APPLICATION OF MACHINE LEARNING TECHNIQUES FOR PROGNOSIS OF
TRAUMATIC BRAIN INJURY PATIENTS IN INTENSIVE CARE UNITS

by

Sina Ehsani

Copyright © Sina Ehsani 2018

A Thesis Submitted to the Faculty of the
DEPARTMENT OF SYSTEMS AND INDUSTRIAL ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN INDUSTRIAL ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

2018
STATEMENT BY AUTHOR

The thesis titled *Application Of Machine Learning Techniques For Prognosis Of Traumatic Brain Injury Patients in Intensive Care Units* prepared by *Sina Ehsani* has been submitted in partial fulfillment of requirements for a master’s degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that an accurate acknowledgement of the source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED: Sina Ehsani

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Vignesh Subbian, PhD
Assistant Professor
Department of Biomedical Engineering
Department of Systems and Industrial Engineering

May 9, 2018
Date
## Table of Contents

ABSTRACT ........................................................................................................................................... 5

CHAPTER 1: INTRODUCTION ........................................................................................................... 6
  1.1 Introduction .................................................................................................................................. 6
  1.2 Research Objectives ..................................................................................................................... 6
  1.2.1 Case Study I Overview ........................................................................................................... 7
  1.2.2 Case Study II Overview .......................................................................................................... 7
  1.2.3 Overall Contribution ............................................................................................................... 8

CHAPTER 2: BACKGROUND .............................................................................................................. 9
  2.1 Biomedical Data and Informatics ............................................................................................... 9
  2.2 Relevant Machine Learning Methods .......................................................................................... 10
  2.2.1 Prior Studies for Prediction of Clinical Outcomes ................................................................. 10
  2.2.2 Prior Studies for Clinical Phenotyping and Pattern Recognition .......................................... 12
  2.3 Application of Machine Learning in TBI ................................................................................... 14

CHAPTER 3 CASE STUDY I: Prediction of Mortality in Critically-ill Traumatic Brain Injury Patients ................................................................................................................................. 15
  3.1 Data Sources .............................................................................................................................. 15
  3.2 Methods ..................................................................................................................................... 15
  3.2.1 Naïve Bayes .......................................................................................................................... 15
  3.2.2 Random Forest ...................................................................................................................... 16
  3.2.3 Artificial Neural Network ....................................................................................................... 16
  3.2.4 Logistic Regression ............................................................................................................... 17
  3.2.5 Sequential Minimal Optimization ......................................................................................... 17
3.2.6 Bagging Algorithm.........................................................................................................18
3.2.7 Voting...............................................................................................................................18
3.2.8 Hyperparameters .............................................................................................................19
3.3 Results .................................................................................................................................20
3.3.1 Data Extraction ................................................................................................................20
3.3.2 Performance Evaluation ..................................................................................................21
3.4 Summary ...............................................................................................................................24

CHAPTER 4 CASE STUDY II: Subspace Clustering of Physiological Data from Acute TBI Patients ..........................................................................................................................25

4.1 Data Sources .......................................................................................................................25
4.2 Methods ...............................................................................................................................25
4.2.1 Subspace Clustering ..........................................................................................................25
4.2.2 Evaluation .........................................................................................................................29
4.3 Results .................................................................................................................................29
4.3.1 Data Extraction ................................................................................................................29
4.3.2 Evaluation of Sub-clustering Algorithms .........................................................................34
4.3.3 Comparison of Subspace Clustering Algorithms ............................................................37
4.4 Summary ...............................................................................................................................37

CHAPTER 5: CONCLUSION AND DISCUSSION ....................................................................38

References .................................................................................................................................39
ABSTRACT

With advances in digital health technologies and proliferation of big biomedical data in recent years, applications of Machine Learning (ML) in healthcare and medicine have gained significant attention. Modern Intensive Care Units (ICUs), in particular, are equipped to generate rich multimodal clinical data on critically-ill patients. In this thesis, we focus on applying machine learning techniques for prognostication of Traumatic Brain Injury (TBI) patients in ICU, which is the leading cause of death and disability among children and adults of age less than 44. We present two case studies to demonstrate the feasibility and applicability of machine learning techniques: one for mortality prediction in TBI patients and the second for extracting patterns from physiological data collected from TBI patients. For the case study I, clinical data including demographics, vital signs, and physiological data for the first 72 hours of TBI patients were extracted from the Medical Information Mart for Intensive Care III (MIMIC III) database. Several traditional supervised machine learning algorithms such as artificial neural network, support vector machine, and logistic regression were employed to construct prediction models. Bagging and Voting techniques were implemented to improve the performance of these algorithms. By comparing the performances of these algorithms, we showed that deploying voting techniques on several different ML models can improve the overall performance. These algorithms obtained the highest Area Under receiver operating characteristic Curve (AUC) of 0.91. For the case study II, an exploratory, secondary analysis of physiologic data of TBI patients from the Phase III trial of Progesterone for Traumatic Brain Injury, Experimental Clinical Treatment (PROTECT) was performed. Subspace clustering was used to extract relationships between various physiologic variables. For both studies, 10-fold cross validation was used for evaluation purposes.
CHAPTER 1: INTRODUCTION

1.1 Introduction

In 2009, the implementation of the Health Information Technology for Economic and Clinical Health (HITECH) Act [1] led to widespread deployment of Electronic Health Records (EHR) across U.S health system. This has increased the availability of digital biomedical data to improve clinical care as well as aid research.

One of the areas noted in the HITECH Act require that EHRs provide appropriate information to help guide clinical decisions at the time and place of care [1]. The lack of EHR systems and reliable digital data in the past have limit the design and deployment of tools such as decision support systems that can give personalized recommendations. Now, with the availability of large volumes of biomedical data and advances in machine learning techniques, designing and evaluating comprehensive analytical tools and computerized algorithms are becoming more viable. However, there exists several challenges in the way that data are gathered and stored, as well as the lack of relevance to specific clinical conditions. While EHRs are generally resourceful, EHR data are hard to mine and interpret, especially when data points are missing, irregularly sampled, and in some cases, are not disease-specific. It should also be noted that different clinical settings such as Intensive Care Units (ICUs), Emergency Department (ED) etc., collect and use data differently based on clinical needs. Therefore, in some cases, specialized studies are more appropriate. In this thesis, we consider Traumatic Brain Injury (TBI) as a context for demonstration purposes.

1.2 Research Objectives

Traumatic Brain Injury is broadly defined as disruption in normal brain function or other evidence of brain pathology as a result of external force or trauma. In addition to increase in head injury incidence, TBI is the most common cause of death and disability in children and adults of age less than 44 [2] [3]. However, there has been a slight decrease in TBI-related deaths as a result of advancements in care delivery, particularly, the use of evidence-based guidelines for TBI
management [4]. Applying data mining and machine learning techniques have also shown to be promising for predicting outcomes and prognostication in TBI patients [5], [6].

