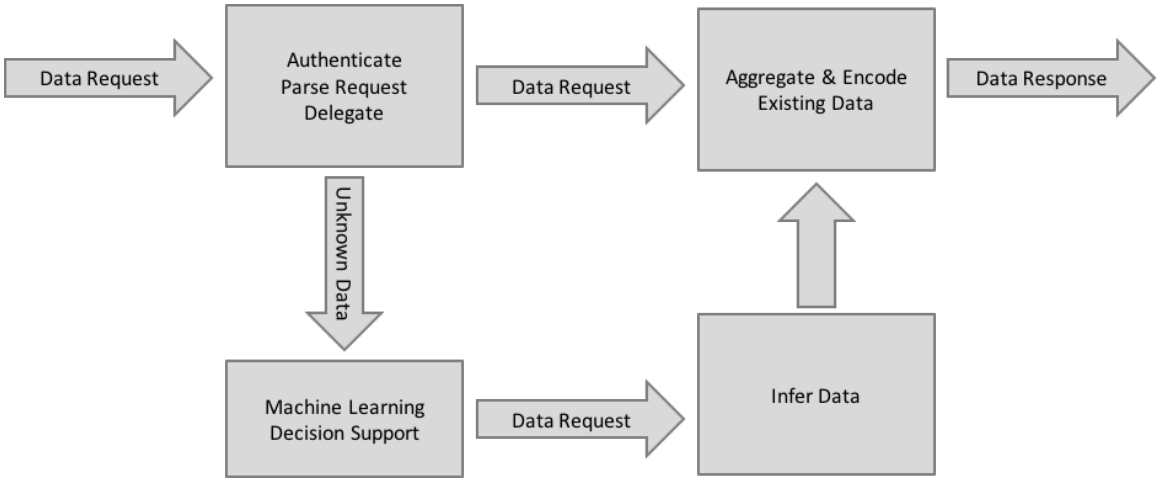

Figure 2b: Data request and preparation from the perspective of distributed data sources.

One important aspect of the data sources is their ability to infer data not explicitly present in database. The inference may include a number of standard approaches such as climbing hierarchies or following semantic relationships in an ontology. For example, the request for data may include concept of heart failure (HF), however the data are encoded using ICD-9 codes. The data source then infers the HF concept by identifying all ICD-9 codes that are part of the HF definition. Another possibility is using Natural Language Processing methods to extract information from unstructured data, such as clinical notes. For example, the data request may include information about smoking status of a patient (i.e., current smoker, past smoker, never smoked) that is not available in the coded data. Through NLP methods that information can be extracted from clinical notes, aggregated, coded and returned to requestor. Finally, the process may include application of classification or regression models (a.k.a., predictive modelling) to infer information such as predicted 6-month mortality or expected complications.

2.1. Communication with Distributed Data Sources

The presented approach is based on a set of Application Programming Interfaces (APIs) and communication schema implemented over standard http/https protocol. The actual data is encoded and transmitted using JSON (JavaScript Object Notation) which easily integrates with most modern programming languages. Two basic types of JSON messages are passed requests (for concept lists or data) and responses to these requests.

The concept list request is a simple query addressed to /resources/concepts with the following content:

```
["GetConcepts"]
```

If approved, the query results in returned JSON message containing the set of available concepts and data fields. It is encoded the following format:

<resource name>:{‘columns’:[<list of available data columns>],
‘concepts’: [<list of available concepts>]}

For example, the message with list of concepts returned by the patient index server (See Section 4) is shown below.

```
{'patients': {'columns': ['ROW_ID', 'SUBJECT_ID', 'GENDER', 'DOB', 'DOD', 'DOD_HOSP', 'DOD_SSN', 'EXPIRE_FLAG'], 'concepts': ['C1717139', '', 'C0079399', 'C0421451', '', '', '', '']}}
```

The request for data is more complicated as it needs to include lists of desired concepts or columns of data, and constraints for obtaining them. The request for data includes the following elements:

[‘GetData’, <resource name>:{‘columns’:[<list of available data columns>],
‘concepts’: [<list of available concepts>],
‘conditions’:[<list of conditions>]}]

For example, the request for patients’ data that include explicit request for two columns (SUBJECT_ID and DOD) along with two concepts ('C0079399', 'C0421451') is generated as follows:

```
['GetData', 'patients', {'columns': ['SUBJECT_ID', 'DOD'], 'concepts': ['C0079399', 'C0421451']}]
```

2.2. Semantic Data Mapping

The most important requirement for the presented method to work is mapping of all data in distributed sources to a common ontology. Such a mapping allows for data querying based on its meaning rather than specific fields and coding. The use of ontology also allows for inference and use of semantic relationships used in that ontology.

In the case of health related data, a good choice is the Unified Medical Language System (UMLS). The Unified Medical Language System is a common ontology that includes over 1 million biomedical concepts and 5 million concept names; it contains hundreds of controlled source vocabularies like SNOMED CT, LOINC, ICD-10, ICD-9CM, RxNorm, HCPT, HL7 and many others as the "Matatheusaurus". This metathesaurus are connected meaningfully through a "Semantic Network", which assigns semantic relationships to concepts from metatheusaurus. The ontology includes a "SPECIALIST Lexicon" that includes information needed for text processing. Each concept in the UMLS has a Concept Unique Identifier (CUI). Since a concept may have several different names, a CUI is used to explain the specific intended meaning of each name in each source vocabulary to link names from all source vocabularies that has same meaning (synonyms). (www.nlm.nih.gov)

The following sections show the mapping process of the data to the UMLS. It uses the example of MIMIC III data described further in Section 4 that describes details of testing of the developed method.

2.2.1. Mapping Variables to MIMIC data

Mapping of attributes was done within PostgreSQL database system and consisted of a set of SQL queries combined with manual searches using UMLS' graphical interface as needed. First step of mapping encompassed identifying chosen attributes in the data, and then locating those attributes in associated tables, as shown in Table 1 below. For each attribute or concept, specific corresponding codes (i.e., ICD9 for diagnosis, LOINC code for lab tests and GSN code are codes given to drugs in MIMIC III, and they correspond to NDDF drug codes dictionary in UMLS) were identified.

