UNIVERSITY OF CALGARY

Leveraging Feature Exploitation to Automate Practical Machine Learning with Text, Image and Tabular Data

by

Fatemeh Sharifi

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA

FEBRUARY, 2021

Abstract

There is a huge growth in the amount of data being generated in forms of tabular, text, and image data. Machine Learning (ML) is a powerful paradigm to support the knowledge discovery process from generated data to the knowledge that is useful in decision making. It is paramount to have methods to find important features on different applications. To this direction, this dissertation investigates four distinct problems related to exploring ML tasks on predicting various types of data including text, image and table.

The first two problems concentrated around tabular data, have the overarching goal of increasing Health-Related Quality of Life (HRQoL) used in treatment and care of prostate cancer patients. Specifically, I first propose a Cluster-based method to particularly exploit the most important features for the desired output. In the second problem, my objective is to identify the minimal set of important features which can predict 1-year follow-up HRQoL while adding interpretability to the proposed model. Using 5093 patients' information with ~1500 measures, the results support the use of the proposed ML technique as an essential tool in identifying predictable features and interpreting the findings.

The third study corresponds to using Natural Language Processing (NLP) to propose a test case failure prediction approach for manual testing that can be used as a specification-based heuristic for test selection, prioritization, and reduction. I show that a simple linear regression model using the extracted NLP-based feature together with a typical history-based feature can accurately predict the test cases' failure in new releases. The comparison of several proposed approaches on 41 releases of Mozilla Firefox over three projects, shows that the NLP-based feature can improve the prediction models.

The last study focuses on image analysis for velocity picking in seismic data. Velocity analysis is a time-consuming task which is mostly performed manually. I develop a novel data-driven ensembling strategy for combining geophysical models with Convolutional Neural Network (CNN), which uses spatiotemporally varying image data for training and predicting purposes. We perform extensive experiments using nine field datasets and evidence better

performance compared to current state-of-the-art method.

Acknowledgements

This PhD study would not be possible without the support of my supervisor, Dr. Behrouz Far. I clearly remember the moment I heard back from Dr. Far that I can do my PhD studies under his supervision which definitely was one of the best memorable moments of my life. Since then, my happiness constantly has been improved. Working with him made me to learn how to be a good person, out of any boundary, both academically and personally and how to become an independent researcher. He is the best supportive supervisor that one can ever have. I also wish to express my appreciation to my co-supervisor, Dr. Emad Mohammed, for sharing his expertise in the doctoral process and research method.

I would like to thank my supervisory committee members, Dr. Kartikeya Murari, Dr. Mohammad Moshirpour and the examiner committee Dr. Elise Fear and Dr. Gautam Srivastava for their helpful feedback on my work, with special thanks to Dr. Moshirpour who was special colleague/mentor for me which I will bring the lessons I learned from him to the rest of my life.

I gratefully acknowledge the funding sources from Dr. Trafford Crump, the Department of Surgery, for both his financial and invaluable academic guidance and support, Dr. Hadi Hemmati for his financial and academic support during chapter 5, and Alberta Graduate Excellence Scholarship.

I would like to thank my managers Elvis Floreani and Gerry Schlosser and all my colleagues specially Dennis Quinn at the Software Engineering Team of Absolute Imaging Inc. for their extensive support, providing the fantastic environment for working, and giving me the flexibility to finish this job.

A heartfelt thanks goes out to my friends Dr. Seyedeh Rezvan Farahibozorg and Dr. Fereshteh Amini and the rest of my friends near and far, whom I would instantly call when I was in hard situation during my PhD process. I am honored to have their support and kindness along the way.

My parents have been my biggest supporters through thick and thin. My mom, Simin Portangestani and my dad, Gholamhossein (Bahador) Sharifi, I love you and thankful to you more than the words can describe. I am so proud to be your daughter. My mom from an early age inspired me to be best in my education and seek new opportunities to do that. I will never forget the sacrifices she made to pursue her own education. My dad has always been an extraordinary example of having the passion to learn more in every aspect of the life. He has always been the one to remind me that I can do anything that I put my mind to.

My little and only sister, Atiyeh, when I think of the sacrifices you have made to cover my absence to support the family, I can instantly burst into tears. You are the joy of my life, the happiness I feel deep in the heart knowing that I have you. If I could do this, the sky is the limit with what you can accomplish and I hope to be off some help on your way. I also would like to thank my uncle, Zakaria Sharifi, who was the first reason that I became computer scientist. I cannot describe how much my personality comes from the great character you have and the wonderful impact and support of you on my life is unforgettable.

And to Elias, my main confidant/friend/and love of my life. I keep saying to my parents that I cannot believe such a wonderful person can even exist and how my experience in life is indescribable with you. From the beginning to the end of the process you went from being my great friend, to fiancé, to husband. I am the luckiest that this PhD also begins and ends with you. Thank you for the unique kind look on your eyes while listening to my anxieties and every moment that you were there for me from allowing me to practice presentation for you and attending the conferences with me to supporting my decisions and thoughts. The

way you adore me, love me and put trust on me, has made me the strongest version of myself I've ever been.

To Mom, Dad, Atiye

May this thesis be a tiny return to all our heartbreaking goodbyes and all the moments I could not be there for you

Table of Contents

List of Figures and Illustrations

List of Tables

List of Symbols and Abbreviations

Symbol or abbreviation	Definition
ML	Machine Learning
$HRQoL$	Health-Related Quality of Life
NLP	Natural Language Processing
CNN	Convolutional Neural Network
EHR	Electronic Health Record
PRO	Patient-Reported Outcome
WHO	World Health Organization
PCa	Prostate Cancer
$GDPR$	General Data Protection Regulation
BN	Bayesian Network
SUT	Software Under Tests
IR	Information Retrieval
AI	Artificial Intelligence
GUI	Graphical User Interface
SA	Simulated Annealing
RNN	Recurrent Neural Network
$LSTM$	Long Short-Term Memory
DNN	Deep Neural Network
PoS	Part of Speech Tagging
$TF-IDF$	Term Frequency Inverse Document Frequency
$PtDAs$	Patient Decision Aids
IOM	Institute of Medicine
IT	Information Technology
EMR	Electronic medical record
PHR	Personal Health Record
$APCaRI$	Alberta Prostate Cancer Research Initiative
XML	Extensible Markup Language
RF	Random Forest
PCA	Principal Components Analysis
$CART$	Classification and Regression Tree
OOB	Out-of-Bag
GI	Gini index
$RDDs$	Resilient Distributed Dataset
$HDFS$	Hadoop Distributed File System

MLP	Multi-Layer Perceptron
ANN	Artificial neural network
LOC	Line of Code
LR	Linear Regression)
NLR	Nonlinear Regression
NN	Neural Network
SVM	Support Vector Machine
TF	Term Frequency
$WSSE$	Within cluster Sum of Square Error
AVG	Average
SD	Standard Deviation
$EPIC-26$	Expanded Prostate Index Composite-26
DAG	Directed Acyclic Graph
CPT	Conditional Probability Table
$MMHC$	Max-Min Hill Climbing
CE	Cross-Entropy
$VPDR$	Velocity Picking Distance to Reference
MSE	Mean-Square Error
MAE	Mean Absolute Error

Chapter 1

Introduction

1.1 Problem Formulations and Proposed Approaches

In real-world applications, data are commonly represented by tabular [50] data, text [84] and images [29].

In this dissertation, I focus on employing predictive machine learning models to help solve the following four real-world, data analysis problems; those are dimension reduction in large healthcare platforms using tabular data analysis, explainable analytics for predictive model in tabular data analysis, predicting manual test case failure using text analysis and automatic velocity picking in seismic data using image analysis.

I first start by proposing a cluster-based random forest method for extracting important features out of high-dimensional data, then I add longitudinality and interpretability to the proposed model by investigating more feature exploitation method such as Bayesian network. I could provide interpretable representations of the predicted model while keeping the ability to have high performance. Further, with the attention to textuality of dataset in the third study, I investigate feature exploitation techniques in the context of text analysis such as TF-IDF, to improve the prediction of the outcome of interest. The last study, uses locally connected features to explore effect of features extracted by Convolutional Neural

Network (CNN) on image data to improve prediction in outcome of interest which mainly uses spatiotemporally varying image data for training and predicting purposes.

1.1.1 Dimension Reduction in Large Healthcare Platforms

Tremendous progress in data collection devices enables hosting vast data modalities in health care. Electronic Health Records (EHRs) are commonly used in the diagnosis and management of patients' diseases. The Office of National Coordinator reported that EHRs have been deployed in more than 95% of hospitals in the United States in 2016 [109]. The interpretation of structured information from EHRs can guide clinical care.

Patient-reported outcomes (PROs) is one example of data that is potentially useful for providing care for patients. PROs can be used to measure HRQOL and monitor symptom severity, which can be used to improve patient care [8][65] and facilitate patient-provider communication [160].

The increasing availability of data in health care has made the need for developing machine learning methods to extract the information and features from the big data a necessity. However, it is usually very difficult to extract useful information from EHRs because the nature of EHR data is highly dimensional. High-dimensional medical datasets present many mathematical challenges and have given impressive rise to new theoretical developments [16]. One of the problems with high-dimensional datasets is that, specifically in medical cases, not all the measured variables are **important** for understanding and predicting the phenomena of interest. Although various recent works have been proposed novel methods with high accuracy from high-dimensional data [15], in health care, it is still of interest to reduce the dimension of the original data to only those that have more contribution in case of the desired output prior to any modeling of the data. The task of dimension reduction is necessary to find specific variables out of all records related to a person considering a specific desired output. In these cases, however, the dimension reduction can be very complex due to the large number of dimensions, incomplete data, and multiple ways to represent the same information.

In this study, I propose a parallel processing technical approach in which the unsupervised and supervised learning are combined to decrease the overall error for predicting HRQOL using diagnostic, clinical, and PRO data by reducing the number of dimensions to the most important features from big health care data.

1.1.2 Explainable Analytics for Predictive Model

According to the World Health Organization (WHO) [112], prostate cancer (PCa) is the second most prevalent cancer in men, with 1.3 million cases per year worldwide [167]. There are a variety of modalities for treating PCa, including: surgery, radiotherapy, chemotherapy, hormone therapy, and active surveillance. All of these treatments are associated with risks for potential side effects [90] that can have negative consequences for patients' Health-Related Quality of Life (HRQoL). HRQoL outcomes have emerged as a major consideration when making treatment decisions for PCa given that the 5-year survival rate for most men with local or regional prostate cancer is nearly 100% [94]. Thus, being able to predict (we use both predict and estimate terms interchangeably, although predict is often equated with classification prediction not regression) patients' HRQoL prior to treatment would presumably help patients make an informed decisions. Therefore, we need accurate prognostic information about patients' future HRQoL prior to treatment. To achieve this goal, I either need clinician judges or predictive models. According to several studies [75][162], clinicians are often poor judges of future patient outcomes while predictive models perform significantly better.

Developing predictive models for post-PCa treatment (henceforth post-treatment) HRQoL poses several challenges. For one, the causes of HRQoL dysfunction (e.g. urinary incontinence and erectile dysfunction) are multi-factorial [97][35] which complicates the development of predictive models. Besides being multi-factorial, modeling HRQoL outcomes is also challenging because of the longitudinal data sets. Among the numerous techniques that are used to create predictive models [108], regression-based methods are the most popular because of their commonality in many statistical software packages and being relatively fast to perform. However, when it comes to big data analysis, the regression-based methods may not have enough appropriate validation in terms of accuracy and speed. Therefore, having a variable extraction/selection method that can identify the influential variables out of big data set is necessary. Furthermore, most of the research studies, to-date, predicting HRQoL do so at the population-level and present results based on the entire observation group. However, patients require individual-level prediction models in order to make treatment decisions. In other words, variables marked with high global importance may not be relevant to predict an individual's risk for negative side effects.

Despite the fact that such complex (black-box) prediction models, are able to achieve high validation result (R-squared in this study), they lack an explanation of the prediction outcome. In May 2018, General Data Protection Regulation (GDPR) requested industries to have a meaningful way to explain any automated decision and after that the interpretability of machine learning models has been receiving huge attention. Noteworthy is that when using predictive analytics in critical domains such as predicting HRQoL outcomes in prostate cancer patients, the explanation power is noticeably more important than the model predictive power.

Despite a variety of clinical interpretation approaches having been explored [168], no work has yet explored the potential of Bayesian Networks (BNs) as a explanation framework to be used on top of Black-box prediction. Bayesian networks [116] which typically reflect the problem structure, can be used to generate explanations of the predictions we already have. To the best of my knowledge, this is the first study to use variable extraction on cluster-based population that can predict the 1-year HRQoL outcomes prior to treatment as a regression prediction not only classification prediction and the first to create the insight into prediction method using Bayesian networks for PCa patients.

1.1.3 Predicting Manual Test Case Failure

Software testing is a time-consuming and costly activity in the software development life cycle. Automation is a promising solution to decrease the cost of testing and increase the quality of the testing process. Test automation can be applied on several testing problems such as test data/case generation, test prioritization, test execution, and test repair. Most of the existing test automation techniques rely on metrics that are extracted either from source code (e.g., code coverage [127]) or specification models (e.g., UML state machines [73][127]) of the software under tests (SUT). However, a large body of testing in practice is done manually. Such test cases are written in natural languages. They describe a scenario to follow on the GUI of the SUT. A typical example of such manual testing is system acceptance testing were the ready-to-ship features are being verified by the manual testers.