The major objective of this thesis is to apply Machine Learning (ML) techniques for prognosis of critically-ill TBI Patients and extract valuable information from clinical data that can be used to potentially guide care givers’ decisions. Two case studies are presented to demonstrate the research objectives. In the following subsection, an overview of each case study and the overall contribution of the thesis are described.

1.2.1 Case Study I Overview

In the case study I, we evaluate and compare multiple machine learning algorithms to predict mortality in TBI patients. While there are several models for predicting mortality in TBI, they can be improved by considering the following two factors. First, several studies use admission data such as age and initial Glasgow Coma Scale (GCS) scores that do not account for time-series data. In TBI, unlike other types of injuries and diseases, most disabilities and deaths are caused by secondary injuries [7]. Since most of the secondary injuries occur days after admission, the initial admission data may be insufficient to prognosticate secondary injuries. Therefore, including time-series (temporal) data would be more beneficial. Second, manually collected data may be error-prone. A four-month mixed methods study in the ICU showed a total of 554 human errors during data gathering [8]. These two factors motivated the use reliable and time-stamped data from a large, publicly-available ICU database (MIMIC – III) [9]. Several machine learning algorithms such as artificial neural network and support vector machine are used to predict the TBI mortality rate at one month after admission.

1.2.2 Case Study II Overview

For the case study II, we apply clustering-based techniques to uncover potential relationships between physiologic data collected as a part of the clinical trial of TBI patients. We compared the performance of different subspace clustering methods (density-based, cell-based, and clustering-oriented). Results from this study will provide the foundation to develop scalable algorithms for further research and validation.
1.2.3 Overall Contribution

The focus of this thesis is to demonstrate the feasibility and application of machine learning algorithms for a specific clinical condition, Traumatic Brain Injury. Throughout the course of this thesis, we present models related to TBI, one that can predict patients’ outcome with high accuracy, and the another that can identify patterns between various physiological parameters. The various machine learning methods presented, together, provide a foundation for further research in clinical applications of computational models.
CHAPTER 2: BACKGROUND

2.1 Biomedical Data and Informatics

Bernstam et al. defined Biomedical Informatics as “the science of information applied to or studied in the context of biomedicine, where information is data with meaning.”[10] In the recent years, the proliferation of electronic health records (EHRs), as a result of the HITECH act [1], has not only provided information tools for clinical care, but also led to a data-intensive environment for biomedical research. Data collected in EHR systems typically includes patient’s demographics, vital sign measurements from bedside, results from laboratory and other physician orders, medication orders, caregiver (nurse, physician, and other clinician) notes, image and imaging reports, diagnosis, and patient outcome data [9], [11]. These data can be broadly categorized into two types: structured and unstructured.

Structured data are organized using predefined data models and enables effective entry, storage, and retrieval of data elements. However, it may not be possible to capture all data in a structured format. Unstructured data are those that do not have a pre-defined format or data models. Medical imaging data and caregiver notes are typical examples of such data.

Biomedical data can also be categorized into static and temporal data. Static data refer to data that are not dependent on time and do not change during the course of the patients’ stay. On the other hand, the temporal data are those that change over time during patients’ stay such as vital signs, physiological data, laboratory test results, medical imaging etc. Overall, different types of data are needed depending on research questions that are being considered. With advances in computational techniques and EHR systems, the potential to generate new knowledge from diverse and multimodal biomedical data to inform clinical care and research are promising [12].
2.2 Relevant Machine Learning Methods

This section is divided into two parts: (1) methods for prediction of clinical outcomes and (2) methods for identification of clinical markers and pattern recognition. For each study or methodology, we report the following: problem objective, dataset and patient population, relevant machine learning methods, evaluation, results, implications, and relevance to this thesis.

2.2.1 Prior Studies for Prediction of Clinical Outcomes

Razavian et al. proposed logistic regression to predict patients at risk for Type II diabetes. The initial population for this research was around 4.1 million individuals with age of at least 18 years. However, this number decreased to around 800,000 patients after applying the exclusion criteria. Patients who had already developed type II diabetes 6-month prior to prediction window were excluded. The study included 11 continuous and binary variables, including age, procedures, physician specialty visits, laboratory orders and results, and medication. Area Under Curve (AUC) was used as an evaluation metric. Logistic regression with L1 regularization yielded an AUC of 0.80 [13].

Henry et al. used supervised machine learning model to validate a targeted real-time early warning score (TREWScore) that identifies patients at high risk for developing septic shock. Data sources included physiological and laboratory data for 13,014 adult patients for model development set and 3011 patients for validation. Cox proportional hazards model was applied to their data with lasso regularization model. The model evaluation showed an AUC of 0.83 (95% confidence interval 0.81 to 0.85), specificity of 0.67, and sensitivity of 0.85, compared to Modified Early Warning Score which had an AUC of 0.73, and specificity of 0.63. The model identified patients in median of 28.2 hours before onset of septic shock, and 2/3 were identified before organ dysfunction [14].

Mundkur et al. used data from an open-access, de-identified dataset from ICU patients to examine the role of race on patient outcomes. The database used in this study, MIMIC II, included patients of at least 18 years of age and admitted to a general ICU. A total of 14,684 patients from the MIMIC II met the inclusion criteria and were included in the study. Multivariate logistic
regression was used to identify the effect of input variables on the outcome. The study showed that age and SAPS-II score varied significantly between different racial groups [15].

As evaluation of a given treatment is difficult for chronic diseases, the development of innovative treatments for these diseases relies on effective disease progression models [16]. Zhou et al. propose a multi-task learning formulation for predicting the disease progression measured by the cognitive scores and selecting markers predictive of the progression. Multi-task learning aims at improving the generalization performance by learning multiple related tasks simultaneously. The key of multi-task learning is to exploit the intrinsic relatedness among the tasks [17]. The dataset used in this study includes 675 patients with MRI data, CSF measurements, and clinical scores in 6-month intervals (for three years). In this study, samples that fail the MRI quality, or have missing entries were removed. Baseline feature values were used to predict progression markers for next three years (in 6-month intervals). The study used multitask learning formulation with group lasso regularization that has two components: an $l_{2,1}$ norm penalty, and a temporal smoothness term. This regularization will ensure that small subset of features are selected for the model at all time points. Results show that the proposed algorithm better captures the progression for Alzheimer’s Disease than other methods such as ridge regression specially in a large future space with small sample size [17].

Sukker et al. introduced a data-driven statistical framework based on Hidden Markov Models (HMMs) to model progression of diseases using the same dataset as the Zhou et al. for patients Alzheimer’s Disease. This study showed that trained HMM is able to model disease progression more precisely than currently defined clinical stages without the need of labeled data [16].

