5-2015

A Computational, Topological Approach to ICU Mortality Rate Prediction with Data Relationship Realization

Adam Michael Aaron
University of Tennessee - Knoxville, aaaron2@vols.utk.edu

Recommended Citation

Aaron, Adam Michael, "A Computational, Topological Approach to ICU Mortality Rate Prediction with Data Relationship Realization. " Master's Thesis, University of Tennessee, 2015.
https://trace.tennessee.edu/utk_gradthes/3341

To the Graduate Council:

I am submitting herewith a thesis written by Adam Michael Aaron entitled "A Computational, Topological Approach to ICU Mortality Rate Prediction with Data Relationship Realization." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mechanical Engineering.

Xiaopeng Zhao, Major Professor

We have read this thesis and recommend its acceptance:

Fernando Schwartz, Jindong Tan, Venugopal Varma

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

A Computational, Topological Approach to ICU Mortality Rate Prediction with Data Relationship Realization

A Thesis Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Adam Michael Aaron
May 2015

DEDICATION

I dedicate this thesis to my wife, Brandi. Without her constant support and encouragement, completion of this thesis would not have been possible.

ACKNOWLEDGEMENTS

I would like to begin by thanking Dr. Xiaopeng Zhao, my advisor for all of his input and guidance through this process. I would like to thank Dr. Fernando Schwartz, who spent many hours teaching this mechanical engineer topological geometry. I would also like to acknowledge Dr. Venugopal Varma and Dr. Mark Noakes for letting me bounce all my ideas off of them and providing valuable feedback.

I would like to thank Robert Kopp, BSN,RN, for his insights on the relationships between the various physiological parameters.

I would lastly like to thank Dr. Xiaopeng Zhao, Dr. Fernando Schwartz, Dr. Venugopal Varma, and Dr. Jindong Tan for serving on my review committee.

ABSTRACT

The objective of this work is to predict the mortality of intensive care unit patients based on their physiological data and understand the relationships between physiological data. Such a model may be used to prioritize care when resources are limited or identify patients that will need significant care in the immediate future. This effort will take a novel approach applying computational topological analysis to classify patients. The algorithm predicting the patient outcomes is trained using an evolutionary algorithm. The dataset used is from the 2012 PhysioNet Computing in Cardiology Challenge. A set containing 4000 records with outcomes was used to train and test the prediction algorithm. The topology extraction algorithm, Mapper, was used to represent the high dimensional data as a 1-D graph of the set topology using a filter. The filter is trained using an evolutionary algorithm to maximize the positive prediction rate and sensitivity. The Event 1 score is the minimum of these two. This algorithm yielded an Event 1 score of 0.42 out of 1.00 for the PhysioNet Challenge. This is comparable to a currently used ICU classification system, SAPS-1 that achieved an event 1 score of 0.30.

Additional developments from this work include an optimized Mapper clustering function that runs in 120 seconds for the complete data set compared to the 2.2 month estimate using the original function. This allowed the rapid iteration needed for optimization in this algorithm. The algorithm developed in this thesis could be more generally applied to analysis and prediction in any feature space for generic problems.

TABLE OF CONTENTS

CHAPTER I Introduction 1
CHAPTER II Literature Review 3
Current Solutions to the 2012 Physionet Challenge 3
Literature on Tools Applied to Solution...11
CHAPTER III Data 13
Data...13
Descriptors...13
Time Series Data 14
Outcomes 15
Outcome Data Analysis 17
Feature Data Analysis 18
CHAPTER IV Method 21
Data Import...24
Feature Extraction 25
Extracted Features 26
Feature Extraction Functions 28
Created Features...28
Calculate BMI 28
Dealing with Missing and Erroneous Values 32
Missing Data 34
Removal of Erroneous Data..34
Replacement...35
Normalize Features 35
Calculate Patient Distance Matrix...35
Generate Filter Function...38
Extract Topology 40
Mapper Inputs...40
Mapper Outputs 41
Post-Processing Topology 43
Calculate inputs for Graphviz...43
Generate Graphviz Input 43
Graphical Representation of Mapper Outputs...44
Assigning Mortality Rates to Cluster Nodes 44
Predict Patient Mortality Based on Clusters 46
Calculate Filter Performance..46
Breed New Filters based on Performance...49
Training of Mapper Filter Function using Evolution Algorithm 49
Optimization Configuration..50
Filter Mask 50
Mapper Configuration 52
Application of Result to Test Patient...52
CHAPTER V Results 54
Interpreting Algorithm Results 54
Performance of In-Hospital Death Optimized Filter 55

Performance of Death14 Optimized Filter 61
Revised Mortality Prediction Algorithm..67
CHAPTER VI Discussion 68
Validity of Model 69
Improvements in Mapper Efficiency 69
CHAPTER VII Conclusions...72
CHAPTER VIII Recommendations...73
Develop Better Prediction Algorithm Using Current Mapper Output 73
Optimize Mapper Parameters to Increase Performance...............................73
Solve for All Cases in Current Space Using Super Computing....................74
Allow for the Application of Weights to Filters...74
Use mortality predictions from multiple filter functions.................................74
LIST OF REFERENCES 75
APPENDIX...79
Appendix A: Modifications to Main Mapper Function......................................80
Vita...82

LIST OF TABLES

Table 1. Summary of literature results for the Event 1 scores 4
Table 2. Feature extraction and creation descriptions 29
Table 3. BMI Class Limits 31
Table 4. Male and female survivability rates based on individual's age..............31
Table 5. Statistics for mortality after leaving the ICU compared to in-hospital death...33
Table 6. Statistics on each feature...33
Table 7. Inputs for the function Mapper. 36
Table 8. List of bits in filter and mask strings with corresponding filters..............39
Table 9. Outputs for the function Mapper..42
Table 10. Outcomes matrix definitions..48
Table 11. Filter mask used to minimize evolution algorithm search space. 51
Table 12. Mapper settings used in evolution algorithm.......................................53
Table 13. Statistics for the optimized Mapper output 56
Table 14. Statistics for topology nodes 57
Table 15. Statistics for the Death 14 Optimized Mapper Output.........................62
Table 16. Statistics for topology nodes for Death14..63

LIST OF FIGURES

Figure 1. Patient deaths based on days survived after admission to the ICU for patients experiencing in-hospital death..16
Figure 2. Patient deaths based on days survived after admission to the ICU for patients experiencing out-of-hospital death. 16
Figure 3. Inter-feature relationships discovered in the data set 20
Figure 4. Feature to mortality relationships found in the data set. Deceased records are selected in red. 20
Figure 5. Process Diagram for ICU Mortality Mapper Prediction Algorithm 22
Figure 6. Method used to predict patient mortality once an optimal solution is found...23
Figure 7. Sample Mapper output after Graphviz Processing 45
Figure 8. Mapper Cluster with 30% Event 1 score for the 2012 Physionet challenge. 56
Figure 9. Data relationships with true positives highlighted in red 59
Figure 10. Distributions for full data set (dark bars) and true positives (light blue bars) 59
Figure 11. Distributions for full data set (dark bars) and mortalities (light blue bars) 59
Figure 12. Correlation matrix for true positives only..60
Figure 13. Correlation matrix for all data points 60
Figure 14. Mapper Cluster for Death14 with 26% Event 1 score for the 2012 Physionet challenge...62
Figure 15. Data relationships with true positives highlighted in red for Death14.65
Figure 16. Distributions for full data set (dark bars) and true positives (light blue bars) 65
Figure 17. Distributions for full data set (dark bars) and mortalities (light blue bars) 65
Figure 18. Correlation matrix for true positives only..66
Figure 19 Correlation matrix for all data points 66
Figure 20. Comparison of Mapper function to revised Mapper function..............71

ACRONYMS

APACHE	Acute Physiology and Chronic Health Evaluation
ALP	Alkaline Phosphatase
ALT	Alanine Transaminase
AST	Aspartate Transaminase
BIL	Bilirubin
BMI	Body Mass Index
BUN	Blood Urea Nitrogen
CRE	Serum creatinine
d	A g x g distance matrix
DiasABP	Invasive Diastolic Arterial Blood Pressure
EDEN	Exploratory Data analysis Environment
F	Feature Vector
FiO2	Fractional Inspired O_2
g	number of patients, 4000
GA	Genetic Algorithm
GCS	Glasgow Coma Score
h	number of features, 26
HCO3	Serum Bicarbonate
HCT	Hematocrit
HR	Heart Rate
ICU	Intensive Care Unit
K	Serum Potassium
MAP	Invasive Mean Arterial Blood Pressure
MechVent	Mechanical Ventilator
Mg	Serum Magnesium
MPM	Mortality Probability Model
Na	Serum Sodium
NIDiasABP	Non-Invasive Diastolic Arterial Blood Pressure
NIMAP	Non-Invasive Mean Arterial Blood Pressure
NISysABP	Non-Invasive Systolic Arterial Blood Pressure
PaCO2	Partial Pressure of Arterial CO_2
PaO2	Partial Pressure of Arterial O_2
pH	Arterial pH
PropT	Troponin-T
RespRate	Respiration Rate
SaO2	O_2 Saturation in Hemoglobin
SAPS	Simplified Acute Physiology Score
SOFA	Sequential Organ Failure Assessment
SysABP	Invasive Systolic Arterial Blood Pressure
Temp	Temperature
TropI	Troponin-I
V	Patient Vector
WBC	Urine Output
WHO	World Health Organization

CHAPTER I
INTRODUCTION

The intensive care unit (ICU) is the area of the hospital where patients in the most critical of conditions are cared for. During care, the hospital tracks many physiological signals and logs them in order to monitor and evaluate the condition of the patient. The precise signals collected may vary by ICU facility and condition of the patient. This data may be mined and analyzed in an attempt to predict patient condition [32].

There are already systems in place to evaluate the condition of a patient based on this information collected. These include: the mortality probability model (MPM), the acute physiology and chronic health evaluation system (APACHE II), and the simplified acute physiology score (SAPS II). MPM is used upon admission and every 24 hours for the first three days to predict mortality of a given patient [14]. APACHE II is used to evaluate the condition of an adult upon admission to the ICU. APACHE II is also used to help medical care professionals select appropriate medications based on the patient condition. This model is based on comparing the mortality rate of a given patient against data from others with similar conditions [8]. Like APACHE II, SAPS II is a score calculated only at the time of admission that describes the likely mortality rate for the patient [7].

The development of a more accurate system for the prediction of mortality rates is desired. A more accurate system could improve the quality of patient care, reduce medical costs [4], and improve the allocation of resources during catastrophic events. Identifying relationships between variables may reduce the problem space for such an algorithm or improve early detection of conditions.

The research conducted herein proposes a new method that exploits the topology of the feature space for a set of patient records in an attempt to classify and predict patient mortality. The algorithm was trained using an evolutionary optimization of the Mapper filter function. The results are comparable to the result for a currently implemented scoring system, SAPS-1, on the same set of data and reveals relationships between physiological features.

The system presented was optimized using one set of criteria and later tested against another set of criteria. The performance improved by 12% on the final model to a final Event 1 score of 0.42 on Set A.

CHAPTER II
LITERATURE REVIEW

In 2012, PhysioNet published a challenge to improve the current prediction of mortality rates in the ICU based on the first 48 hours of physiological data from the patients. The challenge was broken into multiple events, the first of which will be the focus of this thesis. The first event from the challenge scored algorithms based on their ability to positively identify in-ICU mortalities and not misclassify the patients expected to survive. All scores presented are out of a maximum score of 1.00. Random chance will yield a score of 0.14 [5].

Current Solutions to the 2012 Physionet Challenge

The following papers reviewed have presented different solutions to the 2012 Physionet Challenge. This work hopes to build on the published work by increasing the realization of the data and discovering relationships between the data features not previously exploited. In total, 17 papers with submissions to the 2012 Physionet challenge were reviewed. A synopsis of each method, along with the resulting score for Event 1 is included in the discussion. A summary of the results for the Event 1 scores across all of the literature are shown in Table 1. When possible, the Event 1 score for Set A is shown. This is the data set used in the research conducted in this thesis. When not available, the set resulting in the greatest Event 1 score was selected as indicated in the table.

I Silva presented the baseline result for the Physionet challenge using the SAPS-I system. The SAPS-I system is the Simplified Acute Physiology Score. It is a rating system from 0 to 32 in the provided data set that acts as a predictor of

Table 1. Summary of literature results for the Event 1 scores

First Author	Method	Event 1 Score	Set
Johnson, AEW	Bayesian Ensemble	0.54	Set C
McMillan, S	Time Series Motifs	0.50	Set B
Bosnjak, A	Statistics of Physiological Variables and Support Vector Machines	0.30	Set B
Severeyn, E	Simple Correspondence Analysis Approach	0.44	Set A
Marco, LYD	Classification using Variable Distributions	0.55	Set A
Silva, I	SAPS-I	0.31	Set A
Xia, H	Neural Network Model	0.50	Set A
Krajnak, M	Machine Learning with Clinical Rules	0.39	Set A
Hamilton, SL	Logistic Regression	0.57	Set A
Bera, D	Logistic Regression	0.44	Set A
Vairavan, S	Logistic Regression and a Hidden Markov Model	0.52	Set A
Macas, M	Linear Bayes Classification	0.48	Set B
Lee, CH	Imputation-Enhanced Algorithm	0.47	Set A
Yi, C	Histogram Analysis of Medical Variables under Cascaded Adaboost Model	0.38	Set B
Xu, J	Cluster Analysis of Multi-granular Time Series Data	0.23	Set A
Citi, L	Cascaded SCM-GLM Paradigm	0.53	Set B
Pollard, TJ	Artificial Neural Network with Application of Solar Physics Analysis Method	0.38	Set B

mortality. The value is calculated at the time of admission and may not be recalculated throughout the stay. The SAPS-I score was calculated for all of the patients. Based on the prediction of this system, an Event 1 score of 0.31 was achieved [32]. The features extracted in this method could be further used by other algorithms to more accurately predict mortality.

A Bayesian Ensemble method was used by AWE Johnson. As a deep learning method, a set of weak learning algorithms were used in conjunction with one another and trained to be skewed and make decisions based on the observations of the physiological data and the outcomes. Each of the weak learners is a tree that selects a set of physiological data and predicts the outcome based on them. The physiological data selected is based on a Markov chain Monte Carlo sampler. The model used consisted of 500 nodes with depth-2 (depth two meaning two possible outcomes for each). The model was trained using one set and had a success rate on the Event 1 scores for the other two sets of 0.53 and 0.54 [17]. One of the weaknesses of this method is the lack of transparency in how the algorithm is actually making these choices.

Searching for time series motifs was the approach taken by S. McMillan. A time series motif is a short pattern in a time series that, if consistent across multiple records, may be used to predict the patient outcome. All of the time series data is subdivided into bins and assigned a local value of low, medium, or high. This turns every signal into a string of low/medium/high values that can be searched for motifs based on patient outcome. This method achieved an Event 1 score of 0.50 on set B [20]. Features from such a method can be extracted and explored by another method. Unlike other methods discussed, this method can be run on the patient’s physiological data at any point in time using the most recent windows of available data.

A method using statistics of physiological variables incorporated with support vector machines was used by A. Bosnjak. The initial method used the mean and standard deviation for each of the SAPS-I parameters to train the support vector machines to predict mortality rate based off the Set A patient data with outcomes. Additional features were added based on the inputs from physicians. The result was a SVM that was able to achieve an Event 1 score of 0.30 on set B [27].