The mapping process is a combination of manual labor-intensive identification of appropriate concepts which requires strong domain background of the person performing the mapping, with automated search for concepts between different terminologies in UMLS. The latter can be done when original data stored in database are coded using one of standard terminologies, but the final results still need to be verified by human experts.

NAME	CONCEPT	TABLE	FIELD	CODE	CODE FIELD	itemid	field	Range
Serum Na+ (mEq/L)	C0365095	labevents	label	2951-2	loinc_code	50983	itemid	95-215
Blood Urea Nitrogen (BUN)	C0365240	labevents	label	3094-0	loinc_code	51006	itemid	<300
glucose	C0364479	labevents	label	2339-0	loinc_code	50809	itemid	10-1800
glucose Point in time	C0484731	labevents	label	2345-7	loinc_code	50931	itemid	
albumin	C0363885	labevents	label	1751-7	loinc_code	50862	itemid	0.9-6
Total bilirubin level	C0368753	labevents	label	5770-3	loinc_code	51464	itemid	<50
Pao2	C0802809	labevents	label	19994-3	loinc_code	50816	itemid	>15
PaCO2	C0550246	labevents	label	11557-6	loinc_code	50818	itemid	<120
PH	C0550447	labevents	label	11558-4	loinc_code	50820	itemid	6.6-7.7
HCO3 (To calculate PH)	C0364096	labevents	label	1963-8	loinc_code	50882	itemid	
eGFR (creatinine values, age, gender, and race)	C3811844	labevents	label	to_be_calcul	loinc_code		itemid	
Serum Creatinine (for ARF Acute Renal Failure)	C0364294	labevents	label	2160-0	loinc_code	50912	itemid	0-40
Hematocrit	C0366777	labevents	label	4544-3	loinc_code	51221	itemid	10-70.

Table 1: Illustration of the process of locating attributes in MIMIC III data, obtaining their codes and mapping the codes to corresponding CUIs in UMLS.

One important issue concerning the mapping is disambiguation of concepts and need to select the most appropriate one from potentially large number of concepts with intersecting meanings.

Since Metathesaurus aims at preserving meaning, a concept uniquely explains a single meaning in the UMLS. Each concept is uniquely assigned a Concept Unique Identifier (CUI) and it includes several atoms (terms) from all sources. While atoms may have different names and terms, they still have the same meaning (Synonyms) and they all point (map) to the same concept (Cui) (www.nlm.nih.gov)

Concept (CUI)	Terms (LUIs)	Strings (SUIs)	Atoms (AUIs) * RRF Only
C0004238 Atrial Fibrillation (preferred) Atrial Fibrillations Auricular Fibrillation Auricular Fibrillations	**L0004238** Atrial Fibrillation (preferred) Atrial Fibrillations	**S0016668** Atrial Fibrillation (preferred)	**A0027665** Atrial Fibrillation (from MSH) **A0027667** Atrial Fibrillation (from PSY)
		S0016669 (plural variant) Atrial Fibrillations	**A0027668** Atrial Fibrillations (from MSH)
	L0004327 (synonym) Auricular Fibrillation Auricular Fibrillations	**S0016899** Auricular Fibrillation (preferred)	**A0027930** Auricular Fibrillation (from PSY)
		S0016900 (plural variant) Auricular Fibrillations	**A0027932** Auricular Fibrillations (from MSH)

Figure 3: Concepts, terms, strings and atoms in UMLS illustrating the need to select appropriate identifiers for data elements.

On the other hand, atoms may share same name or term, but point to a different concept. This would be due to they have a different meaning, semantic type or context.

Mapping to UMLS included 2 parts:

1. Mapping attributes codes obtained from the previous step to the corresponding CUIs in Unified Medical Language System (UMLS) (i.e., mapping ICD9 or LOINC obtained to the corresponding CUI)

2. Standardized mapping of more general concepts including demographic and administrative data to the corresponding CUIs. The mapping was based on selecting the most general or most clinically sound corresponding CUI. For example; the concept "Age":

Cui1: C0001779: Age: Semantic Types: Organism Attribute.

Looking at relationship of semantic type: Parents: Conceptual Entity. Children: Clinical Attribute

Cui2: C1114365: Age: Time: Point in time:^Patient: Quantitative: Semantic Types: Clinical Attribute

Looking at relationship:

Parent: Organism Attribute.

The general rule used in selecting concept was to use the most general, which in the case of Age is CUI1: C0001779.

The following example in Figure 4 shows selection of a concept among similar terms that share same name but not the same meaning:

```
mimic=> select * from d_labitems where label like '%UREA NITROGEN%';
 row_id | itemid |           label           |      fluid       | category  | loinc_code
--------+--------+---------------------------+------------------+-----------+------------
    304 |  51104 | UREA NITROGEN, URINE      | URINE            | CHEMISTRY | 3095-7
     52 |  50851 | UREA NITROGEN, ASCITES    | ASCITES          | CHEMISTRY | 12265-5
    206 |  51006 | UREA NITROGEN             | BLOOD            | CHEMISTRY | 3094-0
    245 |  51045 | UREA NITROGEN, BODY FLUID | OTHER BODY FLUID | CHEMISTRY | 3093-2
(4 rows)

mimic=> select cui, code, sab, str from mrconso where code = '3094-0' and sab = 'LNC';
   cui    |  code  | sab |                                  str
----------+--------+-----+-------------------------------------------------------------------------
 C0365240 | 3094-0 | LNC | Urea nitrogen [Mass/volume] in Serum or Plasma
 C0365240 | 3094-0 | LNC | Urea nitrogen:Mass Concentration:Point in time:Serum/Plasma:Quantitative
 C0365240 | 3094-0 | LNC | Urea nitrogen:MCnc:Pt:Ser/Plas:Qn
 C0365240 | 3094-0 | LNC | BUN SerPl-mCnc
(4 rows)
```

Figure 4: Meaningful mapping of concepts; selecting the LOINC code that corresponds to Urea Nitrogen in Blood over others explaining Urea Nitrogen in other body fluids while mapping Blood Urea Nitrogen.