Lipton et al. used Long Short-Term Memory (LSTM) based Recurrent Neural Networks (RNNs) to recognize patterns in multivariate time series clinical data. Irregularly sampled clinical measurements from 10,401 PICU episodes were used in this study. Each episode varied in length from 12 hours to several months and included 13 variables. The final network had 2 hidden layers with either 64 memory cells per layer with no dropout or 128 cells per layer with dropout of 0.5. Al models were trained on 80% of the dataset, tested on 10%, and validated using the remaining 10% of the data. Three performance metrics (micro AUC, micro F1, and precision) were used for
evaluation, and LSTM-based RNN model was compared to traditional models such as Multi-Layer Perceptron trained on hand-engineered features and logistic regression. Results showed that LSTM with dropout outperformed other models with micro AUC of 0.86 [18].

Choi et al. presented an algorithm that can capture temporal relations (i.e. disease diagnosis, medication orders, procedure orders, etc.). Gated recurrent units (GRUs) based recurrent neural network (RNN) model was used to detect relationships among time-stamped clinical events. Longitudinal data including diagnosis and medication orders with a 12- to 18-month observation window were included for both cases and controls. A total of 265,336 primary care patients were used for training the medical concept vectors, and heart failure cases and controls were used for all other model training and evaluation tasks. The dataset was divided into training, validation, and testing sets with the ratio of 5:1:1 respectively. The AUC the RNN model was 0.777, compared to AUCs for logistic regression (0.747), multilayer perceptron (MLP) with one hidden layer (0.765), support vector machine (SVM) (0.743), and K-nearest neighbor (KNN) (0.730). With an 18-month observation window, the AUC for the RNN model increased to 0.883 and was significantly higher than the 0.834 AUC for the best of the baseline methods (MLP). Results show that artificial neural network models that are adapted to include time stamped connections (RNN) seem to have higher performance for detection of incident heart failure with a short observation window compared to other models [19].

2.2.2 Prior Studies for Clinical Phenotyping and Pattern Recognition

Several clustering-based and other unsupervised methods exist for clinical phenotyping and pattern recognition. Joshi et al. introduced an unsupervised learning method that can scale and summarize enormous amounts of clinical data to show deteriorations or improvements in multiple organ systems. Traditional clustering models were used with two improvements: (1) removing feature selection at early stages and using feature selection on the unsupervised feature layer (transferring it from first step into an outer layer) and (2) grouping ICU patients based on their characteristic and then scaling it up to high dimensional data. These were due to the fact that the traditional models do not perform well in large scaled data. This study used a
subset of data from MIMIC II database and demonstrated severity states by organ systems. In addition, for secondary analysis, real-time mortality prediction model was constructed and compared to traditional classifiers such as using SAPS II with logistic regression. The model was trained using 70% of the data and tested with the remaining 30%, along with using five-fold cross validation. Results showed that using logistic regression on their proposed model performed with AUC of 0.89 compared to logistic regression on SAPS II (AUC of 0.81). Also, the performance on high severity group was AUC of 0.91 for their model and AUC of 0.77 for SAPS II. The study demonstrated that their model can outperform traditional clustering models when dealing with a high dimensional data [20].

Halder et al. explored the application of k-means cluster analysis for identifying distinct phenotypes or subgroups of asthma patients. Three different patient populations with asthma (less than 500 patients in total) were included in the study. After finding the number of clusters using hierarchical clustering, K-means was used as the principle clustering algorithm. To compare different clusters, statistical tests such as ANOVA for parametric variables, $X^2$ test for proportions, Kruskal-Wallis for nonparametric variables, and t-test for outcome analysis of different cluster arms were used. Results showed the clustering algorithms were able to identify three unique clusters or subgroups of asthma patients [21].

Cohen et al. showed the potential of using machine learning techniques on ICUs. The study used minute-to-minute multivariate physiologic and ventilator ICU data. The final cohort after applying the study criteria and removing missing values included 52,000 data points for 14 different physiological values. Cohen et al. found relations between different physiological variables using hierarchical clustering machine learning method, and showed that clusters were associated with outcome measurements such as incidence of infection, multiple organ failure (MOF), and mortality. Further analysis of the study showed that while some of the clustered physiological lab results relations were already supported by clinical evidence, some new relationships were found in this study. The significant difference between physiological variables between each clusters were founded in this study. They also showed the changes in physiological variables, predicts the patients expire flag [22].
2.3 Application of Machine Learning in TBI

Several data mining and machine learning techniques have been applied to TBI in prior studies. Liu et al. conducted a systematic review on machine learning techniques for predicting outcomes in trauma patients, and summarized 65 observational studies. The review highlighted the impact of ML will require further validation, establishment of common performance criteria, and high-quality evidence about clinical and economic impacts before ML can be widely used and accepted in practice [23].

Lu et al. have used data mining methods while using serial Glasgow Coma Scale (GCS) scores (day 1, 7, and 15 scores), clinical and laboratory parameters to predict mortality rate and functional outcome after a 6-month period on 115 patients. Artificial neural network (ANN), naïve Bayes (NB), decision tree, and logistic regression were compared in this study. Results showed that ANN was the best model for predicting functional outcome with AUC of 96.13%, and, NB was the best predictive model with AUC of 89.73% for mortality prediction [24].

Guiza et al. developed a predictive model using continuous mean arterial pressure (MAP) and intracranial pressure monitoring (ICP) from 264 TBI patients that were admitted to 22 different neuro-ICUs. Glasgow Outcome Score (GOS) of at 6-month after admission was used as the outcome variable. This study had two different goals for its prediction, a short-term (increased intracranial pressure episodes 30 minutes before occurrence) and a long-term (poor neurologic outcome (GOS of 1-2) at six-month period). Multivariate logistic regression and Gaussian processes were able to predict the short term increased intracranial pressure with AUC of 0.87. For predicting the 6-month outcome based on static initial data, the model had 0.72 of AUC. However, by adding the first 24 hours of dynamic data, they were able to significantly increase the AUC to 0.90 [25].
CHAPTER 3 CASE STUDY I: Prediction of Mortality in Critically-ill Traumatic Brain Injury Patients

3.1 Data Sources

Medical Information Mart for Intensive Care III (MIMIC III) [9] is an open-source, de-identified database developed and maintained by the Laboratory for Computational Physiology at MIT for public usage [9]. This database includes a diverse and large population of ICU patients, including 53,423 unique hospital admissions to intensive care units (ICUs) from 38,597 adult patients, collected between 2001 and 2012. MIMIC-III data encompasses a wide range of data (both structured and unstructured) such as demographics, laboratory test results, vital signs, medications data and clinician notes. Additionally, this database is well-documented, and includes an open access repository [26] with codebooks and examples for researchers.

3.2 Methods

The following algorithms were used in this case study to predict the mortality of TBI patients: Naïve Bayes, Random Forest, Logistic Regression, Multilayer Perceptron Neural Network, and Sequential Minimal Optimization.