A simple correspondence analysis approach was taken by E. Severeyn. In this approach the APACHE II, SAPS II, and SOFA scores were used in conjunction with other physiological dada to predict mortality. The primary variables contributing to the simple correspondence analysis are ALP, BIL, BUN, and CRE. The Event 1 score achieved by this method was 0.42 on set A [24].

L.Y.D. Marco presented a logistic regression model using the mean of all features to predict patient mortality. The patients were sorted by mortality and statistics on the mean for each variable. Distributions were created for each variable in each condition. Based on this distribution, predictions were made using a variety of classifiers and tuned to maximize the Event 1 score. The Event 1 score achieved by this method was 0.55 [26].

A neural-network based system developed by H. Xia was used to predict patient mortality rates. The neural network identified 26 features that best predicted the patient mortality and used them to train the neural network to classify the patients. For our system, we will be using the same 26 features used to train the neural network. These 26 features were chosen based on their correlations to mortality as found by H. Xia. This system achieved an Event 1 score of 0.51 [16].

M. Krajnak implemented a fuzzy, clinical rule based system that both takes advantage of machine learning and is also able to explain the significance of the features. This is one of the advantages this system has over neural networks. The initial system consisted of 45 rules across 15 features. The weights of the rules were optimized to achieve an Event 1 score of 0.39 [23].

S.L. Hamilton used logistic regression to predict the mortality rates for the data set. Features for the linear regression included the first and last values, the average, minimum and maximum, and the difference rate between the first and last values. These features were extracted for every one of the 37 time series parameters. Missing data is recovered using the mean for the set of patients experiencing either experiencing in-hospital death or surviving, depending on the outcome of the patient missing the data. The regression performed yielded an Event 1 score of 0.57 [29].

D. Bera also used logistic regression method like S.L.Mamilton. 88 features were extracted from 30 of the patient variables including the minimum, mean, and maximum for each variable. The variable was selected based on availability. Performing the linear regression with the specified set of features yielded an event 1 score of 0.44 [30].

S. Vairavan used a logistic regression model too, with the addition of a hidden Markov model. For the Event 1 problem, 10 features were extracted to predict mortality. The most novel feature compared to other methods presented is the Hidden Markov Model based mortality predictor. Using the time series data, the Markov chain computes the transition between states for each patient. These states are a sequence of “Alive” and “Dead” calculated based on the time series data. These sequences can be calculated for each patient and contribute to the prediction of mortality. This method achieved an Event 1 score of 0.50 [21].

M. Macas used a Linear Bayes Classification method to classify the records and predict mortality using selected features with the Social Impact Theory based Optimizer. Multiple outputs from already established scoring systems were used by this classifier, along with a wide array of features based on the patient variables. In total, 935 features were extracted. Outliers were not removed, but features missing significant amounts of data were not considered. This system achieved an Event 1 score with Set B of 0.48 [25].

C.H. Lee implemented an imputation-enhanced algorithm to predict ICU mortality rates. Features were extracted from the patient data for the first 48 hours and for the last hour specifically. Missing data was recovered based on the mean for individual’s age and gender demographic. Primary features were the last measurement for each variable and standard statistics for each 12-hour bin for each variable including minimum, maximum, mean, standard deviation, and number of observations. The highest Event 1 score achieved for set A was 0.47. A significant finding from this paper is that the prediction based on the last value features outperformed the models that included the minimum and maximum features, showing that additional features do not always yield a more accurate result [18].

C. Yi utilized histogram analysis of the medical variables under a Cascaded Adaboost learning model. All time series data was interpolated across the entire 48 hour domain to establish consistent time across all records with one reading per minute. Then, the mean value is calculated over every 60 minute interval for each patient. This creates a feature vector for each patient in 1776 dimensions. Histograms are created based on each feature for each hour per patient condition. This generates the differentiation metric for the classification algorithm. The Cascaded Adaboost model is then implemented to assign patient mortality rates based on this model. This method achieved and Event 1 score of 0.81. However, the system trained using set a was not well generalized to the classification of set B. Run on set B, the algorithm only yielded an Event 1 score of 0.38. [22] This emphasizes the important of a model having the ability to be generalized.

J. Xu used a cluster analysis of multi-granular time series data. The features used were the minima and maxima from 16 segments of time series data over a 48 hour period. 10 variables were selected to analyze, resulting in 20 features per patient. The Event 1 score for this clustering method using set a was 0.23. [31]

L. Citi used a cascaded SCM-GLM Paradigm to predict the ICU mortality rates. This machine learning algorithm used only set A for training. The data for all variables was split into two 24 hours periods for feature extraction. For each of these periods, the minimum, mean and maximum values were computed. Missing values were replaced with the mean of the data for that feature Support vector machines were trained based off these features to predict the outcomes for set A. This algorithm achieved an Event 1 score of 0.53.[19]

T.J. Pollard trained a neural network to make the prediction with the application of the Solar Physics Analysis Method. Outliers were removed from time series data and patients were sorted based on ICU type prior to neural network training. Features included mean, variance in time series data, and the moments of the gradient in the time series values. Experience in the prediction of solar flare research was exploited to detect perturbations in local minima as a feature. A three layer neural network was used. The algorithm achieved an Event 1 score of 0.27 using set B [28].

Fundamental differences in the approaches include the transparency of the model, whether the model is applied only at the time of admission, or if it may be used for real-time prediction, and the number of features chosen to extract from the data set. The vast majority of the models outperformed the baseline SAPS-I method on Event 1 with scores as high as 0.85. The method implemented in this thesis will attempt to perform as well as the systems presented here and reveal as yet identified relationships between data features. Lessons learned from the literature review include:

1. More features will not always yield a more accurate result
2. It is important to recover missing data and the most common method uses the mean value for the entire population

The primary 26 features presented by H. Xia are used in this research as an extensive feature space leads to extended computation times, inhibiting optimization of the algorithm. Common methods across much of the literature for cleaning and filtering the data before analysis are implemented as well.

Literature on Tools Applied to Solution

Two data analysis tools, one developed by G. Singh and one by C. Steed, will be used to analyze the ICU data set. The topology extraction tool Mapper will be used to create clusters of data in the features space of the records. EDEN will be used to realize relationships in the data prior to analysis and realize relationships in the data after the analysis, with the predicted mortality rates.

A new approach for evaluating high dimensional point cloud data was developed by G. Singh [13] named Mapper. Mapper takes a distance matrix containing the distance measurement between all vectors for a high dimensional point cloud and represents it as a 1-D graph using a filter function to group vectors. The benefits of this approach for clustering are that the number of clusters does not need to be pre-determined, and each cluster is broken into nodes whose shape within the cluster describes the topology of the point cloud. It is a way to take otherwise difficult to visualize high dimension data and view it in a meaningful way that preserves relationships in the data. This approach will be used to analyze the 26-Dimension feature space created from the 26 features identified by H. Xia. A critical part of the mapper function is the filter function, which determines the grouping of the cluster output. A genetic, or evolution, algorithm [15] is used to optimize the filter and find the result yielding the highest Physionet 2012 Challenge Event 1 Score.

EDEN (Exploratory Data analysis Environment) is a powerful tool developed by C. Steed at Oak Ridge National Laboratory that allows for high-dimension data visualization [2]. Large data sets may be loaded into the environment and analyzed using various statistic and visual indicators. It was used to analyze the initial patient data and the cluster/node relationships in the Mapper output.

CHAPTER III
DATA

The following sections describe the records and outcome data available for the 4000 patients provided by PhysioNet with outcomes and the methods used to extract the features from the data. This section will also explain how missing features were addressed. These 4000 data sets represent Set A of the PhysioNet challenge. In total, data for 12000 patients was collected; however sets B and C were blind sets. Outcome data was not available for these data sets and therefore could not be used to evaluate this algorithm. All of the work performed here was performed using Set A.

Data

Three types of data were available: Descriptors, Time Series Data, and Outcomes. Descriptors are simply items describing the patient. The time series data contains physiological information about the patients collected by the ICU. The outcome data is data that was known after the patient's stay at the ICU.

Descriptors

The following are descriptors available for each record:

- RecordID
- Age
- Gender
- Height
- ICUType
- Weight

The Record ID is a unique identifier assigned to each patient record that may be used to track the patient's record and features throughout the process for quality assurance purposes. The Age, Gender, and Height of the patient are single descriptive measurements describing the patient at the time of admittance.

The ICU Type is broken into four categories: Coronary Care Unit, Cardiac Surgery Recovery Unit, Medical ICU, or Surgical ICU. Weight is the patient's weight. It is important to note that for some patients, the weight is not recorded as a descriptor, but as time series data to reflect changes in weight over the length of the stay.

Time Series Data

The following types of Time Series Data are available for each record:

- ALP
- ALT
- AST
- Bilirubin
- BUN
- Cholesterol
- Creatinine
- DiasABP
- FiO2
- GCS
- Glucose
- HCO3
- HCT
- HR
- K
- Lactate
- Mg
- MAP
- MechVent
- Na
- NIDiasABP
- NIMAP
- NISysABP
- PaCO2
- PaO2
- pH
- Platelets
- RespRate
- SaO2
- SysABP
- Temp
- TropI
- PropT
- Urine
- WBC

Each of these fields corresponds to a specific physiological measurement taken from the patient. For more information on the specifics of each of these measurements, please see the 2012 PhysioNet Challenge webpage [5]. The specific details of these signals are not required for this analysis as the desired features are being extracted from them. Outliers from the features are removed during pre-processing, while Missing data is reconstructed.

Outcomes

The following descriptors are available for the patients in the data set:

- RecordID
- Length_of_stay
- Survival
- In-hospital_death

RecordID is used to pair the outcomes with the patient records and track patient records through the algorithm. The Length_of_stay is the period of time spent in the ICU by the patient. Most patients with in-hospital death died within the first three weeks as shown in Figure 1. Survival is the number of days the individual survived after admission to the ICU. In-hospital death is a binary identifier as to whether or not an individual passed away while in the ICU.

In-hospital_death does not account for patients who were recently release and passed away at home; it only accounts for patients who died while still in the hospital. Out-of-hospital deaths are shown in Figure 2. This data follows a similar trend to the data from Figure 1 though it is less obvious. It is possible that a patient’s physiological data indicated what would be an in-hospital death, however they were released from the ICU and passed shortly after leaving the hospital. This would cause the algorithm to classify the patient as a mortality. Based on this discrepancy, the Event 1 score would consider this a misclassification, reducing the algorithm performance. To correct this, Length of Stay and Survival are used to create a mortality outcome DeathX (where X is the number of days after release from the ICU). For example, using Death14, an additional 91 patients (16% increase) were found to pass away within 14 days of leaving the ICU, increasing the deceased count from 554 to 645. Assuming they

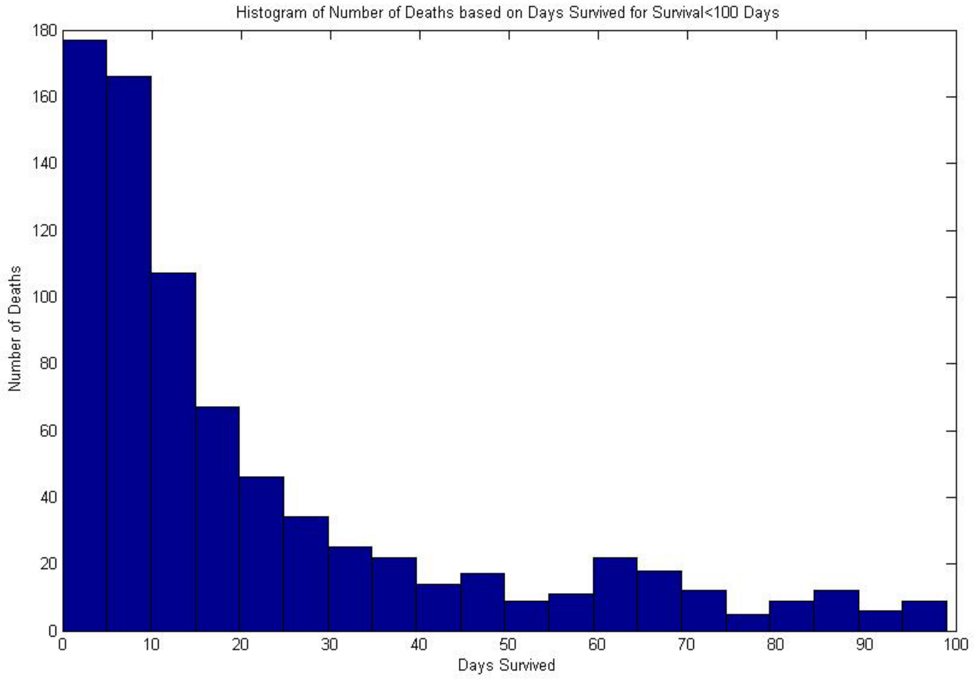


Figure 1. Patient deaths based on days survived after admission to the ICU for patients experiencing in-hospital death

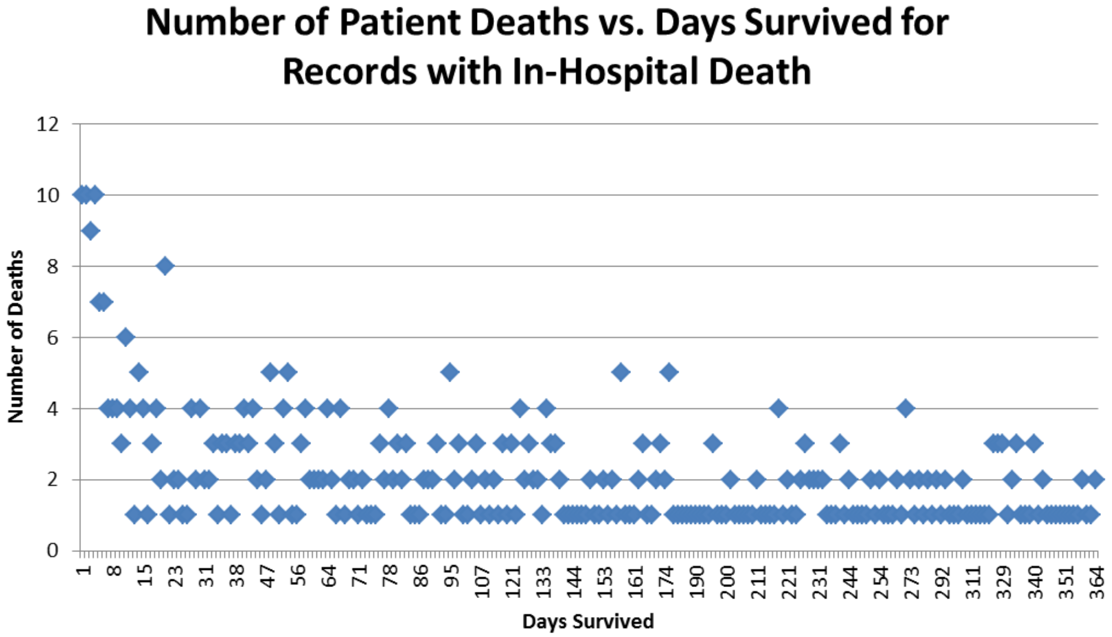


Figure 2. Patient deaths based on days survived after admission to the ICU for patients experiencing out-of-hospital death.

were misclassified by the algorithm as deceased (false positive), this could result in an increase the in Event 1 score of 16% when applied to actual near-term mortalities and not in-hospital death.