2.2.2. Future Work on Semantic Mapping: Automation

The current research effort focuses on automation of the above process in order to minimize expert input needed during the mapping. The general idea is to use existing terminologies to anchor the data inside UMLS, and then follow relationships within the ontology until the most general concepts are retrieved.

Such mapping done in SQL is shown in the Figure 5 below.

NAME	CONCEPT	TABLE	FIELD	CODE	CODE
Prednisone	C0032952	Prescriptions	drug		gsn
Prednisone 5 MG/ML Oral Solution	C0705898	Prescriptions	drug	006745	gsn
Prednisone 1 MG Oral Tablet	C0690120	Prescriptions	drug	006748	gsn
Prednisone 10 MG Oral Tablet	C0690121	Prescriptions	drug	006749	gsn
Prednisone 20 MG Oral Tablet	C0979757	Prescriptions	drug	006751	gsn
Prednisone 5 MG Oral Tablet	C0989249	Prescriptions	drug	006753	gsn
Prednisone 50 MG Oral Tablet	C0690128	Prescriptions	drug	006754	gsn
Prednisone 2.5 MG Oral Tablet	C0690123	Prescriptions	drug	006750	gsn
Cyclosporine	C0010592	Prescriptions	drug		gsn
cyclosporin, Ophthalmic Suspension	C1168858	Prescriptions	drug	051820	gsn
antivenom,puff adder	C1725525	Prescriptions	drug	011683	gsn
cyclosporine, modified 100 MG/ML C	C2609935	Prescriptions	drug	023883	gsn
Cyclophosphamide	C0010583	Prescriptions	drug		gsn
Cyclophosphamide 1000 MG Injectic	C1712128	Prescriptions	drug	008765	gsn
Cyclophosphamide 2000 MG Injectic	C4048562	Prescriptions	drug	008767	gsn

```
WITH RECURSIVE children AS
(SELECT cui1,cui2 FROM mrrel WHERE cui1 = 'C0032952' and rel='CHD'
UNION ALL
SELECT a.cui1,a.cui2 FROM mrrel a  JOIN children b ON(b.cui2 =a.cui1)
WHERE rel='CHD' )
select p.cui2 as cui, max(m.str) as name , r.code as NDDF_code
from children p , mrconso m, rxnconso r
where p.cui2 = m.cui and m.scui =  r.rxcui and r.sab='NDDF'
group by p.cui2, r.code;
```

cui	name	nddf_code
C0690120	PREDNISONE 1 MILLIGRAM In 1 TABLET ORAL TABLET [PrednisONE]	006748
C0690121	prednisone 10 MILLIGRAM In 1 TABLET ORAL TABLET [Prednisone]_#1	006749
C0690123	PREDNISONE 2.5 MILLIGRAM In 1 TABLET ORAL TABLET [PrednisONE]	006750
C0690124	prednisone 5 MILLIGRAM In 1 TABLET ORAL TABLET [Prednisone]_#3	045267
C0690128	PREDNISONE 50 MILLIGRAM In 1 TABLET ORAL TABLET [PrednisONE]	006754
C0705898	PREDNISONE 5 MILLIGRAM In 1 MILLILITER ORAL SOLUTION, CONCENTRATE [PrednisONE]	006745
C0979757	PREDNISONE 20 MILLIGRAM In 1 TABLET ORAL TABLET [PrednisONE]	006751
C0979758	Prednisone Tab 25 MG	006752
C0979761	PREDNISONE PWDR	006744
C0982851	Prednisone Syrup 5 MG/5ML	006746
C0982851	Prednisone Syrup 5 MG/5ML	006747
C0989249	PREDNISONE TABLETS 5 MG	006753
C1814982	prednisone 10 MILLIGRAM In 1 TABLET ORAL TABLET [Prednisone]_#2	006749
C1814982	prednisone 10 MILLIGRAM In 1 TABLET ORAL TABLET [Prednisone]_#2	045268
C1814983	prednisone 10 MILLIGRAM In 1 TABLET ORAL TABLET [Prednisone]_#3	006749
C1814983	prednisone 10 MILLIGRAM In 1 TABLET ORAL TABLET [Prednisone]_#3	045268
C1814984	prednisone 5 MILLIGRAM In 1 TABLET ORAL TABLET [Prednisone]_#2	045267

Figure 5: Applying a rule of automation by mapping a concept that corresponds to a certain drug dose to the general concept of the drug and vice versa (child concept to parent or parent to child)

2.3. Future Direction: Query Language

We are also investigating possibility of using SQL-like query language in the project. While the analytic module communicates with data resources through well-defined protocols and set of APIs, it is still required for users to specify what required data elements are needed for the final analysis. Our initial investigation suggests that a declarative SQL-like language has the greatest potential for use in the distributed system. Users should be able to specify "columns" in the final data they are interested in along with the method of aggregation. The scripting language should automatically combine data manipulation with analysis and prediction capabilities. A pseudocode below illustrates part of such a query.

```
CREATE prediction model ON
    SELECT admission_id CUI=C0184666, age CUI=C0001779, gender CUI=C0079399,
        death_30 (CUI=1148348 - CUI=2361123 <= 30)
        primary_dx CUI=C00332137, number_icu count(CUI=C0583239),
        abnormal_ekg CUI=C0522055 OPTIONAL
    FROM admissions, patients, icu
    GROUP BY admission_id
USING RandomForest (output_attribute=death_30, ignore_attributes=admission_id)
```

Figure 5: Example query code for aggregation of distributed data.

The above query resembles a standard SQL query, but it includes constructs and methods that go beyond simple data manipulation. The three sources (admissions, patients, icu) are not tables/views like in SQL, but rather distributed data sources accessible through APIs. The requested values given by CUIs may not be explicitly stored in these systems, but rather can be calculated from data. Each requested data element (column) is identified by its name and corresponding CUI. The column *abnormal_ekg* is marked as optional, meaning that data is requested only when needed by the learning process. Finally, the selected data is used by the *RandomForest* algorithm to create a model for predicting 30-day mortality.