3.2.1 Naïve Bayes

Naïve Bayes [27] is one of the first, simplest, and efficient machine learning algorithms. Naïve Bayes (NB) uses a conditional probabilistic approach based on Bayes Theory. Using the following equation, Naïve Bayes takes the most probable outcome as its classification:

\[
p(y|x) = \arg\max_{k \in \{1, \ldots, K\}} \{p(y_k) \prod_{i=1}^{n} p(x_i|y_k)\}
\]

Where \(p(y|x)\) is the maxima of all arguments, \(p(x_i|y_k)\) is probability of \(x_i\) (vector of attributes), given \(y_k\) (outcome), this probability is calculated during training process. \(p(y_k)\) is the probability of outcome \(k\). After calculating the probability of different hypotheses, the algorithm selects the hypotheses with the highest probability.
3.2.2 Random Forest

Random Forest (RF) uses multiple decision trees during the training phase and outputs the aggregate result as the output [28], [29]. Using multiple decision trees reduces the possibility of overfitting. The maximum depth of tree is used as a hyperparameter for this method.

3.2.3 Artificial Neural Network

Artificial Neural Network (ANN) is a multi-layer architecture that works similar to neural system of the human brain. Each neuron in our brain has dendrites that collect input data and send those data to nucleus, which in turn processes and send them to axons that work as output units of the neuron. Similarly, ANNs have input units that form the input layer, followed by processing units called hidden layer, and finally an output layer as shown in “Figure 1”.

![Figure 1 Simple Structure of a Neural Network]

ANNs contain several key components including input, output, and hyperparameters as described here:

Input and output:

- $X$: The feature input matrix
- $Y$: The output matrix (labels)

Parameters:

- $b^{[L]}$: Bias is a parameter vector in the $L$th layer
- $W^{[L]}$: Weight is a parameter matrix in the $L$th layer
- $\hat{y}$: The predicted output vector.
Hyper Parameters:

- $n^{[L]}$: Number of hidden units of the Lth layer.
- $L$: Number of layers in the network.
- $g^{[L]}$: Lth layer activation function.
- $a^{[L]}$: Learning rate for gradient decent backward propagation.

The differences between hyper parameters and parameters is that parameters can be changed and optimized in each iteration of the model, while hyper parameters are fixed. Therefore, to get the best results from an algorithm, hyper parameters should be selected carefully. In this study, we used Multilayer Perceptron (MP) algorithm, which is a variation of feedforward ANN algorithm. For more information see [30], [31].

3.2.4 Logistic Regression

Logistic regression is a common linear classification algorithm that models the probability of a variable $Y$, a dependent binary variable, given a vector of independent variables $X$. A logit model is used to find out the probability the example belongs to the positive class:

$$\log\left(\frac{p}{1-p}\right) = \theta_0 + \theta_1X_1 + \theta_2X_2 + \cdots + \theta_nX_n$$

Where $p$ is the probability of $y=1$, $\theta_i$ is a weight vector of variables and $X_i$ is vector of dependent variables. To optimize the results, the logistic regression, use an optimization model on its parameters and their results. The hyper-parameters for this model include the optimization model that was used and its learning rate. The logistic regression algorithm has showed excellent performance in several classification problems. For more information, see [32].

3.2.5 Sequential Minimal Optimization

In this case study, we used a variation of Support Vector Machine (SVM) called Sequential Minimal Optimization (SMO), a supervised machine-learning algorithm. SVM can be used for both regression and classification problems. In SVM, every input is an n-dimensional point (n is number of parameters considered in the model) in such a way that value of each parameter can be
considered as the value of corresponding dimension. Then the algorithm tries to find the appropriate hyper-plane, which separate the space into distinct classes. More information can be found in [33], [34].

3.2.6 Bagging Algorithm

Bagging algorithm [35] divides the dataset into sub-samples and then applies several models on each sub-sample. For example, bagging algorithm can divide the dataset into 10 consecutive sub-sets and apply Naïve Bayes on each sub-set separately. The final results of prediction could be obtained by simple or weighted voting. See “Figure 2”.

![Figure 2 Bagging Technique](image)

3.2.7 Voting

Voting is a machine learning ensemble algorithm that can combine different algorithms to enhance the overall performance. The output of voting is aggregation of outputs of participating algorithms. In this case study, we used average of probabilities for the aggregation rule. See [37] for more details. See “Figure 3”.
3.2.8 Hyperparameters

Common hyperparameters for algorithms used in this case study are number of iterations, batch size, cross-validation. Number of iterations is the stopping criteria for each algorithm. Each iteration includes one forward and one backward propagation. Batch size is the size of input $X$ for each iteration.

Cross-validation is a technique that partitions the original sample data into training and testing sets for evaluation of the model. A $k$-fold cross-validation divides the input data $X$ into $k$ equal partitions with equal size. For the first run of the model only one of the $k$ partitions will be marked as test sample, the other $k-1$ samples will be used for training the model. This process is repeated for each of the partitions (folds). After $k$ runs of the algorithms, $k$ results will be generated, which are then averaged to produce a single outcome.

**Tools:** Waikato Environment for Knowledge Analysis (Weka) was used to implement and test the algorithms. Weka [38] is a free license java-based machine learning software developed and maintained by the University of Waikato, New Zealand.

**Input variables:** In this case study, three types of clinical data were considered as input features: (1) demographic data: age, sex, (2) physiologic data based on laboratory results:
glucose, bicarbonate, chloride, creatinine, potassium, sodium, hematocrit, hemoglobin, Prothrombin Time/ International Normalized Ratio (PT/INR), and platelet count, and (3) general monitoring data: heart rate and O2 Saturation Pulse-oximetry (SPO$_2$). For model evaluation, accuracy, True Positive (TP) rate, True Negative (TN) rate, precision, and Area Under Curve (AUC) were used [39].

3.3 Results

3.3.1 Data Extraction

There were 46,520 unique patients in the MIMIC III v1.4 databases. From these patients, we extracted patients with TBI as their primary diagnoses based on International Classification of Diseases (ICD9) codes. The exclusion criteria were: (1) age under 18 years old, 2) less than three days of ICU length of stay, and (3) patients without at least one value for each parameter. As a result, 2066 TBI patients were included in this case study. Of these, 1828 were alive at one month. (see “Figure 4”). The list of 14 different features after the data extraction are shown in “Table 1”.

![Figure 4 Patient Record Extraction from MIMIC III Database](image)

As a part of the de-identification process in the MIMIC III database, the age all the patients who were older than 89 years were coded as 300 years old. To reduce this bias, we considered
all these patients as 95 years old. Average, standard deviation, min, and max values for all parameters of each patient during first 72 hours of stay were computed. In addition to these parameters, we also included the difference between the initial and the last gathered value in the 72-hour window.