Outcome Data Analysis

This data set consisted of 48 hours of patient data. From this, we attempt to predict in-hospital death of the patients. The in-hospital death could be within days or years of admission depending on the medical condition. It is unreasonable to predict mortalities for patients one year after admission with only 48 hours of data, however predicting mortality within a number of weeks based on this data is far more reasonable. In this section we will analyze and discuss the validity of this challenge using this data set.

One piece of the outcome data is Survival. This number is the number of days an individual survived after being admitted to the ICU. Integer values are given for 1473 (37%) of the patients. All of the in-hospital death records are accounted for in this given population (554 patients). The remaining 919 of these patients who did not experience an in-hospital death experienced an out of hospital death. Plots showing the number of patient deaths vs days survived are shown in Figure 1 and Figure 2. In the cases of in-hospital death, the mortality rate for the first two weeks declines steadily and becomes steady around the three week mark. For the patients who did not die in the hospital, there is a slight trend to indicate greater mortality rates closer to their hospital release date with a decline in mortality rates as the time out of the hospital increases. There are local spikes in the number of out of hospital deaths but there is no correlation between these spikes and the number of days survived out of the hospital.

Feature Data Analysis

The desired features are extracted from the descriptors and time series data and analyzed with EDEN. EDEN is effective at finding correlations within the data and changes in the correlation within data subsets. Many correlations were found between features derived from the same sets of time series data (for example, mean and max GCS had a strong correlation), however the correlations of interest are shown in Figure 3. Within the data provided, a correlation was found between the minima of HCO_3 and BUN, the mean of Na to the minimum of SysABPNISysABP, and the last values of BUN and HCO_3.

The correlation between HCO_3 and BUN may be explained by the regulation of acidity in the body. The NH_4 + HCO_3 buffering process or the NH_3 + CO_2 + H_2O process will yield urea, leading to an increase in the blood urea nitrogen (BUN) [11]. This could explain how the minimum and last HCO_3 value could relate to the minimum BUN value.

The relationship of the mean blood sodium to the minimum blood pressure is a known relationship. Higher sodium intake results in higher blood sodium, and as a result, higher blood pressure [1]. As a result, changes in blood sodium are expected to correlate to the blood pressure value.

Some correlations were found in in-hospital mortality. These are shown in Figure 4. The figures show a relationship between in-hospital death and Urine Sum, GCS Trend, BUN Last, HCO3 Last, and WBCLast. The decrease in the urine output could indicate kidney failure. The decrease in the GCS trend could indicate a worsening in condition over the first 48 hours.

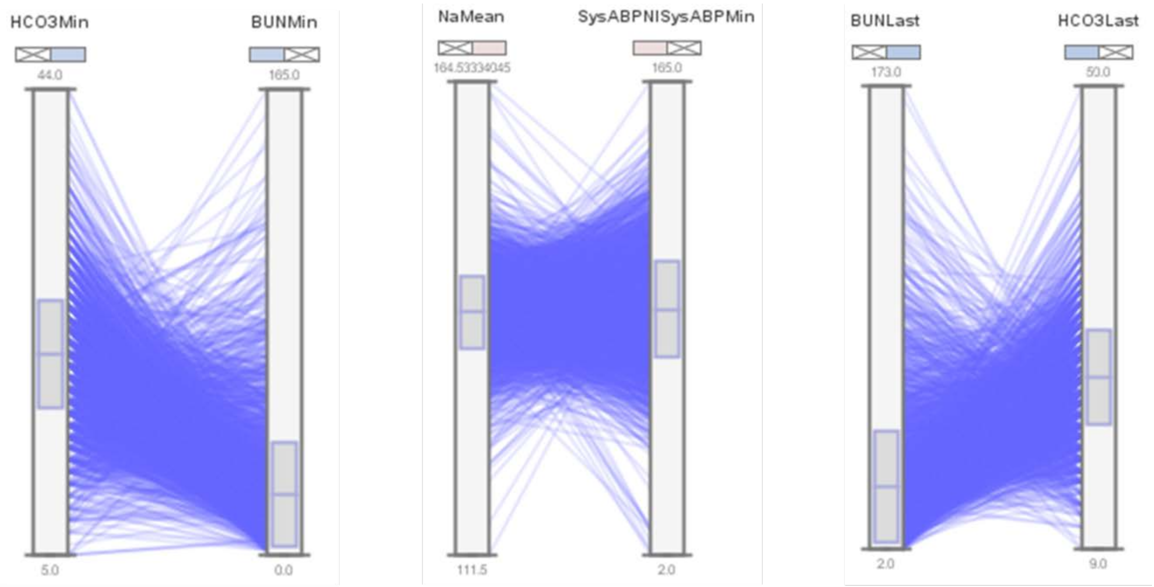


Figure 3. Inter-feature relationships discovered in the data set

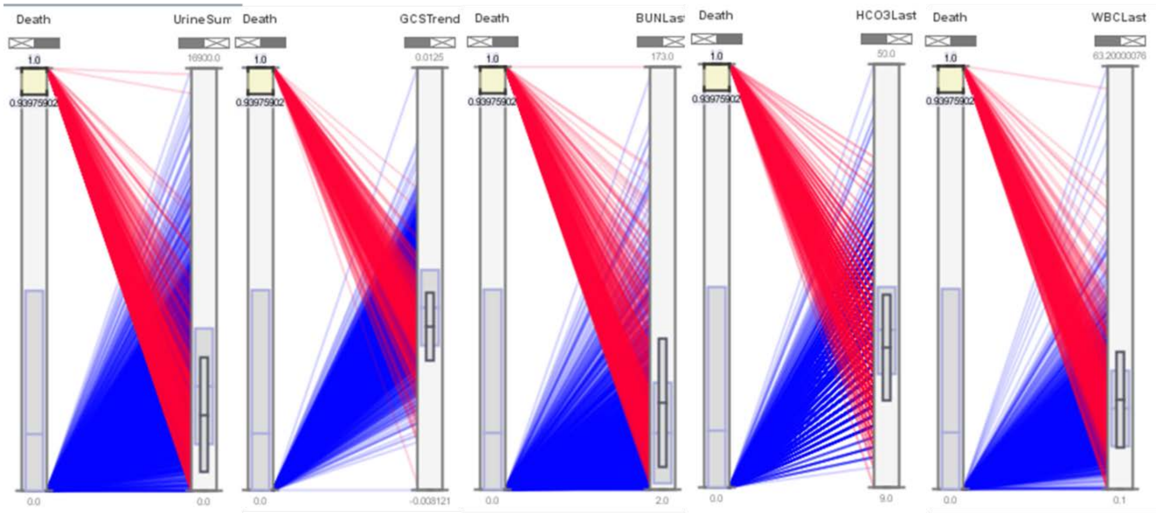


Figure 4. Feature to mortality relationships found in the data set. Deceased records are selected in red.

CHAPTER IV
METHOD

The objective of this project was to improve on the current prediction algorithms used to predict mortality rates in the ICU and understand the relationships between measured data and the outcomes. Relevant data features were extracted from 4000 records. After extraction, any missing values are handled by recovering the missing values. This process yields the Patient Feature Structure. The vectors in the patient feature structure are normalized to prevent feature bias then used to compute the distance matrix required by for computing the topology. The Patient Feature Structure is used to develop the filters that Mapper uses to create the 1D representation of the data. Mapper is run on the Patient Distance Matrix using a composite filter from the Patient Feature Structure in an iterative, evolutionary loop. The filter training process is shown in Figure 5 and the outcome prediction algorithm for a new patient is shown in Figure 6

The feature space is clustered by Mapper. The mortality rate for each node is calculated based on in-hospital death. A prediction algorithm is used to attempt to reproduce the outcomes of the training set. A score is calculated based on the algorithm performance. Well-performing filters move on to the next generation and are bred with the remaining filters in that generation. New filters are added to the population to maintain diversity. This process continues until a 100% prediction rate or a designated number of loops is reached. The Optimized Filter and Mapper Settings are stored for use in a later prediction algorithm for a patient not among this population. This process is shown in Figure 5.

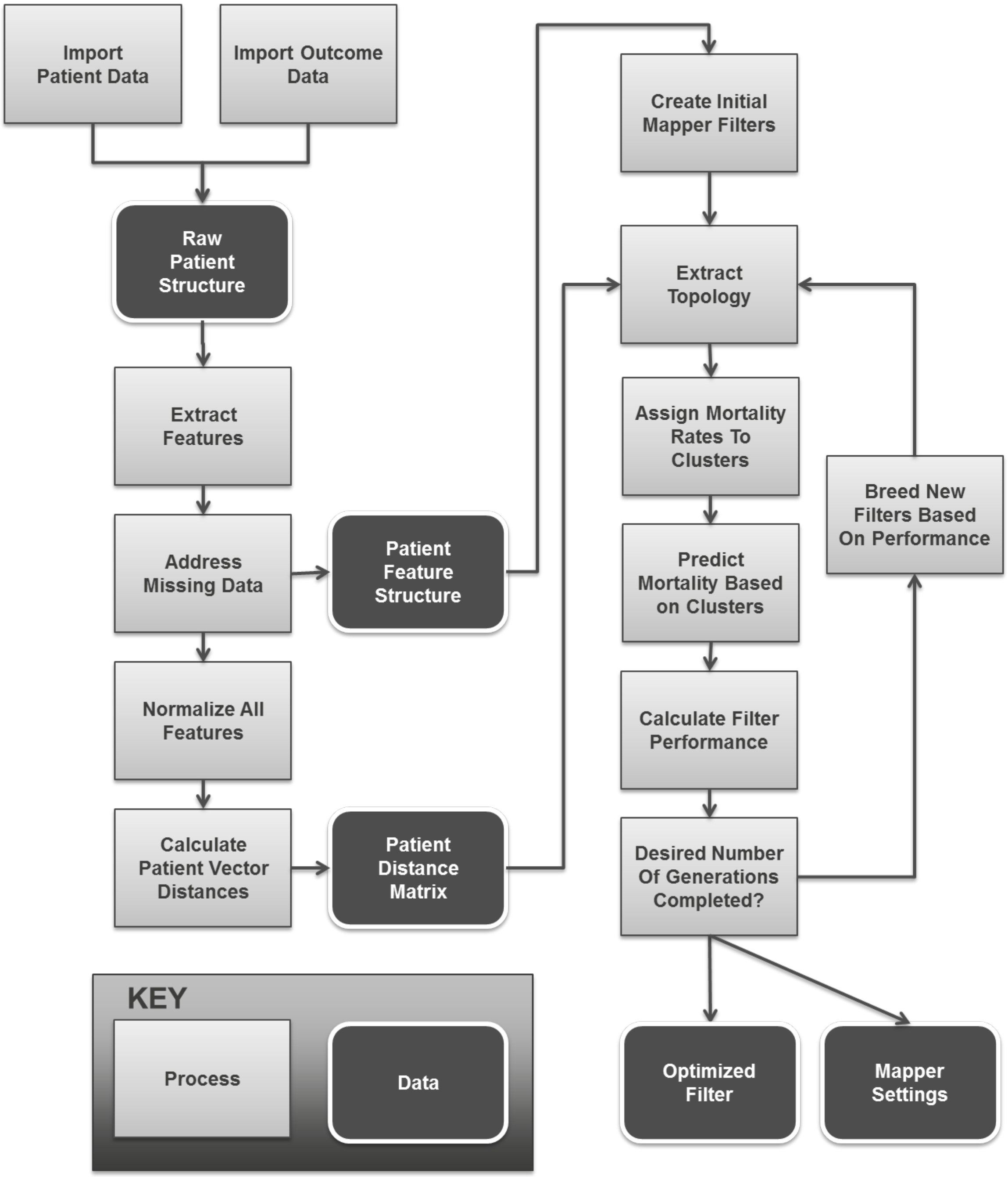

Import Patient Data
Import Outcome Data
Raw Patient Structure
Extract Features
Address Missing Data
Patient Feature Structure
Normalize All Features
Calculate Patient Vector Distances
Patient Distance Matrix
Create Initial Mapper Filters
Extract Topology
Assign Mortality Rates To Clusters
Predict Mortality Based on Clusters
Calculate Filter Performance
Desired Number Of Generations Completed?
Breed New Filters Based On Performance
Optimized Filter
Mapper Settings
KEY
Process
Data

Figure 5. Process Diagram for ICU Mortality Mapper Prediction Algorithm

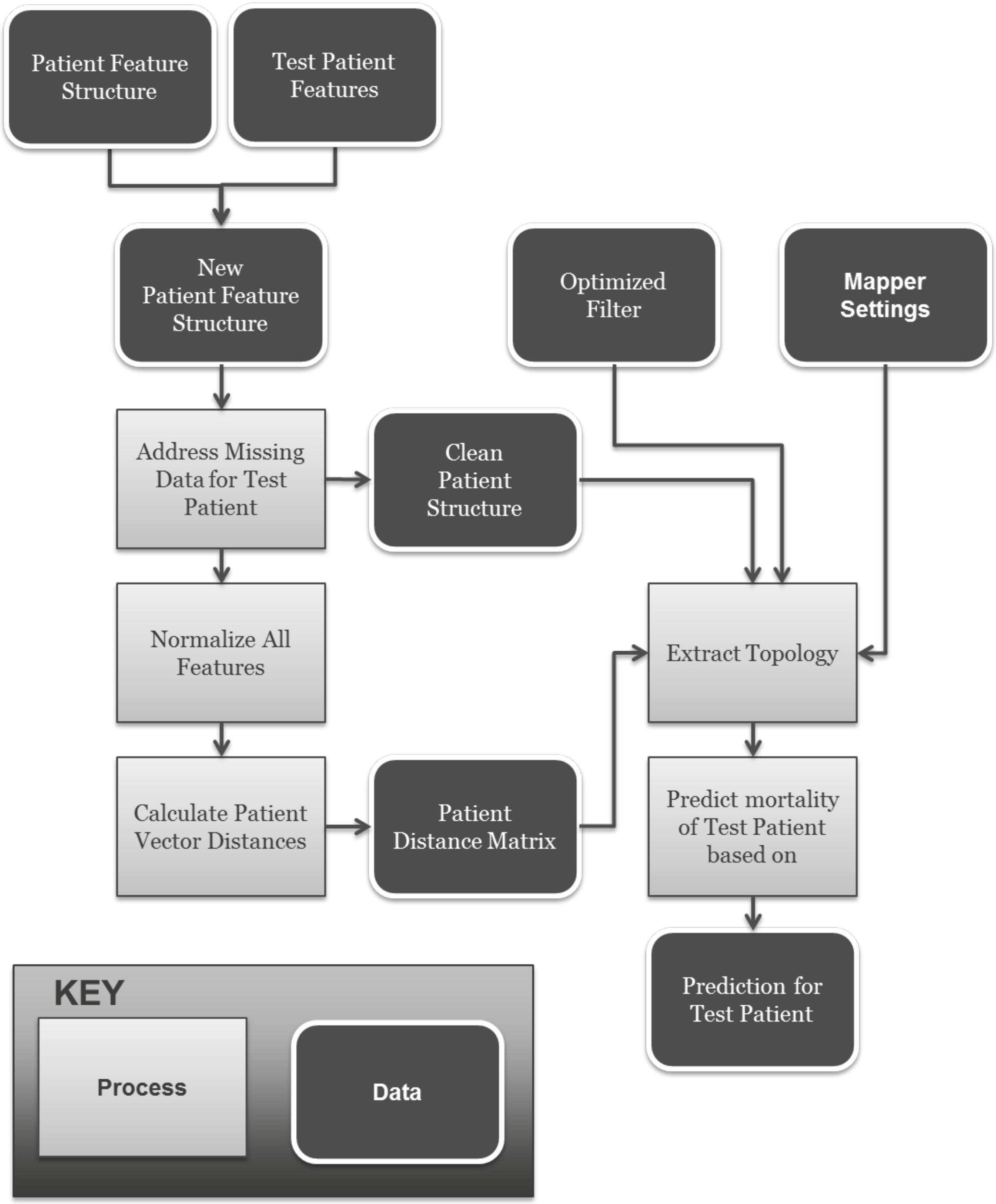

Patient Feature Structure
Test Patient Features
New Patient Feature Structure
Optimized Filter
Mapper Settings
Address Missing Data for Test Patient
Clean Patient Structure
Normalize All Features
Extract Topology
Calculate Patient Vector Distances
Patient Distance Matrix
Predict mortality of Test Patient based on
Prediction for Test Patient
KEY
Process
Data

Figure 6. Method used to predict patient mortality once an optimal solution is found.

