Rashnil Chaturvedi. Interactive Temporal Feature Construction: A User-Driven Approach to Predictive Model Development. A Master's Paper for the M.S. in IS degree. October, 2017. 54 pages. Advisor: Dr. David Gotz

Predictive modeling with visualization techniques can revolutionize the way businesses operate. Analyzing large datasets on high compute machines makes it possible to utilize advance technologies to support data-driven decision making. A wide range of domains deal with data that have random sequence of events (such as real-time verification or health care). Temporal relationship between these events can be highly predictive in nature. However, existing methods of feature selection makes it difficult to identify temporal relationships to enhance the predictive power of models. Often, it requires domain expert's knowledge to identify realistic patterns. Interactive Temporal Feature Construction (ITFC), a visual analytics workflow is designed to enable effective data-driven temporal feature construction. This application provides a new interactive workflow for model building and refinement, and visual representations to support that workflow. Use cases demonstrate how ITFC can result in more accurate predictive models when applied to complex cohorts of electronic health data.

Headings:

Visualization

Machine Learning

Optimization

Prediction

Health Care Analysis

In-Memory Computation

Interactive Temporal Feature Construction: A User-Driven Approach to Predictive Model Development

by
Rashnil Chaturvedi

A Master's paper submitted to the faculty
of the School of Information and Library Science
of the University of North Carolina at Chapel Hill
in partial fulfillment of the requirements
for the degree of Master of Science in Information Science.

Chapel Hill, North Carolina

November 2017

Approved by

_____

Dr. David Gotz

Table of Contents

INTRODUCTION

Advancements in the area of large scale data processing such as availability of non-relational databases, distributed systems and in-memory computational capabilities have enabled collection of larger and complex datasets. This has made possible for many researchers to experiment with variety of data and apply complex statistical and machine leaning techniques to gain insights that can help in taking informed decisions. Businesses have identified the importance of these techniques and is widely being used in almost every sector, be it transportation (Tarko, A et al, 2008) to medicine (Murdoch et al, 2013). One of the most important tool out of these machine learning buzz is the ability to use the past data to predict what may happen in future, called "prediction". This is very important tool that significantly optimizes the way decisions are made today. A prediction model can help to predict whether it may rain today or not, this helps hundreds of people to plan their day or predicting whether the stock price will go up or down, this may have significant impact on the overall economy. Alternatively, a predictive model in the medical domain might be used to determine patients at high risk of being admitted to a hospital based on their recent medical history. As these examples suggest, accurate predictive models can be enormously valuable for both automated and human-performed decision-making tasks. A small improvement in the prediction technique can alleviate many unseen tragedies.

There have been lot of research done in this area in recent past. However, most of the researchers follow the same basic approach. Training models on a set of records (also called training set) that have few set of features (also called predictors or independent variables). Prediction techniques are usually categorized into regression and classification. In regression, output of prediction model is usually a continuous value whereas classification gives probabilities which can then be classified into different classes. Classification is an important machine learning technique under supervised learning, where a label is known beforehand to help in model training and identification of correct class for unknown/unseen data. Shaping the most effective feature vector representation for a complex high dimensional dataset can be very challenging. In these cases, where datasets can contain thousands or even hundreds-of-thousands of variables, significant effort must be made in feature selection. Techniques such as forward-backward selection, AIC and BIC are well known and widely used for this purpose (Guyon et al, 2003).

Feature selection and feature construction in an automated way can prove to be really beneficial in such scenarios. Researchers around the world have been working towards finding new and efficient ways of dealing with such large-scale datasets (Feurer et al, 2015). Many techniques such as correlation estimations, multi-collinearity and dimensionality reduction is widely used in an automated way to identify combination of variables that can provide best fit for the model. Sometimes existing features may not be able to explain the variance in the data. This requires to try other options such as transformation of existing variables or construct new

variables by combining exiting variables to form new features. These constructed features can then be incorporated into the model-building process to enhance prediction power of the model.

Feature construction is more challenging when it comes to temporal events. This is because temporal data not only exhibits relations between features, but also has relations based on the order of occurrence of the events (e.g., patients with heart failure condition may encounter chest pain then chest x-ray followed by hospitalization, which might be highly predictive than having asthma before hospitalization). Time of occurrence in this type of data can give much more insight than the simple combinations of variables (e.g., a patient with a prescription for chest x-ray before hospitalization may be different than prescribing x-ray after admission to the hospital). As we can see here, temporal relations (such as chest pain then x-ray followed by hospitalization shows high predictive power, ignoring multiple intermediate encounters such as prescription for headache or rashes etc.) with numerous event types makes analyzing and predicting future outcomes significantly hard as the number of possible combination grows in multitude. As a result, an exhaustive automated search and evaluation of all possible patterns can become computationally prohibitive. Another important factor to note here is that incorporating domain knowledge into prediction modeling is important in many domains such as medicine. Health care is a very sensitive area where a minor improvement in prediction can lead to saving many lives. However, it is impossible to search through such a vast combination of features that also holds practical relevance

and is excepted by medical practitioners. Therefore, domain expert guidance incorporated in feature construction, or to incorporate additional constraints on the construction process (Malhotra et al, 2016) can be vital in model building.

This paper presents Interactive Temporal Feature Construction (ITFC), a visual analytics technique designed to overcome challenges during the feature construction phase of model building. ITFC provides a user driven workflow to temporal predictive model development and refinement. Two case studies in the later section demonstrate the useful of this approach in real world scenarios by showing quantifiable improvements to model accuracy. This application provides an interactive approach for model building incorporating both the prediction capabilities of machine learning algorithms and domain knowledge of experts.

The remainder of this paper describes ITFC in more detail. Section problem statement begins by posing the challenges faced within the health care domain. Section related work then provides a brief overview of related work. Sections design and methodology review the ITFC workflow, interface design, and underlying algorithms. Sections Prototype and Evaluation describe a prototype implementation of ITFC and highlight two use cases which demonstrate the potential of the proposed technique. Finally, Section Conclusion concludes the paper and suggests topics for future work.

PROBLEM STATEMENT

The use of temporal event data to make predictions is a widely encountered challenge that is found in almost every sector of businesses, whether it is advertising or scheduling of tasks in computer science such as distributed systems. This paper's focus is to address issues faced in health care due to the temporal nature of events and their relations. Patients encounter symptoms at different times. Medical practitioners make diagnoses of specific conditions and/or perform diagnostic procedures. Some diseases might get cured while other may worsen or some get unnoticed and may re-appear in future. There are chronic conditions which may result in multiple visits to the hospital. Clinicians observe changes in patient condition from visit to visit, and often respond to these developments by designing care plans, performing medical procedures, and prescribing medications.

Predictions and forecasting is an important area of research in medicine. Predictions aids preventive medicine and health care strategies, by pre-informing health care practitioners to take appropriate steps to minimize risks. However, this approach requires good quality and reliable data, information and appropriate tools for the prediction of specific health conditions. Electronic Health Record (EHR) records the history of various events that a patient goes through. These records are usually maintained by various medical institutions. This record may contain thousands of data

points per person. Medical record system captures millions of patient's visits and encounters every year. The International Classification of Diseases (ICD) coding system is used widely in the United States and around the world to represent diagnoses. The latest version of this standard (ICD-10) contains more than 60,000 distinct diagnosis codes (Quan et al, 2005). In total, the number of events/patient encounters tracked over time in a modern EHR system can number in the hundreds of thousands. Many different techniques and systems are used to maintain and analyze these records. This may range from using data warehouse or relational databases. However, it is very difficult to find patterns in this ever-growing data. There needs to be an efficient way to search through this data and identify patterns that could help in improving prediction. This can be very daunting for health care analysts. They have to first align these temporal events for all patients, create features by searching through huge combination of event patterns, construct features vectors of all these millions of patients, generate labels for these records and finally, make predictions (Bates, 2014). Even this is not the end of the story. Health Care analysts have to continuously iterate through multiple combinations to identify the best model for predictions. This while process can be very time consuming and inefficient. There may be times that even after long analysis, there are no proper outcomes. (e.g., distinguishing between "a chest x-ray" occurring prior to a hospitalization vs. a chest x-ray occurring during a hospitalization vs. "a chest x-ray'" occurring after a hospitalization; all of which are distinct from the simple combination of requiring both events to be found at any point in a patient's medical history). Due to this vast nature of data, it is practically impossible for health care analysts to identify patterns

and make efficient predictions that can help in decision making. This iterative model building process is what the methods proposed in this paper are designed to support.

The key research contributions presented include: An iterative, user-in-the-loop model development workflow which includes: (1) model evaluation, (2) visualization of event patterns, (3) the construction of new features from patterns identified in the visualization, (4) the training of new models that incorporate the newly constructed features, and (5) a visual comparison of model performance to understand the benefits of the newly constructed features.