Manual testing is specifically interesting, from the automation point of view, because it is one of the most expensive type of testing that requires a human to judge the pass/fail results for every single run of the test cases. This might be still manageable for small systems with limited number of manual test cases, but when the system size grows, running all manual test cases for every feature change will not be possible within the time limits. The time limitation becomes more serious when the development team follows a rapid release (continuous delivery) practice.

"Test case failure prediction" refers to approaches that estimate the likelihood of a test case execution failure. These predictions then can be used in deciding what portion of existing test cases to run if we have a limited budget (test selection). Or in which order should we run test cases to catch defects sooner (test prioritization). Most of test prediction research studies rely on various traditional software metrics, on the code or specification-level, e.g., LOC, complexity, churn, etc. [43] [102]. However, in the context of manual acceptance testing, typically, these artifacts are either not accessible or not linkable to the manual test cases (e.g., a manual acceptance test on the GUI-level is not easily traceable to the underling source code). Therefore, in this work I are exploring other measures for test case failure prediction that do not require software's source code or specification/requirement documents.

One of the best measures that do not need any source code or specification/requirement documents is historical execution result of a test case. There are many studies [37][106], specially in the context of regression testing, which demonstrate that the previous failure of a test case is a very good indicator of its future failure. For example, hemmati et.al. [62], have investigated a history-based technique for prioritizing manual test cases and showed that the history-based approach was by far better than a text analytics-based approach that would only get the textual test case instructions as the input. Their finding is not a surprise, given the extra information (the knowledge about pass or failure of the test cases from previous releases) that the history-based approach has access to, compared to a pure text-based prioritization.

In this study, I propose a feature that extracts some GUI-related information per test case using Natural Language Processing (NLP) and Information Retrieval (IR) methods. The feature is based on the frequency of covering different elements of the GUI in the manual test scripts. I show that a simple linear regression model using the extracted NLP/IR-based feature and a typical history-based feature (previous test execution results) can accurately predict the test cases' failure in new releases. This information is useful for test managers to effectively assign their testing effort.

1.1.4 Automatic Velocity Picking in Seismic Data

To perform seismic imaging, we have to build subsurface velocity model in the whole depth domain, by figuring out near-surface velocity. Due to complex layers of the surface topography, building a high-quality velocity model remains one of the biggest challenges in seismic data processing. Even with the state-of-the-art technology and rapidly increasing modern software, velocity analysis is still considered as a time-consuming task in the seismic data processing. The process of velocity picking can easily take up months to complete since it requires a processor to manually picks a geologically appropriate stacking velocity from the semblance to initialize the process from prior knowledge [104] [153]. To perform the velocity picking, experts use a technique called velocity spectrum [153], in which signal coherence is evaluated in a semblance panel. Semblance panel is a graph of velocity versus two-way zero offset traveltime. They need to determine the anomalies, remove inconsistencies, split the gather in small time windows and compute the signal coherence for different velocity values. The expert then is able to pick the best velocity that leads to the highest coherence values for each time window. All these processes involve experts to visually examine the data, leading to such a time-consuming task in the seismic data processing workflow.

With the improvement of Artificial Intelligence (AI) in every facet of our life [163] [111], many studies have likewise been conducted to automate velocity analysis by using AI and particularly Machine Learning (ML). Other aspects of seismic processing workflow have also seen considerable attention in using AI such as first arrival picking [158] [173], fault classification [171] and denoising [86].

Artificial neural networks (Nonconvolutional) are amongst the first methods suggested for velocity analysis [139] [45] [18]. All these models are using shallow neural networks and as explained in [115], neural networks with many layers (deep) consistently outperform shallow ones on a variety of tasks and datasets [69] [54].

Automatic velocity picking using clustering algorithms has also seen some investigations from K-means [166] [23] to the most recent one DBSCAN [12]. These clustering approaches rely on selecting the semblance peaks in the velocity spectrum as a coherency measure to generate the velocity functions. The studies using K-means do not report a numerical comparison or quantitative analysis with human expert velocity pick. In the case of DBSCAN, although the comparison results is provided, the used data is only synthetic data. Few methods focus on using genetic algorithm and simulated annealing (SA) to solve this problem [71] [72] in which comparison performed by the authors show a high deviation on a sample data.

To address the problem of automatic velocity picking, there has been a recent push to use significantly improved abilities [82] of deep neural network or deep learning. For example, studies of using Recurrent neural networks (RNN) [13] and long short-term memory (LSTM) [176] are carried out to address this problem. However, given that the nature of this problem needs expert vision attention and image processing on one hand, and the superiority of convolutional neural network (CNN) in extracting features from images on the other hand, the state-of-the-art velocity piking methods are based on CNN [44] [114]. Ferreira *et al.* [44] applies an initial velocity picking on semblance and then refines it using CNN. The proposed CNN, however, need both gather image and semblance image along with the true velocities for training. Besides that, they use only one of the existing CNN architecture without having any comparison with other CNN architectures.

Our work is most closely related to the work of Park and Sacchi [114]. The problems mentioned in the previous work are addressed in Park and Sacchi's study [114]. However, their method only focuses on the results of the CNN, without considering the existing geophysical knowledge about the problem. In contrast, the geophysical knowledge is one of the main steps in our hybrid method where it refines the CNN prediction results. Furthermore, their work outputs regions of velocities as a number in range 2000m/s and 4000m/s, whereas our proposed method has the coverage on a wider range of velocities (from 1500m/s to 6000m/s) that includes all the real-world rock velocities.

1.2 Publications

The progress of this research has been presented in the following publications:

1) **F. Sharifi**, E. Mohammed, RT. Crump, B. Far. "A Cluster-Based Machine Learning Model for Large Healthcare Data Analysis" The 5th International Conference on Big Data Innovations and Applications (Innovate-Data), 2019, Istanbul, Turkey; Acceptance Rate 24%. [Part of Chapter 3 of Thesis]

2) F. Sharifi, H. Hemmati, “Investigating NLP-based Approaches for Predicting Manual Test Case Failure” 11th IEEE International Conference on Software Testing, Verification and Validation (ICST), 2018. Västerås, Sweden; Acceptance Rate 24.8%. [Part of Chapter 5 of Thesis]

3) F. Sharifi, EA. Mohammed, RT. Crump, B. Far. “Explainable Analytics to Predict the Quality of Life in Patients with Prostate Cancer from Longitudinal Data” Applied Artificial Intelligence, 2020 (To be submitted). [Part of Chapter 4 of Thesis]

4) F. Sharifi, D. Quinn, G. Schlosser, EA. Mohammed, B. Far. “A Hybrid Geophysical Model with Convolutional Neural Network for Automatic Velocity Picking in Seismic Data” IEEE TRANSACTIONS on geoscience and remote sensing, 2021 (To be submitted). [Part of Chapter 6 of Thesis]

5) F. Sharifi, RT. Crump, EA. Mohammed, B. Far. “Using Machine Learning to Predict the Patient-Reported Outcomes of Patients with Prostate Cancer” (abstract), accepted for presentation in “A Catalyst for Whole-Person Cancer Care (CANPROs)”, 2019, Calgary, Alberta, Canada. [Part of Chapter 3 of Thesis]

6) F. Sharifi, RT. Crump, EA. Mohammed, B. Far. “Using machine learning to predict post-prostate cancer treatment quality-of-life at the Prostate Cancer Centre in Alberta” (abstract), accepted for presentation in “Alberta Prostate Cancer Research Initiative (APCaRI) Fall Symposium”, 2019, Banff, Alberta, Canada. [Part of Chapter 4 of Thesis]

7) F. Sharifi, H. Hemmati. “An NLP-based Neural Network for Predicting Manual Test Case Failure” (poster). Consortium for Software Engineering Research (CSER) 2016 Fall Meeting, Torento, ON, Canada. [Part of Chapter 5 of Thesis]

8) **F. Sharifi**, RT. Crump, EA. Mohammed, B. Far. "Patient Reported Outcome Analysis with Machine Learning-Provides Predictive Information for EPIC-26 Values in Prostate Cancer" (poster). 20th Annual Alberta Biomedical Engineering (AB BME) Conference, 2019, Banff, Alberta, Canada. [Part of Chapter 3 of Thesis]

The other papers that I have published during my PhD studies are:

9) I. Rontu, M. Moshirpour, S. Goli, **F. Sharifi**, E. Mohammadi. "Framework for Teaching Parallel Flipped Classrooms" Canadian Engineering Education Association (CEEA/ACEG), 2019, Ottawa, Ontario, Canada.

10) R. Paul, M. Moshirpour, E. Marasco, SA. Goli, E. Mohammadi, **F. Sharifi**. "Collaborative Autoethnographic Study of a Large-Scale Flipped Classroom Implementation with Multiple Instructors" American Society for Engineering Education (ASEE), 126th annual conference and exposition, 2019, TAMPA, FL; Acceptance Rate 22.5%.

1.3 Gaps in Existing Approaches

This thesis attempts to fill the gaps in existing research concerning the data preparation in the context of tabular data, text and image datasets. First, I discuss the gaps in existing research in this section and how this thesis contributes to the existing literature in the fields is explained in the next Section (i.e. Section 1.4).

After conducting research on dimension reduction in large healthcare platforms for predicting HRQoL in prostate cancer patients, it has become evident that the existing literature about the topic has the following limitations:

- Not including PROs data as predictor.
- They are in population level and overlook the unsatisfied responses which are in minority group.

- They are classification not Regression.
- Not predicting all the domains in EPIC-26.

There is also an existing gap in literature about the explainability power of the proposed predictive models for HRQoL in prostate cancer patients in the previous study. Most of the approaches suffer from the following limitations:

- No work on extracting influential baseline visit features on 1-year follow-up HRQoL output.
- They lack an explanation of the prediction outcome.
- Unable to analyze what-if scenarios in individual level.

Another area of the focus in this thesis is exploiting textual features for predicting manual test case failure. Current literature presents following research problems and challenges in this area of study:

- Most of the existing test case failure prediction techniques are based on the code coverage or requirement coverage which are not usually available in manual acceptance testing.
- No important work on extracting GUI-elements from test case text.
- A few that are based on NLP do not improve the baseline history based approach.

Considering the last study of the thesis, it appears that there are two main research gaps when it comes to automatic velocity picking in seismic data which are:

- They do not cover all the range of real world rock velocities. Their coverage is on a small range of velocities which is not enough for most of the real world fields.
- They only focus on the results of the CNN, without considering the existing geophysical knowledge about the problem.

1.4 Research Contributions

I first describe the findings and contributions pertinent to the first research problem. To the best of my knowledge, this work is the first that proposes to do supervised learning on top of unsupervised learning in high-dimensional semi-structured patient survey data to extract the most important features for EPIC-26 values which can predict it with high coefficient of correlation between predicted and observed values. This study has three main contributions. In the first place, extracting minimal features with the highest possible prediction power out of more than 1500 features for each domain in EPIC-26. Secondly, estimating the actual value of the EPIC-26 on the patients (i.e. a continuous number between 0 and 100) not only the classification (i.e. Yes or No for patient satisfaction). And finally, Conducting a large-scale empirical study on a real-word health care dataset with 5093 patients. The details of all the methods and tools used in this thesis is provided in Section 3.3.

To address the second research problem, I will first explain three main limitations in the most of the existing approaches: 1) Using population-level data set for training the model and presenting the results based on the entire observation group. However, the model should consider patients in minority group as well. Otherwise, it might underestimate the variables that affect the similar patients which are in relatively small group but with similar output. 2) Developing black-box machine learning models which are not interpretable that contribute negatively toward establishing trust and confidence in their predictions for stakeholders. 3) They usually only consider two domains of HRQoL, mostly urinary incontinence and sexual dysfunction. To the best of our knowledge, there are no previous studies that address the problem of estimating a more comprehensive set of side effects, such as bowel and hormonal domains. Therefore, the objective of this study is to propose an approach that fills the gap in research on predicting post-treatment HRQoL for patients with prostate cancer while maintaining the interpretability of the model so that patients and physicians can understand the decisions made by the proposed prediction model.

Given our objective, in the second study, I present a framework to predict and interpret post-treatment (1-year follow-up) HRQoL outcomes for PCa patients out of high dimensional data set. I utilized the k-means algorithm [91] [74] to put similar patients in the same clusters and avoid fitting the model on the entire training data. In the next step, I used Deep Neural Networks (DNNs) [124] to fit each cluster separately and identify the most influential variables in each cluster. This will prevent underestimation of any small group of similar patients which might have same influential variables. Furthermore, I develop a combined Bayesian network method to provide interpretation of prediction model for a variety of HRQoL domains. After aggregating the top results for the patient groups, our results provide insightful recommendations for improving patient experiences.

The third study introduces a text mining-based approach for manual test case failure prediction that improves accuracy of the traditional history-based predictions. The test cases in manual testing are written in natural language. Therefore, the objective of the new metric is extracting some knowledge (features) from the test scripts, which represent the test cases coverage on the GUI. Using these feature vectors, as inputs of prediction models, helps improving the accuracy of traditional history-based approaches, where each test case is represented just by an ID (no feature is mined). Our proposed features are mined using a simple NLP technique, called Part of Speech Tagging (POS) [156], to extract nouns in test cases (as estimates of GUI elements) and then weighting them using a Term Frequency Inverse Document Frequency (TF-IDF) [135] metric.