Table 1 Patient Characteristics based on data gathered during first 72 hours of stay

<table>
<thead>
<tr>
<th>Demographics n = 2066 (782 male and 1284 female)</th>
<th>Average</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>60.86</td>
<td>22.99</td>
</tr>
<tr>
<td>Physiologic data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glucose</td>
<td>132.40</td>
<td>50.74</td>
</tr>
<tr>
<td>Bicarbonate</td>
<td>25.56</td>
<td>3.91</td>
</tr>
<tr>
<td>Chloride</td>
<td>104.73</td>
<td>5.99</td>
</tr>
<tr>
<td>Creatinine</td>
<td>1.04</td>
<td>0.95</td>
</tr>
<tr>
<td>Potassium</td>
<td>4.00</td>
<td>0.58</td>
</tr>
<tr>
<td>Sodium</td>
<td>139.69</td>
<td>5.34</td>
</tr>
<tr>
<td>Hematocrit</td>
<td>31.30</td>
<td>5.4</td>
</tr>
<tr>
<td>Hemoglobin</td>
<td>10.76</td>
<td>2.02</td>
</tr>
<tr>
<td>PT/INR</td>
<td>1.38</td>
<td>0.79</td>
</tr>
<tr>
<td>Platelet Count</td>
<td>269.21</td>
<td>173.52</td>
</tr>
<tr>
<td>Monitoring data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heart Rate</td>
<td>87.09</td>
<td>18.36</td>
</tr>
<tr>
<td>SPO2</td>
<td>97.79</td>
<td>2.98</td>
</tr>
</tbody>
</table>

3.3.2 Performance Evaluation

A 10-fold cross-validation method was used to evaluate the models. The Random Forest (RF) model batch size was set to 100, the maximum depth of tree was set to unlimited, and the number of iterations was set to 1,000. For the Multilayer Perceptron (MP), batch size was set to 200, number of iterations was set to maximum. The number of hidden layers was set to 4 and
the number of hidden units were set as (16,5,5,5) with respect to each hidden layer. The learning rate was set to 0.05, and the activation function of all the layers was sigmoid. For the Logistic Regression (LR) backward propagation, gradient descent with learning rate of 0.05 was used. For the Sequential Minimal Optimization (SMO) the number of dimensions was set to 2 and batch size to 100. For the bagging technique the number of sub sets was set to 10. As shown in “Table 2”, performance of our algorithms (highest AUC, 0.91) were superior to APACHE II, SAPS II, and IMPACT (highest AUC, 0.82) that were conducted in previous studies.

Table 2 Algorithms Performances

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy %</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
<th>ROC Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>92.80</td>
<td>0.99</td>
<td>0.43</td>
<td>0.93</td>
<td>0.91</td>
</tr>
<tr>
<td>MP (ANN)</td>
<td>91.30</td>
<td>0.95</td>
<td>0.57</td>
<td>0.94</td>
<td>0.87</td>
</tr>
<tr>
<td>LR</td>
<td>91.20</td>
<td>0.98</td>
<td>0.47</td>
<td>0.93</td>
<td>0.89</td>
</tr>
<tr>
<td>NB</td>
<td>88.00</td>
<td>0.91</td>
<td>0.68</td>
<td>0.96</td>
<td>0.87</td>
</tr>
<tr>
<td>SMO</td>
<td>92.30</td>
<td>0.99</td>
<td>0.39</td>
<td>0.93</td>
<td>0.69</td>
</tr>
<tr>
<td>B^2-RF</td>
<td>92.70</td>
<td>0.99</td>
<td>0.42</td>
<td>0.93</td>
<td>0.91</td>
</tr>
<tr>
<td>B-LR</td>
<td>92.40</td>
<td>0.98</td>
<td>0.48</td>
<td>0.94</td>
<td>0.90</td>
</tr>
<tr>
<td>V^b-RF, LR, NB</td>
<td>92.30</td>
<td>0.97</td>
<td>0.58</td>
<td>0.94</td>
<td>0.91</td>
</tr>
<tr>
<td>V-MP, LR, NB</td>
<td>90.70</td>
<td>0.97</td>
<td>0.56</td>
<td>0.94</td>
<td>0.90</td>
</tr>
<tr>
<td>V-MP, NB</td>
<td>90.20</td>
<td>0.94</td>
<td>0.63</td>
<td>0.94</td>
<td>0.89</td>
</tr>
</tbody>
</table>

a. B stands for Bagging technique.  
b. V stands for Voting technique.

As shown in the “Table 2”, among the algorithms those used bagging strategy showed slightly better results. However, using voting strategy improved the results of both Multi-Perceptron algorithms (AUC, 0.87) and Naïve Bayes (AUC, 0.87). The combination of MP and NB methods with voting strategy obtained better performance (AUC, 0.89). RF, RF with bagging strategy, and RF combined by voting strategy with LR and NB algorithms obtained the highest AUC (0.91). AUC of other algorithms are shown in “Figure 5”.

22
Among the RF, SMO, and B-RF have best TP rate (0.99). SMO had the least TN rate (0.39). Naïve Bayes has the highest TN rate (0.68). The results for all algorithms are shown in “Figure 6”.

Figure 5 ROC Area Under the Curve for Each Algorithm

Figure 6 TP and NP Rates for Each Algorithm
3.4 Summary

In this case study, different machine learning algorithms to predict the mortality in TBI patients was implemented and evaluated. Of all the algorithms in this study, Random Forest models had the best performance with AUC of 0.91. Results from this preliminary study will inform development of integrative models for TBI prognosis.
CHAPTER 4 CASE STUDY II: Subspace Clustering of Physiological Data from Acute TBI Patients

4.1 Data Sources

The Progesterone for Traumatic Brain Injury, Experimental Clinical Treatment – Phase III (PROTECT III) trial includes a cohort of 882 TBI patients [40], originally recruited for a randomized clinical trial to study the effect of progesterone on patients with acute TBI. Patients were randomly assigned to a treatment group that received progesterone within 4 hours after injury or a placebo group. While the PROTECT III clinical trial showed that there were no difference in patient groups between the two study groups, the longitudinal data from the trial were made available for secondary analyses and research. This dataset includes patient demographics, baseline assessment data, 6-month outcome data including the Glasgow Outcome Scale Extended (GOS-E) survey and mortality status. The temporal data in this study included variables such as daily measurement of Glasgow Coma Score (GCS) for the first 30 days of stay and laboratory test results for first 7 days of stay.

4.2 Methods

In this case study, we implemented and evaluated Subspace Clustering (or subclustering) methods on the PROTECT III data. Specifically, we compared the performance of three subspace clustering methods, density-based, cell-based, and clustering-oriented technique.