3. Implementation & Testing

The developed method has been implemented in a prototype software application and tested on real clinical data. This resulted in an open-source, platform independent implementation that follows the microservice programming paradigm. The implementation of servers and analytical module (client) was done using Python 3 programming language. However, additional servers or clients can be implemented on any platform using virtually any programming language as long as they are capable of communicating with already implemented parts of the system using standardized

interfaces. The following main libraries were used in the implementation:

pandas (servers, client) that provides data structures and querying capabilities for data.

psycopg2 (servers) that provides interface to PostgreSQL database from Python.

json (servers, client) that encodes data needed for transmission.

flask (servers) that is responsible for retrieving and parsing http requests (web server).

requests (client) which is a http client used to query data sources.

matplotlib (client) which is used to visualize results.

In addition, a number of standard Python libraries were used.

During the implementation, Spyder environment has been used for coding as shown in Figure 6. The development was done using Anaconda Python Distribution.

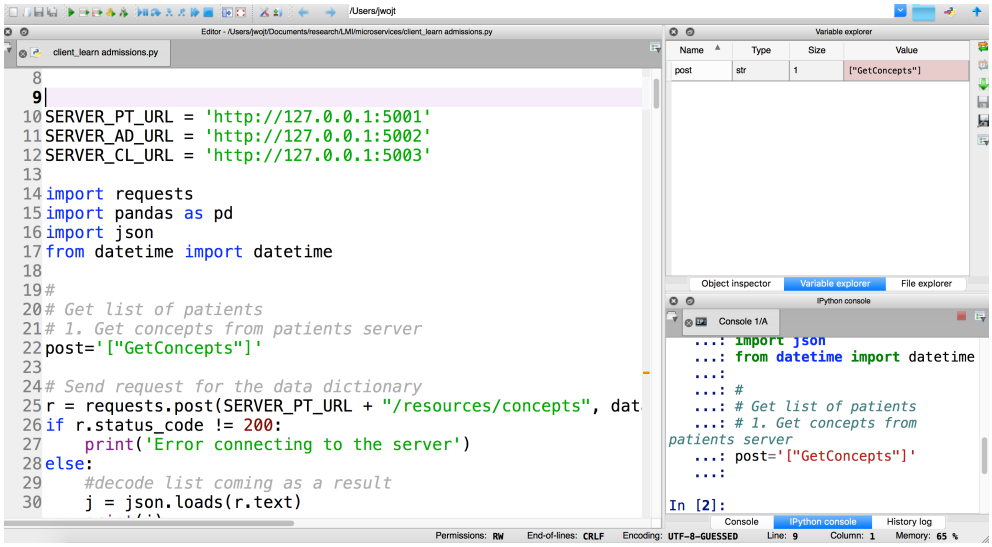

Figure 6: Spyder environment used in programming of the prototype system. The window shows part of the client source code.

In the created implementation, neither servers nor analytic client were equipped with graphical interface. Figure 7 below shows console windows indicating readiness of servers to receive data requests.

Figure 7: Servers indicating readiness to receive data requests.

4. Prediction of 30-day Post-Hospitalization Mortality

4.1. Background

Risk Adjusted Mortality Rates and prediction of mortality are important indicators for care outcome. The applications of risk adjusted mortality are vast. Administrators use mortality rate to compare effectiveness of care among different facilities and hence utilize results in quality improvement efforts. Policy makers and organizations including government agencies, managed care companies and consumer groups (Inouye et al., 1998) use them to evaluate health policy and plan programs. Clinicians are mostly interested in accurate and valid mortality prediction models for many reasons; they can be a strong tool for evaluation of medical effectiveness among treatment groups while controlling for patients' baseline risk. Prediction of mortality index helps clinicians to decide if a patient may benefit from intensive care units and when, and to provide better planning of care. From patient's family perspective, discussing outcome of critically ill patients is always welcomed and appreciated (Rocker et al, 2004).

Several Traditional burden of illness indices are used and adapted to calculate probability of mortality in ICU patients. This includes advanced life support, physical and cognitive functional status of patient (Davis et al, 1995) and intensive care clinicians and bedside nurse estimated probability of intensive care unit survival.

Some illness severity scoring systems that are primarily used to measure severity of illness early in the course of critical illness and its prognosis had been widely used to calculate in-hospital mortality. The Simplified Acute Physiology Score (SAPS), and the Mortality Prediction Model (MPM) use data collected within one hour of ICU admission. Sequential Organ Failure Assessment (SOFA) scoring uses data obtained 24 hours after admission and then every 48 hours. Logistic Organ Dysfunction Score and the Multiple Organ Dysfunction Score [MODS] and the Sequential Organ Failure Assessment Score also had been used recently to measure severity of illness at the time of ICU admission. Acute Physiologic and Chronic Health Evaluation (APACHE) scoring system is widely used to predict risk of in hospital mortality of ICU patients. The instrument uses the worst physiologic values measured within 24 hours of admission to the ICU and requires a large number of clinical variables including age, diagnosis, some laboratory results, prior treatment location, and other clinical variables. A computer generated logistic regression model is run using the resulting score to calculate risk of in-hospital mortality. Most of these illness severity measures gave a baseline assessment of patients and have helped in estimating prognosis and mortality rate. However, the use of these scoring systems as prognostic tools to predict risk of mortality for individual patients is still limited for low accuracy.

Other prediction models using less number of variables had been used to predict mortality within 30 days of hospital discharge. The most commonly used is LACE index. LACE index can use both primary and Administrative data. The name LACE explains variable used in the instrument: length of stay ("L"); acuity of the admission ("A"); comorbidity or diagnoses of the patient (uses Charlson comorbidity score) ("C"); and

emergency department visits (the number of visits in the six months before admission) (“E”). Scores using the LACE index range from 0 (2.0% expected risk of death or urgent readmission within 30 days) to 19 (43.7% expected risk). (Walraven et al, 2010)

A recent study added an extension of the LACE (LACE+) which uses same 4 items of LACE added to age and items unique to Canadian administrative databases (such as the Canadian Institute for Health Information Case Mix Groupings and number of hospital days awaiting alternate level of care arrangements). LACE+ had shown more accuracy in predicting death within 30 days of hospital discharge (c statistic 0.77) while LACE index (c-statistic 0.68). (Walraven et al, 2010). However, both instruments didn’t show sufficient accuracy besides it is not always possible to obtain data on the 4th item of LACE (”E”), as emergency room visits are not necessarily recorded in available data.