We have conducted an extensive experiment with manual test cases of Mozilla Firefox's Desktop, Mobile, and Tablet projects with 13, 14 and 14 releases, respectively.

The contributions of this study are as follows:

- Using a text mining technique (POS) to extract a feature to estimate GUI elements' coverage in manual test cases.
- Using an information retrieval metric, TF-IDF, to build a prediction measure, based on the extracted features.
- Empirically investigating the effect of the NLP-based features, historical data, and

prediction models on the predictions accuracy.

- Conducting a large-scale empirical study on a real-word software system with over 23,085 test cases and 2,457 real bugs.

1.5 Thesis Organization

The remainder of this thesis is outlined as follows. Chapter 2 provides the details of core concepts used in this thesis. Chapter 3 presents the results of an in-depth analysis of a cluster-based machine learning model hosting a large healthcare tabular data analysis. Chapter 4 extends the previous work and describe how I implement explainable analytics for previous predictive model and add longitudinality to the feature extraction. Chapter 5 focuses on explaining text-mining based feature exploitation method used for predicting manual test case failure. Chapter 6 broaden the previous research by including locally-connected features to automatic velocity picking in seismic data using image analysis. Finally, Chapter 7 summarizes my research and discusses future work. A schematic diagram representing the organization of this thesis is provided in Figure 1.1

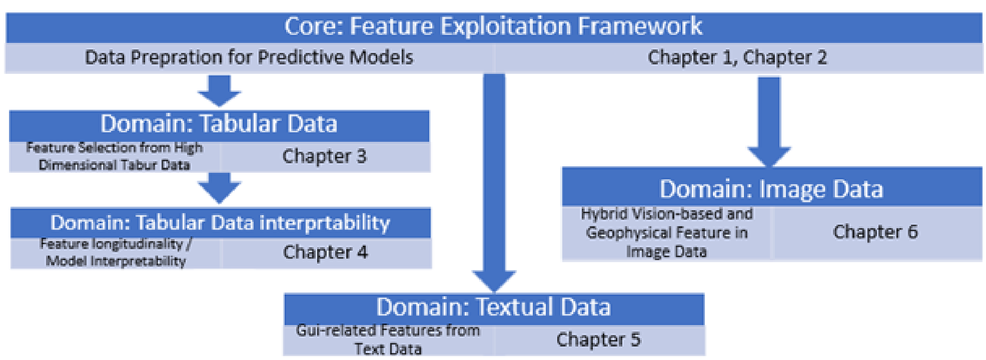


Figure 1.1: Schematic diagram of thesis organization.

Chapter 2

Background and Related Work

The main purpose of this background and literature review chapter is to provide the necessary background information and related work pertaining to this thesis. Section 2.1 explains the concept of patient survey data, clustering and some of existing dimension reduction and prediction techniques in tabular data which are parts of the proposed methods in chapters 3 and 4. Section 2.2 starts with explaining history on test case prediction and continues to demonstrate natural language processing used as core techniques in 5. Section 2.3 discuss deep learning model and geophysical-base equation proposed in literature which are helpful in the last part of the study.

2.1 Dimension Reduction in Patient Survey Data and Explainable Analytics

In this section I explore several machine-learning and data mining algorithms that have been used in proposed patient survey data analysis. First, I start with overall introduction of the work. Second, I continue with explaining the patient survey data concept and the ethos of machine learning in patient survey data. Finally, a brief overview of different machine learning methods applied for dimension reduction, prediction, clustering and explainable analytics will be presented.

It should be pointed out that the areas such as medical image analysis tasks in medical aspects and bioinformatics along with computational biology which are out of the scope than the present thesis are not discussed in this section.

2.1.1 A History of HRQoL Prediction in Prostate Cancer Patients

According to the Canadian Cancer Society, prostate cancer (PCa) is the most common cancer diagnosis in men - 1 in 8 men will be diagnosed with the disease in their lifetime, or approximately 21,500 per year [19]. These men face a decision about their treatment. Surgery, radiation, chemotherapy, and active surveillance are all options. All of these options are associated with risks for potential benefits and harms. Some of the risks of harm can have dramatic consequences on a man's quality-of-life [96][68][79] and result in an increased use in healthcare services [149].

The scientific and clinical evidence as to the "best" PCa treatment option is equivocal [6]. Consequently, it is advocated that the choice of treatment concord with men's preferences for risks[6] [7][113]. This can be achieved through a process of shared decision making between the man, his family, and healthcare providers [88]. To facilitate this process, men must be informed about their options and the associated risks. This is best done through patient decision aids (PtDAs), which not only provide men with balanced information but also elicit their preferences and highlight areas of cognitive dissonance [147].

Conventionally, PtDAs have been disseminated through static, one-way technologies (e.g., DVD with accompanying user guide)[147]. More recently, some PtDAs have enabled two-way interactions. These newer PtDAs allow the user to input personal characteristics based upon which the information provided is tailored to their unique circumstances. However, these characteristics are often very basic, such as age, gender, or race. I have not been able to find a PtDA published in the peer-reviewed literature that incorporates clinical information or Patient-Reported Outcomes to provide highly personalized predictions.

The following literature successfully applied machine learning techniques to make predictions on HRQoL, in the context of prostate cancer, from various domains ranging from urinary incontinence [138] [155][55] [126] [132], sexual [76], fecal incontinence [27] [20] [126], rectal bleeding [27] [126], etc. and methodologies from different regression-based models (Multivariate logistic regression [126] [132], Binary logistic regression [76] [55]), artificial neural network [20], etc. Summary of papers describing predicting tools relating to PROs (September 2016 – July 2019) are shown in Table 2.1. Table 2.2 lists the models for predictive tools in PROs and their explanations. The literature explained here is the extension of two main surveys that introduce the existing approaches in the literature in more details. The first one [108] covers July 2007 to September 2016 and the second one [141] explores papers prior to July 2007.

The objective of this study was to develop personalized predictive risk models for post-PCa treatment PROs. By using Canadian-specific data, it is expected that I can overcome the limitations of existing predictive models. By doing so, I anticipate to be able to use these models in modernized PtDAs that will inform patients about their PCa risk factors so that they may make better decisions with respect to the treatment option that is best for them.

Table 2.1: Summary of papers describing predicting tools relating to PROs (September 2016 – July 2019). NG (Not Given).

Author	Recruited	Country	Population	Outcome	Study Type	Setting	Tool Developed
Wouter Schaake et al. 2018 [138]	2005-2009	Netherlands	243 patients with prostate cancer confined to the prostatic capsule (stage T1-3). All patients were treated with radiotherapy	Urinary incontinence, hematuria, pain and increased voiding frequency;	Prospective	Multicenter	No
Antonio Tienza et al. 2017 [155]	2002 - 2012	Spain	787 consecutive patients that underwent Radical Prostatectomy (RP)	Urinary incontinence	Retrospective	Single centre	Nomogram
Jean-Bernard Delobel et al. 2017 [27]	2000 - 2012	France	972 patients undergoing definitive radiotherapy for localized prostate adenocarcinoma	Rectal bleeding, urgency/tenesmus, diarrhea, and fecal incontinence	Retrospective (487 patients) Prospective (485 patients)	Multicenter	Nomogram
Nikolaos Grivas et al. 2017 [55]	January 2012- March 2016	NG	439	Postoperative incontinence, defined as any involuntary urine loss irrespective of inlay use or amount of urine loss, according to ICIQ-SF	Retrospective	NG	Calculator
Mauro Carrara et al. 2018 [20]	2002 - 2004 and 2010 - 2014	Italy Switzerland	598	Late Fecal Incontinence (LFI)	Prospective	Multicenter	graphical classification tools (GCTs)
Linda Rossi et al. 2018 [126]	NG	NG	820	Rectal bleeding, faecal and urinary incontinence, and nocturia	NG	Multicenter	No
Takuya Sadahira et al. 2019 [132]	2010 - 2017	Japan	571	Urinary incontinence	Retrospective	Single centre	NG
Eugenia Mylona 2019 [100]	May 2008 and July 2018	France	272	Incontinence, retention, dysuria, hematuria, and frequency	Prospective	Multicentric	No
Lawrence H.C. Kim et al. 2019 [76]	2015 - 2016	NG	190	Urinary incontinence	Retrospective	Single centre	No
Robert T. Dess et al. 2018 [31]	2008 - 2013	NG	373	Functional erections	Prospective	Single centre	No

Table 2.2: Explanations of the models for predictive tools in PROs. NG (Not Given).

Author/year	Model Type	Variables	Validation (I/E)	Accuracy	N(events)
Wouter Schaake et al. [138]	Multivariable NTCP (Normal Tissue Complication Probability); R	Dosimetric predictors of all bladder structures, pre-treatment variables diabetes, age, cardiovascular disease, abdominal surgery, adjuvant hormonal treatment and TURP.	InternaL	AUC = 0.66, R-square = 0.10	243
Antonio Tienza et al. [155]	Multivariable logistic regression; SPSS and Orange 2.7	Age, urethral wall thickness, history of transurethral resection of the prostate , and membranous urethral length	No	AUC = 71.7	746
Jean-Bernard Delobel et al. [27]	Regression logistic; SPSS V18 and R,	Total dose, dose per fraction, RT technique	Internal	C-index = 60%	972
Nikolaos Grivas et al. [55]	Cox multivariate analysis and multivariate binary logistic regression; SPSS V22.0	FP score , ILD and MUL	Internal; bootstrap	AUC=0.779	439
Mauro Carrara et al. [20]	Artificial neural network;	Mean dose, previous abdominal surgery, use of anticoagulants, use of antihypertensive drugs, use of neoadjuvant and adjuvant hormone therapy	Internal; 4-fold cross validation	AUC = 0.78	664
Linda Rossi et al. [126]	Multivariate logistic regression; MATLAB, SPSS V21 and R (R 3.3.2)	Age, Baseline, Turp, GI Comorbidity, LRHGE, ContrastNGTDM	Internal; bootstrap	AUC = 0.73	820
Takuya Sadahira et al. [132]	Multivariate logistic regression	MUL and LAM	Internal	NG	571
Eugenia Mylona [100]	Multivariate logistic; Python and R	Whole bladder region and SRS	Internal	AUC = 0.73	272
Lawrence H.C. Kim et al. [76]	Binary logistic regression SPSS version V22.0	MRI variables: age and coronal MUL	Internal	AUC = 0.78	190
Robert T. Dess et al. [31]	Logistic regression	Baseline sexual HRQoLAge	NG	AUC = 0.81	373

2.1.2 Patient Survey Data Concept

The concept of Patient Survey Data stretches back, notably, to the Institute of Medicine's (IOM) 2001 Crossing the Quality Chasm report [170], and has been the subject of considerable Patient-centered care healthcare going forward [134], but its most widespread application to healthcare system is rooted in the implementation of the electronic health record systems in the United States. Despite the fact that, the EHR was designed, at the first place, to document patients' care for reimbursement, a need for secondary use of the accumulated data raised to serve as a platform for research. This collected EHR data, coupled with other health-related data, is a strong potential platform on which useful information can be extracted. However, it is a multidisciplinary field in which there are various sciences such as medicine, computer science, statistics, software engineering, data science, nursing, etc. Using Information Technologies (ITs), electronic related health systems collect information from all healthcare domains and then using computer science combined with medical knowledge, the collected data will be analyzed to improve patients' quality of care which is the main goal of data analysis in health [67].

The sources of healthcare data are as follows:

(a) Electronic Health Record: EHRs are longitudinal data which are collected electronically and are related to the health of or healthcare provided to an individual [4]. It collects, stores, and displays information such as laboratory and radiology test results, demographics, past and active medical history, immunizations, allergies, progress notes, medications, vital signs, pathology reports, and documents related to administration and finance. EHRs were used both in hospitals [98] [148] and in general practice [36] [105], [110] [120] [164] [118].

(b) Electronic medical record (EMR): EMRs are not identical to EHRs and usually pertain to a single practice's digital version of a patient's chart. An EMR contains the patient's information created by a particular physician or care provider.

(c) Personal Health Record (PHR): It is a subset of EHR in which includes medical history, laboratory results, and medications [39].

(d) Patient registry: It is an organized system maintained by health care organizations on their own patients to collect uniform data (clinical and other) to evaluate specified outcomes for the patients defined by a particular disease. Patient registries are often linked to the EHR [51].

Types of Patient survey Data: Patients survey Data are mainly comprised of 3 types of data 1) structured data, such as laboratory instrument readings, medications, demographic data (e.g., age, ethnicity), social information (e.g. alcohol usage), electronic accounting and billings and so on [41]. 2) unstructured data, including office medical records, MRI, CT, clinical notes (e.g. radiology report, surgical note), paper prescriptions, etc [22]. 3) semistructured data, which are a combination of structured and unstructured data. The primary data type used in this study, is the semistructured data format. This format is mainly flow sheets where their interface is like structured data as they have the columns for the attributes and the rows for the participants. However, the value field consists of one of the following 3 parameters. First, a quantitative value, e.g. age. Second, a qualitative value which has been restricted to a set of values. For example, the Alberta Prostate Cancer Research Initiative (APCaRI) [3] survey study has a column for the question "Which of the following best describes your urinary control during the last 4 weeks?" and the answers take the values of "1, No urinary control", "2, Frequent dribbling", "3, Occasionally dribbling" or "4, Total control". Third, they can be used to pull together a description of specific phenomena in the text format. For instance, in APCaRI data set the answer for "Cause of death" is stored as text format such as "Adenocarcinoma of the prostate with metastatic disease to bone and lymph nodes."

