# Dynamic Hierarchical Classification for Patient Risk-of-Readmission

Senjuti Basu Roy†, Ankur Teredesai†, Kiyana Zolfaghar†,
Rui Liu†, David Hazel†⋆, Stacey Newman†, Albert Marinez ◇.
† Center for Data Science, UW Tacoma, ◇ MultiCare Health System, ⋆ Kensci Inc.,
{senjutib,ankurt, kiyana, ruiliu, dhazel, newmsc8}@uw.edu,
albert.marinez@multicare.org, david@kensci.com

## ABSTRACT

Congestive Heart Failure (CHF) is a serious chronic condition often leading to 50% mortality within 5 years. Improper treatment and post-discharge care of CHF patients leads to repeat frequent hospitalizations (i.e., readmissions). Accurately predicting patient's *risk-of-readmission* enables care-providers to plan resources, perform factor analysis, and improve patient quality of life. In this paper, we describe a supervised learning framework, *Dynamic Hierarchical Classification (DHC)* for patient's risk-of-readmission prediction. Learning the hierarchy of classifiers is often the most challenging component of such classification schemes. The novelty of our approach is to *algorithmically* generate various layers and combine them to predict overall 30-day risk-of-readmission. While the components of DHC are generic, in this work, we focus on congestive heart failure (CHF), a pressing chronic condition. Since healthcare data is diverse and rich and each source and feature-subset provides different insights into a complex problem, our DHC based prediction approach intelligently leverages each source and feature-subset to optimize different objectives (such as, Recall or AUC) for CHF risk-of-readmission. DHC's algorithmic layering capability is trained and tested over two real world datasets and is currently integrated into the clinical decision support tools at MultiCare Health System (MHS), a major provider of healthcare services in the northwestern US. It is integrated into a QlikView App (with EMR integration planned for Q2) and currently scores patients everyday, helping to mitigate readmissions and improve quality of care, leading to healthier outcomes and cost savings.

## 1. INTRODUCTION

Hospital readmissions have come to the forefront of healthcare research and discussions in recent years for their recognized universal negative impacts on healthcare systems' budgets and patient loads throughout the world. Within the United States, Centers for Medicare & Medicaid Services (CMS) recently began using readmission rates as a publicly reported quality metric to measure hospital care standards and reimbursements in the fee-for-service model[1]. The severity of the readmissions problem can even be measured in economic terms: the estimated cost of unplanned readmissions is roughly $17.9 billion per year [8]. A significant percentage of these costs are attributable to patients who are often the sickest and most vulnerable: old, critically ill, and suffering from multiple chronic disease conditions. Paradoxically, more than 27% of such readmissions are avoidable [16]. Hence, predicting risk-of-readmission can guide implementation of appropriate interventions to prevent such avoidable readmissions. To that end, this paper investigates the issues faced and challenges addressed in developing and deploying a *framework to predict the risk-of-readmission (RoR)*.

While the deployed framework is extensible by design and includes other chronic conditions, in this paper the chronic condition we focus on is Congestive Heart Failure (CHF) since CHF is one of the leading causes of hospitalization. Studies also indicate that a large percentage of CHF admissions are actually readmissions within a short window of time. The 2005 data for Medicare beneficiaries estimated that 12.5% of Medicare patients admitted due to CHF were followed by readmissions within 15 days, accounting for about $590 million in healthcare costs [12]. In practice, a window of 30-days after discharge for CHF is considered *clinically meaningful* for hospitals and medical communities to take action to reduce readmissions [11] and forms the basis of most readmission risk prediction models.

Current research treats the problem as a binary classification task, where the objective is to identify patients with CHF who are likely to be readmitted within 30 days of discharge as an output of a single binary classifier [21, 19, 18]. A patient readmitted within 30 days = 1 and a patient not readmitted within 30 days = 0. There are several shortcomings to this approach which we address with the DHC framework: the distribution of risk of readmission is highly skewed with a few patients readmitting repeatedly and many patients readmitting once or twice over a period of a year. Traditional classifiers suffer from majority bias and tend to assign patients to the no 30-day-readmission majority class. Moreover, patient characteristics change over time. New diseases and conditions set in; age and vitals continuously change. Thus, including all patients discharged with CHF in the training set to build the classification model introduces noise since patients that were readmitted after a long gap can have characteristics that are very different from pa-


*KDD'15,* August 10-13, 2015, Sydney, NSW, Australia.

DOI: http://dx.doi.org/10.1145/2783258.2788585.

[1]http://www.cms.gov

tients that were readmitted within a short time (30 days) for the same condition. Hence the team of data scientists and the underlying framework has to handle numerous missing values, discretize attributes, collate comorbidities, extract suitable features and employ the right learning algorithm given the constraints.

This research presents a *Dynamic Hierarchical Classification (DHC)* framework to predict RoR for CHF patients. The primary novelty of the solution is to undertake the prediction problem in *several stages or layers*, leading to the formation of a hierarchy of classification models as the overall solution. Each layer aims at predicting the RoR within *certain days* (cut-points). For example, it may be better to predict whether a CHF patient is likely to (ever) readmit or not before calculating her readmission risk within 30 days of discharge. Each such logical consideration constitutes *a classification layer*. Within a given layer, the problem is of a binary classification design layer by layer[2]. Consider a 70 year old female patient suffering from primary diagnosis of congestive heart failure and various associated conditions such as diabetes and renal failure. If at layer one, DHC predicts her to be likely to be readmitted (ever), then the second layer may predict likely to be readmitted within-60-days of discharge (or not). Subsequently, only when she qualifies to be readmitted with a high confidence at both these layers, would the third layer predict her likelihood to be readmitted within 30 days (or not). This may sound simple conceptually, but the challenge of *how many layers are sufficient and how to design the intermediate layers* remain an open problem. Furthermore, the effectiveness of different features may be different in various layers. Then, the question is, how to select the features? We address this by performing layer specific feature selection. The last challenge is how to perform effective binary classification in each layer. We propose threshold tuning of the classification models for that.

In spite of algorithmic automation as proposed, it is not easy to design these layers and always get it right. Clinical teams may provide initial guidance but there is no clear clinical evidence for how many layers (for decisioning) are appropriate for a given patient population or how to define the time-window for readmission cut-points. Naturally, many of these design decisions are also dataset specific, requiring us to discover these layers for a given patient cohort. In this work, we propose an innovative approach: we design multiple algorithms to discover the actual cut-points assuming that the number of layers is specified as an input. In one of our algorithms, we primarily analyze the characteristics of the underlying patient population and propose greedy algorithms to design the cut-points such that the two consecutive layers generated by our algorithm exhibit the highest difference in the characteristics of the patient population. We also non-trivially adapt one of the frequency based popular discretization algorithms, Chi-Merge [7] (that is primarily used to discretize continuous attributes based on class distribution), to generate these cut-points. After that, to maximize a particular metric (such as AUC, precision, recall, accuracy, etc.), we study how to make use of the training data and the trained classification model in a given layer. We propose novel solutions to the above mentioned problems in this work.