4.2. Data

The primary dataset used to test the research question is MIMIC III (Johnson et al., 2016) which is part of PhysipNet project (Goldgerger et al., 2000). The dataset includes a variety of patient and clinical information about hospitalizations, ICU, and patient history. MIMIC III comprises over 58,000 hospital admissions for 38,645 adults and 7,875 neonates. The data spans June 2001 - October 2012. The rationale of using MIMIC III in this project is that it includes much more complex and diverse information than typically found in claims data. One of our goals is to illustrate that learning models from such data using the described method leads to better results than those that can be obtained from claims only data.

From the MIMIC III data, we selected only admissions for patients at least 65 years old. This results in selection of 21,651 admissions. The distribution of selected attributes in the data is presented in Table 2. Within the data, the majority of patients were treated in Medical Intensive Care Units (MICU), followed by Cardiac Surgery Recovery Units (SCRU), Cardiac Care Units (CCU), Surgical Intensive Care Units (SICU) and Trauma Surgical Intensive Care Units (TSICU) as depicted in Figure 8. It can also be observed that the majority of patients were hospitalized only once (Figure 9).

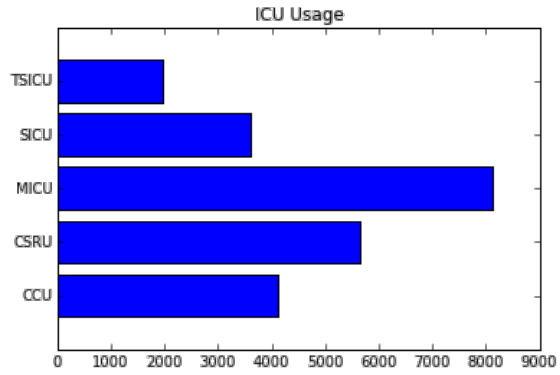

Figure 8: Distribution of types of Intensive Care Units usage in the data.

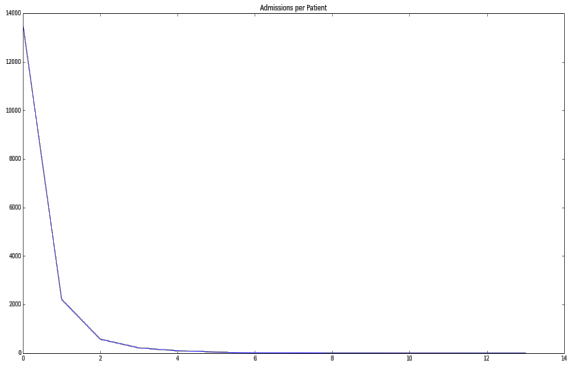

Figure 9: Distribution of numbers of hospitalizations per patient over the analyzed period June 2001 - October 2012.

Table 2: Distribution of selected patient characteristics in the analyzed subset of MIMIC III data. The distributions are calculated per admission.

Variable	Died in 30 days N = 1425		Not died in 30 days N = 20226		LR
Age (mean, SD)	79.33 years (7.26)		76.93 years (7.16)		
Length of Stay					
Hospital	13.73 days (11.33)		10.52 days (9.15)		
CCU (mean, SD)	121.22 days (115.56)	19.79%	72.45 days (86.18)	19.02%	1.05
CSRU (mean, SD)	262.05 days (322.26)	10.74%	92.67 days (132.29)	27.16%	0.32
MICU (mean, SD)	106.10 days (122.87)	57.89%	85.32 days (119.07)	36.14%	2.43
SICU (mean, SD)	143.88 days (222.66)	17.54%	111.51 days (170.28)	16.64%	1.07
Admission Location					
Emergency Room Admit		53.75%		39.22%	1.80
Clinic Referral/Premature		18.95%		19.93%	0.94
Phys Referral/Normal Deli		6.95%		21.73%	0.27
Transfer From Hosp/Extram		18.04%		18.39%	0.98
Transfer From Skilled Nur		1.75%		0.61%	2.89
Transfer From Other Healt		0.49%		0.10%	4.75
Info Not Available		0.07%		0.00%	14.20
Comorbidities					
Cardiac dysrhythmias		42.25%		36.73%	1.26
Acute and unspecified renal failure		37.05%		21.12%	2.20
Essential hypertension		39.16%		52.57%	0.58
Respiratory failure; insufficiency; arrest (adult)		33.40%		17.88%	2.30
Congestive heart failure; nonhypertensive		22.60%		16.28%	1.50
Pneumonia (except that caused by TB or STD)		25.40%		12.66%	2.35
Urinary tract infections		24.70%		16.20%	1.70
COPD		24.84%		17.75%	1.53
Diabetes mellitus without complication		25.47%		24.55%	1.05
Deficiency and other anemia		29.19%		22.87%	1.39
Fluid and electrolyte disorders		27.93%		20.52%	1.50
Disorders of lipid metabolism		26.95%		39.20%	0.57
Coronary atherosclerosis and other heart disease		18.67%		23.09%	0.76

As expected, only small fraction of patients died within 30 days after discharge. The distribution of mortality among patients who were treated in specific ICU units is shown in Figure 10.

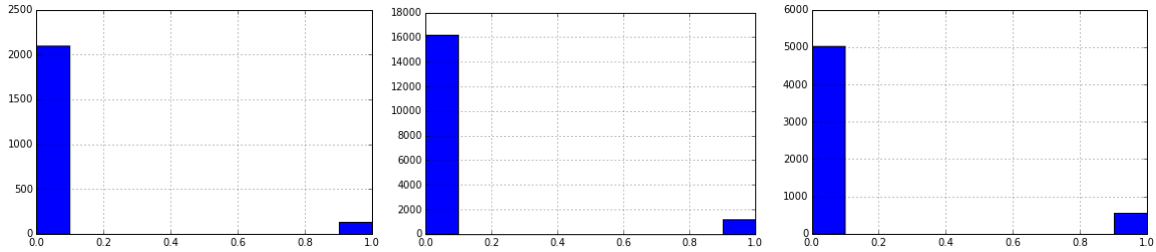

Figure 10: 30-day survival (left bar) vs. mortality (right bar) among patients in the data. The three charts correspond to MICU (left), all patients (center), SICU (right) patients.