4.2.1 Subspace Clustering

Subspace or Projective Clustering is a clustering method that emphasizes on clustering in subspaces of high dimensional spaces i.e., it tries to find clusters in smaller subspaces and builds up to form larger clusters by using overlapping subspaces [41]. Subspace clustering can be classified into three main categories: density-based approaches, cell-based approaches, and clustering-oriented approaches. Density-based approaches define subspaces in dense areas [42]. In cell-based approaches, subspaces are formed by pre-defining the width of grid cells and the number of objects within each cell [43]. Clustering-oriented approaches define properties of the
entire set of clusters, as opposed to definition of the cluster itself, and then assigns objects to the cluster with the most relevant properties [44].

A. Density-based Approaches

The key idea of Density Based Spatial Clustering of Applications with Noise (DBSCAN) [45] is that after detecting a cluster using density-based grids, it looks at the neighborhood of each cluster point in a defined radius; any point that exists in this radius is contributed to the cluster.

Problem Statement: Every cluster \( C \) in a subspace projection is defined by a set of objects \( O \), that is a subset of database \( DB \) and a set of relevant dimensions \( S \) out of the set of all dimensions \( D \).

\[
C = (O, S), O \subseteq DB, S \subseteq D
\]

A clustering result \( R \) is a set of clusters \( k \) found in the respective subspace projections:

\[
R = \{C_1, \ldots, C_k\}, \quad C_i = (O_i, S_i) \text{ for } i = \{1, \ldots, k\}
\]

Definition: A Density-Based Subspace Cluster \((O, S)\) in a 2D space is defined w.r.t. parameters \( \text{minPoints} \) and \( \varepsilon \) – neighborhood \( N_\varepsilon(p) = \{q \in DB | \text{dist}^S(p, q) \leq \varepsilon\} \), where \( \text{dist}^S \) represents a distance function constrained to the dimensions \( S \) [45]:

1. \( \varepsilon \) – neighborhood of a point: Let \( p \) and \( q \), be two points of the sample, and the distance equation between these two points are defined by \( \text{dist} (p, q) \). The distance could be defined as Manhattan distance, Euclidean distance or other different distance methods. The \( \varepsilon \) – neighborhood of a point is defined as:

\[
N_\varepsilon(p) = \{q \in DB | \text{dist}(p, q) \leq \varepsilon\}
\]

2. Directly Density-Reachable: A point \( p \) is directly density-reachable from a point \( q \) with respect to \( \varepsilon, \text{MinPts} \) if

\[
p \in N_\varepsilon(q) \text{ and } |N_\varepsilon(P)| \geq \text{MinPts}
\]

3. Density-Reachable: A point \( p \) is density-reachable from a point \( q \) with respect to \( \varepsilon, \text{MinPts} \), if all the points in a chain of points (including \( q \) and \( p \)) are directly density reachable from each another.
4. **Density Connected**: A point $p$ is density-connected to a point $q$, if only there is point $o$ which both $p$ and $q$ are density-reachable from.

5. **Noise**: The set of points in database $DB$ which are not assigned to any cluster are called noise.

To find a cluster, the DBSCAN Algorithm starts with a random point $p$ and finds all density reachable points with respect to $\varepsilon$ and $MinPts$. DBSCAN also merges two clusters together if the distance between two sets of points is defined as:

$$(S_1, S_2) = \min\{\text{dist}(p, q) \mid p \in S_1, q \in S_2\}$$

**SUBCLU**: Density-connected Subspace Clustering (SUBCLU) is a greedy-algorithm built on an adaption of DBSCAN algorithm for high dimensional data. It computes all density-connected sets hidden in subspaces of high dimensional data. Studies have shown that SUBCLU can outperform other subspace clustering methods based on different measures [46], [45], [43]. SUBCLU is capable of detecting arbitrarily shaped clusters in subspaces.

To use a DBSCAN in each subspace, let $DB$ be a $d$-dimensional feature vectors dataset with $n$ objects $DB \subseteq \mathbb{R}^d$. Let $A = \{a_1, a_2, \ldots, a_d\}$ be the set of all attributes $a$ of $DB$, any subset $S \subseteq A$ is called a subspace. The projection of an object $o$ into a subspace $S$ is denoted by $\pi_S(o)$, and the distance function is denoted by dist. For instance, the $\varepsilon$ – neighborhood of $o$ in $S$ is the same as DBSCAN, but projected in $S$ subspace:

$$N^S_\varepsilon(o) = \{x \in DB \mid \text{dist}(\pi_S(o), \pi_S(x)) \leq \varepsilon\}$$

The core object is defined as:

$$\text{CORE}^S_{\varepsilon,m}(o) \iff |N^S_\varepsilon(o)| \geq m$$

The algorithm begins with generating all 1-dimensional clusters using DBSCAN algorithm. For each detected cluster, it checks whether the cluster also exists in higher dimensions or not. For each $k$-dimensional subspaces $S \in S_k$, the algorithm searches all other $k$-dimensional subspaces $T \in S_k$ having $(k-1)$ attributes in common, and combines them to generate $(k+1)$-dimensional candidate subspaces.

Based on previous studies [46], we choose the $MinPts$ to be in the range from 8 to 128 (with 5 steps), and the $\varepsilon$ – neighborhood from 0.01 to 0.25 (with 9 steps).
initial $Midpts$ value was set to 8 and after each run this value was increased by 30 until it reached 128. The $\varepsilon-neighborhood$ value was initially set to 0.01 and was increased by .03 until a maximum of 0.25.

**B. Cell-based Approaches**

Cell-based clustering is centered on cell guesstimate of the data space. The width of the cells is parametrized by $w$. A cluster $R$ contains a set of cells which each cell contains at least $\tau$ number of data points. One of the popular cell-based is the $MineClus$ algorithm, which describes each of these cells as the objects of the cluster by a hypercube with width $w$. These hypercubes are arbitrarily positioned to define a region with frequent data patterns [43].

Definition: A cell-based subspace cluster $(O, S)$ is defined w.r.t. minimum number of objectives $\tau$ in cells $CS$ of $w$ width specified by intervals $l_i$ per dimension $\forall i \in S$. Each interval is part of common domain $l_i = [l_i ... u_i] \subseteq [0 ... v]$ with lower and upper bound $l_i$ and $u_i$. For all irrelevant dimensions $\forall j \in D \setminus S$ the interval is the full domain $I_j = [0 ... v]$, the cluster objects $O = \{o| o \in DB \cap CS\}$ fulfill $|O| \geq \tau$

**C. Clustering-oriented Approaches**

Clustering-oriented approaches focus on the clustering result $R$ by specifying objective functions. PROCLUS [47] was the first top-down subspace clustering which forms the clusters first and iteratively improves the clustering model. In PROCLUS algorithm the number of clusters and the average dimensionality should be detected, as parameters. PROCLUS partitions the data into $k$ clusters with the average dimension being $l$.

Definition: A clustering oriented approach is defined with respect to objective functions $f(R)$, which is based on the entire clustering result $R$ and an optimal value parameter $optF$ is a result set $R$ with: $f(R) = optF$.