[2]This could be formalized by finding the maximum duration $X$ between two admissions in the dataset; i.e.the task is to predict whether the patient would be readmitted with X days or not.

We present comprehensive experimental results using two real world datasets - we consider 4 years of the State Inpatient Dataset (SID) of Washington State as our large-scale dataset and then we use the real patient dataset provided by MultiCare Health System, a major health system in the northwestern US as the small scale dataset. We empirically evaluate the effectiveness of our proposed solutions with statistical significance analysis using a variety of quality metrics and perform comparative analyses with our hierarchical classifications models against several baseline algorithms. Our results demonstrate that the proposed framework is superior to the baseline algorithms for all quality metrics with statistical significance.

To summarize, we make the following contributions:

1. We initiate the study of designing hierarchical classification models to predict 30-day RoR for CHF patients (Section 2).
2. We propose a *deployed solution for Dynamic Hierarchical Classification (DHC)* framework, empowered with multiple novel solutions to effectively design the prediction hierarchies (Section 3 and and 4).
3. We perform comprehensive experiments using multiple real world patient data sets, demonstrating that the proposed hierarchical model enhances the quality of prediction (Section 5).

## 2. PRELIMINARIES

It is typical in clinical settings for data to be spread across various relational schemas and flat files; a series of (anonymized) patient record data for various windows of time and modalities of measurements are typically extracted from EMRs and secondary hospital sources to form a data warehouse with appropriate dimensions and measures. Such a schema contains granularities of patient measures related to admission and discharge, other diseases, demographic factors, comorbidities, and post-discharge and follow-up-plans, which are all integrated and preprocessed. Each encounter is uniquely identified by an admission ID. Multiple admissions (i.e., readmissions) of the same patient are identified by the same patient ID and different encounter ID. For a given patient and a given hospital admission, we can calculate the number of days between the current admission and her previous discharge and check if it is less than 30.

*Problem definition 1.* Given a patient record, the problem of predicting the RoR within $\gamma$ days of discharge is a hierarchical classification problem. The intuition here is to first predict the readmission window for higher $\gamma$ values before predicting readmission for $\gamma = 30$ days. Each of these *time intervals* are cut-points, demarcating a classification layer in dynamic hierarchical classification (DHC). Since the 30 day interval is clinically meaningful and tied to reimbursements, that interval is the last layer of risk scoring and outcomes. Each layer can constitute a specific readmission window (for example the very first layer predicts whether a patient would be readmitted at all or not for $\gamma$ >365 days in a year). Each layer design is to formulate a binary classification model using historical patient data relevant to that interval to predict the RoR within that readmission interval. How many such layers are to be designed, and how to decide the readmission intervals algorithmically, is the key contribution of this work and described in depth in Section 4.

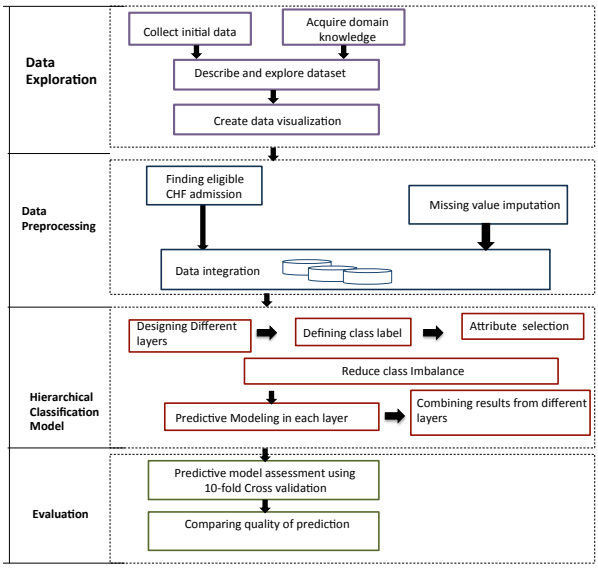


Figure 1: Overall architecture for RoR prediction process includes various stages as seen above including: exploration, preprocessing, modeling, and evaluation.

Data exploration, data pre-processing, hierarchical classification model, and model evaluation are critical for a successful data mining framework, especially at healthcare domain [21, 19, 18]. Figure 1 provides an overview of these major steps. For the DHC framework design we worked closely with cardiologists, nurse practictioners, and other members of the care team to identify critical factors influencing early recurrent readmission. The robustness of these factors and their availability at various points in the care process was an important DHC design consideration.

# 3. DEPLOYING THE DHC FRAMEWORK

Prior research efforts have focused on the accuracy of modeling to predict the likelihood of patients with chronic conditions to readmit within 30 days[15]. Commercial efforts have been limited and mostly restricted to web based mortality tools or risk calculators where data is manually entered to compute the risk. To the best of our knowledge, ours is the first cloud based scalable EMR integrated effort reported in literature that uses data mining fundamentals to continuously monitor risk of patients and issue risk scores to care providers. We have enabled the platform to deliver the per-patient readmission risk score along with actionable insights, not solely as a business intelligence tool sourced from a data warehouse (see 3.1), but on real-time clinical data, delivered within the clinical workflow, at the point of care.

The DHC framework consists of three main components: a) External Layer, b) Communications Layer, c) Analytics Layer. Within these layers there are multiple subcomponents and processes which are tightly integrated in order to accurately and efficiently score patients for 30 day risk of readmission and deliver these risk profiles to the patients electronic medical record (EMR).

## 3.1 QlikView Application

The initial deployment scenario envisioned was a single layer binary classifier exposed through a QlikView Readmission App[3]. This would provide system-wide readmission visualization and reporting with facility level, cohort level, and patient level drill down capability. Additionally, the QlikView Readmission App was used to identify and flag patients (within targeted risk categories) for further investigation and focus. It enables exploration of a detailed patient risk profile (30 day risk of readmission score targeting clinical attributes, psychosocial factors, a composite score, and top contributing factors). With the development of the DHC framework, the QlikView risk scores are now computed by the hierarchical classification with algorithmic multi-layer design. A spapshot of the application is presented in Figure 2.