2.1.3 The Ethos of Machine Learning in Patient Survey Data

Machine learning has impressive impact in analysis healthcare data. However, most of the potential quality of care improvement is still in its infancy, because predictive modeling and simulation techniques for analyzing healthcare data have not yet been adequately developed.

Big Data in Health

"big data" as a term, first time, was used by Michael Cox and David Ellsworth in 1997 [25]. Around 10 years later, although technology was improved to produce large amounts of data, the amount of useful information was comparably less. In early 2000s, big data was developed considerably. A great deal of big data has first been devoted to identifying the content of its so-called 3Vs, volume, velocity, and variety. Moreover, the way in which the software development (such as database management systems, Extensible Markup Language (XML) and recently Spark and Hadoop) has increased, reflected in great attention towards the big data analysis in all areas including healthcare. Collecting and aggregating the clinical data in huge electronic databases were the first steps to make health data storable, reusable, and searchable to help the healthcare providers in practicing more data-based treatment. In recent years, many researchers have tried to investigate the use of big data techniques and tools computationally [172] [66] [87] and biomedically [41] [30] [169] [121]. Quality of healthcare offered to patients can be considerably improved by managing big data in healthcare. Big data not only promises to decrease the cost of care and improve treatments, but also can help doctors and physicians to make personalized decisions. Considering the recent definition of big data which delineate it as high-volume, high-velocity and high-variety information assets requiring cost-effective and innovative forms of information processing for decision-making [48], healthcare data definitely falls into the scope of big data. After the breakthrough in the big data computing [17], stored U.S. health care data alone has reached 150 exabytes [1], mainly in the form of electronic health records and will exceed yottabyte in the near future [121].

The Curse of Dimensionality

Medical big data can be mainly classified into three separate formats, i.e large n and small p (n = sample numbers, p = parameter numbers); small n and large p; and large n and large p [143]. High dimensional data are data with too many parameter numbers compared to the number of observational sample numbers. In high-dimension datasets, there is a non-intuitive behavior which might lead to "curse of dimensionality" [32]. Richard Bellman coined the curse of dimensionality in 1961 in reference to the combinatorial optimization in high dimensions [10]. This phrase has been used to describe a wide range of issues arising from exponential increase in the difficulty of applying various machine learning techniques as the dimension of the input space increases [34]. The curse of dimensionality, statistically, can make it hard to have enough sample from high-dimensional spaces [140]. If I have m dimensions of a standard multivariate Gaussian distribution, $x \sim N(0, I)$, with the mean in the center of the density and an exponential decay as I move outward. Considering m is 1 or 2 to visualize the Gaussian density function, most of the function points should drop around the mean. While it would be reasonable to expect that in the case of random sampling, a high percentage of drawn random samples should be near the mean, this does not keep being true as we increase the dimension. There is an exponential increase in the number of points within a radius $r = 1$ of the center of the distribution.

Curse of dimensionality has also negative impact on some machine learning algorithms, such as the k-nearest neighbor (knn) [157]. In knn method, k samples with the smallest distance to sample x_i, $i = 1, \cdots, n(x_i$ is a particular point of interest) are considered from all the n samples to form a k-nearest neighborhood of x_i, and sample x_j is connected to x_i if it is one of the k-nearest neighbors of x_i. The KNN algorithm assumes that similar characteristics exist in nearby points. The KNN algorithm hinges on this assumption in low dimension however having high dimension this might break down [11].

2.1.4 Dimension Reduction

Big data in its place is indeed a collection of potential rich analytic models in healthcare. Using machine learning algorithms in which the main concern is to learn potential functions which map input data to target outputs, it is critical to determine the main challenges for big data applications in this domain. In other words, when input dataset consists of high dimensional features which might be redundant or noise, the learning capability of machine learning methods will be negatively affected. Therefore, dimension reduction and feature extraction methods are required to be done on the dataset before applying prediction and generalization algorithms. Considering the high dimension data in patient surveys, an effort to improve the data quality by reducing the dimension is necessary.

In this section I cover backgrounds on feature extraction including the basics of deep learning, Random Forest (RF) and Principal Components Analysis (PCA). In addition, the Spark and Hadoop, Neural Network and all other tools and methods that are being used in the first two studies will be explained here.

K-means Clustering

K-means [74] [91] is the most commonly used clustering algorithm which requires no prior information about the associations of data points with clusters [40][125]. K-means algorithm groups the data points into K clusters according to the distance measure. The distance measure it generally uses is Euclidean distance, given by:

$$\sqrt{\sum_{\mathrm{i}=1}^{\mathrm{d}} (\mathrm{x_i} - \mathrm{y_i})^2} \tag{2.1}$$

where d is the number of dimensions, x and y are two points in a d-dimensional Euclidean space. The main idea is to define K centroids, one for each cluster. These centroids should be placed as much as possible far away from each other. First step generates K random centroids. Then it needs to assign each point to the closest centroid. When no point is pending, the first step is completed and clusters are formed. At this point, we should recalculate K new centroids and reassign the points to the clusters. It should be repeated until no more update in centroids. Finally, this technique aims at minimizing an objective function which is called Sum of Squared Error (SSE), given by the following equation:

$$\mathrm{SSE} = \sum\nolimits_{\mathrm{i=1}}^{\mathrm{d}} \sum\nolimits_{\mathrm{x_i \in C_k}} (\mathrm{x_i} - \mathrm{c_k})^2 \tag{2.2}$$

where c_k is the cluster centroid.

Random Forest

Random forest [15] is an ensemble learning method [122] that consists of many decision trees. RF is composed by Classification and Regression Tree (CART) trees, which give their decision for data point. Each decision tree in the forest is constructed using a different subset from the original training data, containing about two thirds of the cases. Normally, about one-third of the cases is left out of the bootstrap sample which is called the out-of-bag (OOB) data for the tree. The OOB data generate a test classification, with an error referred to as the out-of-bag error (OOB error). Gini index (GI) is one of the variable selection functions used in RF, which measures the impurity of an attribute with respect to the classes [16]. GI can be written as following equation for a given training set:

$$P_i = f(C_i, T) / |T| \tag{2.3}$$

$$\mathrm{GI} = \sum\nolimits_{i=1}^{I} p_i \sum_{j \neq i} p_j \tag{2.4}$$

where T is the training set, C_i is a specific class type, P_i is the probability that the selected item (which is patient in our dataset) belongs to class C_i, and I is the number of Classes.

It has been demonstrated to improve the accuracy in comparison to other supervised learning methods [15][150]. RF also provide measures of variable importance for each candidate predictor [15][85].

- Random Forests Variable Importance: Random forests has an embedded feature to identify importance of variables. The importance for each variable, used in section 3.3.3, is calculated by permutation. It describes the difference in prediction accuracy before and after permuting the variable of interest, averaged over all trees [15], by representing the mean decrease in classification accuracy after permuting each input variable over all trees. To determine the permutation importance of a variable, it should be permuted while keeping all other variables in the OOB samples unchanged, then running all trees on the OOB samples and finally calculating how much the squared error improved as a result. The average of this number over all trees in the forest is the importance score for the variable.

Principal Component Analysis

Principal Component Analysis [117] is a technique used to transform a group of correlated variables to uncorrelated variables known as principal components.These principal components are the linear combination of the original variables arranged according to their level of importance in a descending order. In fact, it is the first component which provides the highest account for the variation in the original dataset. These principal components are actually generated through an orthogonal rotation in p-space. PCA is mostly utilized to see whether the first few components are capable of encompassing and accounting for most of the variation in the original dataset.

Spark and H2O

Spark [174] offers a higher level of abstraction by using novel in-memory concepts and techniques ending for momentum for iterative and interactive Big Data processing. In-memory concept in Spark has been introduced by a data structure called resilient distributed datasets (RDDs) to cache data. Spark runtime consists of a driver and multiple workers. Multiple workers should be launched before an iterative operation. Data blocks will be read with the workers from a distributed file system and will be cached in their own memory as partitions of RDD. The RDD feature of Spark is the concept of in-menory processing in which avoids data reloading from disk at each iteration leading to dramatically speeds up the iterative operation.

H2O [57] is a newer framework in machine-learning. It is a framework for predictive and descriptive analytic that can stand alone or on-top of the hahoop and spark. H2O enables the connection between R, an open source statistical programming environment, with a big data infrastructure which is Hadoop Distributed File System (HDFS). H2O provides a platform to implement supervised and unsupervised-learning algorithms which both have been used in this study.

Deep Learning-Dimension Reduction

Deep learning aims at learning feature hierarchies with features from higher levels formed by the composition of lower level features. In this study, I use a deep neural network as a conventional multi-layer perceptron (MLP)) [124]. They include learning methods for a wide array of deep architectures, including neural networks with many hidden layers [161][52], section 2.1.4 contains more information about neural network. Deep neural network can help to provide abstract representations which they are generally invariant to most local changes of the input. In the context of categorical concepts, this is helpful as it can detect categories that cover different kinds of concepts. Deep learning can do abstraction on continuous attributes as well means to have great prediction power used in this study.

- Deep Learning Variable Importance: For Deep Learning, variable importance is calculated using the Gedeon [49] method. It is a modification of Garson method [47] which the hidden layer weights are partitioned into components associated with each input node. After that, the percentage of all hidden node weights assigned to a particular input node is used to measure the importance of that input variable. Mainly it is based on using the weight matrix of the trained neural network itself to determine which inputs are significant.

Artificial Neural Networks Artificial neural networks (ANNs) have been applied to several domains due to their high efficiency in classification and data pattern recognition. ANNs refer to an interconnected group of simple processing neurons functioning similar to human brain neurons [95]. The processing ability of the network is placed in the weights of connected neurons which are achieved and adjusted through a learning process via a set of training patterns. A trained ANN has the ability of classifying multidimensional and non-linear input-output patterns. A simple neuron gets signals from n inputs each possessing their own connection weights. The signals are continuously evaluated by the neuron through combining each input's production, its associated connection weight, and through comparing the summation to its threshold value. Then, the calculated vector will be converted by the transfer function F. This converted value is the neuron's output.

Multi-layer perceptron

A multi-layer perceptron is also called feedforward artificial neural network is used for both classification and regression [48]. It is a Generalization of single-layer perceptron [123], consisting of many neurons and several layers where neurons connect to one another in the adjacent layer. A neuron in the hidden layer and output layer transforms values from the previous layer into a weighted linear sum followed by a non-linear activation function, each of the form:

$$o_j = \sum_{j=0}^{n-1} w_{ji} \times x_j \tag{2.5}$$

Which w_{ji} are the weights and X is the input vector of the data. After that, I can use an activation function for classification purpose. This formula is used for each hidden layer separately. This implies that each layer represents the input in a different way, and as a function of the previous representation. Given this composition, multi-layer perceptron can be trained to minimize any typical loss functions with a simple application of the chain rule, called backpropagation (i.e., of error) for MLPs [128].

2.2 Predicting Manual Test Case Failure

In this section I cover backgrounds on test case prediction including the basics of linear and nonlinear regression techniques. In addition, the NLP and IR methods that are being used in this study will be explained here.

2.2.1 A History on Test Case Prediction

There has been a lot of studies in recent year about defect prediction. To predict a defect, one may use product (source code, test code, etc.), process, and organizational metrics, using several mining software repositories approaches [43]. Testing-related metrics had been also successfully applied on the defect prediction problem [78]. However, in this study, I am not interested in defect prediction and rather want to predict the test case failure. The main difference is that I am focusing on test case-related features as opposed to code-base features (such as Line of Code (LOC), code complexity, and code change), assuming that the code-base may not be available. Common testing features include the size of a test case, code and requirement coverage, and historical test case execution results (pass and fail) [106]. Note that even though, my focus is on test case failure prediction but most of the underlying techniques and approaches are inspired by the defect prediction studies.

Test case prediction techniques can be studied from two perspectives: a) what features or metrics it has access to (available data) and b) what predictive model it uses for estimating the test case failure?

Features in Test Case Prediction

In general, any product, process, or organizational feature that is used in defect prediction can also be used in test case failure prediction. Below I list some of the common features that have been used for test case failure prediction in past.

Code-coverage-based Features In the test failure prediction domain, though, code coverage is the common measure of quality for test cases. For instance, different coverage-based approaches have been used for test case prioritization. However, such approaches assume that we have access to the code coverage information of test cases, either from the previous executions of the test case [106] or by static analysis [99]. In this study, I focus on features that can be extracted only using the test cases in a manual testing settings, on a GUI-Level. Therefore, even code-based coverage info is not available.

History-based Features Another typical test case failure prediction feature, in practice, is the previous execution result per test case. The assumption here is that the historical results of test case executions are recorded and accessible for the prediction method. According to [58] test cases that have detected faults in the past may be regarded as "proven performers" hence should be given more probability to fail again. Therefore, a simple history-based metric would be whether a test case has ever failed in the past or not. The positive answers will be predicted as failing tests. This idea is typically used for regression testing [37]. Most recently, Noor et.al. proposed a history-based prioritization approach where the similarity of modified or new test cases to old failing test cases have been considered as input features for prediction [106] [107]

Text-based Features Text-based features are a newly introduced category that can work with even the most limited datasets with no source code, coverage, or execution information. Basically the only requirement is the textual representation of the test case (can be in programming/scripting/natural languages) [154]. For instance, in [62], the authors have used a topic modeling technique called LDA for representing manual test cases, though that metric did not show good results. In this study, I introduce another text-based metric and improve applicable baseline features.