In this case study, we adapted aforementioned subspace clustering techniques and implemented them on the PROTECT III dataset. Analyses were performed using OpenSubspace [46], [48], an open source framework that extends the WEKA platform [38], [49].
4.2.2 Evaluation

Evaluation of unsupervised learning methods such as cluster analysis is usually informed by domain expertise. However, the domain experts may not agree with each other, and such manual evaluation is challenging for large-scale projects. Therefore, having labeled data, will help evaluation of clustering algorithms.

F1 score, entropy, coverage, and average dimensions were few evaluations metrics that were used in this case study. F1 value is a common metric for evaluating for clustering algorithms. It is the harmonic mean of precision and recall. Entropy is a metric which accounts for clarity of clustering [50]. Coverage characterizes how clusters cover the input data space. Average Dimensions is the average of number of dimensions that the clusters cover in each run. Accuracy of classification compares the patterns detected in the model with outcomes (labeled data). Further details on evaluation metrics can be found in [46].

**Clinical Assessment Scores**: The Glasgow Coma Scale (GCS) is an assessment instrument that is commonly used to grade patients’ level of consciousness using eye, verbal, and motor responses. The GCS motor score ranges from 1 to 6, with lower scores indicating a lower level of consciousness. The verbal score ranges from 1 to 5 and the eye response score ranges from 1 to 4. The total GCS score is a sum of all three components, resulting in a score from 3 to 15. Based on the GCS score, the TBI severity is graded as 3 to 8 being severe, 8 to 12 being moderate, and 13 to 15 being mild [51].

The Glasgow Outcome Scale-Extended (GOS-E) is generally used to assess longer term outcomes such as recovery of patients [52]. The GOS-E is a scale between 1 to 8, where higher score represents better recovery (1 = death, 2 = vegetative state, 3 or 4 = severe disability, 5 or 6 = moderate disability, and 7 or 8 = good recovery).

4.3 Results

This section describes the data extraction process, results from cluster analysis, and comparisons of different subspace clustering algorithms.
4.3.1 Data Extraction

As mentioned in section 4.1, the PROTECT III clinical trial showed that there were no significant differences between the treatment and control groups. For quality assurance purposes, a multiple logistic regression was performed to check if the intervention had impact on features used in this study, which showed that the intervention did not have any significant impact on either the outcome or the features (clinical measurements) as seen in “Table 7”. Therefore, used all 882 study participants were included in this case study. Of these, 643 matched the inclusion criteria as shown in see “Figure 7”.

\[
\begin{align*}
882 & \text{TBI patients} \\
846 & \text{TBI patients} \\
& \text{lived at least three days} \\
643 & \text{TBI patients, that the} \\
& \text{baseline physiological lab} \\
& \text{results were stored.}
\end{align*}
\]

*Figure 7. Study Participants*

The inclusion criteria were (1) subjects were alive for at least three days, (2) subjects were not been excluded from the parent study, and (3) their baseline lab results were available. “Table 3” shows the characteristics of the study subjects at baseline.
Table 3 Characteristics of Study Subjects at Baseline.

<table>
<thead>
<tr>
<th>Characteristic (N=643)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age – average (range)</td>
<td>34 (17-93)</td>
</tr>
<tr>
<td>Male – n (%)</td>
<td>475 (73.9)</td>
</tr>
<tr>
<td>Black – n (%)</td>
<td>105 (16.3)</td>
</tr>
<tr>
<td>Hispanic – n (%)</td>
<td>97 (15)</td>
</tr>
<tr>
<td>Cause of injury – n (%)</td>
<td></td>
</tr>
<tr>
<td>Motor vehicle accident</td>
<td>242 (37.7)</td>
</tr>
<tr>
<td>Motorcycle, scooter, or ATV accident</td>
<td>121 (18.8)</td>
</tr>
<tr>
<td>Pedestrian struck by moving vehicle</td>
<td>78 (12.1)</td>
</tr>
<tr>
<td>Other</td>
<td>202 (31.4)</td>
</tr>
</tbody>
</table>

The PROTECT III dataset included patients with severe to moderate acute TBI based on the Glasgow Coma Scale (GCS). Patients who were intubated at the time of the GCS assessment did not have visual and verbal scores. Only the motor component of the GCS score was considered. Indeed, the motor subscore has been shown to be the most important predictor of long-term outcomes following TBI [53]. The distribution of patients based on their GCS motor score at baseline is shown in “Table 4”.

Table 4 GCS Motor Score Distribution at baseline

<table>
<thead>
<tr>
<th>GCS motor score at baseline</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of subjects – n (%)</td>
<td>13 (2)</td>
<td>45 (7)</td>
<td>80 (12.4)</td>
<td>220 (34.2)</td>
<td>266 (41.4)</td>
<td>19 (3)</td>
</tr>
</tbody>
</table>

Ten different lab results were used in this study. “Table 5” shows results and summary statistics from these different labs, including blood samples, serum chemistry lab results, and blood hematology results at baseline. All values were normalized to a scale between 0 and 10 before inputting in the machine learning algorithms.”.
Table 5 Mean and Range of lab results at baseline.

<table>
<thead>
<tr>
<th>Lab Results</th>
<th>Mean</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glucose -mg/dl</td>
<td>151.6</td>
<td>68-554</td>
</tr>
<tr>
<td>Creatinine -mg/dl</td>
<td>1.015</td>
<td>0.3-4.2</td>
</tr>
<tr>
<td>Potassium -mmol/L</td>
<td>3.667</td>
<td>1.5-5.8</td>
</tr>
<tr>
<td>Sodium - mmol/L</td>
<td>139.8</td>
<td>125-157</td>
</tr>
<tr>
<td>Chloride -mmol/L</td>
<td>105.4</td>
<td>88-130</td>
</tr>
<tr>
<td>Bicarbonate -mmol/L</td>
<td>22.77</td>
<td>8.0-34.0</td>
</tr>
<tr>
<td>HGB(Hemoglobin) -g/dL</td>
<td>13.66</td>
<td>4.9-18.6</td>
</tr>
<tr>
<td>HCT(hematocrit) -%</td>
<td>40.31</td>
<td>14.6-54.2</td>
</tr>
<tr>
<td>Total WBC (White Blood Cells) -×10⁹/L</td>
<td>14.85</td>
<td>3.2-41.40</td>
</tr>
<tr>
<td>Platelets -×10³/mm³</td>
<td>249.7</td>
<td>51-700</td>
</tr>
</tbody>
</table>

Table 6 GOS-E, collected 6 month after patients baseline.

<table>
<thead>
<tr>
<th>Score</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOSE-no</td>
<td>67</td>
<td>12</td>
<td>109</td>
<td>61</td>
<td>63</td>
<td>164</td>
<td>101</td>
<td>66</td>
</tr>
<tr>
<td>GOSE-%</td>
<td>10.4%</td>
<td>1.9%</td>
<td>17.0%</td>
<td>9.5%</td>
<td>9.8%</td>
<td>25.5%</td>
<td>15.7%</td>
<td>10.3%</td>
</tr>
</tbody>
</table>
Table 7 Multiple logistic regression results. LRT stands for Likelihood Ratio Test. AIC stands for Akaike Information Criterion.