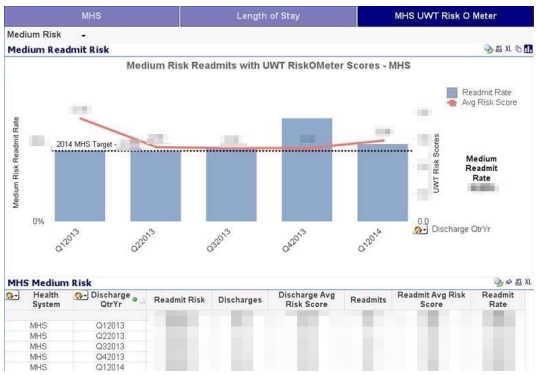


Figure 2: Snapshot of the QlikView Application

Every night it scores discharged patients and creates a ranked list of readmission risk for further review.

## 3.2 EMR Integration

Business intelligence tools 3.1 are useful for population level reporting and risk management but are not oriented towards the clinical workflow and have limited adoption within that context. To fully leverage our developed framework for CHF readmissions, cardiovascular service line leadership recognized the need to deploy DHC directly into their EMR (Epic).

After extensive user research, the team identified two target Epic integration points: 1) the Heart Failure Dashboard, used by cardiology to see the overview of the patients and help drive interventions during the encounter, 2) Care Management Doc Flowsheet, used to support discharge planning (which begins when the patient is admitted) and bring focus to psychosocial factors and followup care.

The EMR integration roadmap consists of: 1) publishing of a single combined score (clinical + psychosocial) as a "result" [4] in Epic, 2) subcomponents of that result to include discrete clinical and psychosocial scores.

## 3.3 Patient Data Flows

DHC is available as an on premise deployment or as a series of services hosted on Microsoft Azure. The overall architecture of the system and flow of messages are illustrated in Figure 3.

[3]http://www.qlik.com/us/explore/products/qlikview

[4]In the same way that lab results can be tracked and recorded over time, the insertion of the score as a result, with primary value and multiple sub-components, enables comprehensive trend analysis, reporting capability and intervention exploration.

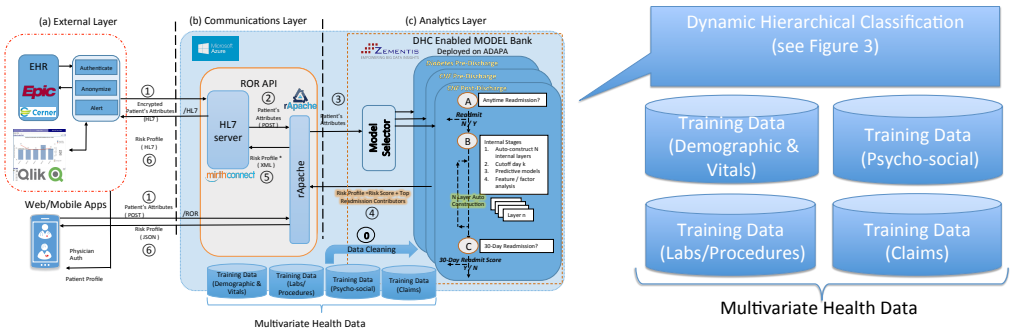


Figure 3: Illustration of the DHC Framework: (a) External Layer, (b) Communications Layer, (c) Analytics Layer. The focus of this paper is the Dynamic Hierarchical Classification in (c) Analytics Layer.

We now return to our example of a 70 year old female patient. She has been admitted to the inpatient setting and within 24 hours her lab results become available. As her medical record within Epic is updated, an HL7 message [5] containing the relevant attributes is generated. This message is then sent from the External Layer (a) to the Communications Layer (b). For cloud deployments, where the data does not stay fully resident within the health system, the tuple of patient data is passed through an anonymization service, which sends only the attributes (+GUID) required for scoring to the API [15].

The HL7 message is then ingested by Mirth Connect service [6] which handles message queuing and translation from HL7 into JSON, used internally by the DHC Enabled Model Bank. For native JSON requests (such as from a mobile device), we would bypass Mirth Connect [15].

After ensuring proper formatting and consumption of the message, it is passed to the Analytics Layer for scoring. We use the Zementis ADAPA Scoring Engine [7] and the DHC Enabled Model Bank. The use of ADAPA brings multiple benefits, among these the ability to scale up to handle a very large volume of messages, as well as the ability to import multiple models (beyond those we have developed for CHF), allowing us to score patients against a number of chronic disease conditions. Additionally, the DHC supports auto selection of cut-points, where the model is trained against MultiVariate Health Data sources outlined in Figure 3 (the significance of this is outlined in Section 4). When a patient tuple is passed to the APIs provided by the ADAPA scoring engine, it is scored against the DHC Model which has been trained for a particular cohort, such as CHF Post-Discharge.

A risk profile, consisting of a risk score as described next in section 4.3 and top correlated factors (identified using Chi-Square test) that contribute to readmission risk, is then generated and returned back to the Communications Layer. In the case of Epic deployment, the JSON response is converted back into HL7 for consumption. This is then returned to the anonymization service, where the risk profile is recoupled with the patient identifier, attached to the patient visit, and published as a result within Epic.

[5]HL7: Messaging format(standards, guidelines and methodologies) by which various health care systems communicate.
[6]http://www.mirthcorp.com/products/mirth-connect
[7]http://zementis.com/products/adapa/

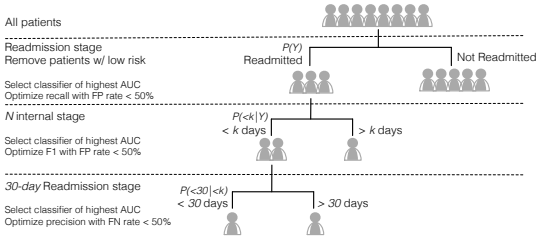


Figure 4: Tree-based hierarchical class structure for predicting 30-day readmission using 1 intermediate layer.

## 4. DYNAMIC HIERARCHICAL CLASSIFICATION (DHC)

We propose *Dynamic Hierarchical Classification (DHC)* to predict the risk of readmission within 30 days, transforming the problem into a tree-structured class hierarchy. Our approach produces a hierarchical set of classifiers in top-down fashion. The hierarchical setting provides the opportunity to make more specific and accurate predictions and improve knowledge about the problem using factor analysis (Chi-squared test). The framework also allows flexible choices of classifiers at different layers. As the distribution of data changes at each layer, DHC allows us to use the best classifier to optimize different learning objectives for each layer.

For our problem (i.e., predicting 30-day RoR), the hierarchical classification task could be restated as follows: first predict if a patient would ever be readmitted or not. For that task, a binary prediction model is designed using appropriate training data. Then, in the *intermediate layers, a set of classifiers* needs to be designed with each constituting a particular readmission window. Each of these is a subtask towards predicting RoR with a specific time window (cut-point). We design binary classification models for each of these intermediate layers. For simplicity, we assume that the number of intermediate layers is predefined and given [8]. The final layer of the classification is to predict whether a patient would be readmitted within 30 days or not. Similar to all previous layers, this sub-task is also treated as a binary classification problem and the appropriate training dataset is used for that purpose.