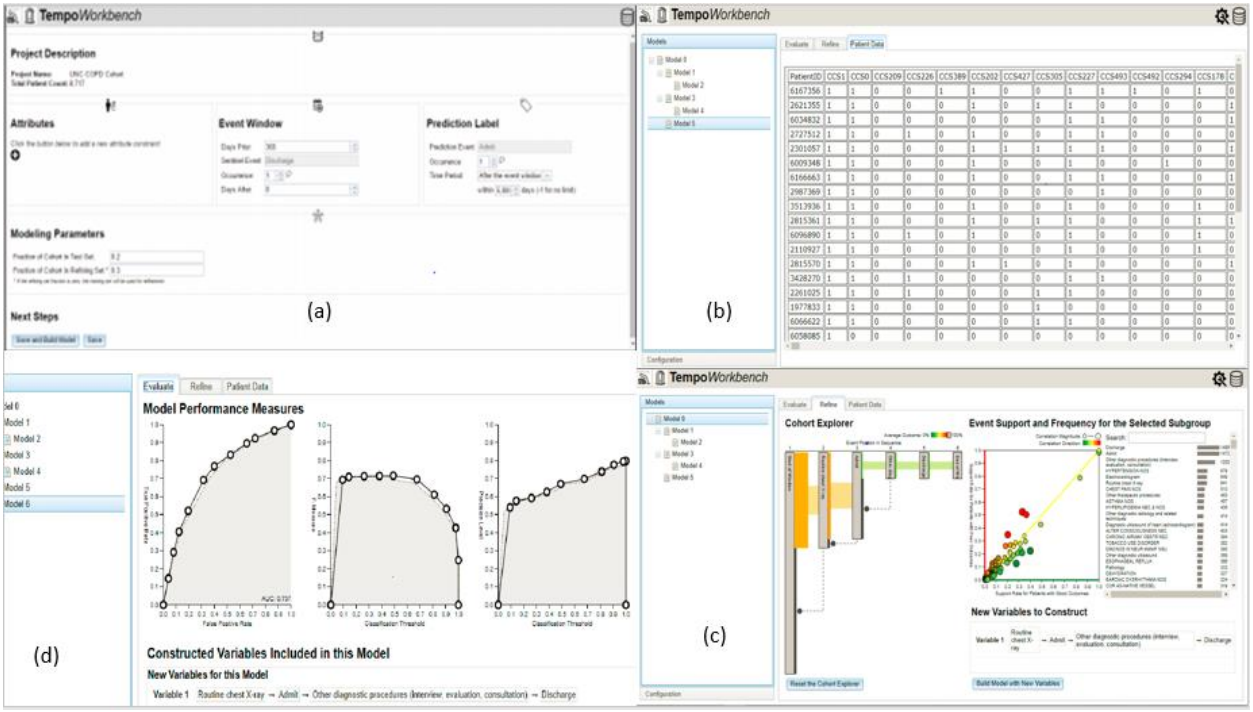


Figure 1: This figure displays the features of ITFC application where (a) is Model Initialization Phase to set the inclusion criteria, (b) is the patinet data (feature matrix) that can be viewed for different models, (c) is the refinement tab that allows users to perform

feature construction and (d) is the evaluation panel for comapring models and their performances on different measures.

RELATED WORK

A wide variety of research is conducted worldwide in the area of prediction, machine learning, feature selection, visualization and temporal events. Areas most relevant to the work presented in this paper include: predictive modeling with high-dimensional data (including health care applications); visualization and prediction methods for temporal event data; and visual analytics methods that support prediction model interpretation and development.

Prediction Modeling

Predictive modeling is both widely studied and broadly applied. As stated in the introduction section, predictions can be made with continuous data using regression or with categorical data using classification techniques (Kuhn et al, 2013) For this paper, the application utilizes the classification technique to predict whether the patient will be diagnosed with a disease of not. In classification, first the data is divided into training set and test set. Training data contains label or outcome for the observations. This data is used to train the model and find the model that performs best on the training data. This model is then used on unseen test dataset to predict outcomes (or possible label). The basic setup for these algorithms is the same. There

is a feature matrix (usually denoted by “X”), a label vector (denoted by “Y”). We regress on X with a set of betas which are the parameter estimates of the feature vector to build model that could predict Y’s for unknown X matrix. Finding the best model majorly depends on these parameter estimates or tuning parameters. Especially for more complex datasets, such as those with high-dimensional data and high levels of interaction between variables, attention to feature engineering is typically required to develop high-performance models (Graepel, 2010). Feature engineering consists of feature selection and feature construction (Liu, 1998). There are many techniques applied for feature selection such as regularization (Lasso), Step AIC/BIC and more.

Performance of these models is measured using various classification evaluation metrics such as Confusion matrix that calculates the true positive rate and false positive rate, precision, recall, and metrics such as ROC and area under the curve AUC or F-measure (Fawcett, 2006). These techniques are widely used in medical field to improve prediction and support decision making (Raghupathi, 2014).

Visual Analytics and Temporal Event Data

Visually analyzing temporal event sequence can significantly enhance the pattern recognition and refinement process. Certain anomalies and outliers become evident in visualization, which cannot be tracked otherwise. Visual analysis plays even more important role when it comes to temporal events. It is challenging to visualize temporal events because of their infinite possible relations. This is the reason visual analytics with temporal data is widely studied topic in recent years and continues to

gain attention (Wang et al, 2008). There are studies in analyzing temporal data by looking and sentinel event and aligning data points around multiple sentinel events. Also, flow based visualization approach has become very popular as it allows a wide variety of exploration techniques to be exploited (Wongsuphasawat etl al, 2011, Wongsuphasawat et al, 201). Health care is the best domain for the applicability of these techniques due to the availability of very high dimensional data points. (Gotz et al, 2012; Gotz et al, 2014).

Beyond pattern finding, temporal data is widely used as the basis for predictive modeling (Laxman, 2006), especially in areas like health care where the patterns over time can be highly informative and keep changing (Moskovitch, 2016). Recent studies have demonstrated that using the relationships in these temporal events can significantly improve the performances of the prediction model (Malhotra et al, 2016). However, they also show that such patterns are hard to obtain without manual feature construction. In this paper, an automated and interactive way of feature construction is proposed.

Visualization techniques have always supported understanding and interpretation of prediction models. It is even more valuable when there are high dimensional temporal data. Visualizations enable to see individual features and helps to decipher the variations explained my them in population data (Krause et al, 2016). Visualization techniques enable researchers to include statistical information gathered from the analysis of data by sampling multiple datasets from the population data. The metrics

such as p-value, R-square, ROC curve etc. give an understanding of the importance of individual features towards the predictive power of the model. There is research work in using visualization for feature selection (Gotz et al, 2014), but there are no/limited work done in using visual analytics for feature construction.

ITFC WORKFLOW DESIGN

In this section, we talk about the design of ITFC application. Building a predictive analytics application requires multiple steps. We need to go through different phases of model building to finally arrive at the desired optimal prediction model. This whole is an iterative process where the users or researchers have to spend lot of time in finding the best fit. This section provides an overview of the ITFC workflow, which is designed to support this iterative model development process.

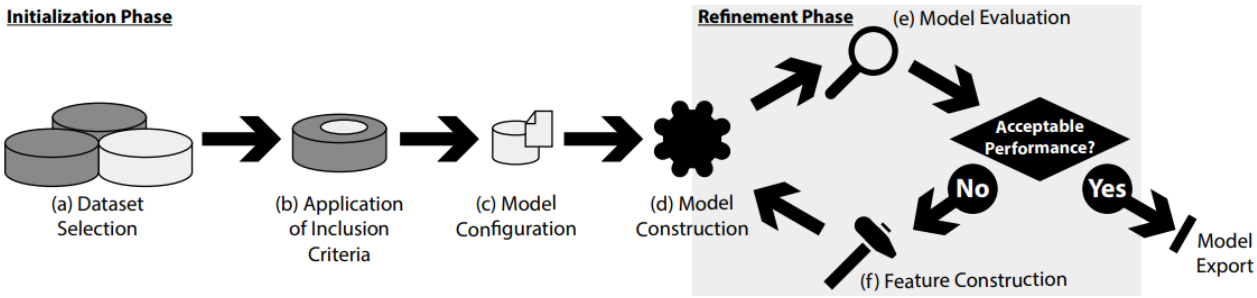


Figure 2: ITFC workflow includes two high-level phases. At the start, initialization phase results in the construction of an initial predictive model. Then refinement phase supports iterative feature construction to drive model improvements.

Initialization Phase of the ITFC Workflow

The initialization phase consists of four basic steps which result in the production of an initial predictive model. As shown in Figure 2, step (a) involves selection of data cohort. This is the refined data set under investigation. Users can either selected the preloaded data or they can upload the data to be analyzed. Once the data is selected, ITFC parses the data and gathers information on various features or variables that are part of the dataset. In the second step, users are provided with an interactive dashboard that allows them to set various inclusion criteria for the current analysis. Users can select which portion of the data should be included for the analysis by specifying the event window start and end. This window extracts the data points based on number of days specified in inclusion criteria. There are other optional variables that can be included such patient demographics. The event window comprises of number of days prior to sentinel event, days after the sentinel event and the number of occurrence for the event. This dataset is then used for model training, refinement and predictions.

The third step includes specifying the parameters for model training, called “Model Configuration”. This includes selecting the sentinel event and selecting the prediction label. Prediction label is the target event that needs to be predicted for new data points. It will be used for training then used for making predictions. Other criteria at this stage includes the event occurrence, i.e. which occurrence of the prediction label we want to predict (for eg, predict whether the patient will be admitted to the hospital second time). Then we have time window with number of days. Time window is

basically when we would like to predict the occurrence of prediction event. It can be before the event window, during the event window or after the event window. For example, we might want to predict that a heart failure may happen after 30 days of discharge from the hospital. In this case, time window is after the event window and number of days is 30. So, our algorithm will try to make predictions for the future using the past data.

Finally, model configuration step also includes an important selection criterion for dividing the original dataset into training set, refine set and test set. The training set is used to build initial model and iteratively search for a model that performs the best on the refine set. This helps to eliminate the possibility of overfitting. When a model is identified with minimum deviation on the refine and training set, it is used on the test set to make predictions. Refinement set is used to visualize prediction errors and support pattern discovery during the Feature Construction step. The training set and refinement sets have labels tagged to the data points which helps in the training of the model as well as checking the performance. The user must specify what percentage of data records from input dataset to include in each of these datasets, and the system will randomly distribute without replacement the data records. This random distribution of data into three distinct datasets is widely used approach to avoid overfitting (Fawcett, 2006).

The screen capture in Figure 3 (a) depicts the user interface in our prototype system for specifying both the Inclusion Criteria and Model Configuration. As the figure

shows, the process is enabled using a relatively simple combination of traditional interface widgets.

Once the data selection is performed, final step in this phase of the workflow is Model Construction. This is the steps that builds our initial model based on the inclusion criterion and configurations specified. In this phase, training data *(D_{train})* is transformed into a feature matrix *(F_{train})* with events as columns (these events are diagnoses, procedures or hospital admissions as recorded in medical record) and patents as rows. In this initial step, temporal relations between events is not considered. Feature matrix only preserves the information that whether an event occurred or not for each patient. This feature matrix gives us our baseline matrix that can be used for model building. In addition, a label vector *(v_{train})* is created containing one label per data item of the training set. The feature matrix and label vector are then used to train an initial predictive model *$m(F_{train}, v_{train})$*. Multiple prediction algorithms were tried for model building such as Naïve Bayes and Gradient Boosted Trees. However, the algorithm that is used in ITFC is logistics regression with BFGS optimization.