<table>
<thead>
<tr>
<th></th>
<th>Deviance</th>
<th>AIC</th>
<th>LRT</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Physiological labs at baseline</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chloride</td>
<td>768.51</td>
<td>772.51</td>
<td>0.79378</td>
<td>0.373</td>
</tr>
<tr>
<td>INR</td>
<td>768.72</td>
<td>772.72</td>
<td>0.5818</td>
<td>0.4456</td>
</tr>
<tr>
<td>Sodium</td>
<td>769.01</td>
<td>773.01</td>
<td>0.2959</td>
<td>0.5865</td>
</tr>
<tr>
<td>Glucose</td>
<td>769.14</td>
<td>773.14</td>
<td>0.16976</td>
<td>0.6803</td>
</tr>
<tr>
<td>Platelets</td>
<td>769.14</td>
<td>773.14</td>
<td>0.16026</td>
<td>0.6889</td>
</tr>
<tr>
<td>aPTT</td>
<td>769.18</td>
<td>773.18</td>
<td>0.12664</td>
<td>0.7219</td>
</tr>
<tr>
<td>HGB</td>
<td>769.23</td>
<td>773.23</td>
<td>0.0777</td>
<td>0.7804</td>
</tr>
<tr>
<td>T.WBC</td>
<td>769.23</td>
<td>773.23</td>
<td>0.07173</td>
<td>0.7888</td>
</tr>
<tr>
<td>HCT</td>
<td>769.27</td>
<td>773.27</td>
<td>0.03225</td>
<td>0.8575</td>
</tr>
<tr>
<td>Potassium</td>
<td>769.3</td>
<td>773.3</td>
<td>0.00731</td>
<td>0.9318</td>
</tr>
<tr>
<td>Bicarbonate</td>
<td>769.3</td>
<td>773.3</td>
<td>0.00502</td>
<td>0.9435</td>
</tr>
<tr>
<td>Creatinine</td>
<td>769.3</td>
<td>773.3</td>
<td>0.00018</td>
<td>0.9894</td>
</tr>
<tr>
<td><strong>Outcomes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expire_Flag</td>
<td>767.94</td>
<td>771.94</td>
<td>1.36374</td>
<td>0.2429</td>
</tr>
<tr>
<td>GOSE</td>
<td>768.46</td>
<td>772.46</td>
<td>0.84946</td>
<td>0.3567</td>
</tr>
</tbody>
</table>
4.3.2 Evaluation of Sub-clustering Algorithms

By applying the Density-Based Subspace clustering SUBCLU algorithm on baseline lab results, the following relationships were identified:

- **International Normalized Ratio (INR):** The INR which characterizes the clotting tendency of blood was identified as one of the clusters.

- **INR relation with Chloride and Creatinine:** The clustering models showed a strong relation between INR with chloride and creatinine.

- **Creatinine level, Hemoglobin level, Hematocrit percentage and Sodium level:** In the models with higher dimensions, a relationship between creatinine level, hemoglobin level, hematocrit percentage, and sodium level were noted.

- **Potassium level and Total White Blood Cells:** The relationship between potassium level, and total white cells were noticeable.

Two of the total 45 experiments using SUBCLU algorithm are shown in “Figure 8”. These examples demonstrate some of the above observations. “Figure 9” shows results from applying MineClus and PROCLUS algorithms on the PROTECT dataset.
Figure 8 Two different experiments using SUBCLU algorithm on PROTECT III dataset. The color coding based on object ranking visualization shows the value of each cluster and potential relationships. The first experiment found clusters in three spaces of INR, Chloride and Creatinine. The second figure also shows the clustering of INR and Chloride.
Results using MineClus and PROCLUS algorithms. These results also support the findings generated by the SUBCLU algorithm.
4.3.3 Comparison of Subspace Clustering Algorithms

“Table 8” compares the different subspace clustering methods. Mortality status was used as an outcome. Density-Based (SUBCLU) algorithms had higher F1 and coverage. Cell-based algorithms (MineClus) had a good performance on the F1 measure while having lower number of clusters. Clustering-oriented algorithm (PROCLUS) performed reasonably in terms of accuracy and entropy, while it had the lowest F1 compared to other models.

Table 8 Comparison of Subspace Clustering Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Density-Based/ SUBCLU (min-max)</th>
<th>Cell-Based/ MineClus (min-max)</th>
<th>Clustering-Oriented/ PROCLUS (min-max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>0.45 - 0.69</td>
<td>0.42 - 0.64</td>
<td>0.36 - 0.44</td>
</tr>
<tr>
<td>Entropy</td>
<td>0.45 - 0.59</td>
<td>0.44 - 0.55</td>
<td>0.48 - 0.63</td>
</tr>
<tr>
<td>coverage</td>
<td>0.9 - 1</td>
<td>0.78 - 0.97</td>
<td>0.43 - 0.82</td>
</tr>
<tr>
<td>Num. Cluster</td>
<td>6 - 1024</td>
<td>6 - 64</td>
<td>8 - 32</td>
</tr>
<tr>
<td>Avg. Dim.</td>
<td>2.3 - 9</td>
<td>3.2 - 6.1</td>
<td>2 - 9</td>
</tr>
<tr>
<td>Accuracy (%)</td>
<td>81 - 88</td>
<td>88 - 88</td>
<td>88 - 88</td>
</tr>
<tr>
<td>Runtime (s)</td>
<td>367 - 745785</td>
<td>58 - 194</td>
<td>155 - 402</td>
</tr>
</tbody>
</table>

4.4 Summary

In this case study, subclustering algorithms were applied to a TBI clinical trial data set. The secondary analyses showed specific clusters of physiological features in the dataset. Different subspace clustering approaches including density-based, cell-based, and clustering-oriented methods were compared and evaluated.
CHAPTER 5: CONCLUSION AND DISCUSSION

In this thesis, we explored the application and feasibility of machine learning techniques for a specific clinical condition (TBI) using data from two specific sources (one from a randomized clinical trial and another from a publicly-available ICU database collected from a single hospital). Lack of access to multiple data sources has limited further external validation of the proposed methods. In addition to data limitations, results from subclustering algorithms are yet to be validated by clinical domain experts.

Applying machine learning algorithms for specific clinical problems can provide potentially valuable insights into development of clinically-relevant tools. The two case studies presented in this thesis serve as a demonstration for such applications. Results from this study will provide the foundation to develop scalable algorithms for further research and validation. As a next step, more focus on temporal data and algorithms such as Recurrent Neural Network (RNN) and other methods for time-series analyses are warranted.
References