[8]However, in our experimental analyses, we empirically decide the number of intermediate layers.

For the purposes of illustration, Figure 4 describes the overall process of predicting 30-day RoR using 1 intermediate layer. The framework involves three stages: *Readmission Stage*, *Internal Stage*, and *30-day Readmission Stage*. Even though the number of layers in the internal stage are given, we still have to identify *the appropriate readmission windows, i.e., cut points* for those layers. Our proposed solutions to design cut-points are described in Section 4.2.

**Algorithm 1:** DHC Algorithm

**Require:** Training dataset $D$, number of internal layers $N$, cutoff day $k > 30$, $Max$ is the maximum readmission time based on the dataset $D$

```
[Stage 1] Design Max-day readmission prediction model using D
m = 31 , n = Max
for i = 1 to N − 1 do
    Extract D′ ⊂ D, where D′ contains only patients who are readmitted within (m, n)-days in D
    k ← CutpointFinder(D′, m, n)
    [Stage 2] Design (m, k)-day readmission prediction model using D′
    D ← D′
    m = k + 1
Design (m, Max)-day readmission prediction model using D′, where D′ contains only patients who were readmitted within (m, Max) days in D
Extract D′′ ⊂ D, where D′′ contains only patients who were readmitted within (0, 30) days in D
[Stage 3] Design (0, 30)-day readmission prediction model using D′
```

Algorithm 1 formalizes the DHC framework. Similar to the multistage screening in medical practice, we impose a maximum error bound while maximizing the recall and precision at different stages. The error bound (i.e. false positive rate) is determined by domain experts. Table 1 provides a tabular view of *Readmission Stage*, *Internal Stage*, and *30-day Readmission Stage*.

## 4.1 Readmission Stage

The first *Readmission* stage predicts whether a patient will be readmitted at all after being discharged from the hospital. The learning objective is to maximize AUC and recall [7] in order to capture all patients with readmission risk. The maximum false positive rate is set to be 50%. Since this is the root (very top level) of the hierarchical classification model, the goal is to filter out patients who are unlikely to be readmitted.

## 4.2 Internal Stage

The objective is to maximize AUC and recall while controlling the false positive rate to be lower than 50%. We build $N$ internal layers and, for each layer, derive a cutoff day $k$. Even when $N$ is a given integer number, we still have to determine $N$ different readmission windows (cut-points). Intuitively, what we intend to do is to choose those cut-points, such that, if one can *create $N$ non-overlapping partitions of the entire patient population based on those cut-points, patients that fall across the stratum are as divergent as possible*. We propose a heuristic algorithm towards that end which iteratively creates $N$ different cut-points.

Algorithm 2 describes the *hill climbing* search heuristic to identify a single cutoff day $k$ that constitutes a layer. Without loss of generality, to find a cut-off day $k$ between a time window $(m, n)$ (note that if $N = 1$, then $m = 31$, $n = Max$, where, for a given dataset $D$, the highest readmission day is denoted to be $Max$), we call a pseudo-random number generator between $(m, n)$ to get a $k$, such that $m < k < n$. We select two candidate cut-points: $k' = k + 5$ and $k'' = k - 5$. After that, we compute two divergences: one based on $k'$ and another based on $k''$. $k$ then gets updated based on whichever cut-point gives rise to higher divergence. We continue the search until no further improvement is possible. Once a single cut-off point $k$ is found, we run Algorithm 2 twice to find two other cut-off points, one between 31 and $k$ and the other between $k + 1$ and $Max$. This process is repeated until all $N$ internal layers are identified. Next, we describe two divergence calculation methods used by Algorithm 2.

### 4.2.1 Entropy-based Divergence Methods

Recall Algorithm 2 and note that to select a single cut-point between $(m, n)$ window, the hill climbing algorithm needs to compute the divergence of patient population that constitutes two readmisison windows between $(m, k)$ and $(k + 1, n)$ based on any $k$, $m < k < n$. In order to use Entropy-based divergence methods, one has to represent each of these patient populations as a probability distribution. To do that, we obtain the *center of the individual patient group*, where the center is the *average* for the numeric attributes and *mode* for categorical attribute, as described in Algorithm 3 (DIVCAL).

In our implementation, we consider two popular divergence methods: Kullback-Leibler Divergence (KL) and Jensen-Shannon Divergence (JSD) [3]. The main difference between the two methods is that unlike KL, JSD is symmetric and always defined. The divergence between two centers $p$ and $q$ of $m$ dimensions, for Kullback-Leibler(KL) and Jensen-Shannon(JSD) methods are calculated as follows [3]:

$$KL(p, q) = \Sigma_{\forall i=1}^{m} ln(\tfrac{p(i)}{q(i)}) \times p(i)$$

$$JSD(p, q) = \tfrac{1}{2}\left(\Sigma_{\forall i=1}^{m} ln(\tfrac{2p(i)}{p(i)+q(i)}) \times p(i) + \Sigma_{\forall i=1}^{m} ln(\tfrac{2q(i)}{p(i)+q(i)}) \times q(i)\right)$$

**Algorithm 2:** Subroutine CutpointFinder$(D, m, n)$

**Require:** Dataset $D$, lower boundary of readmission window $m$, upper boundary of readmission window $n$

```
k ← RANDOM_NUMBER(m, n)
repeat
    k′ ← k + 5
    k′′ ← k − 5
    currentDIV = DIVCAL(k)
    for all i ∈ {k′, k′′} do
        if DIVCAL(i) > currentDIV then
            k = i
            currentDIV = DIVCAL(i)
until DIVCAL(k) > DIVCAL(k′) and DIVCAL(k) > DIVCAL(k′′)
return k
```

### 4.2.2 Chi-Merge Based Solutions

In addition to Entropy-based methods, we intelligently adapt the popular frequency based Chi-Merge algorithm [10] to design the internal layers.

| | Readmission Stage | Internal Stage | 30-day Stage |
|---|---|---|---|
| *Objective* | AUC & Recall | AUC | AUC & Precision |
| *Error Bound* | FP Rate $< 50\%$ | FP Rate $< 50\%$ | NA |
| *# of Layers* | 1 | N | 1 |
| *Probability* | $P(Y)$ | $P(< k\|Y)$ | $P(< 30\| < k)$ |

Table 1: Comparison of three stages

**Algorithm 3:** Subroutine DIVCAL(k)

**Require:** Dataset $D$, cutoff day $k$, center of data $V$
1: $D' \leftarrow$ subset of $D$ where readmission day between $m, k$
2: $D'' \leftarrow$ subset of $D$ where readmission day between $k+1, n$
3: $p \leftarrow$ center of $D'$
4: $q' \leftarrow$ center of $D''$
5: $d \leftarrow KL(p,q) Or JSD(p,q)$
6: **return** $d$