Notation	Description
D_{input}	The entire input dataset
$D_{test}, D_{train}, D_{refine}$	The testing, training, and refining sets
t_b, t_a	The length of the event window before (t_b) and after (t_a) the sentinel event's timestamp; determines which time span of events are included from each event sequence when building the feature matrices
$v_c = \{e_1 \rightarrow e_2 \rightarrow \cdots e_n\}$	A constructed variable with n event types in the specified order
F_d^b	Feature matrix containing only baseline features (individual events) for $d \in test, train, refine$
F_d^c	Feature matrix containing only constructed features (event patterns) for $d \in test, train, refine$
l_d	Label vector for $d \in test, train, refine$
m	A predictive model
$m(F_{train}^b, F_{train}^c, l_{train})$	A predictive model trained using baseline and constructed features as well as a label vector
$m(F_{train}^b, l_{train})$	The initial predictive model, trained using only baseline features and a label vector

Table 1. Summary of notation

Table 1: Table with various notations used to represent different datasets and features.

Refinement Phase of the ITFC Workflow

Refinement phase allows users to interact with the ITFC application and perform guided search to identify the best model. Once the baseline model is trained, refinement phase begins. As Figure 2 illustrates, this phase consists of three iterative steps with a final model export decision step. It is an iterative process which terminates once an optimal model is identified. The steps consist of model evaluation, model export, feature construction and model construction. At the model evaluation and export step, performance of the models is compared and the best model is exported.

Model Evaluation is the first step in the refinement step and it calculates variety of metrices for performance measurement. These metrics are ROC curve, area under the curve AUC, F-measure and Precision at different levels of classification threshold. ROC curve gives model's true positive rate, false positive rate etc. These metrics are compared against each other using visualizations and charts. At this step users can analyze and export the feature matrix data from the Patient Data tab (Figure 3). Figure 3 displays the evaluation tab and the patient data tab in detail. Evaluation tab consists of three charts, leftmost of ROC curve that plots the true positives on the Y axis vs. false positives on the X axis, middle one is for F-Measure metric against various classification threshold levels and finally rightmost of precision curve that demonstrates the precision of prediction at different classification thresholds. Small circles on these charts displays measurements at different thresholds and lines are drawn to connect the circles, providing a visual interpolation of intermediate values. Users can mouse over the circles to retrieve exact measurements in a tool tip, and the ROC plot is augmented with a label in the bottom right displaying the Area Under the Curve (AUC) of the RUC plot. These statistics are traditional model performance measures and used widely within the machine learning community (Fawcett, 2006). Moreover, they are provided as part of the default model evaluation suite included in Apache Spark's Machine Learning Library (Ryza et al, 2017).

After assessing the performance of the model, health practitioners can make a decision on the acceptance of the model. If the model performance is acceptable, model can be exported and used in production for making predictions. However, if

the performance is not acceptable, further exploration needs to be performed to find new features and models. This exploration is supported with the use of multiple visualizations in the “Refine” tab (Figure 3).

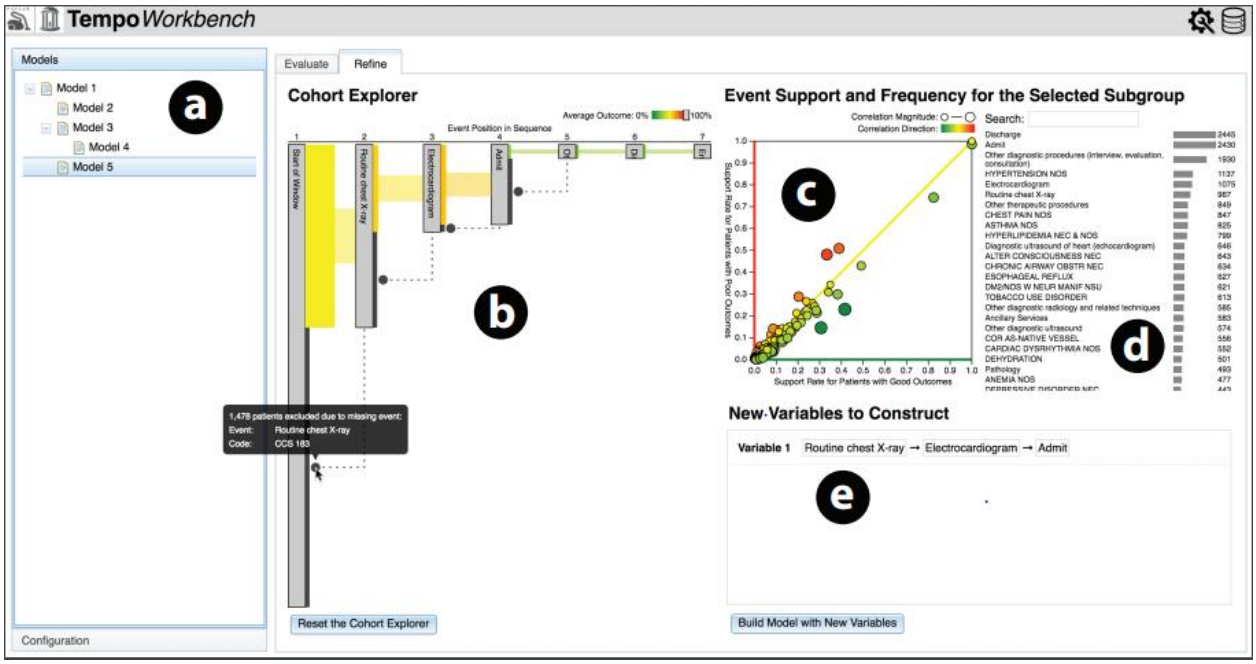


Figure 3: A screenshot taken during the development of a model to predict hospital readmission. Users refine (a) an existing model by (b) exploring patterns of events over time, along with linked views of (c) event-label correlations and (d) event frequency statistics. (e) New variables are constructed as temporal combinations of events, and these new variables are incorporated into updated models. The new models extend (a) the model tree, which allows users to compare the performance of a new model against its predecessor.

Feature Construction is the most critical part of the ITFC workflow. It allows users to build new features by incorporating domain knowledge into model refinement. The approach behind this design is to iteratively identify events which were wrongly predicted, build new features by combining sequences of events that would help in improving predictions for these events and finally expand the feature matrix to include these newly constructed features. This newly formed matrix is then used to build a new model.

Thus, ITFC is designed to (1) visualize the data records for which the current model was incorrect, and (2) help users identify temporal event patterns within those records which are strongly associated with the correct label. These patterns are then used to define newly constructed features which can be used to train a revised model. In order to retrain model, the current model is tested against refine date (D_{refine}) using feature matrix (F_{refine}) and labels from the refine set. This gives us a class of outcomes, incorrectly predicted as positive and incorrectly predicted as negative. These two set of data points are of interest in feature construction. We display these two wrongly predicted data points using scatter plot as shown in Figure 3 (c). Figure 3 shows various visualizations made available for the users to assist in feature construction. These three coordinated views on the refinement panel are designed to support user-driven e pattern discovery. That is, to help users discover through a combination of domain expertise and statistical summaries, patterns of events within the incorrectly predicted data which might help a model better discriminate between classes during the prediction task.

The three coordinated views include: (1) the Cohort Explorer Figure 3 (b); This visualization is the main driving element of this panel. Cohort explorer displays event window selected as part of the initialization phase. In the middle, we have sentinel event and the highlighted sections on both side shows the events before and after the sentinel event. The grey bars represent the start, end of the window. Since all events are between the window, these bars are at 100%. The two color-coded bars on both sides of sentinel event is called as time graph (Figure 4). This graph displays the average time delay between milestones. The color of those bars represents the proportion of the data records with the label of interest for prediction (e.g., the percentage of patients who were hospitalized). In our prototype we adopt a green-to-yellow-to-red gradient to reflect users' intuitive cultural associations of “good” and “bad”. When users select any of the time edges in the cohort explorer, the other two visualizations get updated. The time edge shown in the below figure constitute many events that may have occurred between the two milestones (which is sentinel event and start of window in the Figure 4 (a)).

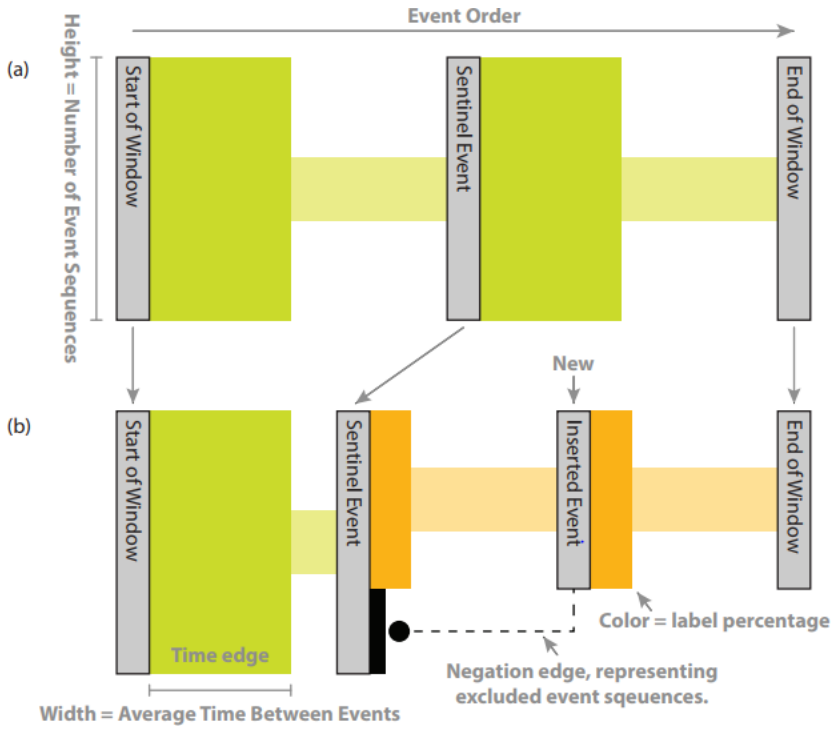


Figure 4: The cohort explorer starts (a) by showing the sentinel event along with the start and end of the event window. (b) Users insert events to define patterns, as in this example's "Sentinel Event" "Inserted Event". The negation edge (dotted black) represents sequences without "Inserted Event", and the change in color indicates that a higher percentage of sequences with the pattern had a negative label.