The Patient Feature Structure and the Test Patient Features are combined into the New Patient Feature Structure. This structure undergoes the same process to generate the Clean Patient Structure and the Patient Distance Matrix. These are now fed into a single Mapper run along with the Optimized Filter code and the Mapper Setting used to generate it. Mapper runs, and the mortality rate for the patient is predicted, yielding the Prediction for Test Patient, a 0% to 100% value predicting the probability of in-hospital death for the patient. This process is shown in Figure 6.

This algorithm may be trained to predict an overall mortality rate, (not just in-hospital) by training the filter function to one more or multiple of the DeathX outcomes. It would be implemented by predicting the mortality for the patient across multiple versions of this model and extrapolating the expectancy curve. For this research, we will be focusing on making just a single prediction.

Data Import

The primary data set consists of data for 4000 patients collected for the first 48 hours after being admitted to the ICU. Some of the data is static: it does not change during the 48 hours. Other data is dynamic, having time series data for the first 48 hours. A summary of the types of data is given in the following list:

- RecordID
- Length_of_stay
- Survival
- Death
- Age
- Gender
- Height
- ICUType
- Weight
- ALP
- ALT
- AST

- Albumin
- BUN
- Bilirubin
- Cholesterol
- Creatinine
- DiasABP
- FiO2
- GCS
- Glucose
- HCO3
- HCT
- HR
- K
- MAP
- MechVent
- Mg
- NIDiasABP
- NIMAP
- NISysABP
- Na
- PaO2
- Platelets
- RespRate
- SaO2
- SysABP
- Temp
- TroponinI
- TroponinT
- Urine
- WBC
- pH

All of the raw patient data was imported without any subsampling or filtering. All of the patient record data and patient outcome data was imported into a Matlab structure Raw Patient Structure that is capable of handling the mix of data types in a single construct. The available data is a mixture of single values and 2-D arrays containing patient physiological data. For any case where patient data was not available, a flag was added to that field in the structure to indicate the lack of data. This flag is necessary for the feature extraction and calculation functions to properly deal with missing values instead of attempting to compute the specified feature.

Feature Extraction

From the raw data and outcome data, h = 26 features were extracted. By extracting features, patients may then be represented by a 1 x h vector in a g x h matrix containing the 26 desired features where g is the number of records (4000). This vector is later used to compute the Patient Distance Matrix. These features, along with some additional extracted features and some calculated features may be used to create the 35 x 4000 filter function space that is used by Mapper to generate the 1D cluster representation of the patients for in-hospital death prediction. The Feature Extraction section discusses how the 35 features were extracted. The following are three representations for the patient feature matrix:

$$FeatureMatrix$$
$$= \begin{bmatrix} Patient_1 Feature_1 & \cdots & Patient_1 Feature_{26} \\ \vdots & \ddots & \vdots \\ Patient_{4000} Feature_1 & \cdots & Patient_{4000} Feature_{26} \end{bmatrix}$$
$$= \begin{bmatrix} V_1 \\ \vdots \\ V_{4000} \end{bmatrix}$$
$$= [F_1 \quad \dots \quad F_{26}]$$

where V represents each patient vector, a 1 x 26 vector in the feature space and F represents the column vector of each feature, a 4000 value list of a specific feature for each patient.

Extracted Features

Three sets of features were extracted from the data set: 26 Distance Features for the distance matrix, additional features for use in Mapper filter functions, and outcome features.

The 26 Distance Features were chosen based on the results of [16] showing that these 26 features have the greatest correlation to mortality based on the neural network prediction. The extracted features are as follows:

1. GCSLast
2. GCSMean
3. GCSMax
4. HCO3Min
5. UrineSum
6. GCSTrend
7. HCO3Max
8. BUNMax
9. BUNLast
10. HCO3Last
11. BUNMin
12. HCO3Mean
13. BUNMean
14. SysABPMean
15. WBCLast
16. SysABPLast
17. FiO2PaO2Ratio
18. WBCMean
19. TempMean
20. GlucoseMax
21. NaMean
22. NaMax
23. SysABPNISysABPMin
24. Age
25. LactateLast
26. TempLast

Additional features are extracted for use in the Mapper filter function and for filtering and sorting the data. These are not part of the distance matrix calculation, but they may be used to group the patient data in Mapper. The values extracted are as follows:

1. RecordID
2. Length_of_stay
3. Survival
4. Death
5. Age (redundant)
6. Gender
7. Height
8. ICUType
9. Weight

Age is extracted both as a Distance Feature and as an Additional Feature. It is extracted in this step as it is part of the basic demographic information describing the patient. A later filter mask is used to prevent it from being used twice in the filter function. For cases where Weight is represented as time-series data, the first value is taken to represent the patient weight.

Survival represents the number of days an individual survived after admission to the ICU. A -1 Survival value represents individuals that have no record of death. A non-“-1” value is known for all patients that experienced in-hospital death. If left as-is, the clustering algorithm will group all of the long-term survivors with those who deceased quickly. The good values for the data set had a maximum close to 2600 days for survival. Also, none of the raw data had a value of 2600 days for survival. As a result, any individuals with unknown survival are assigned a survival value of 2600 days. This simultaneously allows these individuals to be clustered with other long term survivors, and allows us to know they are the group who had their value assigned.

Feature Extraction Functions

Functions were created to extract desired data from the patient fields. Table 2 summarizes all of the features extraction functions used. In all cases, the function is run on a selected field for the patient (for example, GCS). The function extracts the feature indicated by the function name and adds it to the patient structure. All features are resolved to a single floating point value to allow them to be used as vector components or filters. Any features that could not be created due to missing features are identified and later replaced by the feature's mean value as described in the Dealing with Missing and Erroneous Values section.

Created Features

Additional features were computed for the data set and used for filtering, training, and understanding the data set in EDEN. The following features are computed for the data set:

1. BMI
2. BMI Class
3. SurvivalRate
4. DaysHome
5. Death7
6. Death 14

Calculate BMI

This function adds the field BMI and BMIClass to the patient structure. BMI is added as a feature to the patient structure due to its correlation

Table 2. Feature extraction and creation descriptions

Feature	***Description***
Static	returns the first value of the data from the field
Sum	returns the summation of all of the data from the field
Ratio	returns the ratio of the mean of the first field divided by the mean of the second field
ABMin	returns the minimum of the mean of the first field and the mean of the second field
Min	returns the minimum of the data from the field
Max	returns the maximum of the data from the field
Mean	returns the mean of the data from the field
Trend	returns the slope of the data from the field using Matlab polyfit() with order 1
MaxSum	returns the sum of the local maxima
First	returns the first value of the data from the field
Last	returns the last value of the data from the field. Invalid if fewer than 2 data points.
BMI	returns the body mass index based on the height and weight of the patient
BMIClass	returns the body mass index class based on the height and weight of the patient
AgeBasedSurvival	returns the expected annual survival rate based on World Health Org. data for a specific age and gender

to mortality rates [12]. The BMI is calculated using Equation 1.

$$BMI = \frac{Weight}{Height^2}$$

Equation 1

Where Height is the patient height in meters and Weight is the patient weight in Kilograms. The patient's BMI class is assigned a value from 0 to 3 based on their BMI as shown in Table 3. Each of these values designates a specific BMI class.

Calculate Age Based Survival Rate

This function adds the field SurvivalRate to the patient structure. The survival rate is calculated based on data provided by the World Health Organization (WHO) in [9]. The WHO study provides data on the probability that an individual would become deceased the following year based on their current age and gender, worldwide. Other data sets were provided in the study, based on geographical region as well, however, the geographical location of our patients is unknown. Thus, the data set for world averages was used. The values given in the lookup table were converted into survival rates. The lookup tables used are shown in Table 4.

Days Home

This function takes the difference between the Days Survived and Days in ICU. This is used to estimate the number of days an individual was outside of the ICU prior to being deceased. For large values, we cannot expect our prediction algorithm to be accurate. If a person left the ICU one year ago then died, it is unreasonable to predict this mortality based solely on 48 hours of ICU data.

Table 3. BMI Class Limits

BMI Range	*Description*	*Identifier*
BMI < 18.5	Under Weight	0
18.5 ≤ BMI < 25	Normal Weight	1
25 ≤ BMI < 30	Over Weight	2
30 ≤ BMI	Obese	3

Table 4. Male and female survivability rates based on individual's age

Age	Male Survival Rate	Female Survival Rate
0	94.15%	94.53%
1	99.30%	99.25%
5	99.81%	99.81%
10	99.88%	99.88%
15	99.82%	99.82%
20	99.74%	99.75%
25	99.69%	99.71%
30	99.62%	99.69%
35	99.56%	99.68%
40	99.44%	99.63%
45	99.26%	99.53%
50	98.93%	99.30%
55	98.45%	98.98%
60	97.56%	98.37%
65	96.35%	97.53%
70	94.39%	96.03%
75	91.61%	93.92%
80	87.25%	90.39%
85	79.83%	83.17%

However, a person dying within a few weeks of leaving the ICU may have died due to complications from their condition that originally took them to the ICU in the first place. This value is calculated to analyze those groups in both Mapper and EDEN.

Death X

The DeathX feature is a feature indicating that the individual died in or within X days of leaving the ICU. This feature was used to assess if the individual who recently left the ICU may have been misclassified as a result. An individual may have been mistakenly released from the ICU, or released from the ICU to go home and spend the rest of their life with friends and family who otherwise would have died in the hospital. This parameter adjusts for that and can be used instead of the Death value imported from the Outcomes data set (where Death indicated in- hospital death). The number of positive cases available based on this new parameter is outlined in Table 5. Results are presented later on in this document based on the Death14 Parameter. Using this parameter, there is a 17% increase in the number of positive cases, making it 614 up from 548.

Dealing with Missing and Erroneous Values

Once the features have been extracted, erroneous values are removed and are considered missing. Missing values are then recovered using the mean value for the good data in the specific field. Statistics on the missing values and outliers for the 26 features are in Table 6. The number of manipulations

Table 5. Statistics for mortality after leaving the ICU compared to in-hospital death.

Days after Leaving ICU	*Deceased*	*Additional*	*Percent Increase*
0	548	0	0%
7	614	66	12%
14	640	92	17%
21	659	111	20%

Table 6. Statistics on each feature

Field	Outliers	Missing	Field	Outliers	Missing
RecordID	0	0	**BUNMax**	89	64
Death	0	0	**BUNLast**	92	112
Length_of_stay	77	60	**HCO3Last**	36	149
Survival	0	0	**BUNMin**	101	64
Age	0	0	**HCO3Mean**	44	76
Gender	0	3	**BUNMean**	93	64
Height	16	1894	**SysABPMean**	26	1201
ICUType	0	0	**WBCLast**	38	176
Weight	44	326	**SysABPLast**	49	1219
BMI	1	1895	**FiO2PaO2Ratio**	43	1458
BMIClass	0	1895	**WBCMean**	43	73
SurvivalRate	0	3	**TempMean**	46	64
GCSLast	0	66	**GlucoseMax**	65	113
GCSMean	0	64	**NaMean**	45	75
GCSMax	53	64	**NaMax**	44	75
HCO3Min	54	76	**SysABPNISysABPMin**	25	134
UrineSum	16	124	**Age**	0	0
GCSTrend	19	1418	**LactateLast**	28	2535
HCO3Max	49	76	**TempLast**	4	67

performed on each record was tracked for filtering purposes. Removing highly modified records may lead to a more accurate result.

There is a discrepancy between the missing counts for some of the values, such as GCS Mean and GCS Last. This occurs because the Last feature requires that more than 2 values exist for the Last feature to be valid.

Missing Data

From the initial set of data imported, there was data missing from certain physiological time series. In these cases, the feature was flagged for replacement. Information on missing data by feature is shown in Table 6.

Removal of Erroneous Data

Outliers (>+/- 3 Standard Deviations from mean) were removed for some of the analyses from all fields and later replaced. In other cases, outliers were selectively removed. Without very specific analysis of every physiological signal, this is the easiest way to remove errors for the full data set. An example is with the height parameter.

Height was reported in centimeters. The maximum height value reported was 431.8 cm. This is the equivalent of just over 14 feet. It is easy to believe this is incorrect. Re-interpreting the 432 value in alternative units or decimal places do not yield reasonable results for the height of a human.

With the amount of manual time it would take to comb through 26 features across 4000 records, it is easiest to just remove an outlier for the current algorithm.

Replacement

For patients who are missing a value for a particular feature, the mean value for the entire population is used to replace the missing value. Once replaced, the final patient feature matrix is prepared for Mapper input extraction. Alternative methods were considered, such as using the mean value interpolated based on age and gender, however, such relationships were not present in this data set. Such a method would not have yielded any benefit as a result.

After all of the missing and erroneous values have been addressed, the resulting Patient Feature Structure is exported and stored for use in the optimization routine. This same structure was then normalized and prepared for the distance calculation.

Normalize Features

Before computing feature space distances, all of the features are normalized from 0 to 1 across their range to ensure that the individual values of a given feature do not mistakenly weight its contribution to distance in the feature space. The features are normalized based on Equation 2.

$$F_{normalized} = \frac{F_{in} - \min(F_{in})}{\max(F_{in}) - \min(F_{in})}$$

Equation 2

such that any value in $F_{\text{normalized}}$ is always less than or equal to 1.

Calculate Patient Distance Matrix

As described in Table 7, the distance matrix is a g x g symmetric distance matrix where g is the number of records. The cell i,j contains the value

Table 7. Inputs for the function Mapper.

INPUTS	
d	n x n distance matrix where cell i,j contains the distance between vectors i and j
filter	n x 1 array of real numbers used to decompose the space
resolution	Number of samples in each interval
overlap	Percentage of overlap between each consecutive interval
magicFudge	Number that "fudges" the number of clusters that will tend to be created by the algorithm

representing the distance between the ith and jth record vectors in the feature space. This distance matrix is required by Mapper to determine which records are within the same node or cluster.

There exist many options for the metric or distance function. These include the Euclidean metric, the squared Euclidean metric, or the L1 distance. They are listed in order of computation intensity from high to low. Since this analysis is not being performed in real time, we have chosen Euclidean metric. If real-time computation is desired on a much larger data set, the L1 could be considered. If more clusters are desired, the squared Euclidean metric could be used. By not taking the square root of the distance function, the distance between points that are far from one another will be exaggerated.

The distance matrix d is generated from the 26 desired features. For each record, the features represent a 26 dimension vector. Before measuring the distance between the vectors, all of the features are normalized so as not to weight one more heavily than the other based on the inherent physiological data values. Once the features have been normalized, the Euclidean distance between each feature is calculated based on Equation 3.

$$d(V_1\ to\ V_2) = \sqrt[2]{\sum_{i=1}^{g} (V_{2i} - V_{1i})^2}$$

Equation 3

In Equation 3, V represented the gx1 vector in the feature space and g represents the dimension of the feature space.