Chi-Merge is traditionally used to discretize continuous attributes based on the class distribution frequency. On the other hand, our objective is to create a set of $N$ intermediate layers (and cut-points thereof) for the variable readmission, where this variable itself is also the class label. This precludes direct adaptation of Chi-Merge in our settings. In fact, to be able to apply Chi-Merge, we need to extend our settings in the following way that we illustrate using a simple example next. Imagine that we have a population of 100 patients in total. Out of these 100 patients, 20 got readmitted within 30 days and 10 never got readmitted. The remaining 70 patients create the positive instances who got readmitted within 30 and $Max$ days. Similarly, Chi-Merge also needs negative instances. If a set of $X'$ patients are readmitted on a day $d$, then we set the negative instances using the remaining $70 - X'$ patients. After this modification, we apply Chi-Merge to decide the $N$ intermediate layers.

## 4.3 30-day Readmission Stage

The final stage predicts the readmission $\leq 30$ days. The sample size has been reduced as we move down toward this final leaf node. We use local information from this subset to train classifiers and assess the likelihood of a given patient being readmitted to the hospital within 30 days – i.e. $P(Readmit \leq 30 | Readmit \leq k)$. The objective of this stage is to maximize the AUC and precision so that the medical resources can be efficiently invested on the right patients.

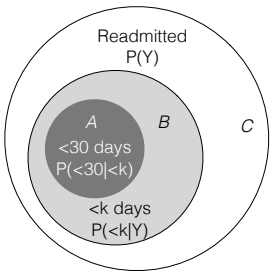


Figure 5: Relationships of predicted probability of three stages considering 1 intermediate layer

Figure 5 shows the relationship of predicted probability of three stages, where circle $A$ is the final stage, circle $B$ is the internal stage (for simplicity we consider 1 internal stage in this example), and circle $C$ is the readmission stage. Intuitively, the predicted probability of 30-day readmission should be the highest in $A$, lower in $B$, and the lowest in $C$. Given the definition of the problem for predicting 30-day readmission, the goal is to estimate $P(\leq 30)$, as below.

$$P(\leq 30 | \leq k) = \frac{P(\leq 30 \wedge \leq k)}{P(\leq k)}$$

Since $\leq 30$ is a subset of $\leq k$, we obtain:

$$P(\leq 30 \wedge \leq k) = P(\leq 30)$$

Therefore,

$$P(\leq 30) = P(\leq 30 | \leq k) P(\leq k)$$

Similarity, we can obtain $P(k)$ as follows:

$$P(\leq k|Y) = \frac{P(\leq k \wedge Y)}{P(Y)}$$

Since $\leq k$ is a subset of $Readmission = Y$, we obtain:

$$P(\leq k \wedge Y) = P(\leq k)$$

Thus,

$$P(\leq k) = P(Y) P(\leq k|Y)$$

Therefore,

$$P(\leq 30) = P(\leq 30 | \leq k) P(\leq k) = P(\leq 30 | \leq k) P(\leq k|Y) P(Y)$$

Given the equations above, we can compute $P(\leq 30)$ for all instances in $A$ in Figure 5. We then combine the results using normalization techniques to ensure the correct sequence of instances: $A \prec B \prec C$, if the list is sorted in descending order by the predicted probability. In Figure 5, we identify three mutually exclusive areas – the dark grey area in $A$, the light grey area in $B$, and the white area in $C$. For all instances in $A$ (the dark grey area), we normalize the predicted probability in to the range (0.7,1]. For all instances in the light grey area in $B$, we normalize the predicted probability to the range (0.3,0.7]. Finally, we fit the predicted probability from the white area in $C$ into [0,0.3]. We compute our evaluation metrics (e.g. AUC, recall etc.) using the combined normalized probability vector.

# 5. EXPERIMENTS AND EVALUATION

We evaluate performance of proposed DHC framework on Washington State Inpatient Dataset and the Heart Failure cohort data from MultiCare Health Systems (MHS).

## 5.1 SID Dataset

The State Inpatient Databases (SID) [9] of Washington State (referred as SID-WA) is available for four years 2009-2012. It is discharge abstract data that includes inpatient

[9] http://www.hcup-us.ahrq.gov/sidoverview.jsp

discharge records from community hospitals in the State of Washington with all-payer, encounter-level information.

Each year comprises four files that are associated with hospital encounters: core file (CORE), charges file (CHGS), diagnosis and procedure groups file (DXPRGRPS), and disease severity measures file (SEVERITY). Total 596 attributes for each encounter with an unique identifier KEY, that can be used to link records across files (but NOT across years). We construct a heart failure cohort with patients whose primary or secondary ICD9-CM diagnosis codes are heart failure. Our cardiology partners identified a total of 91 attributes to be related to CHF readmission from a mix of demographic variables such age, gender, and race; as well as comorbidities related to diagnosis information derived from primary and secondary diagnosis codes for that patient. Many features related to utilization services such as thew emergency room, ICU, electrocardiograms services and so on are also included in our prediction models.

The SID-WA attribute DaysToEvent is used to compute the days between two consecutive hospital admissions for each patient. The days since previous hospital discharge is computed using the days between two hospital admissions minus the length of stay of the first admission. We exclude admission records in which the patient passed away while in the hospital, and in which the patient age is less than 1 year (age=0). The dataset then contains a total of $2,051,105$ readmission instances.

## 5.2 Dataset from MultiCare Health System

The Cardiovascular datamart is our primary source within the MultiCare Health System data. It has about 8,600 patients diagnosed with Heart Failure, serviced over $14,200$ hospital encounters from 2009 to 2013. We consider only patients who are discharged to home, excluding inter-hospital transfers. Encounters where the patient expired are not included. The post cleaning and processing set we used for these experiments includes 6,348 patients diagnosed with CHF and 14,170 hospital encounters. A total of 49 attributes were determined to be related to CHF admission. These are mainly clinical. The key socio-demographic factors related to readmissions are: gender, race, and marital status. Other important factors pertinent to CHF are ejection fraction(EF) [10], blood pressure, primary and secondary diagnosis indicating comorbidities, and APR-DRG codes [11] for severity of illness and risk of mortality. Information about discharges, such as discharge status, discharge destination, length of stay and follow-up plans, are also found to be correlated to CHF readmissions. In addition, we include 34 cardiovascular and comorbidity attributes.