(2) Scatter plot summarizes the event support and correlation to the prediction label as shown in Figure 3 (c); x-axis displays the support rate or correlation for patients with good outcome whereas y-axis holds information on the support rate for patients with bad outcomes. Events on the extreme ends of the x or y axis and the size of the dot on scatter plot shows higher magnitude of correlation with the prediction label. Color of the dot represents the correlation with prediction label. Events that are close to the diagonals are the events that do not explain much variations in the prediction of

label. The statistics reflected in this graph is the information on the pattern seen in wrongly predicted data records. (3) The third visualization on this panel is the histogram Figure 3 (d). This histogram shows the frequency of events occurring for the selected time edge in the cohort explorer. There is a search box at the top that allows the users to enter the event name and select the required event.

Users use their domain expertise and statistical knowledge to identify an event of interest, that have correlation with the prediction label as well as have relevance in the real world. For example, viral fever followed by heart failure may have less significance compared to chest pain followed by heart failure. Therefore, these judgements can be made by experts to improve the predictive power of the model. Once the event is identified, users can right click on the dots or they can select the event of interest from the histogram to add it to cohort explorer. This event is then added to the time line as shown in Figure 4 (inserted event). We can see that the color of the time edge associated with the new milestone also changes to a different color. The dashed lines represent the negative edge, which are the events that were lost because they were not present for few patients. This selection of event causes the dataset to further reduce so that we are left with a dataset that is relevant for the analysis. Similarly, more and more milestones can be added to the time graph, which will reduce our problem to a set that represent strong association between the predicted label and the event sequences. The reset button can anytime bring back the cohort explorer to its initial state.

Once the desired sequence of events is reached, a new constructed variable can be created using the select event node. Figure 3 (e) demonstrate this functionality. Users can then initiate a new model using the selected variable or they can construct multiple variables and use in them in model building. Once the variables are created, users can initiate the creation of new model by clicking on Build Model button. This will trigger an event that will use the newly constructed variables as part of the feature matrix to build a refined model. Figure 5 demonstrate the overall flow in terms of matrix operations. First (Figure 5 (a)) sentinel events are identified for each patient, if the sentinel event never occurred for that patient then that patient is excluded in the filtering process (b). Out of the remaining patients, their events are windowed based on the inclusion criteria set by the users (c). Finally, the events are aligned such that their sentinel events fall on time 0 (d). This aligned data is then transformed into a feature matrix of ones and zeros as shown in Figure 5 (e).

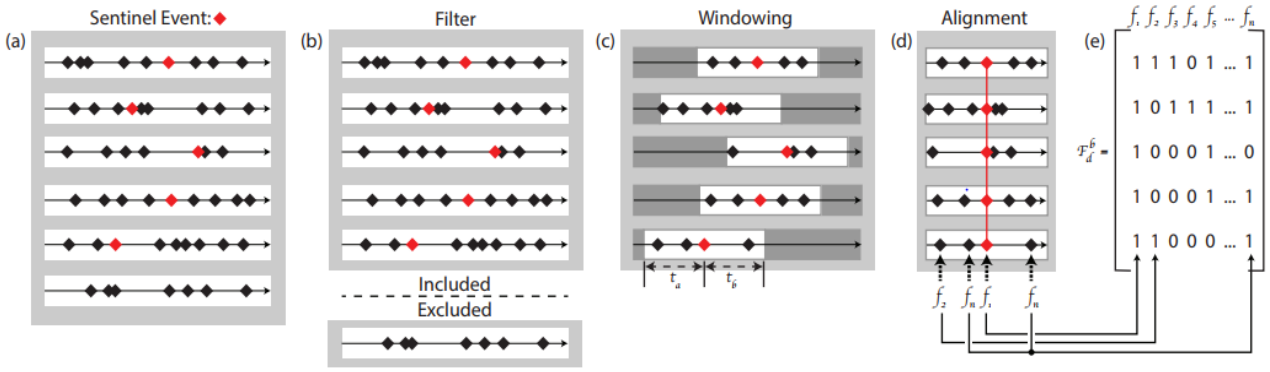


Figure 5: The raw event sequence data is transformed into time-aligned fixed-duration time windows using a four-stage process of (a) sentinel event identification, (b) filtering, (c) windowing, and (d) alignment. The aligned data is then (e) used to constructed the base feature matrix.

Refined Model Construction: A refined model is the model created after adding newly constructed features to the feature matrix through the refinement tab. This new model has all the features of its parent model as part of the matrix along with the newly constructed features discovered using pattern search through visualizations in refine tab, especially the Cohort Explorer. A revised model is constructed with combination of feature matrix with parent variables and newly constructed features $m(F_{train}, F_{constructed}, l_{vector})$ i.e. parent features and the derived features. This new revised model again goes through the training process and is tested again the test dataset (D_{test}). The evaluation for this model is performed in the same way it was done for the parent mode. Therefore, we get the performance measures of ROC, AUC, F-Measure and the Precision at different classification thresholds. This allows users to compare different models against each other on different metrics. Since both models are trained on the same training set and evaluated on the same test set, the differences in performance are attributable entirely to the difference in constructed features.

This whole process can be repeated multiple times in an interactive manner at real time to provide more refined prediction models. This parent-child hierarchy is maintained using model tree architecture as shown in Figure 3 (a). This tree can be used to navigate the multiple models created during a typical ITFC session. Users can select any of the models in this tree to compare the selected model's performance to

its parent. In addition, users can click back and forth on the various models in the tree to make rough comparisons in performance across arbitrary models within the tree.

DATA TRANSFORMATION

In order to achieve the iterative process of model refinement and feature construction, the data needs to be transformed in multiple steps to bring it the finest form for making predictions. These steps (1) forming event sequence window and align events so that the sentinel event is at time zero for all the patients; (2) the construction of the base feature matrix; and (3) the creation of the constructed feature matrix formed from the user-specified event patterns defined via the Cohort Explorer interface. This whole process is demonstrated using Figure 5.

Event Sequence Filtering, Windowing, and Alignment: As we discussed in Work Flow Design section, users need to specify the inclusion criteria during the model configuration phase. These inclusion criteria may be related to the demographics of the patient such as age, gender etc. or they can be model configuration specific parameters such as sentinel event (which can be diagnosis, procedure or any hospitalization), prediction label (which again can be any of the events from patient's medical record), event window start and end date (we annotate then as t_b time before and t_a time after (Table 1). These dates are the start and end of the period that user wants for their analysis. For example, users may want to take data for each patient

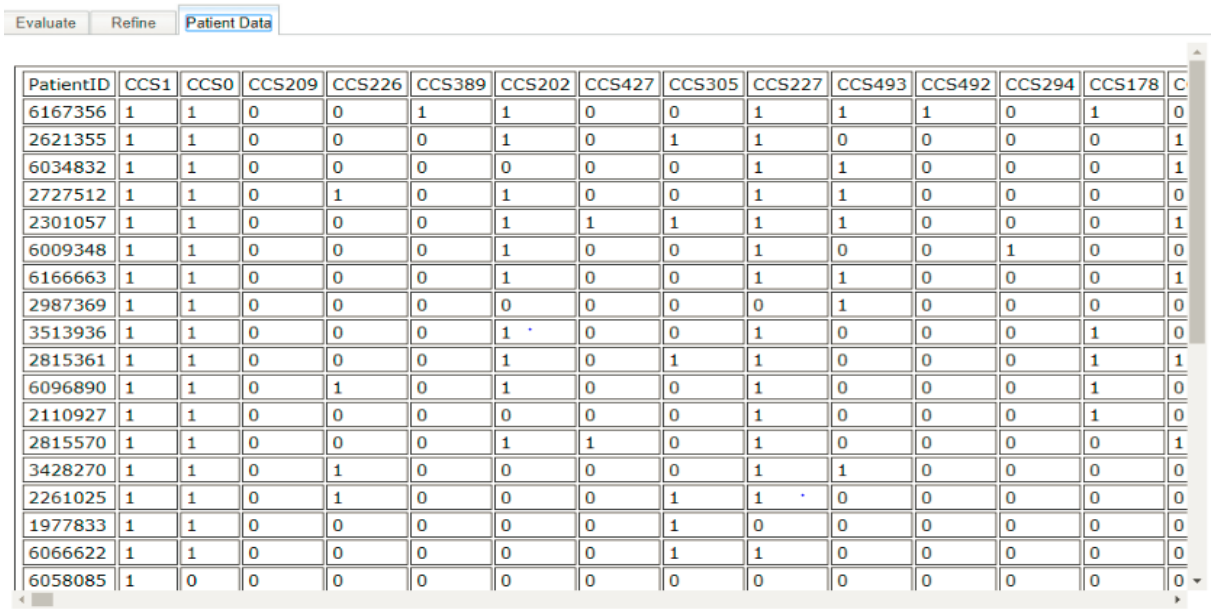
365 days prior to sentinel event and 100 days after the sentinel event to perform modeling). Finally, the occurrence provides ability to select which occurrence is the interest of analysis (For example, users may be interested to analysis patients who had at least two heat failure situations). The values at the inclusion stage sets the base matrix which is our input matrix *(D_{input})* used for initial model building. This matrix is used for training the models, testing, and in the refinement phase to expand to include constructed variables.

Once the inclusion criteria are specified, ITFC performs a search on the complete dataset to find patients that matches this inclusion criteria. As shown in Figure 5 (a-b), step (a) identify patients with matching sentinel event and step (b) filters out patients that do not have the selected sentinel event. This leaves us with the data with patients, all of who have sentinel events and the occurrence for that sentinel event. In the third step (c), start time period and end time period is used to construct an event window for each patient. For example, if the start period is selected as 365 days, then for each patient events prior to 365 days from the sentinel event are discard for each patient. Similarly, the filtering happens for events after the sentinel event looking at the end period selected. This is performed by using the event timestamp from the datasets. Each patient has a date and time of event in the dataset, which can be used to filter out data points that do not match the inclusion criteria. This operation leaves us with equal length window for each of the patients as shown in Figure 5 (c). However, the absolute times associated with each of these windows varies depending on the time of occurrence for the corresponding sentinel event. For example, in a dataset of

electronic health data, the date upon which each patient is diagnosed with heart failure can vary broadly. At this time, we would like to align all these events so that sentinel event for all patients is aligned to time zero (t=0). So, we have sentinel event aligned to a single axis as shown in Figure 5 (d). Number of events for each patient may vary depending on number of medical records are available for that patient in the specified window. This approach allows users to analyze data with respect to the sentinel event irrespective of the date that event actually happened.