Test Case Prediction Models

Classifiers such as linear and nonlinear regression (LR and NLR) and Machine learning models such as neural network (NN) and support vector machine (SVM) have been extensively used in bug triage and prediction studies [56] [119] [28]. In this study, I use the same insights and apply LR, NLR and NN, as the most common techniques, to learn a classification model from the extracted features per test case. Ithen use that model to predict future test cases' failures based on the current attribute values.

Linear Regression A linear regression method is one of the most common approaches in statistics and it works in application of numeric prediction especially where both the output class and the features are numeric [136]. Basically, the output class is expressed as a linear combination of the attributes with their corresponding weights. The LR function is given by:

$$\mathbf{y} = b_0 + \sum\nolimits_{i=1}^{n} b_i x_i \tag{2.6}$$

where y is the output, x_i is the input variable, b_i is the regression coefficient of explanatory variable i, and b_0 is the value of the intercept in the linear fitting.

Quadratic Regression Quadratic regression model is used besides linear regression in this study. While LR gives a clear analysis of the relationship between output and each single input, quadratic regression takes the interactions between input variables into account [37], thereby, making it a better fit for nonlinear systems. Quadratic regression is generated from

LR by adding more terms to equation 2.6. The quadratic regression function is:

$$\mathbf{y} = b_0 + \sum\nolimits_{i=1}^{n} b_i x_i + \sum\nolimits_{\substack{1 \le i \le n-1 \\ i+1 \le j < n}} b_{ij} x_i x_j + \sum\nolimits_{i=1}^{n} b_{ii} x_i^2 \tag{2.7}$$

with same definition as per equation 2.6.

2.2.2 Natural Language Processing

Overview

Natural Language Processing techniques, which can extract structured information from free text, can be very useful in software automation when the input data is text or speech. Parsing natural languages is one of the most basic requirements for automating any system that depends on the input from human in the form of text or speech. Natural Language Processing is a popular topic in Computational Linguistic which deals with interfacing computer with human languages. NLP specifically deals with extracting context out of natural language text by using grammar inference, relating the words with each other and using this information to address natural language semantics [101]. There are several sub-problems in NLP one of which is POS tagging (assigning part of speech tags to words), which is being used in this study.

Part-of-Speech Tagging

Part-of-Speech tagging [26] [145] is the task of labeling each word in a sentence with its appropriate syntactic category, called part of speech. Part-of-speech taggers, identify POS of a word and tag it as a noun, verb, preposition, etc. and then parse the tagged words into grammatical phrases to help distinguish the semantics of the component words. This technique is a very important preprocessing task for almost all systems with the input from human speech or text. POS tagging has received great attention from software engineering researchers developing text-based software engineering tools [2].

In the context of software engineering, NLP is not a new topic and has been used for decades. Main applications of NLP in software engineering are in the areas of Specification Mining and Requirement Engineering. In 1989, Saeki et al. [133] used NLP techniques to derive formal specifications from informal ones using similar idea as presented in this study; extracting nouns and verbs as representations of classes, attributes and methods in the code base. Hudaib et al. [73] presented a two-way approach using NLP techniques; to extract UML state machines from English natural language requirements, and then back from UML model to English natural language requirements.

Term Frequency-Inverse Document Frequency

TF-IDF is a very common metric in the context of Text Retrieval. It is a measure that is assigned to a term in a corpus of documents. The first part, Term Frequency (TF), simply uses the term's frequency to record the number of times that it appears in a document normalized by the total number of terms in that document:

$$tf(t,d) = \frac{count(t,d)}{\sum_{v \in d} count(v,d)} \tag{2.8}$$

Inverse Document Frequency (IDF) which is the second part of TF-IDF measures how rare or common this term is across all documents in the corpus. It is calculated by dividing the total number of documents (D) by the number of documents that contain that term (df). IDF is typically presented as the logarithmic version of the above calculation.

$$idf(t) = log\frac{|D|}{|\{df\}|} \tag{2.9}$$

The TF-IDF of a term t is simply t's tf multiplied by its inverse document frequency.

$$tfidf(t,d) = tf(t,d) \cdot idf(t) \tag{2.10}$$

where,

tf is the term frequency.

df is the document frequency, i.e., the number of documents containing the word.

D is the number of documents.

In this research, TF-IDF is used to measure the strength of features extracted from POS tagging.

2.3 Automatic Velocity Picking in Seismic Data

2.3.1 Deep Learning Methods for Prediction

Deep learning refers to re-emergence of artificial neural networks in machine learning which mainly is a set of ML-based models that uses patterns gained directly from data, to make considerable progress in the way computers extract information from data [48]. Traditionally, they are simply described as networks with many layers that transform input(s) to output(s). I will proceed with the variation's algorithms of deep learning.

Convolutional Neural Networks

Convolutional Neural Network imitates how the visual cortex of the brain processes images and recognizes them, therefore it is specified for the visual tasks. They contain the neurons with weights and biases which are learnable. Each neuron is capable of optionally performing a non linear operation optionally. The layers can understand the semantics of different type sub-image pattern which can occur at any spatial position in the overall image. For instance, the co-occurrence of two eyes and a nose can occur at any position in an image frame. However, it is not clear which specific patterns the models should look for, so a learning process is needed to learn them from a large set of examples and reasonably it works well if there is a large amount of labeled data [130]. CNN-based image classification has developed rapidly and excelled at perceptual tasks. AlexNet [80] and GoogLeNet [151], as two state-of-the-art CNNs, ranked at the top in the ImageNet large-scale visual recognition competition (ILSVRC). In addition to classification, CNNs are used for object recognition [80] and image [89]. The first steps of CNN typically contain two types of alternating layers that are doing the "feature extraction" task: convolutional layers and pooling layers. The convolutional layer is the essential component of CNNs in which I have a learned set of patterns or filters. They take a stack of inputs (e.g. color in an image) and convolve each with a set of learnable filters to produce a stack of output feature maps. Hence, each feature map is simply a filtered version of the input data and is tasked with learning how to extract a specific feature from the input, such as edges, corners, parts of objects, etc. Pooling layers [38] task is to reduce the dimensionality of feature maps using aggregation functions to calculate a summary statistic across a small, local region of the input. Max pooling [9] as an example, calculate the maximum output located within a rectangular region of the input, then reduces that rectangular region to a single value equal to the maximum. A common choice is to divide each feature map into non overlapping 2×2 grids of pixels, then reducing each to a single pixel, converting a feature map from, e.g. 32×32 pixels to 16×16 pixels. As a result, only the most pronounced features in each rectangular region are forwarded to the deeper layers of the model.

2.3.2 Dix Equation

The Dix Equation [33] is used for estimating unknown formation velocities, from measured stacking velocities, assumed to be valid proxies for RMS-velocities. The formula is as below:

$$\nu_i = \sqrt{\frac{V_i^2 t_i - V_{i-1}^2 t_{i-1}}{t_i - t_{i-1}}} \tag{2.11}$$

where V_i is interval velocity of layer i, and t_{i-1} and t_i are 2-way traveltimes to the top and bottom of layer i respectively. V_{i-1} and V_i are the RMS-velocities associated with layers $i-1$ and i respectively.

Chapter 3

A Cluster-based Machine Learning Model for Large Healthcare Data Analysis

To address the first research problem explained in chapter 1.1.1, this chapter characterizes dimension reduction in large healthcare platform by investigating the features, out of such high-dimensional data, that could best represent desired output and explain them so that such features can be used in the future at the point of care.

This chapter is organized as follows: Sections 3.1 and 3.2 explain data input and output in this experimental study. Section 3.3 describes the methodology in which patient clustering, classification and prediction are explained. Section 3.4 presents the results and discussion on different research questions. Section 3.5 summarizes this chapter.

3.1 Data

This study is based on a secondary analysis of data prospectively collected from participants enrolled in the Alberta Prostate Cancer Research Initiative study [3] which includes 5093 patients with 1592 dependent and independent variables. The APCaRI study is enrolling the population of men undergoing diagnosis for prostate cancer in Calgary and Edmonton, Canada. These patients were referred for a diagnostic biopsy based on conventional clinical guidelines (e.g., elevated PSA and/or abnormal digital rectal examination). To be eligible for the APCaRI study, men must be over 18 years of age, speak English or have a translator available, and not have had a prior prostate cancer diagnosis. Participants in the APCaRI study consent to having their data used for secondary studies. Only participants whom have provided this consent are included in the analytic data set used for this study. To be eligible for this study, participants had to have been diagnosed with prostate cancer and completed the EPIC-26. This study is approved by the Conjoint Health Research Ethics Board at the University of Calgary.

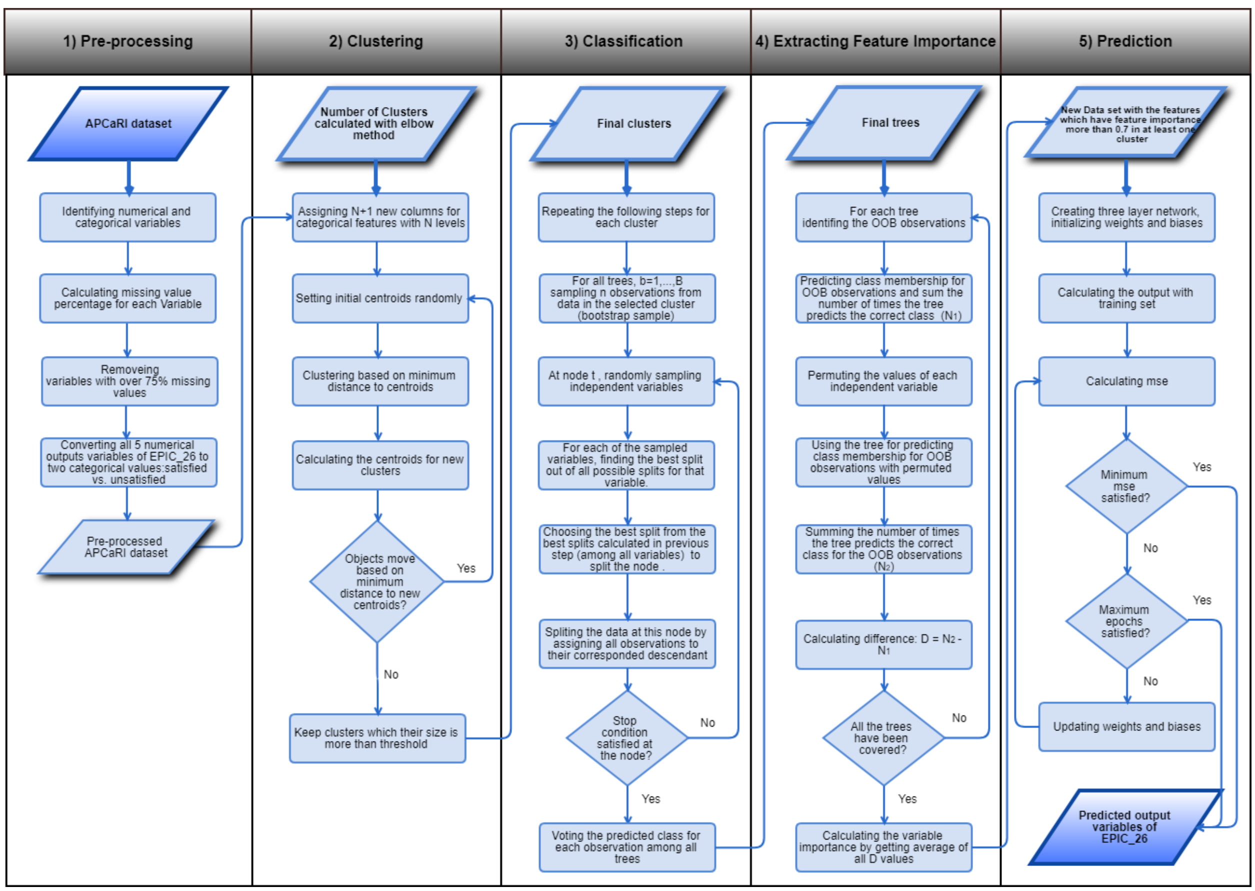


Figure 3.1: Flowchart of the approach described in this chapter. After collecting the APCaRI data, the steps are 1) Pre-processing 2) Clustering 3) Classification 4) Extracting feature importance and 5) Prediction.

3.1.1 Data Collection

Participants' demographic characteristics were collected during the in-person interview prior to biopsy. Trained data collectors abstracted participants [19]; clinical data from laboratory, pathology, or treatment reports. All data were entered into a REDCap database [59] . All personal identifiers were removed from the APCaRI study database to our research team for analysis.

3.2 Desired Output, EPIC-26

The EPIC-26 questionnaire is a valid and reliable subjective measure HRQOL in men undergoing prostate cancer treatment [152]. The items of EPIC-26 are responded to using either a four- or five-item Likert scale. Responses are transformed to a 0-100 scale, with higher scores representing less symptom severity. Using the available scoring instructions [137], items are grouped into one of five domains: urinary incontinence (UI) (4 items), urinary irritative/obstructive (UIO) (4 items), bowel (B) (6 items), sexual (S) (6 items), and hormonal (H) (5 items) [165]. As shown in table 3.2 there are 26 items in which one item (i.e.,"Overall, how big a problem has your urinary function been for you during the last 4 weeks?") is not included in any domain. For each domain, item scores are averaged to calculate the domain summary score. Participants completed the EPIC-26 either in-person (if the participant is willing to come in to the clinic), over a secure website, or by filling the questionnaire mailed to them.