## 5.3 DHC Evaluation

Our modeling for SID-WA data and corresponding analysis is conducted on Microsoft Azure over virtual machines with 16 cores and 112GB of RAM. For the modeling environment we use R and Python both. Models are then exported in PMML and deployed using the Zementis ADAPA scoring engine. We evaluate quality using measures such as Area Under the Curve(AUC), Precision, Recall, and Accuracy [7].

Depending on the final consumption of RoR predicted score for a patient population, different evaluation measures are appropriate. AUC measure is typically interesting when the dataset is imbalanced and we want to ascertain comparison of performance with other competing tools for risk stratification. Precision is critical if there is a high cost related to incorrectly predicting patients' risk to belong to the *Readmission* class with accompanying reimbursement penalties associated with the decision. Recall is relevant if using the DHC framework to identify patients who belong to *Readmission* class from within an overall population for post-discharge care management. Besides these, we also compute the confusion matrix for all the experiments (each fold) consisting of true positive count (TP), false positive count (FP), true negative count (TN), and false negative count (FN).

In general, we select the classifier that renders the highest AUC for each layer for reporting in this paper. In actual deployment we may vary this choice depending on the endpoint where the score is being consumed. We also tune the threshold to optimize the selected evaluation metric for each layer in DHC.

**Predictive Models:** We use five different predictive models in general. The first one is Logistic Regression (LR) [7], a method that models the outcomes (class labels) as a so-called logit function of the predictive variables. The second one is Naive Bayes (NB) [7], a well-known simple statistical classifier that assumes attribute independence. We also use Random Forests and Adaboost [5], as popular ensemble algorithms (we use 500 decision trees in random forests and 50 CART decision trees in Adaboost). The last implemented model is the Support Vector Machines (SVM) [7] using well-established RBF kernel. For each layer of DHC described in Section 4, we run a 10 fold cross-validation procedure on train set to determine the best classifier.

### 5.3.1 Comparative study on SID dataset

We follow the holdout method to test the models built on SID-WA data. The data is first partitioned into two independent train and test sets. Three years of data 2009-2011 are allocated to the training set which consists of 1,543,131 admissions and the last year of data (2012), consisting of 507,974 records, is allocated to the test set.

The $N$ internal layers are discovered using either entropy-based divergence calculation methods (Kl-divergence, Jensen Difference), or appropriately adapting frequency based discretization method (Chi-merge), as described in Section 4.

**Statistical Significance Test:** Wherever applicable, we also perform paired t-test [7] to further understand the statistical significance of the obtained results. To be able to run t-test, we partition the test population into 20 folds, then calculate the quality metric for the competing methods in each fold, and finally run the t-test to investigate whether results are significantly different from each other on each metric. Unless otherwise stated, the significance level is set to p-value $< 0.05$.

**Effect of Number of Internal Layers:** We carry out experiments to find the best number of internal layers for each cut-point selection method. We vary the number of layers from 3 to 12. The logic for this range was simply to see if we auto-discover less than 30, 30 to 60, and more than 60 as natural layers. Based on the experiments, the best number of internal layers depends on the cut-point selection methods. However, we observe that the increase in number of layers negatively affects the true positive rate. This

[10] EF is the volumetric fraction of blood pumped out of the ventricle with each heartbeat

[11] All Patient Refined Diagnosis Related Groups Definition

can lead to a significant decrease in recall, while the AUC and the accuracy do not change considerably. The reasoning behind this is as follows: with an increase in the number of layers, more patients are pruned in the initial layers (i.e., predicted readmission is "no") and consequently less patients get promoted to the 30-day readmission layer (which has to be the last layer given the problem formulation). Empirical analyses suggests Chi-Merge to propose 1 (i.e., $N = 1$), and Jensen Divergence optimize at 3 layers overall, while KL-Divergence shows the best performance for 5 internal layers. Table 2 demonstrates the readmission interval for the internal stages as well as the classifier chosen for each stage based on 10 fold cross-validation procedures on training data. Classifiers were chosen based on performance as measured by the appropriate metric from Table 1. It can be seen that Logistic regression (LR) is the dominant classifier for most of the cut-points.

**Effect of Cut-Points Selection Algorithms:** Recall Section 4 and note that given the number of internal stages as an input, we propose different algorithms to automatically select the $N$ cut-points for the internal stage. The aim of these proposed solutions is to choose cutoff days that discriminates two groups of patients with distinct characteristics as much as possible. As described in previous sections, we vary the value of $N$ experimentally and observe its qualitative effect on different internal hierarchy selection algorithms (Entropy Based Divergence vs Frequencey Based Discretization) and observe that these algorithms lead to different readmission windows (cut-points) for the internal stages. Based on our exhaustive empirical analyses, divergence methods have more stable results in comparison with Chi-Merge, especially when we increase the number of internal stages. Table 3 enlists the best result of each cutoff algorithm based on the best number of layers. The results indicates that, for Recall and AUC, KL Divergence significantly outperforms Jensen Divergence as well as Chi-Merge.

**Effect of Threshold Tuning:** As discussed in section 4, since the distribution of data changes from layer to layer, we run model selection algorithms at each layer in order to select the classifier that renders the highest AUC. Different data distributions and classifiers at each layer can lead to different probability distribution which may cause some inconsistency in final predictions. DHC allows us to apply scaling, as well as threshold tuning at each stage. This not only solves the problem of varied probability distributions but also enables us to optimize the selected evaluation metric for each stage. The default threshold is set to 0.5. In general, we expect that increasing the threshold throughout the hierarchy will lead to eliminating more patients at upper layers as no-readmission cases. Conversely, decreasing the threshold leaves the classification decision to lower layers. Among evaluation metrics, we focus on recall and AUC, as they provide insight into the performance of classifiers on skewed datasets. We tune the thresholds through the layers such that we improve the overall prediction quality, with the aim of increasing the appropriate objective (see Table 4 ).

**Baseline Algorithms:** For comparison, we implement five baseline solutions that employ a single classification model on the full dataset without using a hierarchical structure as proposed in DHC. We consider Logistic Regression, Naive Bayes, SVM, Random Forest, and Adaboost as five classifiers to build our baseline models (prior efforts have demonstrated them to work well for risk of readmission) [21]. Comparing the results in Table 5, the Naive Bayes classifier outperforms other models - it has high AUC, and reasonable recall.