Base Feature Matrix. Once the user initializes model configuration, three base matrices are produced by ITFC. These base matrices are created when the user specifies the training, test and refinement ratios. These matrices are used, respectively, for model construction, model evaluation, and feature construction. All these matrices follow the same architecture where the rows are the distinct patients and columns represents unique events occurred at least once among patients in the analysis. If a patient encountered a particular event, then the cell value for that event is marked as one otherwise it is zero. For example, if a patient went through x-ray, but not admission to the hospital, then the value in column x-ray is marked as 1 whereas column for hospitalization will be marked zero. This hospitalization column would be marked as one for another patient who was admitted to the hospital. In the design of feature matrices, maximum value that a cell value can take is one. In medical science, severity of any particular diagnosis cannot be attributed to the number of occurrence of that event. It may be possible to have multiple diagnosis for chest pain, but it not necessary that it may have direct correlation with any particular

disease. Therefore, single occurrence of a symptom in medical health record holds the same significance as the multiple occurrence of that particular event. Domain specific modifications can be made to the design to incorporate higher values of event occurrences. In that case, the cell value may be more than one and will represent the number of times an event occurred for a particular entity under consideration. User can view the feature matrix data along with patient using the patient data tab in the ITFC application as shown in below figure.



PatientID	CCS1	CCS0	CCS209	CCS226	CCS389	CCS202	CCS427	CCS305	CCS227	CCS493	CCS492	CCS294	CCS178	C
6167356	1	1	0	0	1	1	0	0	1	1	1	0	1	0
2621355	1	1	0	0	0	1	0	1	1	0	0	0	0	1
6034832	1	1	0	0	0	0	0	0	1	1	0	0	0	1
2727512	1	1	0	1	0	1	0	0	1	1	0	0	0	0
2301057	1	1	0	0	0	1	1	1	1	1	0	0	0	1
6009348	1	1	0	0	0	1	0	0	1	0	0	1	0	0
6166663	1	1	0	0	0	1	0	0	1	1	0	0	0	1
2987369	1	1	0	0	0	0	0	0	0	1	0	0	0	0
3513936	1	1	0	0	0	1	0	0	1	0	0	0	1	0
2815361	1	1	0	0	0	1	0	1	1	0	0	0	1	1
6096890	1	1	0	1	0	1	0	0	1	0	0	0	1	0
2110927	1	1	0	0	0	0	0	0	1	0	0	0	1	0
2815570	1	1	0	0	0	1	1	0	1	0	0	0	0	1
3428270	1	1	0	1	0	0	0	0	1	1	0	0	0	0
2261025	1	1	0	1	0	0	0	1	1	0	0	0	0	0
1977833	1	1	0	0	0	0	0	1	0	0	0	0	0	0
6066622	1	1	0	0	0	0	0	1	1	0	0	0	0	0
6058085	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6: Users can view the patient data in the “Patient Data” tab. This data has patient ID as rows and columns represents CCS code category. This figure represents the base matrix view of input data.

This matrix is our input matrix D_{input}, which is used in all phases of the ITFC workflow process. A key observation is that this process produces a base feature matrix in which each column represents information for a single independent variable (e.g., a single medical diagnosis code). The base feature matrix does not in any way

capture information about event patterns over time (e.g., a procedure occurring before diagnosis vs. the same procedure occurring after diagnosis) or any other inter-variable relationship.

Building the Constructed Feature Matrix

On every refinement phase, user constructs a new feature using the refine tab and the cohort explorer discussed in the above sections. When the user has identified an appropriate feature to construct, the sequence of events is selected and formed into a new variable. Users build the new model using these newly constructed variables. During the refinement phase, three new constructed feature matrices are created D_{train}, D_{test} *and* D_{refine}. These matrices are used for model construction, model evaluation and feature construction phase. At every stage of exploration, parent matrix is preserved and if there is a new child model is created, it is created by transforming the parent matrix and adding new constructed features. This new version of matrix is then used for further exploration. This allows users to quickly get back to any of the metrices and perform further analysis.

All the three feature matrices constructed using new features have identical representation. Row represents unique patients and columns are the constructed features. The values for these constructed features is populated with one if the event sequence selected by the user is present otherwise zero. For example, if the user constructs a new feature *(*F_c), which is sequence of chest pain followed by x-ray followed by hospital admit, then for the new constructed column cell value will be

one if this sequence of events is present for the patient and zero if not. Importantly, the pattern events need not be directly sequential as long as they appear in the correct temporal order. These features matrices are updated on each iteration of model refinement process and new columns are added to the matrix. These newly constructed features will enhance the predictive power of model and can help in decision making.

PROTOTYPE INTERFACE

A prototype application was developed as part of this research to demonstrate the workflow and techniques described in the above sections. This section describes the technology stack adopted by the prototype implementation as well as the real-world data to which the prototype was applied.

Multiple components are put together in order to develop a prototype of ITFC. ITFC is a web based application with front end developed using HTML, CSS and JavaScript. Data driven document (D3) (Bostock et al, 2011) is used for creating visualization application and Dojo for both interface widgets and layout containers (Holzner et al, 2008). As a result, the interface is accessible from all modern desktop web browsers. Server-side implementation is done using Java server pages (JSP) and Tomcat is used for deploying the web app. Data transformation performed during the

ITFC workflow is outside the computation capabilities of single machine. Therefore, in order to process such large datasets and perform machine learning, Apache Spark was utilized. Apache spark is a framework built on top of HDFS and Hadoop. Underlying implementation of spark is same as map reduce, but it provides greater flexibility from a programming perspective. Moreover, Spark provides in-memory optimizations and lazy computation which together allow it to perform up to hundred times faster than the Hadoop map-reduce framework (Zaharia et al, 2010). Spark executes by forming a lineage of execution cycle. This is maintained using Spark's resilient distributed dataset design (RDD), which allows for the caching and reuse of data in memory. This in-memory is suitable for ITFC as it helps in performing iterative execution of matrix operations during the refinement phase in an efficient and scalable manner. The Spark Machine Learning Library (MLlib) is used for all predictive modeling and evaluation (Meng et al, 2016)

EVALUATION

The ITFC prototype was applied to a variety of real-world data sets from the medical domain. Data was obtained (with IRB approval) from the UNC Health System's Clinical Data Warehouse (CDW) (Mostafa et al, 2015), which is a large-scale data repository containing a rich variety of electronic health information gathered from both hospital and outpatient facilities within the UNC Health System since mid-2004. Data contained in the CDW includes a vast array of variables ranging from patient hospitalization to diagnosis to different procedures. Basically, this dataset has information on various events occurred for patients in the medical history and as recorded by the UNC health system.

In this paper, two use cases were used to demonstrate the application of ITFC in the real world. These two examples incorporate data from CDW and has information of various events for real patient encounters. The data used for testing ITFC includes (1) Heart failure dataset: This dataset cohort includes patients that encountered heart failure at least once. The dataset includes all the events present in the medical history of patients who encountered heart failure situations. (2) The second dataset used was Chronic obstructive pulmonary disease (COPD). COPD is a common lung disease which involves damage to the lungs over time. The events are recognized with the use of ICD-codes. ICD codes are universally accepted codes that tie difference diagnosis

conditions. Similarly, procedure codes are identified using CPT codes provided as part of CDW datasets. This data also includes information on the hospital encounters as well as various lab events. This data comprises of close to 22,000 patients and all their events from 2008-2014. The data is anonymized in order to protect the identify of individual patients. Original CDW data has millions of patients and hundred-and-thousands of heath records, but for this study the data is sampled from the larger population to test the usability of ITFC. For each use case, CDW data from 2008-2014 was obtained for all patients with: (1) at least one encounter with the UNC hospital system, (2) at least one encounter with the UNC outpatient medical system, and (3) a history of being diagnosed with the target condition that is the focus of corresponding use case. Criteria 1 and 2 ensure that each included patient uses the health system for both inpatient and outpatient care, while criterion 3 focuses the dataset to the use case's target medical condition.

These datasets include tens-of-thousands of data points that possess temporal relationships. These two datasets highlight the features of ITFC and demonstrate various phases of the workflow. They also help in showcasing the improvements in predictions by incorporating an iterative search to find patterns, and by constructing new features with help of domain expertise. This section describes two such use cases, each focused on a cohort of patients with specific conditions: (1)~chronic obstructive pulmonary disease (COPD) (2)~heart failure (HF).