Generate Filter Function

The filter is any value that may be assigned to each patient that maps the patient into a 1D space. This filter could be any feature or combination of features for each patient. In our algorithm, we will attempt to exploit this property by creating linear combinations of the available features to map the point cloud data into a 1D space that may be used to predict mortality rate. A list of all considered filter function components are listed in Table 8.

Filters are created using a linear combination of features. Before combining the features, all of the features are normalized, mapping the minimum value to 0 and maximum value to 1. After a set of features are selected, their values are summed and divided by the number of features. This ensures that in all cases, the mapper function uses a filter function from 0 to 1 for each patient, making all filters comparable.

A filter mask was used to decrease the evolution algorithm's search space for an optimal filter. The filter mask works by preventing specific filters from being activated. As there are multiple features that draw from the same data type (for example GCSMean, GCSMax, et.), it is reasonable to pick only one filter per data type to start. A bit string similar to the filter bit strings is used to allow or disallow a specific filter from being activated. The Filter Mask section shows the filter mask configuration used for this project.

Table 8. List of bits in filter and mask strings with corresponding filters.

Bits 1-8	**testType bits**	**Bits 9-34**	**Top 26 features(cont.)**
1	Age	**19**	BUN Min
2	Gender	**20**	HCO3 Mean
3	Height	**21**	BUN Mean
4	ICUType	**22**	SysABP Mean
5	Weight	**23**	WBC Last
6	BMI	**24**	SysABP Last
7	BMI Class	**25**	FiO2PaO2 Ratio
8	Survivability Rate	**26**	WBC Mean
Bits 9-34	Top 26 features	**27**	Temp Mean
9	GCS Last	**28**	Glucose Max
10	GCS Mean	**29**	Na Mean
11	GCS Max	**30**	Na Max
12	HCO3 Min	**31**	SysABPNISysABP Min
13	Urine Sum	**32**	Age
14	GCS Trend	**33**	Lactate Last
15	HCO3 Max	**34**	Temp Last
16	BUN Max	**Bit 35**	Distance
17	Bun Last	**35**	Distance from Patient 1
18	HCO3 Last		

Extract Topology

The Mapper function is a function designed to take a distance matrix for a high-dimensional data set and apply a filter function that maps the data set into a 1-D representation of the data, preserving the topology. Connectedness and shape (number of branches, holes) are preserved in this representation. This allows the high dimension data set to be analyzed by nodes and clusters. This section will lay out the requirements for the mapper program, describe how they have been calculated, and describe how the mapper outputs are used.

Mapper Inputs

The inputs for the Mapper function are defined in Table 7. Mapper requires a g x g distance matrix and a g x 1 filter function. The distance and filter inputs are constructed using the information from the patient feature matrix. The resolution, overlap, and magicFudge are parameters used to configure how mapper groups and projects the clusters.

Mapper Configuration

There are three parameters used to tune the output of Mapper:

Resolution, Overlap, and Magic Fudge

Resolution determines the length of each interval. In our case, we will typically use the inverse of the number of samples desired to calculate this value. The number of samples is related to the number of nodes that exist in each cluster.

Overlap is the percent overlap between adjacent nodes in a cluster. This is what forms the connectedness of the data set. Increasing the overlap percent typically results in fewer nodes as more data is shared between adjacent nodes. Decreasing the overlap percent typically results in more nodes.

MagicFudge is used to “fudge” the desired number of clusters. Increasing this number will increase the number of clusters and decreasing this will decrease the number of clusters. This allows control over the number of clusters but does not require us to pre-determine the number of clusters to be created. This is one of the benefits of using Mapper over other cluster creation tools.

Mapper Outputs

The outputs for the Mapper function used are defined in Table 9. The adjacency matrix contains information on the relationship between the nodes. The node info contains info on the interval level, the filter value for the node, and the set of members in the node. levelIDx contains the list of nodes in the subinterval of the filter.

Adjacency Matrix

The adjacency matrix is a sparse matrix containing binary data indicating which nodes overlap to form a cluster. So, if a 1 exists at cell 5,7, this indicated that nodes 5 and 7 are connected in the topology. A 0 would indicate that they are not connected. While a 1 implies that they belong to the same cluster as the other node, a 0 does not imply they are not a member of the same cluster as multiple nodes string together to form a single cluster.

Table 9. Outputs for the function Mapper

OUTPUTS	
adja	The adjacency matrix of the output graph
nodeInfo	Cell array containing the information listed below.
nodeInfo.Level	The interval index for the node
nodeInfo.Filter	The max filter value belonging to all of the nodes in this cluster
nodeInfo.Set	The set of all of the members belonging to this cluster.
levelIDX	List of nodes belonging to each subinterval of the cluster.

Node Info

The node info is a structure containing three sets of data: level, filter, and set. The only information pertinent to our optimization is the set, an array containing all of the members in that node. This information is used to extract info from the patient feature structure and describe the node.

Post-Processing Topology

In order to create a graphical representation of the output of mapper, it must undergo some post processing. In our case we take advantage of this to show how well or poorly our algorithm groups individuals with similar mortality. The graphical representation of the data topology is achieved using a program Graphviz that was supplied with Mapper. Graphviz imports the mapper output and generates a graphic.

Calculate inputs for Graphviz

Graphviz requires the node relationship information contained in the Mapper adjacency (adja) output, the size for each node, the color for each node, and the desired labels for the bottom of the graph.

The adjacency info is taken directly from the mapper output. The set size is calculated by counting the number of values in the nodeInfo.Set array. The color is determined based on the mean mortality for the patients in the set.

Generate Graphviz Input

The function writeDotFile provided with Mapper takes the information from the previous section and converts it into a .dot file that Graphviz can use to generate the images. The images may then be used for quick inspection of a solution to describe its performance.

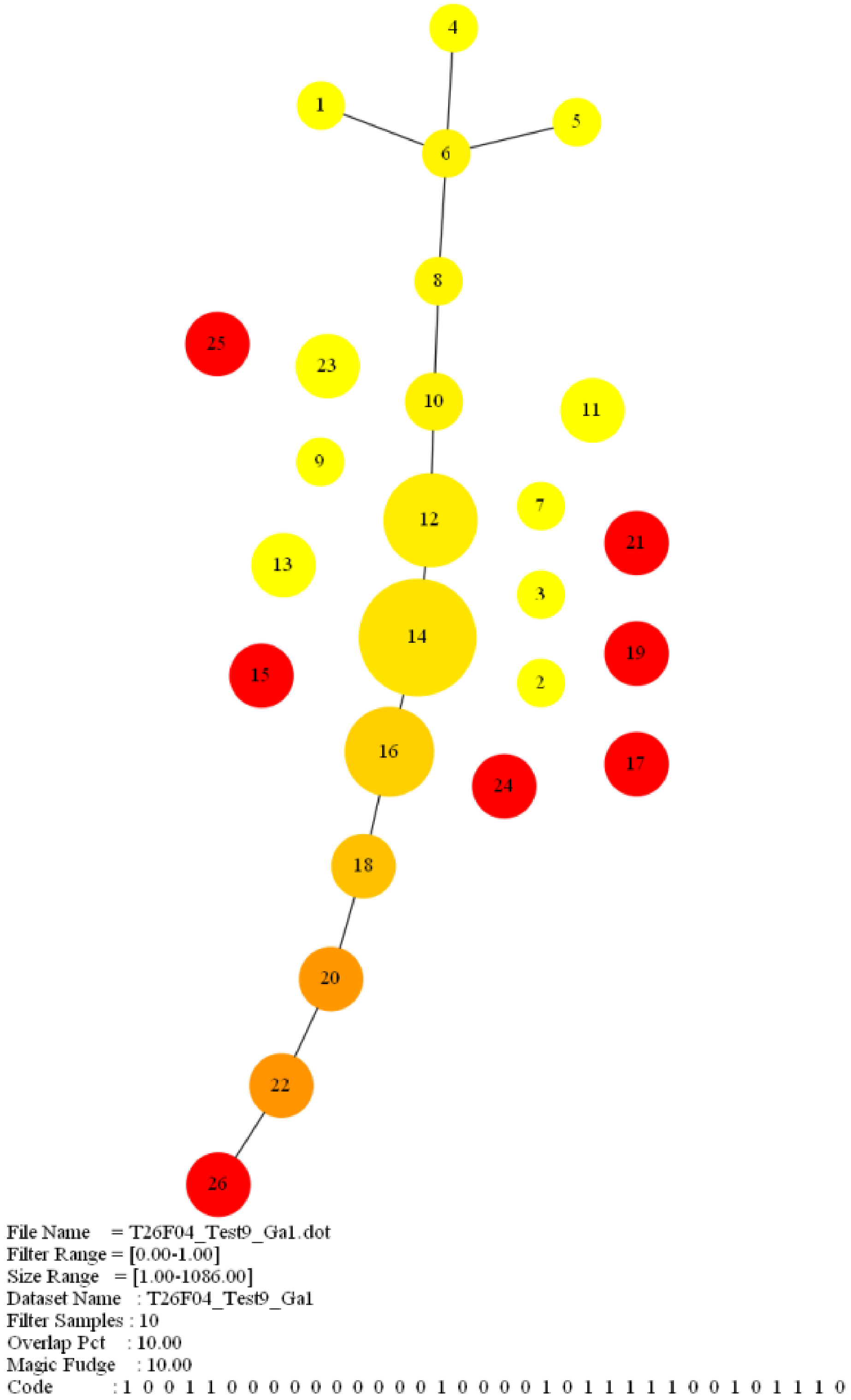
The labels chosen include the file used to generate the mapper output, data used as inputs for mapper, mapper parameters, and the filter code used. With this information, it is possible to fully recover this result from the original patient feature structure. Later results also include the filter's Event 1 score as defined by [32].

Graphical Representation of Mapper Outputs

The outputs of Mapper undergo some minor post processing to format the data for Graphviz. Graphviz is a Matlab script that takes the post-processed mapper output and converts it into a graphical representation as shown in Figure 7. In this figure, the color represents the mortality rate, scaling from 0% (Yellow) to 100%(Red). The size of each node is a representation of the number of patients in the cluster. The lines connecting some of the nodes represented overlap between adjacent nodes. Outside of this information, the shape or relative position of the nodes has no meaning or representation.

Assigning Mortality Rates to Cluster Nodes

After mapper has created all of the nodes and clusters, each node is assigned a mortality rate based on its members. The mortality rate assigned is the mean mortality rate for all patients in that node.



```
File Name    = T26F04_Test9_Ga1.dot
Filter Range = [0.00-1.00]
Size Range   = [1.00-1086.00]
Dataset Name   : T26F04_Test9_Ga1
Filter Samples : 10
Overlap Pct    : 10.00
Magic Fudge    : 10.00
Code           :1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 1 1 1 0
```

Figure 7. Sample Mapper output after Graphviz Processing

The nodeInfo output from mapper is a structure containing information about each node. NodeInfo.set contains the list of all of the members in a given node. The in-hospital death outcome is extracted from the patient structure for each patient and the mean is calculated on the resulting array.

Predict Patient Mortality Based on Clusters

A simple algorithm was constructed to predict patient mortality based on the node data.

1. Identify all nodes containing the patient
2. Look up the mortality rate for each of these nodes
3. Choose the greatest mortality rate
4. If that mortality rate is greater than or equal to 50%, predict that the patient is deceased
5. If the mortality rate is less than 50%, predict that the patient survived.

Other prediction algorithms could be created for this step use a different topology or filter data in a given node so that a specific gender or age range was analyzed separately. This algorithm was chosen for simplicity and its general applicability as the focus of this research was optimizing the filter function. This particular prediction algorithm is prone to over-predict mortality (false positives).

Calculate Filter Performance

The fitness function for the algorithm was based on the Event 1 score from the 2012 Physionet Challenge using data set A. The performance is based on the minimum of the sensitivity (Se, Equation 4) and the positive predictivity (+P, Equation 5). True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) are defined in Table 10 and are used in Equations 4-6.

$$Se = \frac{TP}{TP + FN}$$

Equation 4

$$+P = TP/(TP + FP)$$

Equation 5

$$Score1 = \min(Se, +P)$$

Equation 6

A perfect score is achieved when all of the deceased patients are positively identified by the prediction algorithm and none of the surviving patients are predicted as deceased. Random chance will yield a core of 13.9% and the currently used SAPS-1 system will yield a score of 29.6% [32]. The algorithm discussed in this research achieved a score of 30.0%

Table 10. Outcomes matrix definitions

		Observed	
		Deceased	Survive
Predicted	Deceased	True Positive (TP)	False Positive (FP)
	Survive	False Negative (FN)	True Negative (TN)

Breed New Filters based on Performance

Each new generation is calculated after the previous generation has been evaluated. First, the previous population is sorted based on member performance. The top five members from this population are selected and preserved for the next generation. Next, ten new members are generated by breeding the top members with all other members of the population.

Two randomly generated 1x10 arrays are created; one with random values from 1 to 5 (to select one of the first 5 members) and one containing values from 1 to 20 (to randomly select one of any of the previous generation's members). Once the breeding pairs have been selected, the bit strings were combined by averaging their respective values. This method was chosen over the method presented by Whitley [15] in order to preserve favorable features more often. The concern is then raised if enough variation is being maintained in the population. This is addressed by injecting completely random individuals into the population. Any features in both are maintained, and any features absent from both are maintained. Any discrepancies are settled by random selection. The final five members are randomly generated and added to the population. Any redundant members are replaced with a randomly generated member.

Training of Mapper Filter Function using Evolution Algorithm

As defined in the Mapper Inputs section, any number of filters can be created from combinations of the filter features. Our optimization routine will activate or deactivate each individual filter in an attempt to subdivide the space into meaningful clusters used to predict the mortality rates in each node. An evolution algorithm was chosen for this task because it is a convenient way to search a space that can be constructed as a binary string of features for the topic of interest, in our case, a filter.

Mapper takes between 30 seconds and one minute to run with the parameters we have defined. It would take just under two years to compute every possible filter and filter score on a single core machine. The code has been optimized to run on multiple cores to reduce computation time.

Optimization Configuration

A population of 20 members was chosen to allow the algorithm to complete a generation quickly, giving a user multiple opportunities to evaluate algorithm progress and ability to proceed or start with a new population. The process continues until the number of desired generations is reached. It may immediately be run again after completion with the current population to improve on the current result.

Filter Mask

A filter mask was implemented that would restrict the search space of the breeding algorithm. The filter mask used is described in Table 11. With this mask, our search space is limited to 2^{20}-1 or just over one million possible filters as opposed to the 34 billion possibilities. The 2^{20} possible filters come from the 20 possible features accessible by the feature mask and the -1 as a filter of all

Table 11. Filter mask used to minimize evolution algorithm search space.