Table 5 compares the result of the baseline models and the best DHC result which was obtained by KL-divergence method with 7 layer (see Table 3 and Table 4 ). As stated above, we also report the statistical significance of quality results (significance level is set to p-value $< 0.05$). It can be easily observed that our proposed framework achieves significantly better results for all metrics (AUC, recall, precision and accuracy) compared to the baseline with statistical significance. It is interesting to note that the highest improvement is for the recall and AUC, which are considered as the main metrics when dealing with skewed datasets.

| | Random Forests | Adaboost | Naive Bayes |
|---|---|---|---|
| Readmit Stage | 0.6803 | 0.6539 | 0.6000 |
| Internal Stage | 0.6014 | 0.5830 | 0.5600 |
| 30-day Stage | 0.6245 | 0.5900 | 0.5833 |

Table 6: MHS data: Classifier Comparison based on AUC for three stages

### 5.3.2 *Comparative study on MultiCare Health System dataset*

On MHS internal data, the presented results are representative. We primarily focus on describing the utility of the deployed DHC solution designed using KL-Divergence (as that is the best performing algorithm based on the SID dataset) and compare it with the single stage baseline classification algorithms.

Based on Kl-Divergence, we select one layer in internal stage (i.e., $N = 1$), thereby constituting 3 layers overall (readmit, internal, and the 30-day yes no layer). After getting results from Algorithm 2, the readmission interval for the internal stage was determined to be 65 days. Hence, the problem of predicting 30-day readmission risk is formalized using hierarchical classification as follows: *Readmit Stage*: predicting readmission within 1200 days (based on our dataset, $Max = 1200$), *Internal Stage*: predicting readmission within 65 days, and *30-day Stage*: predicting readmission within 30-days.

**DHC Internal Predictive Models for MultiCare Health System:** The following three classifiers exhibit the best and somewhat comparable classification performance – Random Forests [2], Adaboost [5], and Naive Bayes [7]– for each layer of DHC described in Section 4. Table 6 shows the AUC of the three classifiers in each layer. As shown in Table 6, Random Forests achieves the highest AUC for all three layers. Therefore, we present the results of Random Forests for all three stages for the rest of our experiments.

Table 7 shows the experimental results for the proposed DHC approach and the prediction quality for each stage. Since, Random Forests achieves the best performance, for brevity, we therefore present our results by comparing DHC with a single-stage baseline Random Forests classifier (500 decision trees). DHC outperforms the baseline on true positive count, recall, and AUC. It is noted that the highest improvement takes place in the *Readmit Stage*, as our predictive model achieves higher AUC at that particular stage. This observation is intuitive as it appears that the quality of prediction can be improved by eliminating negative in-

| | layers | Cutpoints |
|---|---|---|
| Chi-Merge | 3 | Readmit(Adaboost),280(LR),30(LR) |
| KL divergence | 7 | Readmit(Adaboost),355(svm),240(LR),195(LR),150(LR),110(LR),30(LR) |
| Jensen Divergence | 3 | Readmit(Adaboost),280(LR),30(LR) |

Table 2: SID Data: Selected cutoffs and classifiers for each cutoff selection methods

| | layers | TP | FP | TN | FN | Prec | Recl | Acc | AUC |
|---|---|---|---|---|---|---|---|---|---|
| Chi-Merge | 3 | 34750 | 103275 | 329928 | 40021 | 0.252* | 0.465 | 0.718* | 0.699 |
| KL Divergence | 7 | 38818 | 118427 | 314776 | 35953 | 0.247 | 0.519* | 0.696 | 0.701* |
| Jensen Divergence | 3 | 35762 | 106907 | 326296 | 39009 | 0.251 | 0.478 | 0.713 | 0.698 |

Table 3: SID data: Best DHC Results for cutoff selection methods.* denotes whether the result is significantly better than the others among the three variations of cutoff selection methods based on paired-t-test.

stances at an early stage. The overall quality improvement becomes less at the internal stage. This is due to the fact that the data in the internal stage is more homogeneous and no distinct patterns among the patients are identified.

We also have observed that for this dataset the difference in KL-divergence across different $k$ (cut-point) values do not vary greatly (most of them approximately 0.057). After further analysis we realize that such low KL divergence and homogeneity arises due to our use of *mean value* to determine the centroid of each group. In the future we explore alternatives to calculate group centroids, potentially leading to larger values of $k$ for the internal stages. Nevertheless, based on Hill-Climbing we selected $k = 65$.

**Factor Analysis:** We apply the Chi-Squared test [7] to determine the significant factors. The $p$-value is set to 0.05 in all cases. We observe that the number of significant factors vary at each stage. There are 32 influential factors for *Readmission Stage* and *Internal Stage* and 16 for *30-day Stage*. Table 8 enlists top-10 most influential factors sorted based on increasing $P$-values, based on the Chi-Squared test. The importance of a particular factor is different at each stage. For example, *Length of Stay* is an important factor for *Readmission Stage* and *Internal Stage*, but not for *30-day Stage*. It implies that, given that a patient is likely to return to the hospital, care managers can concentrate more on risk factors closely associated to 30-day readmission. Such insights are tremendously useful to the domain experts and the physicians for appropriate prognosis.

| Readmission | Internal | 30-day |
|---|---|---|
| Severity Of Illness | Risk-Of-Mortality | Severity Of Illness |
| Risk-Of-Mortality | Severity Of Illness | Risk-Of-Mortality |
| Length-Of-Stay | Length-Of-Stay | Acute coronary syndrome |
| Renal Failure | Secondary ICD9 | Discharge Followup Category |
| Anemia | Cardio-respiratory Failure | Stroke |
| Ejection-Fraction | Fluid Disorder | Cardio-respiratory Failure |
| Pneumonia | Acute coronary syndrome | Fluid Disorder |
| Fluid Disorder | Discharge Followup Category | Chronic atherosclerosis |
| Dialysis | Ejection Fraction | Malnutrition |

Table 8: MHS data: Factor analysis of three different stages of DHC model

## 6. RELATED WORK

To the best of our knowledge, no hierarchical classification techniques for risk of readmission risk prediction have been yet reported in literature. Very few cloud based deployed solutions exist today in healthcare and none are for scoring patients for their readmission risk on multiple chronic conditions, in real-time, integrated with EMRs.

**Readmission Risk Prediction:** An early research result for predicting RoR for CHF patients was termed the Yale Model [11]. Logistic regression was used for 30-day all-cause readmission risk for 65+ year old HF patients. More recently [1], administrative claim data was used to build a regression model on 24,163 patients from 307 hospitals on patients 65 years or older. In collaboration with MultiCare Health Systems, our prior efforts have resulted in accurate solutions [15, 21, 19, 18, 20] to predict 30-day RoR. However, none of these solutions generalizes for any time-interval and automated design, which is one of the primary contributions of the proposed research.

**Hierarchical Classification:** Hierarchical classification is extensively used in various application areas such as in text mining for web page classification applications [4], Information Retrieval [6], and in signal processing [14]. The use of multistage classification for Bayesian combination of classifiers is explored in literature [13], K-Nearest Neighbor [17], and hierarchical SVMs [9]. Thus we turn to a hierarchical or multi-stage classification process for predicting risk of readmission. The design though is non-trivial to optimize overall prediction quality and ensure good generalization, particularly for high risk patients.