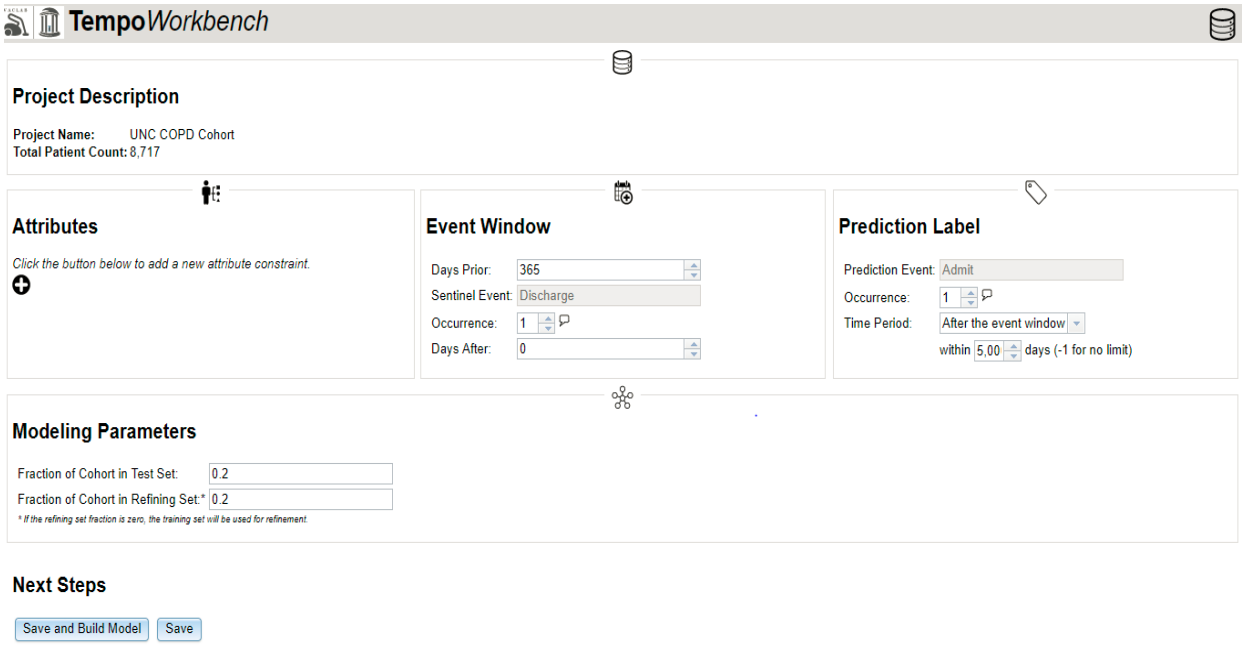


Figure 6 (a): Screen captures of the prototype ITFC system for COPD cohort showing(a)~inclusion criteria and model configuration.

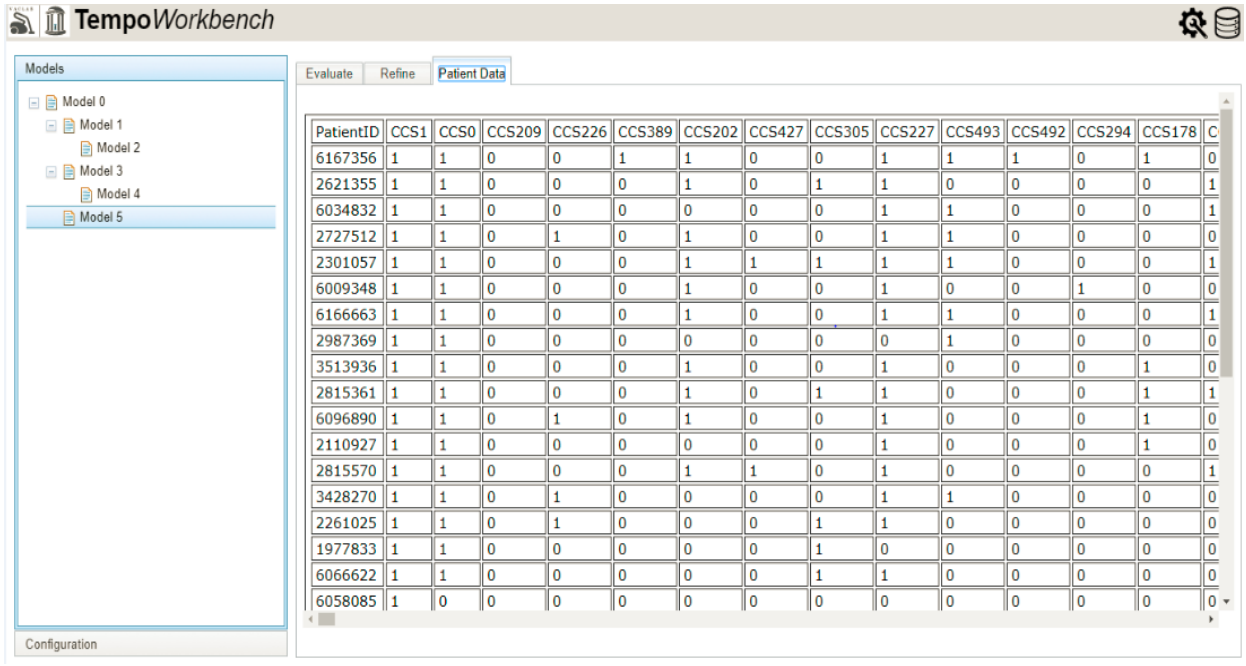


Figure 6 (b): Screen captures of the prototype ITFC system for COPD cohort showing (b)~initial model evaluation

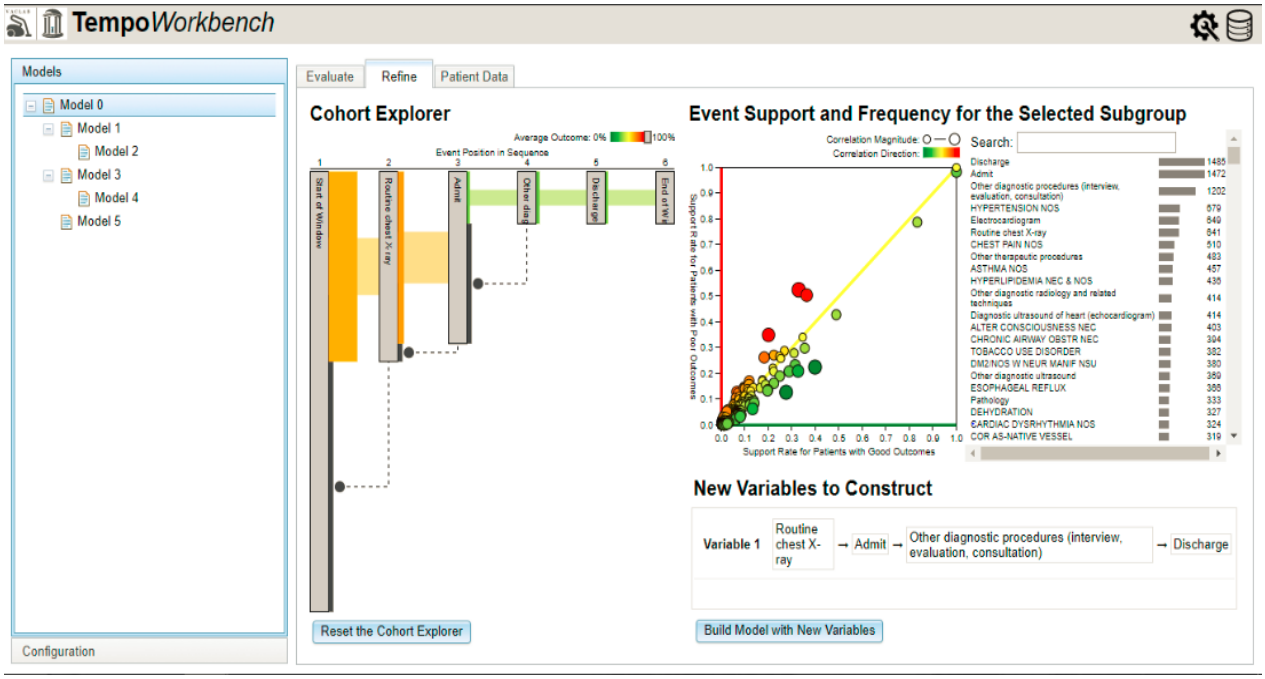


Figure 6 (c): Screen captures of the prototype ITFC system for COPD cohort showing (c)~feature construction.

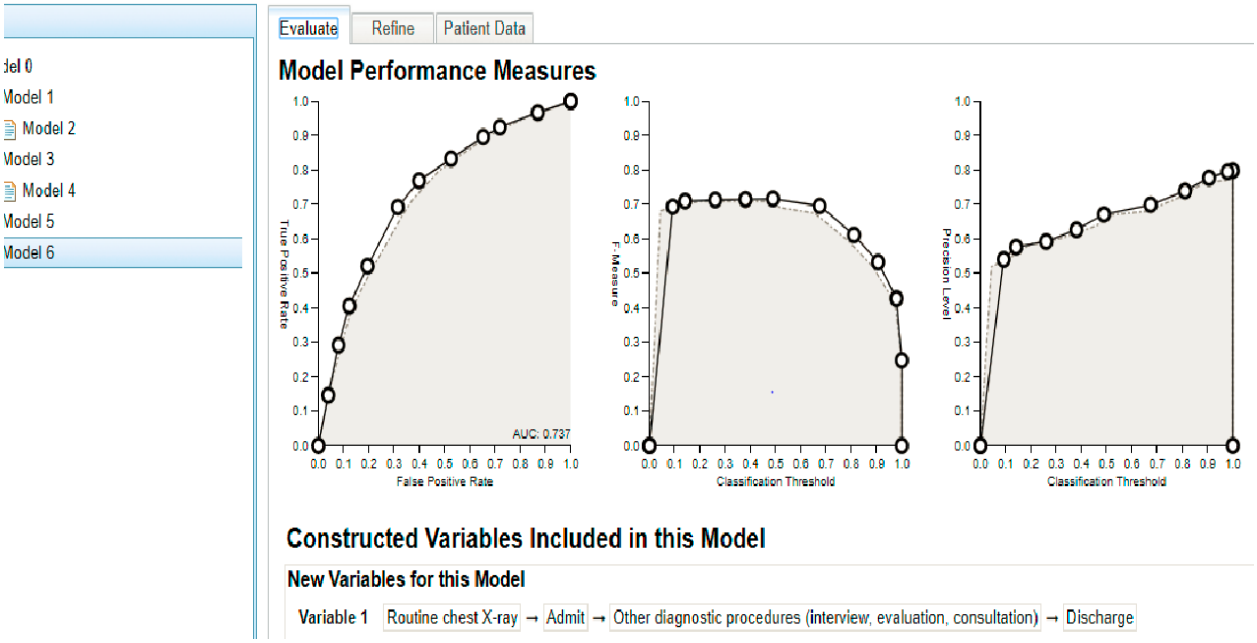


Figure 6 (d): Screen captures of the prototype ITFC system for COPD cohort showing (d)~refined model evaluation showing (e)~a ROC curve comparison with the prior model showing improvements in performance

Use Case 1: COPD

First use used for testing ITFC workflow was for patients who encountered COPD condition at least once and had data points on hospital admission encounters. In this use case, COPD data cohort was carved out of CDW datasets. This data then went through couple of pre-processing steps where the data was filtered and unwanted information was removed before the modeling processes. The data was finally loaded into ITFC application. The comprised of 8,717 unique patients who had a history of COPD-related diseases (COPD was represented using the family of 490. * - 496. * ICD-9 codes). This data contained information on various diagnosis, procedures and hospital encounters with approximately 140 events per patient, with a total of 1.25 million data points for 8717 patients. Figure 6 above demonstrate the overall execution of ITFC workflow for COPD cohort.