Mask Value	Feature	Mask Value	Feature
1	Age	**0**	BUN Min
1	Gender	**0**	HCO3 Mean
1	Height	**0**	BUN Mean
1	ICUType	**1**	SysABP Mean
1	Weight	**1**	WBC Last
0	BMI	**1**	SysABP Last
0	BMI Class	**1**	FiO2PaO2 Ratio
0	Survivability Rate	**0**	WBC Mean
0	GCS Last	**1**	Temp Mean
1	GCS Mean	**1**	Glucose Max
0	GCS Max	**1**	Na Mean
1	HCO3 Min	**1**	Na Max
0	Urine Sum	**1**	SysABPNISysABP Min
0	GCS Trend	**0**	Age
0	HCO3 Max	**1**	Lactate Last
1	BUN Max	**1**	Temp Last
0	Bun Last	**1**	Distance from Patient 1
0	HCO3 Last	**KEY: (1) Filter Active (0) Filter Inactive**	

0’s is considered invalid and is recalculated. Each 1 and 0 in Table 11 represents if a particular feature is “on” or “off.” This is reasonable based on the inter-data relationships found using EDEN.

Mapper Configuration

Different sets of parameters were tested with Mapper. The specific configuration parameters chosen may greatly affect the outcome of the clustering algorithm. Different sets of parameters were tested. These sets of parameters are shown in Table 12. Configuration set 2 was used for the solution presented in the results section.

Application of Result to Test Patient

After an optimized filter has been trained, the optimized filter along with the Mapper parameters and Patient Feature Structure are ready to be used to predict the patient’s mortality. The new patient is added to the Patient Feature Structure and the same is applied as before, without the optimization step. This tool not only predicts if the individual is likely to be deceased in the ICU, but also is capable of providing a percentage chance of survival by extracting the mortality rate from the node the patient is grouped with.

Statistics on the specific node for life expectancy and confidence level may be extracted as well, based on the days survived data stored in the Patient Feature Structure. This algorithm could be re-trained based on the first 24 hours of patient data, or the first 12 hours to provide a more accurate, short term prediction capability.

Table 12. Mapper settings used in evolution algorithm.

Generations	*Resolution*	*Overlap*	*Magic Fudge*
Default	5	50	10
Set 1	25	10	10
Set 2	20	10	10
Set 3	30	10	10

CHAPTER V
RESULTS

This section will present the algorithm output, the performance of the optimized filter, the statistics on the results, and the validity of the models created. The results were presented for the models trained to the in-hospital death outcome and the Death14 outcome.

Interpreting Algorithm Results

The Mapper results are color plotted using mortality data. A deep red node would indicate a node that indicates mortality. A completely yellow node would indicate a node with only survivors. The size of the circle indicates the number of records in the node. A line joining two adjacent nodes indicate an overlap between the adjacent nodes of approximately the percent overlap specified in the Mapper configuration.

By visual inspection, a successful mapper algorithm for the prediction of mortality rates will contain either of the following:

1. A large cluster that shows a gradient from high mortality rate to low mortality rate. For this to be significant, this cluster must contain a high percentage of the total population and contain a significant number of nodes.
2. A few large clusters contain all the deceased patients and very few surviving patients.

Performance of In-Hospital Death Optimized Filter

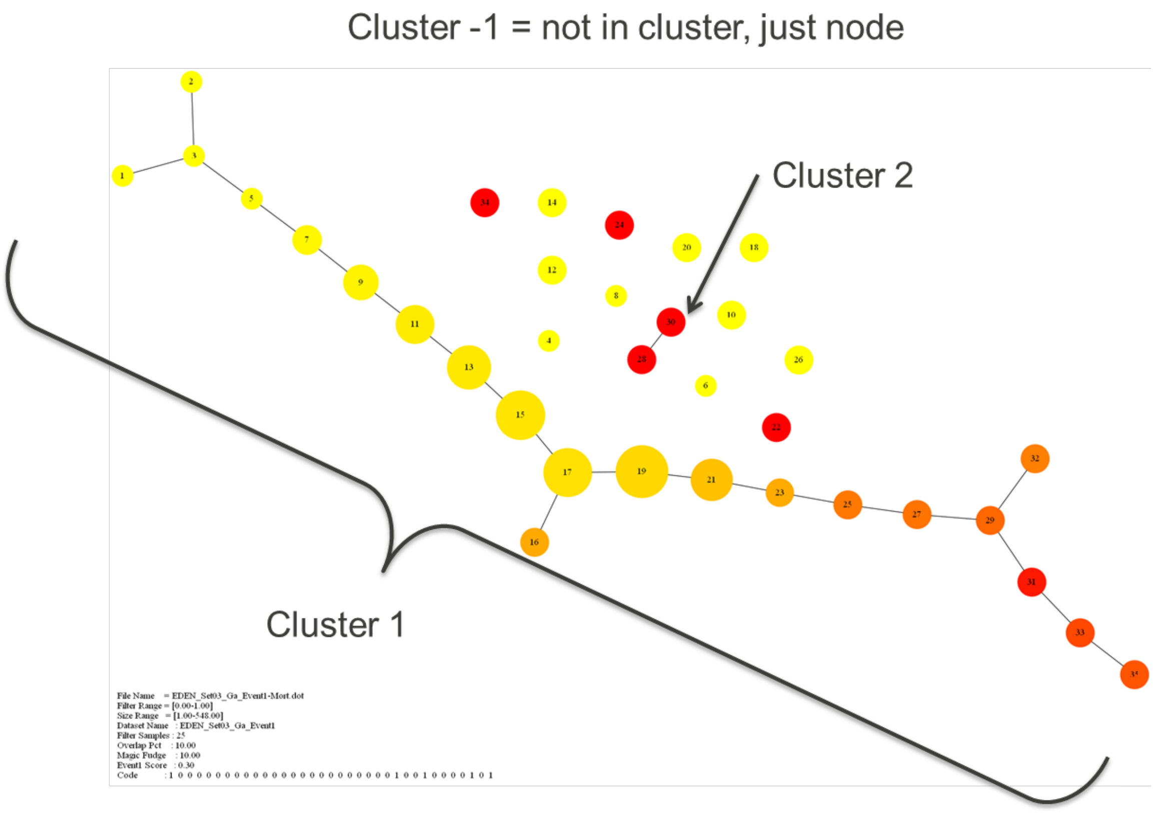
The algorithm was allowed to evolve for 100 generations and was able to yield a 0.30 Event 1 score. The cluster generated is shown in Figure 8.As for the broader applicability of this model, the statistics for this clustering are shown in Table 13. Statistics for individual clusters are in Table 14.

Also of note, the sum of the deceased and survivors is 695 dead + 3760 surviving = 4455 total patients > 4000 initial patients because of the overlap between the nodes. A single patient may be in more than one node, and for the totals in Table 13, the values were summed by node, leading to artificially inflated values.

The optimized filter generated is created based on the following features:

- Age
- FiO2PaO2 Ratio
- Glucose Max
- Lactate Last
- Distance from Patient 1

A linear combination of these features was used as the filter function input to Mapper to yield the cluster shown in Figure 8. One primary cluster was created with multiple single record nodes. Cluster 2 was mis-identified as a cluster by the Mapper algorithm; only a single record exists in the entire cluster. To understand how the filter utilized these values, the data for all of these features was analyzed against the true positives in EDEN. All of the patients were assigned a



Cluster -1 = not in cluster, just node

Cluster 2
Cluster 1
File Name = EDEN_Set03_Ga_Event1-Mort.dot
Filter Range = [0.00-1.00]
Size Range = [1.00-548.00]
Dataset Name : EDEN_Set03_Ga_Event1
Filter Samples : 25
Overlap Pct : 10.00
Magic Fudge : 10.00
Event1 Score : 0.30
Code : 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 1

Figure 8. Mapper Cluster with 30% Event 1 score for the 2012 Physionet challenge.

Table 13. Statistics for the optimized Mapper output

Statistic	Value	Statistic	Value
Total Nodes	35	Nodes with <5 members	17
Total Deceased	695	Members in nodes with <5 members	23
Total Survivors	3760	Deceased in nodes with <5 members	17

Table 14. Statistics for topology nodes

Node	Cluster	Members	Number of Deaths	Mortality Rate
28	2	1	1	100%
30	2	1	1	100%
1	1	67	1	1%
2	1	1	0	0%
3	1	127	1	1%
5	1	192	4	2%
7	1	279	9	3%
9	1	353	17	5%
11	1	389	31	8%
13	1	444	40	9%
15	1	510	52	10%
16	1	3	1	33%
17	1	496	61	12%
19	1	548	85	16%
21	1	420	103	25%
23	1	269	89	33%
25	1	162	85	52%
27	1	89	49	55%
29	1	47	28	60%
31	1	20	18	90%
32	1	4	2	50%
33	1	14	10	71%
34	1	1	1	100%
4	-1	1	0	0%
6	-1	1	0	0%
8	-1	1	0	0%
10	-1	2	0	0%
12	-1	1	0	0%
14	-1	1	0	0%
18	-1	1	0	0%
20	-1	1	0	0%
22	-1	1	1	100%
24	-1	1	1	100%
26	-1	1	0	0%
35	-1	6	4	67%

single node based on their proximity in the feature space to the nodes they were members of. This allowed for each patient to be assigned an explicit mortality rate based on their most appropriate node. This enables the filtering of the data in EDEN based on the true mortality value and the predicted mortality rate for that patient. The results of this analysis are shown in Figure 9 through Figure 13.

Emerging from this analysis are the following trends for true positive patients identified by the trained algorithm as shown in Figure 10:

1. There is a notable shift upward in the mean age for the patients
2. There is a notable shift upward in the mean value for the FiO2PaO2 Ratio
3. There is a notable shift upward in the max glucose value
4. There is a significant shift upward in the last lactate value
5. There is a differentiation that occurs based on the distance to patient 1.

Similar tendencies are observed in Figure 11 for the set of all mortalities, though they are not as pronounced. This reveals that the patients positively identified by this algorithm are identified based on changes in the features indicated. These patients can be removed from the set and the training routine can be repeated to determine a filter function better trained to identify the remaining patients. The prediction can then be configured to choose one topology or another based on thresholds in the indicated features.

Intra-feature relationships are shown in Figure 12 and Figure 13 for the filter parameters found. A completely blue square indicates a strong inverse relationship between the features. A completely red square indicated a strong positive relationship between the features.

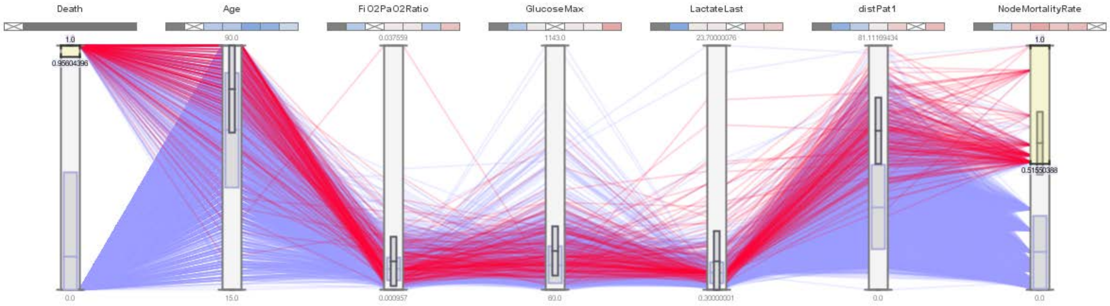


Figure 9. Data relationships with true positives highlighted in red

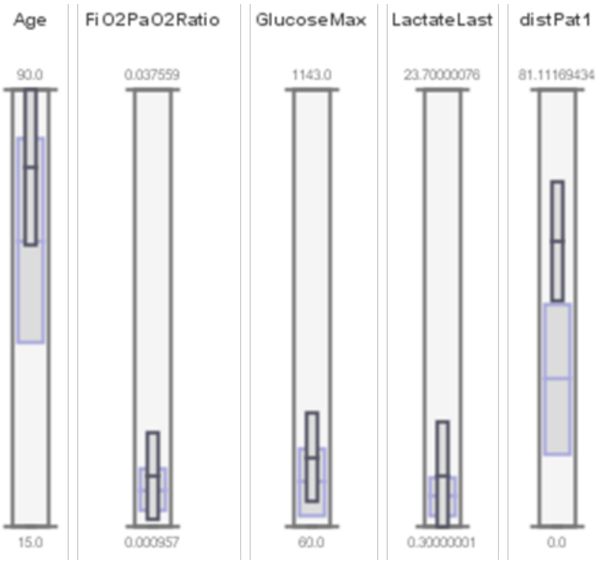


Figure 10. Distributions for full data set (dark bars) and true positives (light blue bars)

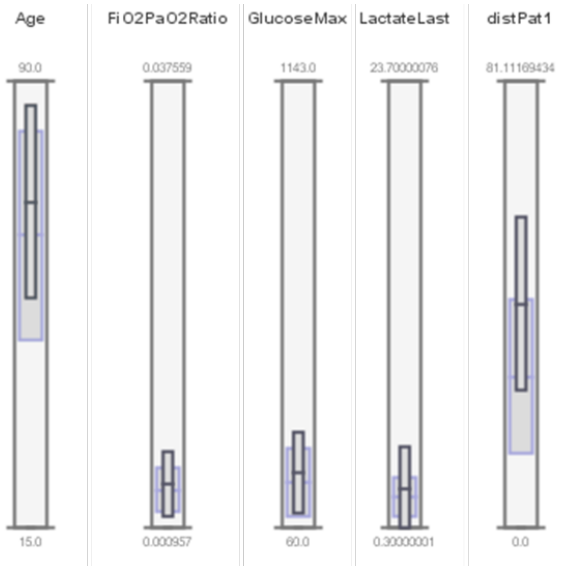


Figure 11. Distributions for full data set (dark bars) and mortalities (light blue bars)

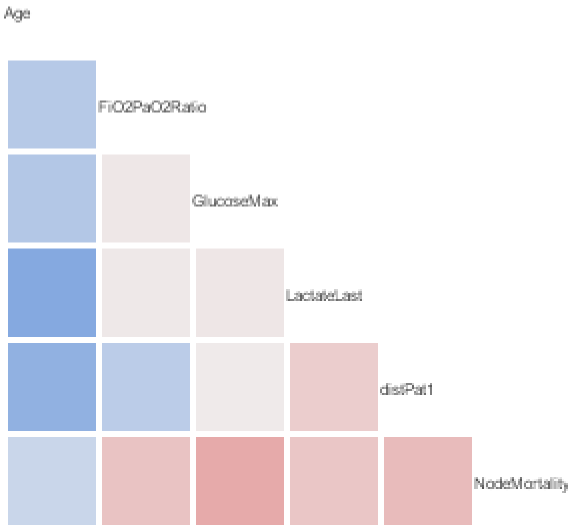


Figure 12. Correlation matrix for true positives only

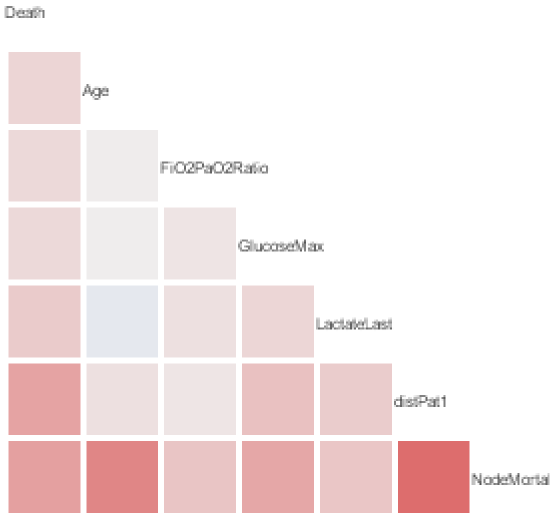


Figure 13. Correlation matrix for all data points

Performance of Death14 Optimized Filter

The algorithm was allowed to evolve for 100 generations using Death14 as the desired outcome and was able to yield a 0.26 Event 1 score. The cluster generated is shown in Figure 14.As for the broader applicability of this model, the statistics for this clustering are shown in Table 15. Statistics for individual clusters are in Table 16.