## 7. CONCLUSION

Predictive models for risk of readmission can significantly improve quality of care. With an increase in the number of patients suffering from chronic conditions, demand for actionable, accurate, and cost-effective solutions to be deployed also increases. In this paper, we describe the algorithm design, deployment challenges, architecture of our

| | layers | Thld | TP | FP | TN | FN | Prec | Recl | Acc | AUC |
|---|---|---|---|---|---|---|---|---|---|---|
| Chi-Merge | 3 | .55,.55,.45 | 37921 | 112700 | 320503 | 36850 | 0.252 † | **0.507*** | 0.706 † | 0.699 |
| KL Divergence | 7 | .4,.55,.55,.4,.4,.45,.4 | 40044 | 121483 | 311720 | 34727 | 0.248 | **0.536*†** | 0.692 | 0.696 |
| Jensen Divergence | 3 | .55,.55,.45 | 39116 | 118093 | 315110 | 35655 | 0.248 | **0.523*** | 0.697 | 0.7† |

Table 4: SID Data: Threshold tuning for result on table 3.† denotes whether the result is significantly better than the others among the three variations of cutoff selection methods(after threshold tuning).* denotes the result that is significantly higher after threshold tuning compared to result before that. The Thld column describes the selected threshold values from the readmission stage to 30-day stage.

| | TP | FP | TN | FN | Prec | Recl | Acc | AUC |
|---|---|---|---|---|---|---|---|---|
| logistic regression | 52139 | 264117 | 169086 | 22632 | 0.165 | 0.697 | 0.435 | 0.593 |
| Naive Bayes | 37247 | 120142 | 313061 | 37524 | 0.23666 | 0.498 | 0.689 | 0.678 |
| SVM | 3190 | 2575 | 430628 | 71581 | 0.553 | 0.043 | 0.854 | 0.634 |
| AdaBoost | 0 | 75799 | 0 | 432175 | NAN | 0 | 0.851 | 0.696 |
| Random Forests | 3150 | 1434 | 430741 | 72649 | 0.687 | 0.041 | 0.854 | 0.724 |
| DHC (Proposed Technique) | 40044 | 121483 | 311720 | 34727 | **0.248**† | **0.536**† | **0.692**† | **0.696**† |

Table 5: SID-WA data: Single Layer Baseline classifier comparisons. Naive Bayes is the best performing baseline solution.

| | Total | Thld | TP | FP | TN | FN | Prec | Recl | Acc | AUC |
|---|---|---|---|---|---|---|---|---|---|---|
| Baseline (Random Forests) | 14,170 | 0.5 | 376 | 139 | 10,995 | 2690 | 0.7301 | 0.1226 | 0.8007 | 0.6492 |
| DHC-combined | 14,170 | 0.5 | **1,272** | 2,084 | 9,020 | 1,794 | 0.3790 | **0.4148** | 0.7263 | **0.6596** |
| DHC-Readmit | 14,170 | 0.5 | 5,119 | 3,025 | 3,794 | 2,232 | 0.6285 | 0.6963 | 0.6290 | 0.6803 |
| DHC-Internal | 8,144 | 0.46 | 2,749 | 4,360 | 743 | 292 | 0.3866 | 0.9039 | 0.4287 | 0.6014 |
| DHC-30Day | 6,332 | 0.7 | 861 | 1,230 | 3,220 | 1,021 | 0.4117 | 0.4574 | 0.6445 | 0.6245 |

Table 7: DHC Results on MHS data: Each stage contains different number of instances (Total column) and uses different threshold (Thld) for classification.

*cloud-based deployed framework* DHC. MultiCare Health System uses the described DHC framework to predict the 30 day post discharge risk of readmission for their heart failure collaborative. We demonstrate that the proposed framework clearly outperforms baseline solutions for congestive heart failure (CHF). DHC automatically discovers and defines the layers by leveraging the underlying historical patient data. Detailed experimental evaluations on two sizeable real-world datasets statistically validate the utility of DHC and the deployed engineering efforts demonstrate the real-world impact of the framework.

## 8. ACKNOWLEDGEMENT

We acknowledge MHS and Microsoft Azure for Research for their generous supports in this research.

## References

[1] H. BG et al. Incremental value of clinical data beyond claims data in predicting 30-day outcomes after heart failure hospitalization. *Circ Cardiovasc Qual Outcomes*, 2011.

[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.

[4] S. Dumais et al. Hierarchical classification of web content. In *SIGIR*, 2000.

[5] J. Friedman et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 2000.

[6] M. Granitzer. Hierarchical text classification using methods from machine learning. 2004.

[7] J. Han and M. Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.

[8] S. F. Jencks, M. V. Williams, and E. A. Coleman. Rehospitalizations among patients in the medicare fee-for-service program. *New England Journal of Medicine*, 360(14):1418–1428, 2009.

[9] T. Joachims. Text categorization with support vector machines: Learning with many relevant features, 1998.

[10] R. Kerber. Chimerge: Discretization of numeric attributes. In *Proceedings of the tenth national conference on Artificial intelligence*, pages 123–128. Aaai Press, 1992.

[11] H. M. Krumholz et al. *30-day heart failure readmission measure methodology. Report prepared for the Centers for Medicare & Medicaid Services.*

[12] P. E. Krumholz HM. REadmission after hospitalization for congestive heart failure among medicare beneficiaries. *Archives of Internal Medicine*, 157(1):99–04, Jan. 1997.

[13] M. W. KurzyÅĎski. On the multistage bayes classifier. *Pattern Recognition*, 21(4):355 – 365, 1988.

[14] U. Libal. Multistage pattern recognition of signals represented in wavelet bases with reject option. In *MMAR*, 2012.

[15] V. Rao et al. Readmissions score as a service (raas). 2014.

[16] C. v. Walraven et al. Proportion of hospital readmissions deemed avoidable: a systematic review. *Canadian Medical Association Journal*, 2011.

[17] Y. Yang and X. Liu. A re-examination of text categorization methods. pages 42–49. ACM Press, 1999.

[18] K. Zolfaghar et al. Big data solutions for predicting risk-of-readmission for congestive heart failure patients. In *IEEE Bigdata*, 2013.

[19] K. Zolfaghar et al. Exploring preprocessing techniques for prediction of risk of readmission for congestive heart failure patients. In *DMH Workshop*, 2013.

[20] K. Zolfaghar et al. Predicting risk-of-readmission for congestive heart failure patients: A multi-layer approach. *CoRR*, abs/1306.2094, 2013.

[21] K. Zolfaghar et al. Risk-o-meter: an intelligent clinical risk calculator. In *KDD*, 2013.