The data has also been processed by resolving missing values, converted to numeric variables, and split between training (80%) and testing (20%). The final dataset has been processed and passed to machine learning module. All learning, attribute selection, and other operations have been done using only training data. Testing data has been used only to calculate final accuracy of the model.

This work reports accuracy of models in terms of Area Under receiver-operator Curve (AUC), often referred to in as c-statistic.

4.3. Results

4.3.1. Method Selection

The first set of results concern selection of the most appropriate method that can handle the data. A number of methods have been investigated as well as different settings of these methods. Table 3 shows comparison of accuracy of six methods applied to complete dataset. The results indicate that Support Vector Machine (SVM), Decision Tree and Random Forest, perfectly describe training data. However, it is clear that all three methods suffer from overfitting. Naïve Bayesian model seems inappropriate for the data, and Logistic regression provide reasonable, but low accuracy.

Despite overfitting, Random Forest significantly outperforms other methods when applied to testing data. Thus, in the further set of experiments Random Forest is used.

Table 3: Comparison of Methods applied to complete dataset

Method	**AUC (training)**	**AUC (testing)**
Logistic	0.73	0.663
SVM	1.0	0.5
Linear SVM	0.522	0.512
Bayesian	0.514	0.512
Decision Tree	1.0	0.543
Random Forest	1.0	0.743

4.3.2 Use of Administrative and Clinical Data

In the second set of experiments we tested if addition of clinical data (lab values) to administrative data (coded diagnoses) improves accuracy of prediction of 30-day mortality. Inclusion of lab values is consistent with existing models such as APACHE II.

The results indicate that addition of clinical data makes small difference in the accuracy. The AUC increases from 0.72 to 0.74. The ROC for combined administrative and clinical data is consistently above the one for administrative data only, as shown in Figure xxx. Interestingly, when applied to Medical Intensive Care Unit (MICU) and Surgical Intensive Care Unit (SICU) patients only, the accuracy worsens. While contradictory to the fact that these are two distinct types of patients and separate modeling should improve accuracy, this discrepancy can be explained by the amount of data available and thus overfitting of models.

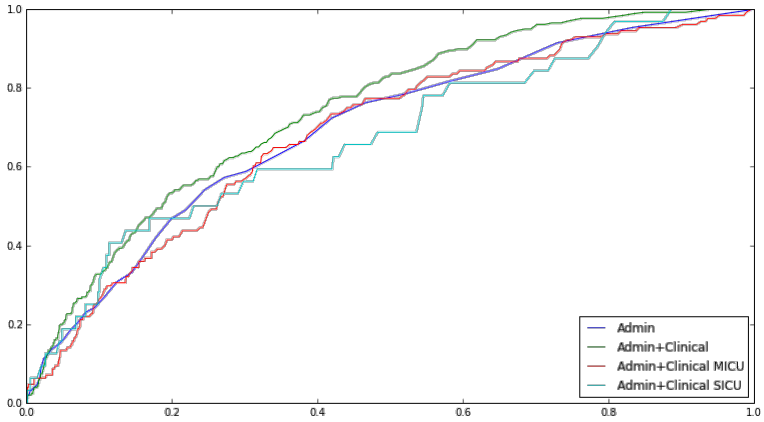

Figure 11: Receiver-operator curves for models trained using administrative data only, administrative and clinical data, and specialized models for MICU and SICU only.

4.3.3 Minimum C-LACE Model

Finally, we investigated possibility of reducing number of attributes needed to accurately predict 30-day mortality. Such a reduction is important for simplification of model and, as described in Section 5, allows for creation of online calculator in which data can be entered manually.

All 308 attributes used in the full model were ranked based on their weight in the Random Forest model. We created a set of models while increasing number of attributes until the accuracy became comparable to one in full model. This resulted in selection of top 20 attributes listed in Table 4 along with their weights.

Table 4: Top 20 most predictive attributes in the data ordered by importance calculated by Random Forest.

Feature	Importance
age	0.0452
HOSPITAL_LOS	0.0346
MICU_LOS	0.0320
CCU_LOS	0.0177
CCS 106	0.0176
CCS 157	0.0169
CCS 98	0.0159
ADMISSION_LOCATION	0.0157
CCS 131	0.0152
CCS 108	0.0145
CCS 122	0.0133
SICU_LOS	0.0130
CCS 159	0.0129
CCS 127	0.0127
CCS 49	0.0127
CSRU_LOS	0.0126
CCS 59	0.0123
CCS 55	0.0123
CCS 53	0.0110

The AUC of model based only on age was 0.516 which is basically a random guess based on prior class distribution. Similarly, the AUC of model based on Age and Length of Hospital Stay was 0.576. Interestingly models based on 5 and 10 top attributes performed very close to each other with AUC values of close to 0.7. Finally, the model based on 20

attributes performed only slightly worse than one based on all 308 attributes (AUCs about 0.74). Figure 12 below illustrates ROC for these models.

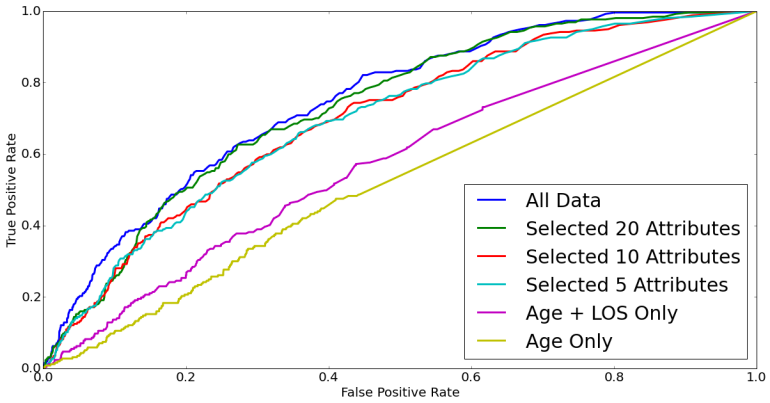

Figure 12: Accuracy of models for different selection of attributes given as ROC.