The main motive of this analysis was use COPD data and identify patients who were admitted to the hospital and then use that information to build a predictive model that helps us in estimating whether a set of patients will be readmitted to the hospital again. Once the data was loaded, inclusion criteria were decided. Since the interest for this experiment was to filter out patients who were discharged and readmitted to the hospital in a certain time frame, sentinel event was chosen as “Discharge” and prediction label was set as “Admit” (i.e. patients who were admitted after discharge). Millions of dollars are wasted due to hospital readmission and is a very important area of research. Therefore, hospital discharge is chosen for sentinel event. Number

of days prior to sentinel event is set to 365 days i.e. one year and zero days are selected for days after. For this experiment, the interest was to consider patients with at least one occurrence of hospital admission in COPD cohort. Finally, the dataset was divided into test set with 50% of randomly picked data points, 30% of the dataset set for testing and the remaining 20% for refinement process. Figure 6 (a) displays this inclusion criteria in the configuration window.

After setting the inclusion criteria, initial model was built by pressing "Save and Build Model" button. As we can see from Figure 6 (b), the initial model without any constructed features is built with an AUC of 69.7%. This model is further analyzed using refinement tab where first event identified to show high correlation with the label event is selected (Asthma → Discharge). This relation is used for constructing a new feature and a new model is built (model 2). Since this model did not show much improvement over the initial model, further explorations are performed and a new event sequence is identified (x-ray → x-ray → hospital admission → routine procedures → discharge) as shown in Figure 6 (c). A child model is built with parent as model 2 to obtain model 3. This model shows a significant improvement of 1.8% over the initial model with AUC 71.5 (Figure 6 (d)).

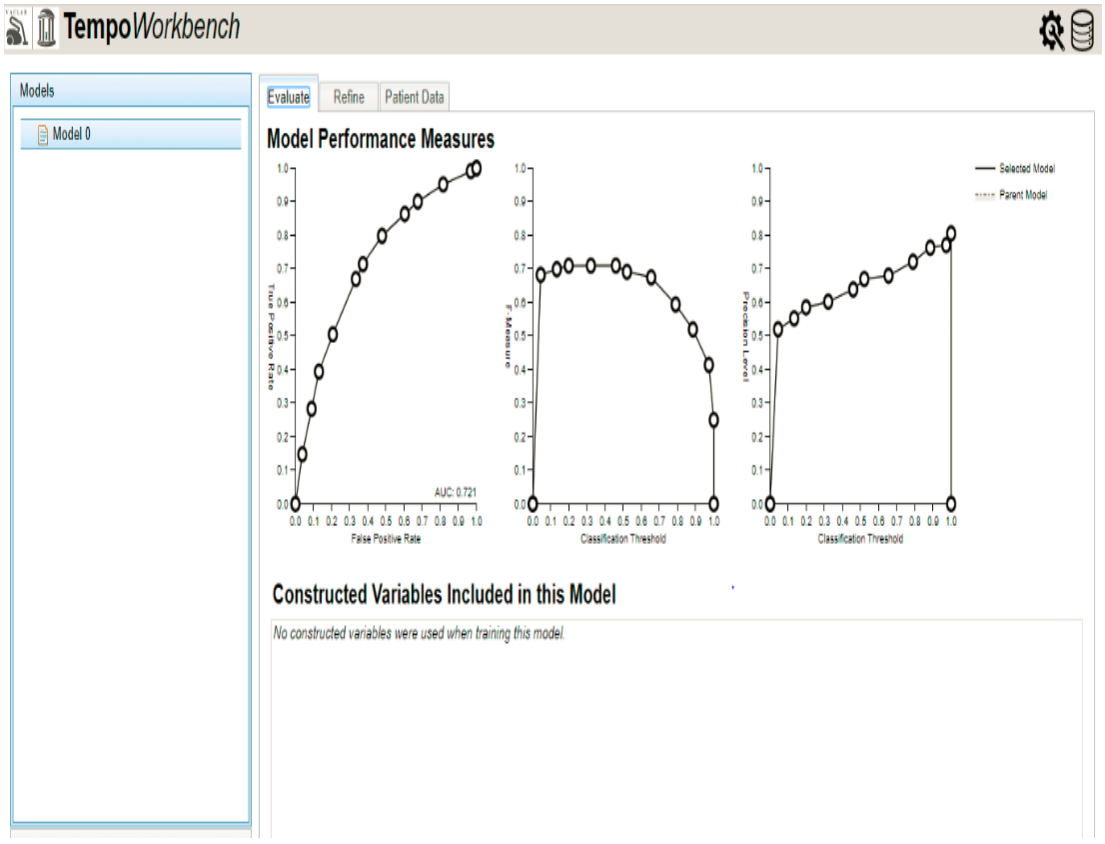


Figure 7 (a): Screen captures of the prototype ITFC system for COPD cohort showing (a) the initial model (Model 1) with 72.1% AUC

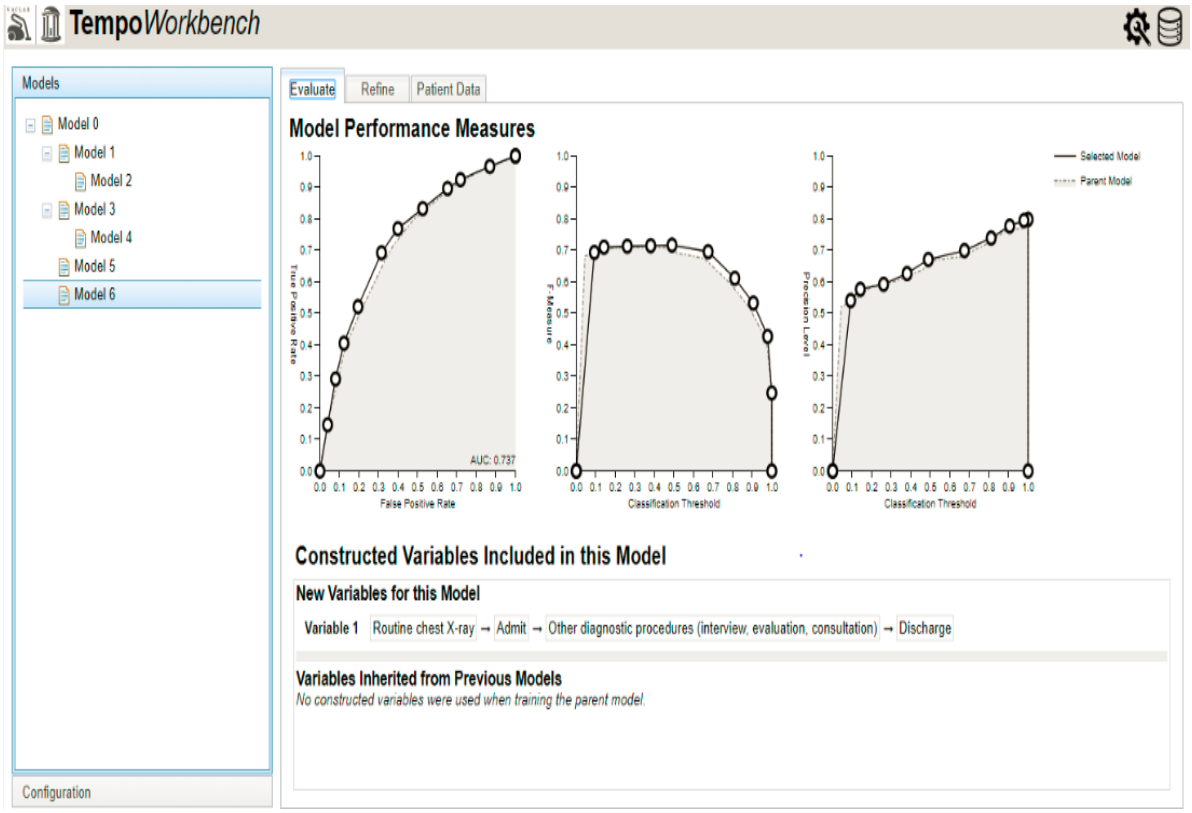


Figure 7 (b): Screen captures of the prototype ITFC system for COPD cohort showing (b) final model (Model 6) obtained after multiple search and explorations with AUC 73.7%, an increase of 1.6%. This also showcase the constructed features used for model building.

Figure 7 shows another experiment with the same COPD dataset. Figure 7 (a) shows the initial model built using a different model configuration, but same sentinel event and prediction label. This time a new data is partitioned into 20% test set, 20% refinement set and 60% training set. The AUC of initial model is 72.1%. After this initial model, there are multiple explorations are performed by forming variety of constructed variables. This gave us six different models with their respective performance measures as can be seen from Figure 7 (b). The final model (sixth model) obtained after these exploratory searches gave us an improvement of 1.6%

and that too by including minimum event sequences in the newly constructed features. It proved that sometimes, combination of few selected features can have significant impact on the overall performance of model. This model, which is formed using the sequence chest x-ray → Admit → other diagnosis → Discharge has proved to show significant improvement over other models which had additional events such as hypertension, or a second chest x-ray.

The inherited pattern variables shown in Figure 6 (d) represent the variables constructed during the refinement of Model 0 to create Model 1. The new variables are those introduced for the first time when creating the current model (Model 2). Finally, the performance measure plots in Figure 7 (b) allow for the comparison of performance between the current model (Model 6) and its parent (Model 0). Here we can see that the addition of the new Variable 1 has produced a bump in the middle of the ROC curve (when false positive rates are between 0.3 and 0.6). The Precision curve also shows improvements to demonstrate higher precision ratio for the newly constructed model. This shows that the new model is indeed is helpful, allowing us predict more accurately some of the harder to classify patients.