3.3 Method

Before I start to explain proposed method, it is necessary to understand what characteristics are desired for the model. Epic-26 values for each domain for all the patients are a continuous value between 0 and 100, with 0 indicating the least satisfaction and 100 shows the highest.

Table 3.1: Summary of some features collected in APCaRI dataset.

Variable / Field Name	Form Name	Field Type(s)
in_001 to in_031	intake	radio
up_000 to up_007	data_entry_update	checkbox
ic_000a to ic_000e & ic_001 & ic_missing_01	informed_consent	text
rx_001 to rx_121 &	medications_and_supplements	yesno
bl_001 to bl_233 & sampleid & bl_missing_01 to bl_missing_03	biosample	text
ps_008 to ps_013	psa	radio
ib_001 to ib_083 & ib_provider	biopsy_diagnostic	text
im_001 to im_009	imaging	yesno
tp_000 to tp_101 & tp_accession1 to tp_accession6 & tp_provider1 & tp_provider6	turp	radio
ac_000 to ac_279 & ac_provider	ambulatory_care	radio
rp_000 to rp_036	prostatectomy_pathology	checkbox
su_001 to su_015	surgery	radio
cy_000 to cy_012	cryotherapy	radio
hi_000 to hi_024	hifu	radio
fl_000 to fl_024	focal_laser	radio
pt_000 to pt_024	photodynamic_therapy	radio
af_001 to af_089	biopsy_posttreatment	checkbox
r_000 to r_160	radiation	radio
ct_001 to ct_149	chemotherapy	checkbox
pr_001 to pr_035	progression	radio
eq_001 & ep_001 & ip_001 & er_001 & eq_missing_01 to eq_missing_03 & ep_missing_01 to ep_missing_03 & ip_missing_01 to ip_missing_03 & er_missing_01 to er_missing_03	survey_info	yesno
eq_002 to eq_007 & ep_002 to ep_027 & ip_002 to ip_008 & er_002 & mi_002 to mi_007	pro_survey	slider

All of these 26 features which are the influential features for Epic-26 domains values are hidden in the 1592 features I have in the dataset. By taking this into account, in this section, I explain the cluster-based random forest approach for health related quality of life prediction, as per Fig. 3.1, by first clustering the patients and then using random forest method for feature engineering and finally applying the prediction technique on the patients considering only the extracted features to evaluate the feature extraction method.

3.3.1 Patient Clustering

The purpose of performing clustering before classification can be explained as follows. The patient survey data set is an almost highly imbalanced data. Table 4.1 demonstrate that there are 2.62%, 1.19%, 0.69%, 26.97% and 1.2% dissatisfied responses (i.e. less than 50) for UI, UIO, B, S and H, respectively. Therefore, to prevent classification model to overlook the unsatisfied responses which are minority, I utilized K-means which is a popular clustering methods. Using the clustering, patients which are similar (i.e. unsatisfied patients) would probably end up in the same cluster. Then considering the unsatisfied clusters separately for classification, will help to find the influential features on minority responses as well as on majority responses in which include the satisfied patients (i.e. there EPIC-26 values are more than 50). In this study, I assigned N new columns for categorical features with N levels so can be used in k-means. In the K-means clustering, first, k centroids are randomly selected. The number of k, the cluster numbers, is decided with the elbow method. Each point which is patient in this study, decides its centroid nearest to it according to the Euclidean distance. After each of the patients in the dataset is assigned to one of the k clusters, the centroid of each cluster is updated. This procedure is repeated until the centroids are not changed anymore. At the end of this part the clusters are formed.

Elbow Method: Elbow method [103] is a visual method to determine approximately the best number of clusters. Reason being is that starting with K=2, and increasing by 1 in each step, I want to select a K where the cost, which is within cluster sum of squares (WSSE) in this study, before that drops significantly but beyond the "elbow" it reaches a plateau when you increase it further. In the Fig. 3.2 we can see in our cases using 18 clusters should be the right number of clusters for Epic-26 variables, as it is the elbow of the curve.

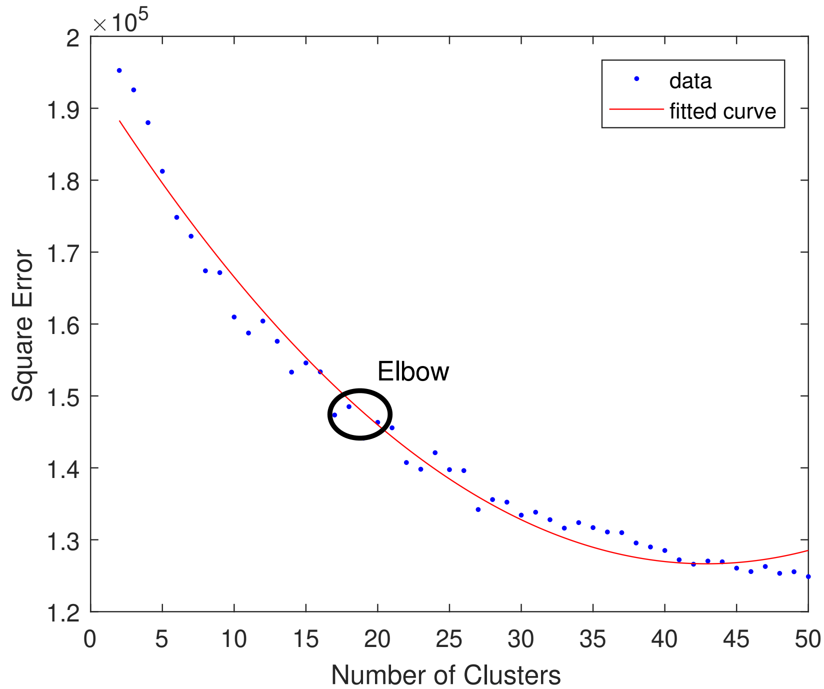


Figure 3.2: Elbow method for finding the best number of clusters.

3.3.2 Patient Classification

At the end of previous step I ended up having patients in 18 different clusters. To enhance the classification output based on the satisfaction of the patients, I assigned value "Yes", indicating their satisfaction, to the Epic-26 variables if their value is grater than or equal to 50 and "NO" if their value is less than 50. The new features will be used as outcome of interest only for training our classifier not for using it in the prediction part. In other words, the prediction algorithm will estimate the actual values of EPIC-26 (i.e. the continuous number between 0 and 100) not the classified values (i.e. Yes or No). The selected classifier in this step is random forest as it is a reasonable suggestion for our semistructured high-dimensional dataset which most of its variables are categorical (i.e. radio, checkbox, yesno, etc.). Random Forest is a tree-based algorithm that combines the results of many weak trees/learners to have improvement in predication accuracy.

The RF mainly requires the definition of two parameters to generate importance of features, the number of classification trees desired (ntree) and the number of columns to randomly select at each level (mtry). In this study, several RF models were constructed using the selected features, each with ntree of 1000 and mtry equaling to the square root of the number of input variables [21]. Furthermore, RF implementation on H2O [57] was used because of its flexibility and processing speed.

3.3.3 Extracting Important Features

Random Forest model provides an expert way to measure features' importance as explained in section 2.1.4. In this study, the top important features which are considered, are those with importance more than 0.7 (it is selected by try and error to get the most efficient threshold which the extracted features does not exceed 10 features) in at least one of the clusters.

3.3.4 Prediction of the EPIC-26 Actual Values

Artificial Neural Networks have been applied to the extracted features from the previous step to to predict the actual number of EPIC-26 for each domain. The data were divided to 70%, 15% and 15% for training, validation and testing, respectively. I repeated the above training and testing process 10 times and the results are reported in Table 3.4.

Neural Network Set up: The NN I employed in this study is a simple multi-layer perceptron with 3 layers containing 10 neurons in hidden layer. The network has 10 inputs which are the minimal set of important features extracted by each of the methods, and one output is a continuous number for EPIC-26 domain value. I use the logistic function as the activation function inside the neurons with backpropagation as the learning mechanism.

3.4 Results

3.4.1 Baseline

Among different methods for feature extraction and dimension reduction, I study the use of: i)Random Forest on the dataset without clustering; ii) PCA on the dataset without clustering; iii) Deep learning on the clustered data. For the baseline, a 3 layer NN is used as the prediction method (as a powerful paradigm for prediction) for all the feature extraction/dimension reduction methods.

3.4.2 Evaluation Approach/Study Design

In this section, I explain the evaluation of the proposed cluster-based feature extraction technique. I use NN for predicting EPIC-26 exact values to evaluate if the extracted features are predictive or not.

Objective

The objective of this experiment is to propose a parallel processing technical approach in which the unsupervised learning and supervised learning are combined to decrease the overall error for predicting HRQOL using diagnostic, clinical, and PRO data by reducing the number of dimensions to identify the most important features from the dataset.

Evaluation Metric

As I am predicting a continuous value for each domains of EPIC-26, the metric used in this chapter for assessing the effectiveness of the predictive model is Coefficient of correlation (R) between predicted and observed values in EPIC-26 domains. In case of using R-squared, it can be calculated as $R*R$ or R^2. Correlation coefficient reflects the level of direct relationship between two components, i.e., how much the variables are associated, and the values are in the range of +1 to - 1. An association of +1 suggests that there is an extraordinary positive direct relationship between elements. In other words, the predicted values is the same as the actual value, while a negative association demonstrates disagreement between two values.

Research Questions

RQ1: Can a basic Random Forest algorithm with a simple NN prediction method improve the effectiveness of a traditional dimension reduction, PCA, with the same prediction method? In this research question, I compare the prediction results of Random Forest as part of the proposed method with the PCA method.

RQ2: Does Clustering the patients first and then extracting the features, improve the prediction coefficient of correlation? In this research question, I upgrade the simple RF method, which applies the extracting feature importance on the whole data, to a cluster-based Random Forest (CB-RF). The proposed method first starts with clustering the patients and then applies the feature extraction models on the patients in the clusters separately. In this research question I compare CB-RF with original random forest (ORF) on the whole data not the clustered one.

RQ3: Keeping the clustering part of the method, are the features extracted with Random Forest, better than features extracted with another powerful feature extraction method, deep learning which has been applied on the clustered data as well? In this research question, I investigate the EPIC-26 prediction ability of Random Forest vs. Deep learning; However, both the algorithms are applied on the clustered patients for the sake of consistency.

In RQ3, I define a standard deep learning with multilayer perceptron with 5 hidden layers, each containing 100 hidden nodes. The activation function used in the layers are $Tanh$. I use stochastic gradient descent using backpropagation as training algorithm. For improving generalization I specify the input layer dropout ratio as 0.2.

3.4.3 Case Study Results

In this section, I explain and discuss the results of the experiment and answer the three research questions.

RQ1:

To answer this research question, I compare the original RF used in the proposed method with the PCA as a standard statistical technique for dimension reduction. In both of the methods the whole dataset is used without any clustering, and then I use NN as the prediction model in both methods. The idea is to see the effect of RF without giving much advantages to the clustering. Looking at Fig. 3.3, our first observation is that overall the PCA's R values are very low. Moreover, the figure shows that the RF approach is quite effective in the predicting quality of life variables.

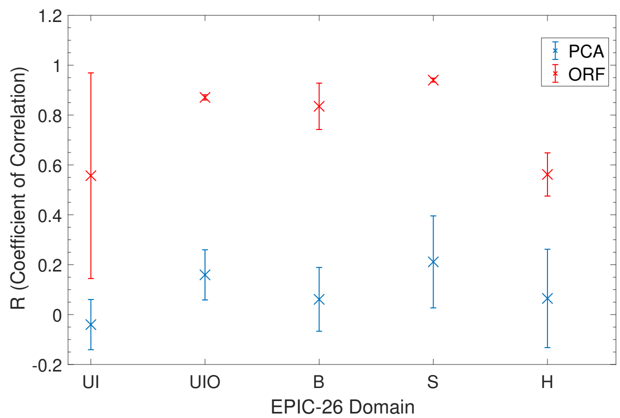


Figure 3.3: Mean and standard deviations of coefficient of correlation for PCA vs a simple (Original) Random Forest (RF). EPIC-26 Domains are UI (urinary incontinence), UIO (Urinary Irritative/Obstructive), B (Bowel) and S (sexual).

The previous analysis show that the results of the Random Forest is better than PCA which is in favor of the RQ1. It also indicates that I can improve the basic PCA feature extraction technique using a simple RF with NN prediction model. This is quite interesting since it shows that PCA cannot work effectively on dataset which mostly is categorical and a simple RF brings extra knowledge about the features. This knowledge is basically the importance of the features which can not be seen in the PCA model. However, the RF that was used in RQ1 was quite simple. So the next question will be whether a more complex feature extraction can help in prediction or not, which is studied in RQ2.

RQ2:

The objective of RQ2 is to study the cluster-based measure in more details. I want to know whether RF can be improved if I cluster all the patients and then extract feature importance in each cluster, or not. This basically tries to improve the RF without using new feature extraction models. Focusing on RQ2, given that in many outputs RF already provides high coefficient correlation, Fig. 3.4 shows that CB-RF can improve it effectively, meaning that the most significant information, in terms of the important features that contribute to predicting the desired output, are common in patients in the same clusters, explaining why CB-RF is outperforming simple RF itself.