5. Online Calculator

In order for other researchers to test the developed mortality prediction models, we developed an online calculator which includes Minimum and Full C-LACE models. The minimum model is available through a web form that can be used by entering data, as well as Application Programing Interface (API) for automated use. The full model is available only through an API, since it is unlikely for anyone to answer 308 questions on a web form. At this stage, the online calculator is intended only for research purposes and not for clinical use, since additional validation is needed.

Simple online form (Figure 13) is used to enter patient and hospitalization characteristics. The entry is split into sections related to length of stay in hospital and specific ICUs, age, admission location and selected conditions most predictive of 30-day mortality. After submitting the form, user is provided with estimated probability of 30-day mortality. Because of the way the data was analyzed, the calculator is intended to be used at the time of hospital discharge.

Within the scope of this project it was impossible to completely test the calculator and in particular assess its impact on patient care. Thus, the site contains a disclaimer that the calculator is intended to be used only for research purposes.

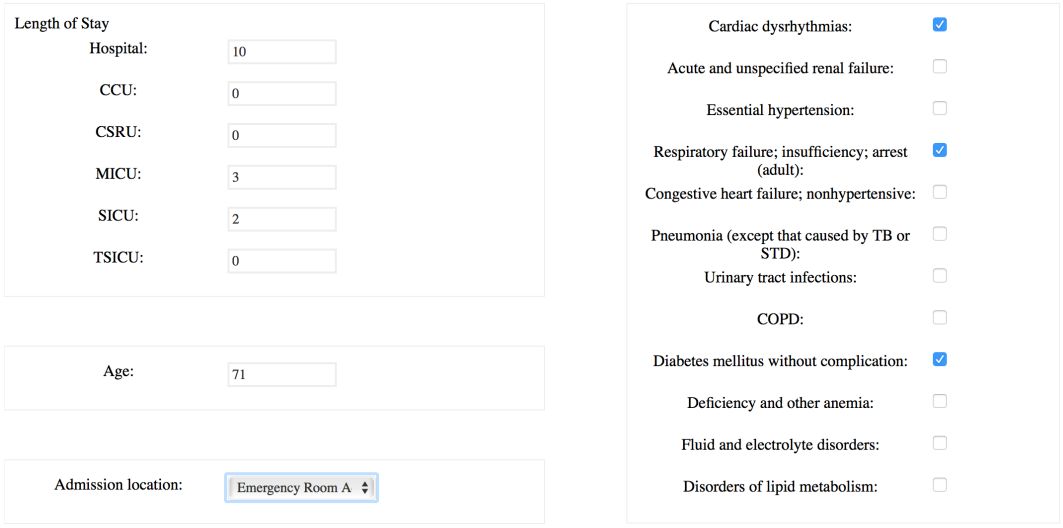

Figure 13: Design of the simple form used to enter patient and hospitalization information.

6. Conclusion

The presenter report described results of the project in which a new method for integrating data has been developed and implemented in a prototype computer software. The method relies on the idea that there is no need to create data warehouse or integrate or invest in large data infrastructure, but rather data can be pulled together ad-hoc as needed.

The obtained results indicate that it is possible to learn from distributed health data without prior construction of data warehouse or large scale integration. The team's experience within the project shows that analysis of data in the distributed system was easy and did not require more effort than within a centralized data warehouse. In fact, some aspects of the analysis were easier due to the use of semantic concepts and reduced need to rely on specific field names and coding used in various systems. Organizations that rely on analysis of data from multiple distributed sources are most likely to benefit from the use of the presented methods.

On the other hand the team recognizes that there may be situations in which more traditional approach to first integrate data and then analyze it may be more suitable. Large organizations that rely on heavy analysis of own data that can afford construction of a warehouse are most likely better off with the centralized approach.

6.1. Project Timeline and Accomplishments

The project was completed over period of one year from October 2015 to September 2016. Figure 14 shows main stages of the project as well as deliverables. The first stage of the project was to perform a detailed review of existing technologies for data integration and warehousing. It also included selection of application area along with dataset suitable

for testing the method. In the second stage of the project the data integration methodology was created. At the same time, data access and permissions were obtained, and the dataset was downloaded to GMU site. The third main stage consisted of software implementation and semantic mapping of data. Finally, the implemented method was applied to the obtained and mapped data and prediction results were obtained/analyzed.

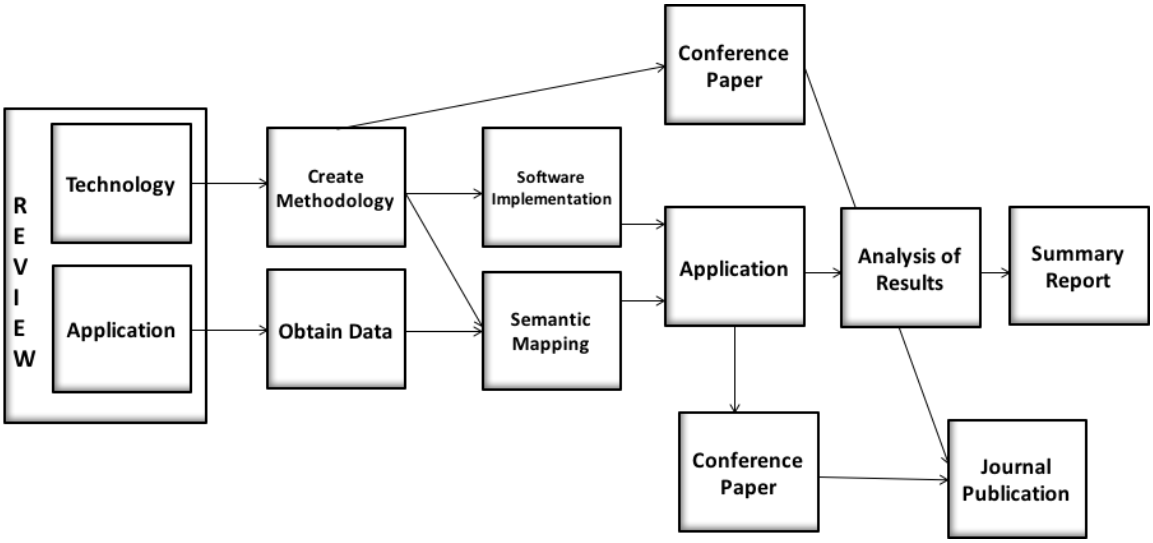

Figure 14: Project stages and deliverables.