Use Case 2: Heart Failure

Similar to the COPD use case, heart failure data cohort was used to test the generalization of ITFC approach to a wider variety of datasets. Same model configuration settings were chosen to predict hospital readmission risk for HF patients at the time they are being discharged from the hospital. The first steps in the

ITFC workflow are dataset selection and the application of inclusion criteria. Dataset for this cohort comprised of 5,804 patients. These patients were selected because they had at least one occurrence of heart failure in their medical records (family of 428. * ICD-9 codes). This dataset was loaded to ITFC and model configurations were set. For this use case the prototype was connected to a dataset with a cohort of 5,804 HF patients. This dataset contains around a million events for these 5,804 patients that includes diagnosis, procedures and hospital encounters. In the model configuration step, we first specified the sentinel event. Hospital discharge was chosen as the sentinel event. Moreover, since the goal was to predict risk of readmission at the time of discharge, the event window was set to only include events prior to the time of discharge. In this case, prior days were set to 365 and days after were set to 0. Admit was selected as the prediction label. The objective was to test whether our model can predict unseen patients with similar medical history accurately for readmission. All these parameters were set and the dataset was partitioned into 50% train, 30% test and 20% refine.

Once the settings were configured, initial model was built by clicking on “Save and Build Model”. This initiated the training process and gave us with the base model. The performance measures were displayed with RIC curve, F-measure and precision curve. Initial model gave an unsatisfied AUC of 65.6% Further explorations were performed using refine tab and an event sequence was identified “chest x-ray → hospital admit → routine procedures → discharge”, very similar to the last use case. This event sequence was used to create a new constructed feature. A refined model

was created and it was seen that this new constructed feature was strongly associated with readmissions to the hospital.

When the new model was trained, the performance results were visualized in the Evaluation panel and displayed along with the performance measures from the initial model which allowed direct comparison. The new model, incorporating the new pattern-based variable that we had discovered, boasted a revised AUC of 67.2%, an AUC improvement of roughly 1.8%.

DISCUSSION

This paper has opened new doors to real time user-driven health analytics. As we have seen, this approach has many advantages over the existing solutions available in the market. In this section, we will discuss various benefits of using ITFC and also highlight few weaknesses of this workflow approach.

ITFC highlights strong concepts that lack in the present predictive analytics applications. This workflow approach can significantly assist medical practitioners in taking informed decision that can be trusted. ITFC can be beneficial in multiple ways such as (1) provides a user-driven workflow approach that guides medical practitioners throughout the lifecycle of the model building, (2) users can visually compare different models in real-time to identify the best predictive model, (3) there are multiple visualizations to support users in their search and identification of hidden patterns in huge datasets with thousands of features, (4) flexible and can easily adapt to different domains, (5) realistic and trusted approach where domain knowledge is incorporated in model building and refinement process to get models that can help in real life decision making.

On the other-hand, implementing ITFC can be challenging. ITFC builds an initial model which is identified using machine learning techniques. However, in order to improve the performance of the model, domain experts have to spend time in the identification of sequences of events that may help in improving predictive power of models. Moreover,

this may lead to series of models and make it difficult to analyze models in a single view. This issue can be addressed by devising new visualization approach for model comparison. Another major challenge is the implementation of ITFC workbench. ITFC provides a real-time interactive workflow which requires in-memory data processing and model refinement. This requires some effort in infrastructure setup and technical expertise to build such low latency application. A highly distributed and scalable architecture will be required for such applications. Regardless of all these challenges, ITFC can significantly enhance the model building and refinement process in health care systems.

CONCLUSION

This paper presented a novel approach to automated pattern discover and visual analytics technique. Interactive Temporal Feature Construction (ITFC), supports an interactive and guided approach for users to analyze large amount of complex data. This application has proved to be a productive approach in handle large dataset with temporal relationships. As we have seen, this approach is valuable in health care analysis where there are millions of combinations and relations between different events. ITFC presents a user-in the loop approach for feature construction, model refinement and model evaluation. ITFC enables users to use their domain expertise and refine models to get most efficient and realistic predictions. ITFC provides an interactive application for analyzing and visualizing model predictions that enables informed decision-making process. Performance measure graphs such as ROC, AUC, F-measure and precision curve provides users with visuals to compare between multiple models on a single screen.

ITFC is novel contribution towards the visual analytics community. It has proven to present a different approach towards analyzing large datasets. ITFC shifts the paradigm from traditional way of doing data analysis to an automate user-in-the-loop workflow approach. This workflow allows users to quickly load their data, set inclusion criteria, decide model configuration parameters and perform explorations in

a guided manner using multiple visualization tools. This workflow includes an iterative refinement phrase in which model errors are visualized for exploration, and new features are constructed based on user-identified temporal event patterns that are shown to be associated with the prediction target within the incorrectly predicted data records. To support this workflow, a number of new algorithms and visual representations are proposed.

This research work resulted in a prototype application that implements the ITFC workflow and it was applied to the real-world datasets obtained from UNC health care system. There were three cohorts used for testing the applicability of ITFC workflow. These were heart failure data, diabetes data, and COPD. These use cases demonstrated that using a guided search on complex data, we can identify patterns that may lead us to much improved prediction models. These research results were also shared with IEEE and medical practitioners to get the feedback and asses the usability. This work has shown lot of promise with different users and people understand the importance of research in this area of visual analytics. These initial results are promising, but several challenges remain to be addressed in future research.

Future plans include more comprehensive user evaluations, including both usability experiments as well as longer-term case studies exploring (1) how these tools are used in practice, and (2) how they can be used in combination with other existing predictive modeling tools. Another area of interest for future work is to explore the

integration of other types of constructed features, such as hierarchical aggregation of event types, to help user's additional informative structures that could further improve performance. Finally, we hope to experiment with larger cohort sizes by deploying the Spark-based prototype within a large high-performance compute infrastructure. This aim is made difficult by the protected nature of health data and the need for a correspondingly secure compute platform. However, the use cases in this paper provide preliminary results that encourage the further efforts toward larger-scale experiments.

REFERENCES

Tarko, A., Inerowicz, M., Ramos, J., & Li, W. (2008). Tool with road-level crash prediction for transportation safety planning. *Transportation Research Record: Journal of the Transportation Research Board, (2083)*, 16-25.

Murdoch, T. B., & Detsky, A. S. (2013). The inevitable application of big data to health care. Jama, 309(13), 1352.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research, 3(Mar)*, 1157-1182.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. *In Advances in Neural Information Processing Systems (pp. 2962-2970).*

Malhotra, K., Navathe, S. B., Chau, D. H., Hadjipanayis, C., & Sun, J. (2016). Constraint based temporal event sequence mining for Glioblastoma survival prediction. *Journal of biomedical informatics, 61*, 267-275.

Bates, D. W., Saria, S., Ohno-Machado, L., Shah, A., & Escobar, G. (2014). Big data in health care: using analytics to identify and manage high-risk and high-cost patients. *Health Affairs, 33(7),* 1123-1131.

Quan, H., Sundararajan, V., Halfon, P., Fong, A., Burnand, B., Luthi, J. C., ... & Ghali, W. A. (2005). Coding algorithms for defining comorbidities in ICD-9-CM and ICD-10 administrative data. *Medical care,* 1130-1139.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters, 27(8),* 861-874.

Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare: promise and potential. *Health information science and systems, 2(1),* 3.

Kuhn, M., & Johnson, K. (2013). Applied predictive modeling *(Vol. 810). New York: Springer.*

Graepel, T., Candela, J. Q., Borchert, T., & Herbrich, R. (2010). Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. *In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 13-20).*

Liu, H., & Motoda, H. (Eds.). (1998). Feature extraction, construction and selection: A data mining perspective (Vol. 453). *Springer Science & Business Media*.

Wang, T. D., Plaisant, C., Quinn, A. J., Stanchak, R., Murphy, S., & Shneiderman, B. (2008, April). Aligning temporal data by sentinel events: discovering patterns in electronic health records. *In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 457-466). ACM.*

Wongsuphasawat, K., Guerra Gómez, J. A., Plaisant, C., Wang, T. D., Taieb-Maimon, M., & Shneiderman, B. (2011, May). LifeFlow: visualizing an overview of event sequences. *In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 1747-1756). ACM.*

Wongsuphasawat, K., & Gotz, D. (2012). Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *IEEE Transactions on Visualization and Computer Graphics, 18(12),* 2659-2668.

Gotz, D., & Wongsuphasawat, K. (2012). Interactive intervention analysis. *In AMIA annual symposium proceedings (Vol. 2012, p. 274). American Medical Informatics Association.*

Gotz, D., & Stavropoulos, H. (2014). Decisionflow: Visual analytics for high-dimensional temporal event sequence data. IEEE transactions on visualization and computer graphics, *20(12),* 1783-1792.

Laxman, S., & Sastry, P. S. (2006). A survey of temporal data mining. Sadhana, 31(2), 173-198.

Moskovitch, R., Wang, F., Shahar, Y., & Hripcsak, G. (2016). Temporal data analytics. *Journal of Biomedical Informatics, 62(C),* 276-277.

Malhotra, K., Navathe, S. B., Chau, D. H., Hadjipanayis, C., & Sun, J. (2016). Constraint based temporal event sequence mining for Glioblastoma survival prediction. *Journal of biomedical informatics, 61,* 267-275.

Krause, J., Perer, A., & Ng, K. (2016, May). Interacting with predictions: Visual inspection of black-box machine learning models. *In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (pp. 5686-5697). ACM.*

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters, 27(8),* 861-874.

Ryza, S. (2017). Advanced analytics with spark: patterns for learning from data at scale. *O'Reilly Media, Inc.*.

Bostock, M., Ogievetsky, V., & Heer, J. (2011). D³ data-driven documents. *IEEE transactions on visualization and computer graphics, 17(12),* 2301-2309.

Holzner, S. (2008*). The Dojo Toolkit: Visual Quickstart Guide. Peachpit Press.*

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: Cluster computing with working sets. *HotCloud, 10(10-10),* 95.

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., ... & Xin, D. (2016). Mllib: Machine learning in apache spark. *The Journal of Machine Learning Research, 17(1),* 1235-1241.

Mostafa, J., & Moore, C. (2010). *The North Carolina Translational and Clinical Sciences Institute. Clinical and translational science, 3(3),* 71-72.