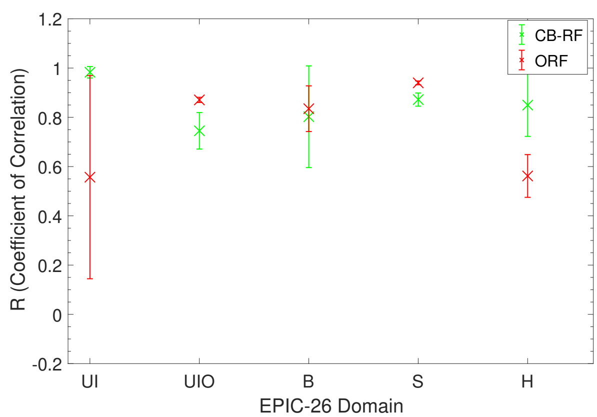


Figure 3.4: Mean and standard deviations of coefficient of correlation for RF vs the proposed method, Cluster-Based Random Forest (CB-RF). EPIC-26 Domains are UI (urinary incontinence), UIO (Urinary Irritative/Obstructive), B (Bowel) and S (sexual).

RQ3:

In RQ3, I study the same question as RQ2, but in the context of different feature extraction models. In other words, we try to improve the CB-RF by a better feature extraction model.

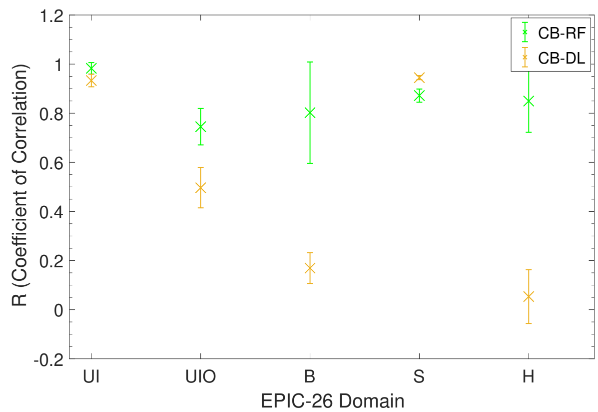


Figure 3.5: Mean and standard deviations of coefficient of correlation for Cluster-based Deep Learning (CB-DL) vs the proposed method, Cluster-Based Random Forest (CBRF). EPIC-26 Domains are UI (urinary incontinence), UIO (Urinary Irritative/Obstructive), B (Bowel) and S (sexual).

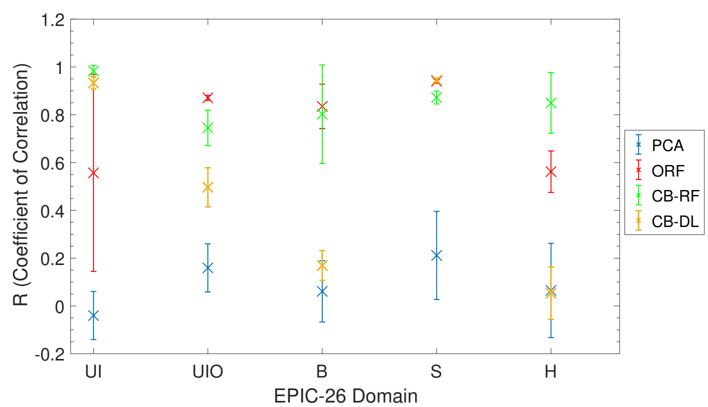


Figure 3.6: Mean and standard deviations of coefficient of correlation for Cluster-based Random Forest (CB-RF) vs other methods: Cluster-based Deep Learning (CB-DL), PCA and Random Forest (RF). EPIC-26 Domains are UI (urinary incontinence), UIO (Urinary Irritative/Obstructive), B (Bowel) and S (sexual).

To answer RQ3, I replace RF with Deep Learning, retaining the cluster-based idea, in other words, I will cluster the patients and then use DL and RF to extract important features from clusters. Looking at Fig. 3.5, we see that the two feature extracting models, in the context of NN prediction, have different levels of coefficient of correlation. It shows that only being cluster-based is not enough even when we have an accepted and popular method which is deep learning in this case. Another observation is that the proposed method is the best method with the prediction R (correlation coefficient) of 85% (R-squared 72%).

This conclusion can also be drawn by looking at Fig. 3.6 and Table 3.4, the improvement is on average 76%, 10% and 33% on PCA, RF and CB-DL, respectively. In addition, the low p-values reported in Table 3.5, all less than 0.05, show that the improvements are not random or by chance.

3.5 Summary

There is huge growth in the amount of patient survey data being generated in healthcare industries and hospitals. Curse of dimensionality is a barrier to extracting useful information from patient survey data which can help in the treatment and care of patients. It is paramount to have methods to find importance of features based on such huge volumes of stored information for the desired outputs. The health-related quality of life is a powerful paradigm to help reaching such a desired output, measuring as patient satisfaction. In such scenarios, it is difficult to investigate the features, out of such high-dimensional data, that could best represent desired output and explain them so that such features can be used in the future at the point of care. In this study I propose a Cluster-based Random Forest (CB-RF) method to particularly exploit the most important features for the desired output, which is Expanded Prostate Index Composite-26 (EPIC-26) domain scores. EPIC-26 is being used for assessing a range of HRQOL issues related to the diagnosis and treatment of prostate cancer. Different feature extraction methods are applied to extract features and the best method is the proposed CB-RF model which could find the most important features (10 or less) out of over 1500 features that can be used to accurately estimate patient with their EPIC-26 values with on average 85% coefficient of correlation between predicted and observed values of real dataset including 5093 patients.

Table 3.2: The questions related to each EPIC-26 values. EPIC-26 Domains are UI (urinary incontinence), UIO (Urinary Irritative/Obstructive), B (Bowel) and S (sexual).

Epic-26 Name	The domain related to Epic Name	The question related to Epic Name
ep-002	UI	Over the past 4 weeks, how often have you leaked urine?
ep-003	UI	Which of the following best describes your urinary control during the last 4 weeks?
ep-004	UI	How many pads or adult diapers per day did you usually use to control leakage during the last 4 weeks?
ep-005	UI	Dripping or leaking urine?
ep-006	UIO	Pain or burning on urination?
ep-007	UIO	Bleeding with urination?
ep-008	UIO	Weak urine stream or incomplete emptying?
ep-009	UIO	Need to urinate frequently during the day?
ep-010	-	Overall, how big a problem has your urinary function been for you during the last 4 weeks?
ep-011	B	Urgency to have a bowel movement?
ep-012	B	Increased frequency of bowel movements?
ep-013	B	Losing control of your stools?
ep-014	B	Bloody stools?
ep-015	B	Abdominal/Pelvic/Rectal pain?
ep-016	B	Overall, how big a problem have your bowel habits been for you during the last 4 weeks?
ep-017	S	Your ability to have an erection?
ep-018	S	Your ability to reach orgasm (climax)?
ep-019	S	How would you describe the usual QUALITY of your erections during the last 4 weeks?
ep-020	S	How would you describe the FREQUENCY of your erections during the last 4 weeks?
ep-021	S	Overall, how would you rate your ability to function sexually during the last 4 weeks?
ep-022	S	Overall, how big a problem has your sexual function or lack of sexual function been for you during the last 4 weeks?
ep-023	H	Hot flashes?
ep-024	H	Breast tenderness/enlargement?
ep-025	H	Feeling depressed?
ep-026	H	Lack of energy?
ep-027	H	Change in body weight?

Table 3.3: Statistical characteristics of patients for all the EPIC-26 domains score quartiles. EPIC-26 Domains are UI (urinary incontinence), UIO (Urinary Irritative/Obstructive), B (Bowel) and S (sexual).

Range of Score	No. Patients for each EPIC-26 domain				
	UI (%)	UIO (%)	B (%)	S (%)	H (%)
[0-25)	17 (0.78)	2 (0.09)	0 (0)	279 (12.82)	5 (0.23)
[25-50)	40 (1.84)	24 (1.1)	15 (0.69)	308 (14.15)	21 (0.97)
[50-75)	179 (8.23)	251 (11.53)	69 (3.17)	356 (16.36)	128 (5.88)
[75-100]	1372 (63.05)	1630 (74.91)	1879 (86.35)	872 (40.07)	1076 (49.45)
Missing Values	568 (26.1)	269 (12.36)	213 (9.79)	361 (16.59)	946 (43.47)

Table 3.4: Coefficient of correlation between predicted and observed values in EPIC-26 domains. Neural Network is used for predicting EPIC-26 domains using important features calculated with different features extraction techniques:PCA, ORF (Original Random Forest), CB-DL (Cluster-based Deep Learning), CB-RF (Cluster-Based Random Forest) - EPIC-26 Domains are UI (urinary incontinence), UIO (Urinary Irritative/Obstructive), B (Bowel) and S (sexual). For NN, R values of 10 runs and their average (AVG) and Standard Deviation (SD) are reported.

Method	EPIC-26 Domain	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	AVG	SD
PCA	UI	-0.0944	-0.1095	0.1393	-0.1256	-0.0627	-0.1049	0.0096	-0.1703	0.0466	0.0701	-0.0402	0.1006
	UIO	0.2387	0.1768	0.0585	0.1067	0.1667	0.3139	0.1754	0.2796	-0.0041	0.0813	0.1594	0.1006
	B	-0.0122	-0.0073	-0.1144	-0.1075	0.1474	0.2904	0.1535	0.0166	0.1223	0.1225	0.0611	0.1279
	S	0.3601	0.088	0.1011	0.2488	0.1474	-0.1366	0.4373	0.149	0.235	0.4843	0.2114	0.1844
	H	0.1065	-0.3202	0.053	0.3192	0.2827	-0.1631	0.0604	0.2468	0.0188	0.0437	0.0648	0.1972
ORF	UI	0.7204	0.7081	0.7032	0.5946	0.7403	0.6875	0.7871	-0.6036	0.6008	0.6299	0.5568	0.4123
	UIO	0.8536	0.8545	0.879	0.8719	0.874	0.8786	0.8608	0.873	0.8796	0.882	0.8707	0.0106
	B	0.8538	0.588	0.8782	0.8024	0.8558	0.8879	0.8878	0.872	0.9093	0.8143	0.8350	0.0929
	S	0.9431	0.9354	0.936	0.9351	0.9276	0.9479	0.9365	0.9479	0.9534	0.9366	0.9400	0.0078
	H	0.403	0.5316	0.4414	0.5906	0.6003	0.5648	0.6446	0.6824	0.6163	0.5433	0.5618	0.0867
CB-DL	UI	0.9258	0.9645	0.9329	0.965	0.9548	0.9069	0.956	0.9229	0.901	0.9014	0.9331	0.0256
	UIO	0.5046	0.472	0.4939	0.4142	0.5399	0.4604	0.5321	0.6296	0.3404	0.5767	0.4964	0.0819
	B	0.1858	0.193	0.189	0.1504	0.2134	0.269	0.0512	0.0895	0.2009	0.1514	0.1694	0.0624
	S	0.9303	0.9474	0.9441	0.9562	0.9455	0.9463	0.9438	0.9415	0.9577	0.935	0.9448	0.0083
	H	0.1119	0.105	0.0525	-0.1691	0.1855	0.1354	0.1086	-0.0785	-0.0203	0.1021	0.0533	0.1096
CB-RF	UI	0.993	0.9878	0.9917	0.9873	0.986	0.9936	0.9175	0.9915	0.988	0.9921	0.9829	0.0231
	UIO	0.7724	0.8073	0.7585	0.789	0.7866	0.7201	0.6928	0.43441	0.7938	0.7715	0.7451	0.0741
	B	0.8679	0.8622	0.8698	0.2506	0.928	0.6722	0.9212	0.8841	0.895	0.8722	0.8023	0.2065
	S	0.88207	0.8543	0.8786	0.9061	0.895	0.8741	0.9069	0.8411	0.8325	0.8488	0.8719	0.0267
	H	0.8711	0.8896	0.8182	0.8886	0.895	0.9135	0.4976	0.8946	0.9102	0.9163	0.8495	0.1268

Table 3.5: P-value of Cluster Based Random Forest (CB-RF) and and other dimension reduction techniques.

Method	PCA	ORF	CB-DL	CB-RF
PCA	1	1.92E-25	8.37E-10	7.52E-28
ORF	1.92E-25	1	0.000173	0.018507
CB-DL	8.37E-10	0.000173	1	3.76E-08
CB-RF	7.52E-28	0.018507	3.76E-08	1

Chapter 4

Explainable Analytics to Predict the Quality of Life in Patients with Prostate Cancer from Longitudinal Data

The next part of my thesis extends the study in chapter 3, in which, the proposed predictive models were used to predict the HRQoL in prostate cancer patients. Despite the fact that such complex (black-box) models are able to achieve high accuracy, they lack an explanation of the predicted outcome. Thus, this chapter explores enormous potential to discover new patterns from massive amount of data and transform those new black-box models into human-understandable knowledge.

The rest of the chapter is organized as follows. I present the data set and EPIC-26 measurement method first in Section 4.1. Then the proposed methods for prediction and explanation are demonstrated in Section 4.2. After that I show the results and insights in Section 4.4. Finally, I present remarks and discussions of this study in Section 4.5 followed by the chapter summary in Section 4.6.