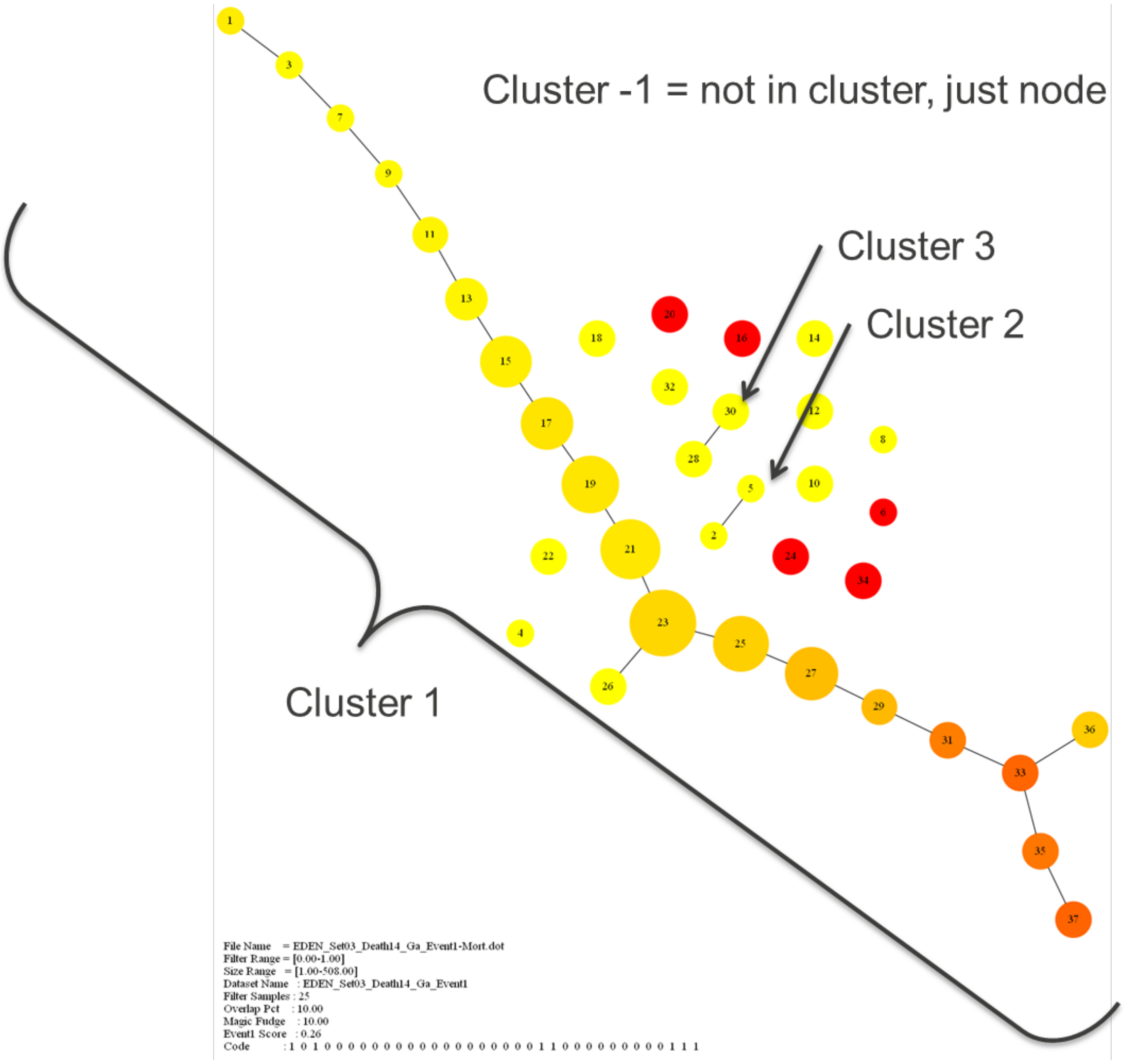
Based on the statistics, there are not a large number of members in the non-clustered groups, and there are enough nodes to spread out the deceased and survivors to allow the algorithm to differentiate. Based on these two facts, the model is considered valid for greater application.

Also of note, the sum of the deceased and survivors is 692 dead + 3705 surviving = 4397 total patients > 4000 initial patients because of the overlap between the nodes. A single patient may be in more than one node, and for the totals in Table 15, the values were summed by node, leading to artificially inflated values.

The optimized filter created is based on the following features:

- Age
- Height
- SysABP Mean
- WBC Last
- Lactate Last
- Temp Last
- Distance to Patient

A linear combination of these features was used as the filter function input to Mapper to yield the cluster shown in Figure 14. One primary cluster was created with two small clusters and multiple small nodes. Clusters 2 and 3 were misidentified as a cluster by the Mapper algorithm; only a single record exists in the entire cluster. To understand how the filter utilized these values, the data for



```
File Name    = EDEN_Set03_Death14_Ga_Event1-Mort.dot
Filter Range = [0.00-1.00]
Size Range   = [1.00-508.00]
Dataset Name   : EDEN_Set03_Death14_Ga_Event1
Filter Samples : 25
Overlap Pct    : 10.00
Magic Fudge    : 10.00
Event1 Score   : 0.26
Code           : 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 1
```

Figure 14. Mapper Cluster for Death14 with 26% Event 1 score for the 2012 Physionet challenge

Table 15. Statistics for the Death 14 Optimized Mapper Output

Statistic	Value	Statistic	Value
Total Nodes	37	Nodes with <5 members	18
Total Deceased	692	Members in nodes with <5 members	19
Total Survivors	3705	Deceased in nodes with <5 members	5

Table 16. Statistics for topology nodes for Death14

Node	Cluster	Members	Number of Deaths	Mortality Rate
28	3	1	0	0%
30	3	1	0	0%
2	2	1	0	0%
5	2	1	0	0%
37	1	23	14	61%
33	1	81	49	60%
35	1	41	22	54%
31	1	140	71	51%
29	1	247	68	28%
27	1	390	101	26%
36	1	5	1	20%
25	1	410	76	19%
23	1	508	81	16%
17	1	388	42	11%
19	1	428	42	10%
21	1	451	44	10%
1	1	13	1	8%
15	1	379	26	7%
11	1	248	11	4%
13	1	304	13	4%
9	1	163	5	3%
3	1	43	1	2%
7	1	97	2	2%
26	1	1	0	0%
6	-1	1	1	100%
16	-1	1	1	100%
20	-1	1	1	100%
24	-1	1	1	100%
34	-1	1	1	100%
4	-1	2	0	0%
8	-1	1	0	0%
10	-1	1	0	0%
12	-1	1	0	0%
14	-1	1	0	0%
18	-1	1	0	0%
22	-1	1	0	0%
32	-1	1	0	0%

all of these features was analyzed against the true positives in EDEN. All of the patients were assigned a single node based on their proximity in the feature space to the nodes they were members of. This allowed for each patient to be assigned an explicit mortality rate based on their most appropriate node. This enables the filtering of the data in EDEN based on the true mortality value and the predicted mortality rate for that patient. The results of this analysis are shown in Figure 15 through Figure 19.

Emerging from this analysis are the following trends for true positive patients identified by the trained algorithm as shown in Figure 16:

1. There is a notable shift upward in the mean age for the patients
2. There is a slight shift downward in the mean SysABP value
3. There is a notable shift upward in the WBC Last value
4. There is a slight shift upward in the last lactate value
5. There is a differentiation that occurs based on the distance to patient 1.

Similar tendencies are observed in Figure 17for the set of all mortalities, though they are not as pronounced. This reveals that the patients positively identified by this algorithm are identified based on changes in the features indicated. These patients can be removed from the set and the training routine can be repeated to determine a filter function better trained to identify the remaining patients. The prediction can then be configured to choose one topology or another based on thresholds in the indicated features.

Intra-feature relationships are shown in Figure 18 and Figure 19 for the filter parameters found. A completely blue square indicates a strong inverse relationship between the features. A completely red square indicated a strong

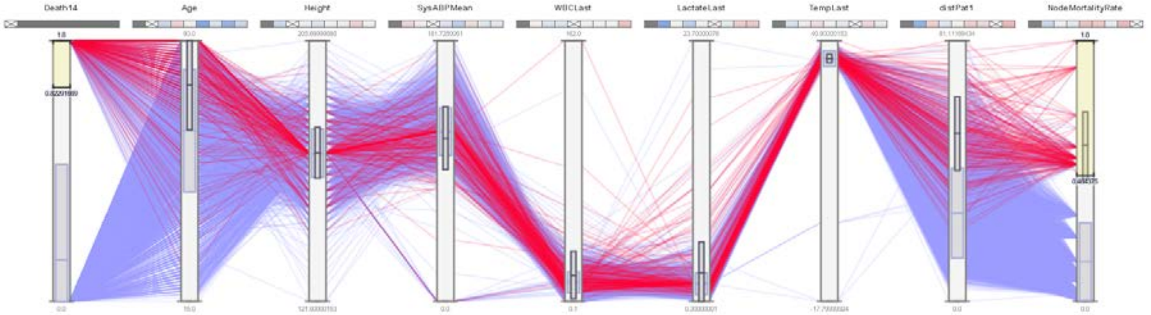


Figure 15. Data relationships with true positives highlighted in red for Death14

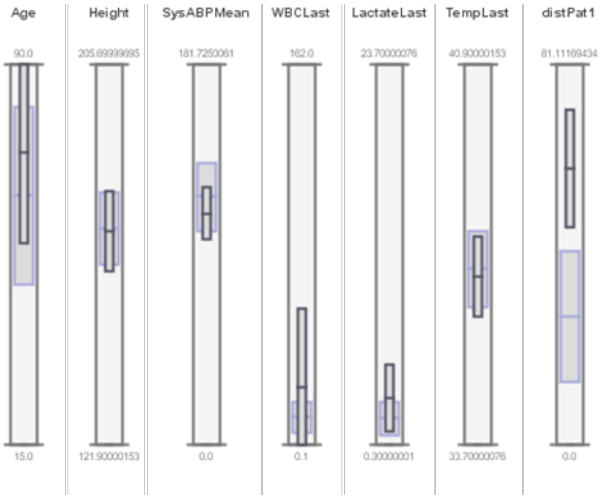


Figure 16. Distributions for full data set (dark bars) and true positives (light blue bars)

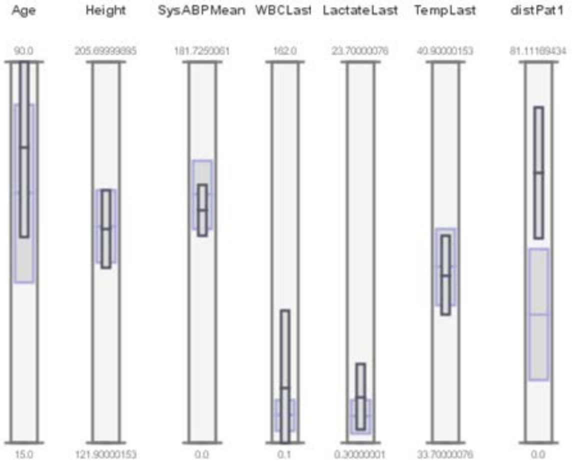


Figure 17. Distributions for full data set (dark bars) and mortalities (light blue bars)

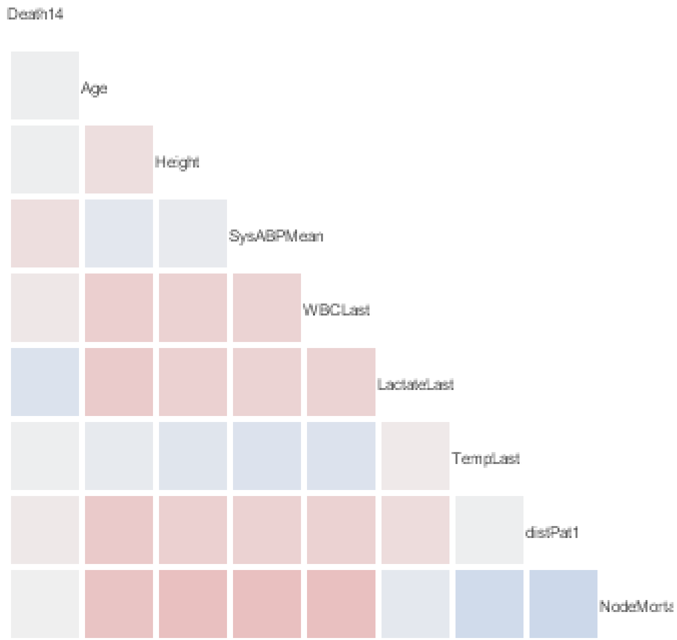


Figure 18. Correlation matrix for true positives only

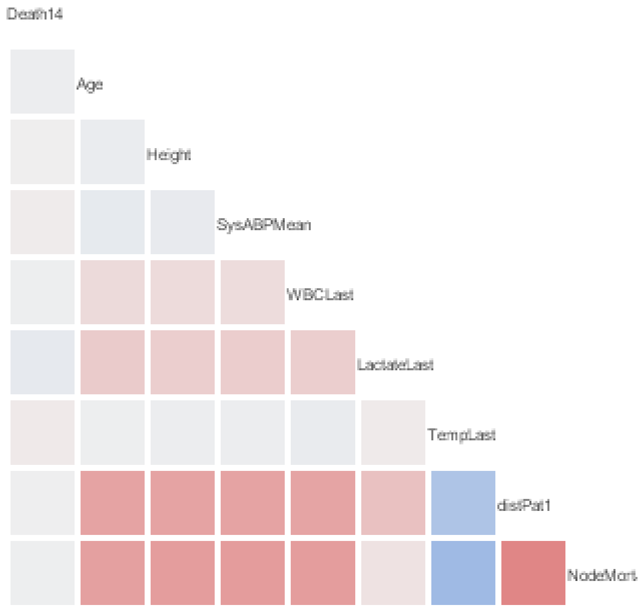


Figure 19 Correlation matrix for all data points

positive relationship between the features.

Revised Mortality Prediction Algorithm

The mortality prediction algorithm used to predict mortality was basic and not optimized. A revised algorithm was implemented as follows:

1. Calculate centers for all nodes
2. Assign each patient to a single node based on their proximity to the node centers
3. If the mortality rate for the node is >=threshold, predict the patient is deceased
4. If the mortality rate for the node is <threshold, predict the patient survives

A threshold is calculated for each model to maximize the score. Using this updated prediction algorithm with the optimized clusters, the following revised Event 1 scores were achieved:

- In-hospital Death: 0.42 with threshold = 0.3
- Death14: 0.37 with threshold = 0.1

CHAPTER VI
DISCUSSION

The model created using in-hospital death as the trained outcome has the following findings. A significant inverse relationship between age and the last lactate value was found for the true positives. A significant relationship was also found between the node mortality and maximum glucose value. These features of the cluster could be used to identify the reliability of the data in the node as these relationships are not present for the greater group of mortalities as shown in Figure 13. Based on the statistics, there are not a large number of members in the non-clustered groups, and there are enough nodes to spread out the deceased and survivors to allow the algorithm to differentiate. Based on these two facts, the model is considered valid for greater application.

The model created using Death14 as the trained outcome has the following findings. The last values for WBC last and Lactate last show slightly elevated levels in deceased patients. They are also well-differentiated by their position from patient 1.Based on the statistics, there are not many members in the non-clustered groups, and there are enough nodes to spread out the deceased and survivors to allow the algorithm to differentiate. Based on these two facts, the model is considered valid for greater application.