6.2. Future Research

The current efforts of the team can be divided into three main directions:

- Methodological aspects of machine learning in distributed environments. Specifically, the plan is to work on methods that request data "on the fly" while the learning algorithm is being executed. The method starts with partial data and limited information and requests additional information (new attributes) for cases for which it is needed.
- Migration of the developed methodology into standard computing platform such as Hadoop. High performance computing platforms allow for analysis of very large datasets and may allow for scalability of the proposed solution beyond what is possible with simple prototype implementation presented in this report. The solution will rely on some combination of Hadoop's map-reduce paradigm, with API-based approach presented here.
- Application of the method to large scale data analysis. The work is specifically intended to target analysis of extremely large claims databases, such as those maintained by the Centers of Medicare and Medicaid Services (CMS), or clinical data from large organizations such as the Department of Veteran's Affairs.

References

Bache, R., et al. "An eligibility criteria query language for heterogeneous data warehouses." *Methods of information in medicine* 54.1 (2015): 41-44.

Ghaleb, M, and Farag A "Towards designing a federated database framework for disease outbreak notification systems." Open Source Software Computing (OSSCOM), 2015 International Conference on. IEEE, 2015.

Meeker, Daniella, et al. "A system to build distributed multivariate models and manage disparate data sharing policies: implementation in the scalable national network for effectiveness research." *Journal of the American Medical Informatics Association* 22.6 (2015): 1187-1195.

Moreno, RP, Metnitz, P.G., Almeida, E, et al., "SAPS 3--From evaluation of the patient to evaluation of the intensive care unit. Part 2: Development of a prognostic model for hospital mortality at ICU admission." Intensive Care Med. 2005; 31:1345.

Inouye, S.K., Peduzzi, P., Robison, J., Hughes, J., Horwitz, R., Concato, J., "Importance of Functional Measures in Predicting Mortality Among Older Hospitalized Patients." JAMA. 1998;279(15):1187-1193.

Rocker, G., Cook, D., Sjokvist, V., Weaver, B., Finfer, S., McDonald, E., Marshall, J., Kirby, A., Levy, M., at al., "Clinician Predictions of Intensive Care Unit Mortality." Crit Care Med. 2004;32 (5)

Iezzoni LI., "Risk Adjustment for Measuring Health Outcomes Ann Arbor." Mich: Health Administration Press 1994.

Davis, RB, Iezzoni LI, Phillips RS, Reiley P, et al., "Predicting in-hospital mortality: the importance of functional status information." Med Care.33:906-921.

Kuzniewicz MW1, Vasilevskis EE, Lane R, Dean ML, Trivedi NG, Rennie DJ, Clay T, Kotler PL, Dudley RA., "Variation in ICU risk-adjusted mortality: impact of methods of assessment and potential confounders." Chest. 2008;133(6):1319-27.

Vasilevskis EE, Kuzniewicz MW, Cason BA, Lane RK, Dean ML, Clay T, Rennie DJ, Vittinghoff E, Dudley RA., "Mortality probability model III and simplified acute physiology score II: assessing their value in predicting length of stay and comparison to APACHE IV." Chest 2009: 136(1):89-101.

Ho KM, Dobb GJ, Knuiman M, et al., "A comparison of admission and worst 24-hour Acute Physiology and Chronic Health Evaluation II scores in predicting hospital mortality: a retrospective cohort study." Crit Care 2006; 10:R4.

Van Walraven, C., Dhalla, IA, Bell, C., et al., "Derivation and validation of an index to predict early death or unplanned readmission after discharge from hospital to the community." CMAJ 2010; 182(6): 551–557.

van Walraven C, Wong J, Forster AJ., “LACE+ index: extension of a validated index to predict early death or urgent readmission after hospital discharge using administrative data.” Open Med. 2012: 19;6(3):e80-90.

Au, AG, McAlister, FA, Bakal, JA, et al., “Predicting the Risk of Unplanned Readmission or Death Within 30 Days of Discharge After a Heart Failure Hospitalization.” American Heart Journal. 2012;164(3):365-372.

Levy, C.,, Kheirbek, R,, Alemi, F., Wojtusiak, J., Sutton, B,, Williams, A.R. and Williams, A., "Predictors of six-month mortality among nursing home residents: diagnoses may be more predictive than functional disability." Journal of Palliative Medicine, 18(2), 100-6, 2015.

Ngufor, C., Wojtusiak, J., Hooker, A., Oz, T. and Hadley, J., "Extreme Logistic Regression: A Large Scale Learning Algorithm with Application to Prostate Cancer Mortality Prediction," Proceedings of the The 27th International Florida Artificial Intelligence Research Society Conference, 2014.

Rose, Sherri. "Mortality risk score prediction in an elderly population using machine learning." American journal of epidemiology 177.5 (2013): 443-452.

Cooper, Gregory F., et al. "An evaluation of machine-learning methods for predicting pneumonia mortality." Artificial intelligence in medicine 9.2 (1997): 107-138.

Taylor, R. Andrew, et al. "Prediction of In-hospital Mortality in Emergency Department Patients with Sepsis: A Local Big Data–Driven, Machine Learning Approach." Academic Emergency Medicine (2016).

Johnson AEW, Pollard TJ, Shen L, Lehman L, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, and Mark RG. “MIMIC-III, a freely accessible critical care database”. Scientific Data (2016). 10.1038/sdata.2016.35.

Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. “PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals”. Circulation 101(23):e215-e220

A publication of the *Machine Learning and Inference Laboratory*
College of Health and Human Services
George Mason University
Fairfax, VA 22030-4444 U.S.A.
http://www.mli.gmu.edu

Editor: J. Wojtusiak

The *Machine Learning and Inference (MLI) Laboratory Reports* are an official publication of the Machine Learning and Inference Laboratory, which has been published continuously since 1971 by R.S. Michalski's research group (until 1987, while the group was at the University of Illinois, they were called ISG (Intelligent Systems Group) Reports, or were part of the Department of Computer Science Reports).