Comparing the two models, in-hospital mortality and Death14, both performed at similar levels of accuracy (0.42 and 0.37) While these were low compared to many of the other models used, they achieved the intent of discovering relationships within the data set with performance comparable to the currently a currently used system. This system could have been trained to predict the given set and achieve a score of 1.00, however the models would not have been more broadly applicable.

Validity of Model

Only a single data set was available for this project, so, the training and test sets are identical. The minimal in-hospital death records does not allow the data set to be separated into representative training and test sets. Because of this, it is important to evaluate the validity of the model. On one end of the spectrum, we could have created a 4000 cluster space that would have essentially worked as a lookup table for all of the patients and would have yielded 100% prediction accuracy, but would not be applicable to another data set. On the other end, we would have a single cluster with all the patients in it that would have 0% prediction accuracy. A reasonable model in our case will have minimal single-patient nodes with the vast majority of the patients in nodes with more than ten to twenty patients. This creates a model that is general enough that it can be used to predict the mortality of a test patient introduced to the population. The solution presented satisfies this requirement.

Improvements in Mapper Efficiency

In the process of developing this algorithm, there was the need for a more efficient Mapper function to allow for rapid iteration with the full data set. Mapper was executed for subsets of the available data. As the number of patients increased, the runtime increased exponentially. The data trend indicated a two month run time when evaluating Mapper using all 4000 patients. This was due to the way the matrices were being handled within the function. There existed a matrix definition that was later converted to a sparse matrix. This code was replaced by code that initially defined the matrix as a sparse matrix. The results of the Mapper code were compared to the original mapper code. The result is a mapper function that can fully execute in about 120 seconds. This is an improvement of four orders of magnitude on runtime. All results are identical for a sample set of 20 different subsamples of the total data set. A comparison of algorithm performance is shown in Figure 20. This code change is described in Appendix A.

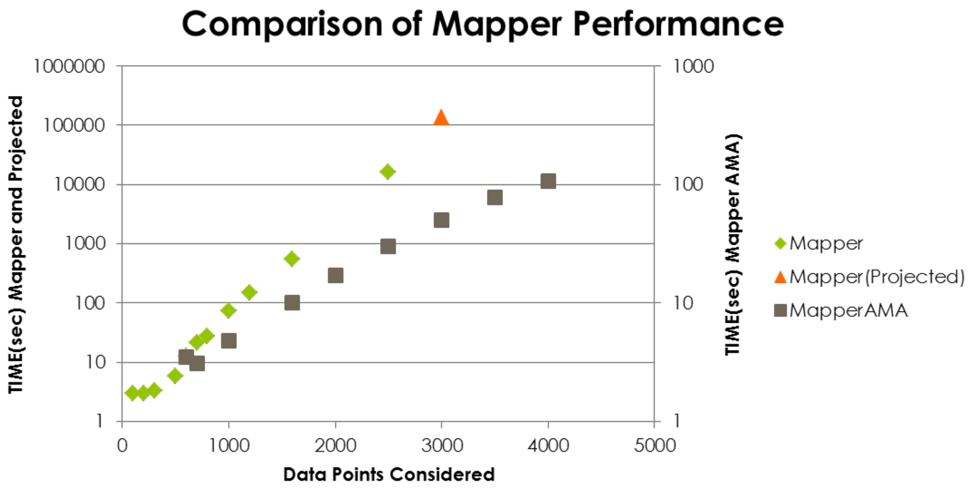


Figure 20. Comparison of Mapper function to revised Mapper function.

CHAPTER VII
CONCLUSIONS

The system developed was able to achieve a score of 0.42 for the PhysioNet Event1 Score after only a few generations of optimization. At this time, this model performs as well as a currently used patient condition evaluation system, but not as well as the neural network solutions presented in the literature review that regularly achieved scores in excess of 0.50. This process however does reveal relationships within the data and provide differentiation features for the true positive data, should this method be applied in a diagnostic setting. This topological approach can be further developed to achieve better performance. Future developments are discussed in the following section.

CHAPTER VIII
RECOMMENDATIONS

The following are recommendations to improve on and further develop this work. A combination of these approaches could be applied to improve the performance of this method. This method could also be applied outside of the ICU setting with any set of features for a data set.

Develop Better Prediction Algorithm Using Current Mapper Output

Such an algorithm will be aware of the data cluster shape, if the point was located at one end of a chain or the other, and be aware of any gradient or tendency along that path. This information can be extracted from the adjacency matrix.

Optimize Mapper Parameters to Increase Performance

The mapper configuration can be optimized to minimize the number small nodes disconnected from a cluster. This will either cause the smaller nodes to coalesce into the main cluster or into smaller clusters. This will improve the validity of the algorithm.

The mapper configuration can also be optimized to increase the number of nodes in the main cluster. This can be done by increasing the number of filter samples.

Solve for All Cases in Current Space Using Super Computing

If a super computer were available, it would be possible to simply solve for all possible filters for a specific mapper configuration and see if a better performing solution was available.

Allow for the Application of Weights to Filters

All of the features involved in the filters were either completely on or completely off. Future work could consider applying weights to the filters instead of simply activating or removing them.

Use mortality predictions from multiple filter functions

A single 1-D topology from Mapper was used to predict mortality. This process worked to optimize a single filter. This can be extended to use a set of different filters, extract the mortality rates from each topology, and compare them. One can imagine that a specific topology may predict mortality for a specific age, gender, or ICU type well, but another topology may work better for another case. This scoring function can apply weights to each graph based on ICU type, age, or gender for the patient.

LIST OF REFERENCES

[1] 2015 USDA Nutrition Evidence Library. Adults and Sodium: What is the relationship between sodium and blood pressure in adults aged 19 years and older?. Printed on: 03/18/15 - from: http://www.nel.gov/evidence.cfm?evidence_summary_id=250164

[2] Chad A. Steed, Galen Shipman, Peter Thornton, Daniel Ricciuto, David Erickson, and Marcia Branstetter. “Practical Application of Parallel Coordinates for Climate Model Analysis.” In Proceedings of the International Conference on Computer Science, June 2012, pp. 877-886. DOI: 10.1016/j.procs.2012.04.094.

[3] Cios K, Moore G. Uniqueness of medical data mining. Artif Intell Med 2002;26:1-24.

[4] Dasta JF, McLaughlin TP, Mody SH, Piech CT. 2005 Daily cost of an intensive care unit day: the contribution of mechanical ventilation. Crit Care Med 2005;33(6):1266-71.

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

[6] Hand D, Mannila H, Smyth P. Principles of data mining. Cambridge, MA, USA: MIT Press, 2001.

[7] Jean-Roger Le Gall, MD; Stanley Lemeshow, PhD; Fabienne Saulnier, MD. (1993). A New Simplified Acute Physiology Score (SAPS II) Based on a European/North American Multicenter Study. JAMA. 1993;270:2957-2963

[8] Knaus WA, Draper EA, Wagner DP, Zimmerman JE (1985). "APACHE II: a severity of disease classification system". Critical Care Medicine 13 (10): 818–29. doi:10.1097/00003246-198510000-00009. PMID 3928249.

[9] Lopez AD, Salomon J, Ahmad O, Murray CJL, Mafat D. Life Tables for 191 Countries: Data, Methods, and Results. World Health Organization. GPE Discussion Paper Series: No. 9

[10] MATLAB Release 2014a, The MathWorks, Inc., Natick, Massachusetts, United States.

[11] Rodes J, Benhamou JP, Blei A, Reichen J, Rizzetto M. The Textbook of Hepatology: From Basic Science to Clinical Practice, 3rd Edition. August 2007, Wiley-Blackwell. ISBN: 978-1-4051-2741-7

[12] Solomon CG, Manson JE. Obesity and Mortality: a review of the epidemiologic data. Am J Clin Nutr 1997;66(suppl):1044S-50S

[13] Singh G, Memoli F, Carlsson G. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. Point Based Graphics 2007, Prague, September 2007.

[14] Soares M, Fontes F, Dantas J, Gadelha D, Cariello P, Nardes F et al. (2004). "Performance of six severity-of-illness scores in cancer patients requiring admission to the intensive care unit: a prospective observational study.". Crit Care 8 (4): R194–203. doi:10.1186/cc2870. PMC 522839. PMID 15312218.

[15] Whitley, Darrell. A Genetic Algorithm Tutorial. 1993

[16] Xia H, Daley BJ, Petrie A, Zhao X. A neural network model for mortality prediction in ICU. Computing in Cardiology (CinC), 2012 , vol., no., pp.261,264, 9-12 Sept. 2012

[17] Alistair EW Johnson, Nic Dunkley, Louis Mayaud, Athanasios Tsanas, Andrew A Kramer, Gari D Clifford. Patient Specific Predictions in the Intensive Care Unit using a Bayesian Ensemble

[18] Cheng H Lee, Natalia M Arzeno, Joyce C Ho, Haris Vikalo. An Imputation-Enhanced Algorithm for ICU Mortality Prediction

[19] Luca Citi, Riccardo Barbieri. PhysioNet 2012 Challenge: Predicting Mortality of ICU Patients using a Cascaded SVM-GLM Paradigm

[20] Sean McMillan, Chih-Chun Chia, Alexander Van Esbroeck, Ilan Rubinfeld, Zeeshan Syed. ICU Mortality Prediction using Time Series Motifs

[21] Srinivasan Vairavan, Larry Eshelman, Syed Haider, Abigail Flowers, Adam Seiver. Prediction of Mortality in an Intensive Care Unit using Logistic Regression and a Hidden Markov Model

[22] Chucai Yi, Yi Sun, Yingli Tian. CinC Challenge: Predicting In-hospital Mortality of Intensive Care Unit by Analyzing Histogram of Medical Variables under Cascaded Adaboost Model

[23] Michael Krajnak, Joel Xue, Willi Kaiser, William Balloni. Combining Machine Learning and Clinical Rules to Build an Algorithm for Predicting ICU Mortality Risk

[24] Erika Severeyn, Miguel Altuve, Francisco Ng, Carlos Lollett, Sara Wong. Towards the Prediction of Mortality in Intensive Care Units Patients: a Simple Correspondence Analysis Approach

[25] Martin Macas, Jakub Kuzilek, Tadeáš Odstrčilík, Michal Huptych. Linear Bayes Classification for Mortality Prediction

[26] Luigi Y Di Marco, Marjan Bojarnejad, Susan T King, Wenfeng Duan, Costanzo Di Maria, Dingchang Zheng, Alan Murray, Philip Langley. Robust Prediction of Patient Mortality from 48 Hour ICU Data

[27] Antonio Bosnjak, Guillermo Montilla. Predicting Mortality of ICU Patients using Statistics of Physiological Variables and Support Vector Machines

[28] Tom J Pollard, Louise Harra, David Williams, Steve Harris, Demetrio Martinez, Kevin Fong. 2012 PhysioNet Challenge: An Artificial Neural Network to Predict Mortality in ICU Patients and Application of Solar Physics Analysis Methods

[29] Steven L Hamilton, James R Hamilton. Predicting In-Hospital-Death and Mortality Percentage using Logistic Regression

[30] Deep Bera, Mithun Manjnath Nayak. Mortality Risk Assessment for ICU Patients using Logistic Regression

[31] Jianfeng Xu, Dan Li, Yuanjian Zhang, Admir Djulovic, Yu Li, Youjie Zeng. CinC Challenge: Cluster Analysis of Multi-Granular Time-Series Data for Mortality Rate Prediction

[32] Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, Roger G Mark. Predicting In-Hospital Mortality of Patients in ICU: The PhysioNet/Computing in Cardiology Challenge 2012

APPENDIX

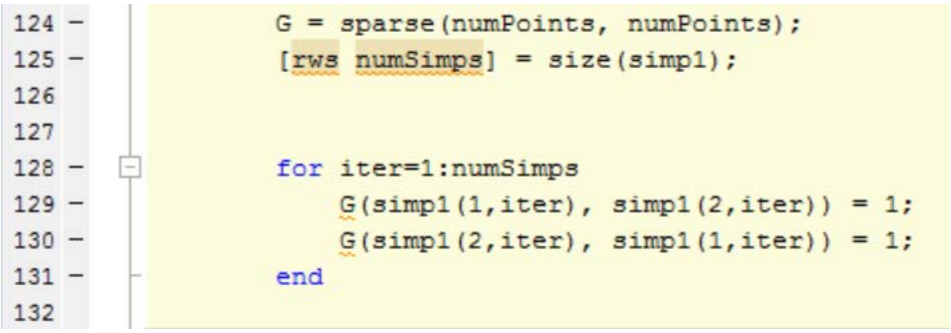
Appendix A:
Modifications to Main Mapper Function

The main Mapper function has been modified in the following ways:

Previously, the main function Mapper.m created by Gurjeet Singh contained the code in Figure A1.

This definition of the matrix and traversal of the matrix incurs a large number of unnecessary computations. This code has been replaced by the code in Figure A2:

This change in code leads to the definition of the sparse matrix with the required data. The for loop from the origonal code is no longer needed to populate the matrix. This results in the code running in 120 sec on average for this data set vs. the projected two months it would have taken based on the results shared in the document above.

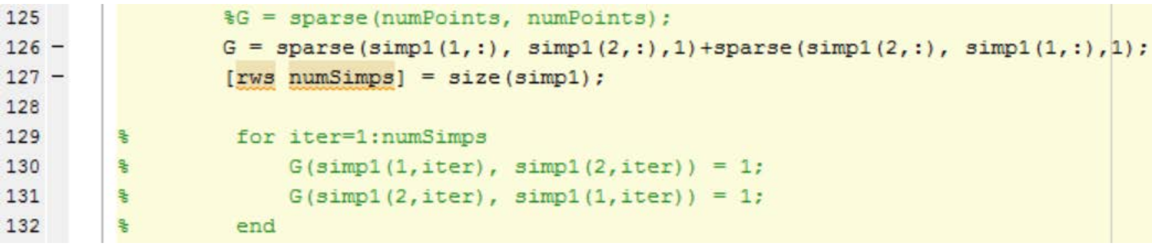


```
        G = sparse(numPoints, numPoints);
        [rws numSimps] = size(simp1);


        for iter=1:numSimps
            G(simp1(1,iter), simp1(2,iter)) = 1;
            G(simp1(2,iter), simp1(1,iter)) = 1;
        end

```

Figure A1: Origonal Mapper Code



```
        %G = sparse(numPoints, numPoints);
        G = sparse(simp1(1,:), simp1(2,:),1)+sparse(simp1(2,:), simp1(1,:),1);
        [rws numSimps] = size(simp1);

%        for iter=1:numSimps
%            G(simp1(1,iter), simp1(2,iter)) = 1;
%            G(simp1(2,iter), simp1(1,iter)) = 1;
%        end
```

Figure A2: Optimized Mapper Code

VITA

Adam M. Aaron was born in Saint Marys, Pennsylvania. After graduating from Saint Marys Area High School in 2005, he attended Carnegie Mellon University in Pittsburgh, Pennsylvania. Upon graduating with his B.S. in Mechanical Engineering in 2009, Adam moved to Knoxville, TN to begin his career as a Researcher at Oak Ridge National Laboratory in Oak Ridge, Tennessee. He enrolled at University of Tennessee, Knoxville in 2009 in pursuit of his M.S. in Mechanical Engineering. This thesis, published in May of 2015 is the culmination of his M.S. in Mechanical Engineering.