4.1 Quality of Life in Prostate Cancer Patients and Data

To demonstrate proposed approach to predict post-treatment HRQoL for patients with PCa based on their pre-treatment prognostic factors, I am using data collected as part of the Alberta Prostate Cancer Research Initiative (APCaRI). The APCaRI [3] is enrolling the population of men undergoing diagnosis for prostate cancer in Calgary and Edmonton, Canada. These men were referred for a diagnostic biopsy based on conventional clinical guidelines (e.g., elevated PSA and/or abnormal digital rectal examination). I filtered the data to include only those patients without a prior PCa diagnosis (i.e. excluding recurrence), which resulted in 2,670 patients and involving 433 unique variables collected prior to treatment (i.e. a study "baseline"). Given that I want to predict 1-year HRQoL, I further excluded those patients who did not have 1-year data completed. This process resulted in a data set of 1,214 patients. Desired output is EPIC-26 that measures HRQoL as explained in Section 3.2 of the Chapter 3.

4.2 Cluster-based Deep Neural Network Prediction Model

Patients' severity for each of the EPIC-26 domains is a temporal continuous score between 0 and 100, with 0 indicating the highest severity and 100 indicating the least. All of the dependent variables and most of the independent variables are temporal whose states change with time (i.e. the variable values are different in baseline and 1-year follow-up). CB-DNN aims at predicting the EPIC-26 values for next year (1-year follow-up) for the patients at their baseline visit in which the independent variables at baseline visit should be used. Considering that the dimension of the data set is large, the CB-DNN algorithm starts by finding the most important variables which can have adequate description of the data by having an acceptable prediction of the quality of life for 1-year follow-up. The method's steps are detailed below.

Firstly, a K-means [74][91] is used to group the data points into K clusters according to the distance measure. The Euclidean distance is calculated as $\sqrt{\sum_{\mathrm{i=1}}^{\mathrm{d}}(\mathrm{x_i}-\mathrm{y_i})^2}$ where d is the number of dimensions (433 in this study), x_i and y_i are two patients in a d-dimensional Euclidean space. The main idea is to define K centroids, one for each cluster. These centroids should be placed as much as possible far away from each other. The Elbow method [103] is used to find the best number of the clusters, K. It is a visual method to determine approximately the best number of clusters. The EPIC-26 values in 1-year follow-up is a highly imbalanced data. Table 4.1 demonstrate that there are 7.4%, 12.19%, 3.62%, 37.56% and 3.79% in first three quartiles (i.e. less than 75) in 1-year follow-up for UI, UIO, B, S and H, respectively. Therefore, I prevent the classification model to overlook these more severe responses which are minority by using clustering. Patients which are similar (i.e. less satisfied patients) would probably end up in the same cluster. Then considering these clusters separately for variable extraction will help to find the influential variables on minority responses as well as on majority responses in which include the patients in the less severe quartile (i.e. there EPIC-26 values are more than 75).

Secondly, for each resulting cluster from the previous step, a deep neural network is trained to extract the important variables per cluster. Deep learning aims at learning variable hierarchies with variables from higher levels formed by the composition of lower level variables. In this study I use the deep neural network as a conventional multi-layer perceptron (MLP) [124]. Variable importance in DNN is calculated using the Gedeon [49] method explained in Section 2.1.4. In this study, the top important variables which are considered are those with importance more than 0.85 (it is selected by try and error to get the most efficient threshold which the extracted variables does not exceed 10 variables) in at least one of the clusters.

DNN is well suited for APCaRI data set of this study. There are two main reasons for

Table 4.1: Statistical characteristics of patients for all the EPIC-26 domains score quartiles in 1-year follow-up. EPIC-26 Domains are UI (urinary incontinence), UIO (Urinary Irritative/ Obstructive), B (Bowel), S (sexual) and H (Hormonal).

Range of Score	No. of Patients for each EPIC-26 Domain				
	UI (%)	UIO (%)	B (%)	S (%)	H (%)
[0-25)	2 (0.16)	1 (0.08)	0 (0)	204 (16.80)	2 (0.17)
[25-50)	10 (0.82)	13 (1.07)	9 (0.74)	132 (10.87)	6 (0.49)
[50-75)	78 (6.42)	134 (11.04)	35 (2.88)	120 (9.89)	38 (3.13)
[75-100]	768 (63.26)	893 (73.56)	1033 (85.09)	238 (19.61)	1006 (82.87)
Missing Values	356 (29.32)	173 (14.25)	137 (11.29)	520 (42.83)	162 (13.34)

this. In the first place, DNN can help to provide abstract representations which they are generally invariant to most local changes of the input which is helpful if other hospitals want to reproduce the results based on their data set. Moreover, In the context of categorical concepts, given that most of the independent variables in this study are either binary or ordinal, DNN is helpful as it can detect categories that cover different kinds of concepts. In DNN, the tanh (hyperbolic tangent) activation function, ADADELTA optimizer [175], and five hidden layers with 100 neurons each were used.

Thirdly, The important variables extracted from previous step which is for patients in baseline visit are fed as input to an Artificial Neural Networks for each domain separately. The 1-year follow-up EPIC-26 variable are considered as output of the ANN. The data were divided to 70%, 15% and 15% for training, validation and testing, respectively. The ANN used for prediction is a simple multi-layer perceptron with 3 layers containing 10 neurons in hidden layer. The network has 10 inputs which are the minimal set of important variables extracted by each of the methods, and one output is a continuous number for EPIC-26 domain value. I used the logistic function as the activation function inside the neurons with backpropagation as the learning mechanism. The ANN model is used as non-linear prediction model in the last step of the proposed model. That is because unlike traditional machine learning models such as linear or logistic regression which has been used widely in the literature for predicting quality of life [27][76][31], ANN models can learn more complex non-linear data representation.

Note that, in this study, in order to have a fast solution, both clustering and deep neural network implementation are on H2O. H2O [57] is a tool in the machine learning for predictive and descriptive analytic that can be stand alone or executed on-top of the Hadoop [14] or spark [174] frameworks. H2O handles billions of data rows in-memory by using in-memory compression [81], it also enables the connection between R, an open source statistical programming environment, with a big data infrastructure which is Hadoop Distributed file System.

4.2.1 Evaluation Metric

The proposed techniques is evaluated using the coefficient of determination (R^2). It gives a quantitative assessment of the ability of the model to explain the variance in unseen data by measuring how well the model is compared to the average model. This is a well established measure of the strength of the relationship between the predicted outcome and actual outcome in regression problem. This coefficient is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(\hat{Y}_i - Y_i\right)^2}{\sum_{i=1}^{n} \left(Y_i - \bar{Y}\right)^2} \tag{4.1}$$

where $\hat{Y}_i$ denotes the value of the outcome variable predicted by the proposed method for the i^{th} data point. Y_i and $\bar{Y}$ are the actual value of the outcome variable and its mean respectively. The coefficient of determination can have the value from 0 to 100%, higher coefficient values suggest that the predicted values and the actual values are more the same. In this study, the Minimally Important Differences (MID) reported in [144] is applied before calculating the R^2 values. MID provides a measure of the smallest change in the EPIC-26 values that patients perceive as important. All errors less than the MID value of the EPIC-26 domain are considered as zero for calculating coefficient of determination.

4.3 Bayesian Network-based Explanation Model

A Bayesian network [116] represents a class of graphical models as a Directed Acyclic Graph (DAG) over **X** (a set of random variables). Such that random variables are associated with the nodes in DAG. In this section, a method is developed for learning the structure of a Bayesian network. It is important to note that I adhere to the standard assumption behind the Naıve Bayes rule which mentions that there is no interaction between the input variables given the outcome [77][146][5][61].

EPIC-26 may be modeled using one Bayesian network for each of the five domains. The structure of the BN prediction model is for two time slices: baseline visit and 1-year follow-up visit. All the nodes included in the model are temporal except in_002 (How would you classify your race?) whose state does not change with time. It is also noteworthy to mention that some of the temporal nodes such as urinary incontinence, urinary irritative/obstructive, bowel, sexual and hormonal variables are included with their 1-year follow-up values, while others that are extracted from CB-DNN are considered with their baseline values. The reason behind utilizing the BN as the base for the proposed explanation model is to find the relationship between the baseline variables proposed by CB-DNN as important variables that can predict the 1-year EPIC-26 values, and 1-year EPIC-26 output values. Therefore, time lags in each BN model include a one year lag in the effect of variables extracted by CB-DNN on 1-year EPIC-26 output values. Baseline variables extracted by CB-DNN are reported in Table 4.2, 4.3, 4.4, 4.5 and 4.6, for UI, UIO, B, S and H, respectively. As I only need the arcs which end up to 1-year EPIC-26 value, I do not need the latent variables to reduce the conditional probability tables (CPTs) for the BN. However I applied some minor changes on BN learning process to get the best explanation results for proposed model.

Algorithm 1: Greedy Bayesian network Algorithm

Input: a data set $\mathcal{D}$ from **X**, an initial DAG $\mathcal{G}$ (usually the empty DAG), Schwarz Information Criterion as score function Score($\mathcal{G}, \mathcal{D}$)

Output: the DAG $\mathcal{G}_{max}$ that maximises Score($\mathcal{G}, \mathcal{D}$)

1. Compute the score of $\mathcal{G}$, $S_{\mathcal{G}} = \text{Score}(\mathcal{G}, \mathcal{D})$.
2. Set $S_{max} = S_{\mathcal{G}}$ and $\mathcal{G}_{max} = \mathcal{G}$
3. **Hill climbing**: repeat while S_{max} increases:
   - for every possible arc addition, deletion or reversal in $\mathcal{G}_{max}$ that the arc ends up to the 1-year follow-up EPIC-26 domain value and resulting in a DAG:
     (a) compute the score of the modified DAG $\mathcal{G}^*$, $S_{\mathcal{G}^*} = \text{Score}(\mathcal{G}^*, \mathcal{D})$:
     (b) if $S_{\mathcal{G}^*} > S_{max}$ and $S_{\mathcal{G}^*} > S_{\mathcal{G}}$, set $\mathcal{G} = \mathcal{G}^*$ and $S_{\mathcal{G}} = S_{\mathcal{G}^*}$.
   - if $S_{\mathcal{G}} > S_{max}$, set $S_{max} = S_{\mathcal{G}}$ and $\mathcal{G}_{max} = \mathcal{G}$.
4. **Tabu search**: for up to Tabu list size:
   - repeat step 3 but choose the DAG $\mathcal{G}$ with the highest $S_{\mathcal{G}}$ that has not been visited in the last Tabu list size steps regardless of S_{max};
   - if $S_{\mathcal{G}} > S_{max}$, set $S_{max} = S_{\mathcal{G}}$, and $\mathcal{G}_{max} = \mathcal{G}$ and restart the search from step 3.
5. **Max-min hill climbing**: First add an arc from most important variable extracted by CB-DNN to its correspondence EPIC-26 domain to the DAG $\mathcal{G}$ with the highest $S_{\mathcal{G}}$ from step 4, then for every arc $\mathbf{X}_t$ in it:
   - $PC_{\mathbf{X}_t} = MMPC(\mathbf{X}_t, \mathcal{D})$ [159]
   - repeat step 3 but for arc addition only add $Y \rightarrow \mathbf{X}_t$ if $\mathbf{X}_t$ is in $PC_{\mathbf{X}_t}$

The implementation of BN learning which is proposed in this study for explaining the results of the CB-DNN is shown in Algorithm 1. Steps 1 and 2 are initialisation phases followed by a hill climbing search (step 3), which is then refined with Tabu search [53][131] in step 4 and Max-min hill climbing (MMHC) [159] in step 5. The variations of each algorithm have been used in large parts of the literature on score-based BN structure learning [46][159]. Arc additions and deletions is being used in hill climbing to explore the neighbourhood of the current candidate to find if there is any DAG that increases the score over $\mathcal{G}_{max}$. Being an optimisation heuristic, hill climbing cannot guarantee theoretically that $\mathcal{G}_{max}$ is a global maximum. In order to escape from local maximum, tabu search is applied. This greedy search tries to move away from $\mathcal{G}_{max}$ by allowing additional local moves (up to tabu list size). If a new DAG with a score larger than $\mathcal{G}_{max}$ is found in the process, that DAG is taken as the new $\mathcal{G}_{max}$ and it reverts to hill climbing in step 3. There is a possibility that no such DAG is found by tabu search. MMHC then tries again to escape the local maximum by first adding a mandatory arc from the most important variable extracted in CB-DNN part to the outcome of interest and then updating as follows. MMHC algorithm can be categorized as a hybrid method [159]. It combines two main approaches exist for learning BNs, search-and-score approach [24] and constraint-based approach [146]. It first use a local discovery algorithm to learn the skeleton of BNs and then a greedy Bayesian-scoring hill-climbing search. If the DAG that was resulted with tabu search, if any, was indeed the global maximum, the assumption is that MMHC will also identify it as the optimal DAG, in which case the algorithm terminates. For sake of complexity and considering the objective of this study, I treat the proposed model in a way that keeping only arc for $x_i \rightarrow E$ in which x_i is any of the baseline variables extracted from CB-DNN and E is the EPIC-26 domain which the current BN is trained for. In some cases those that end up in the final structure might have dependency between themselves.

4.4 Results

CB-DNN determines important variables at the global level (i.e. per patient group), while BN can provide reasons at individual patient level. By combining the global and local method, I achieve the objective of the research: to make good HRQoL prediction for 1-year follow-up with high R^2 score while maintaining decent degree of interpretability. In the following sections I show a comparative analysis of the proposed methods, which helps both physicians and patients to understand the results and get the right insights towards better

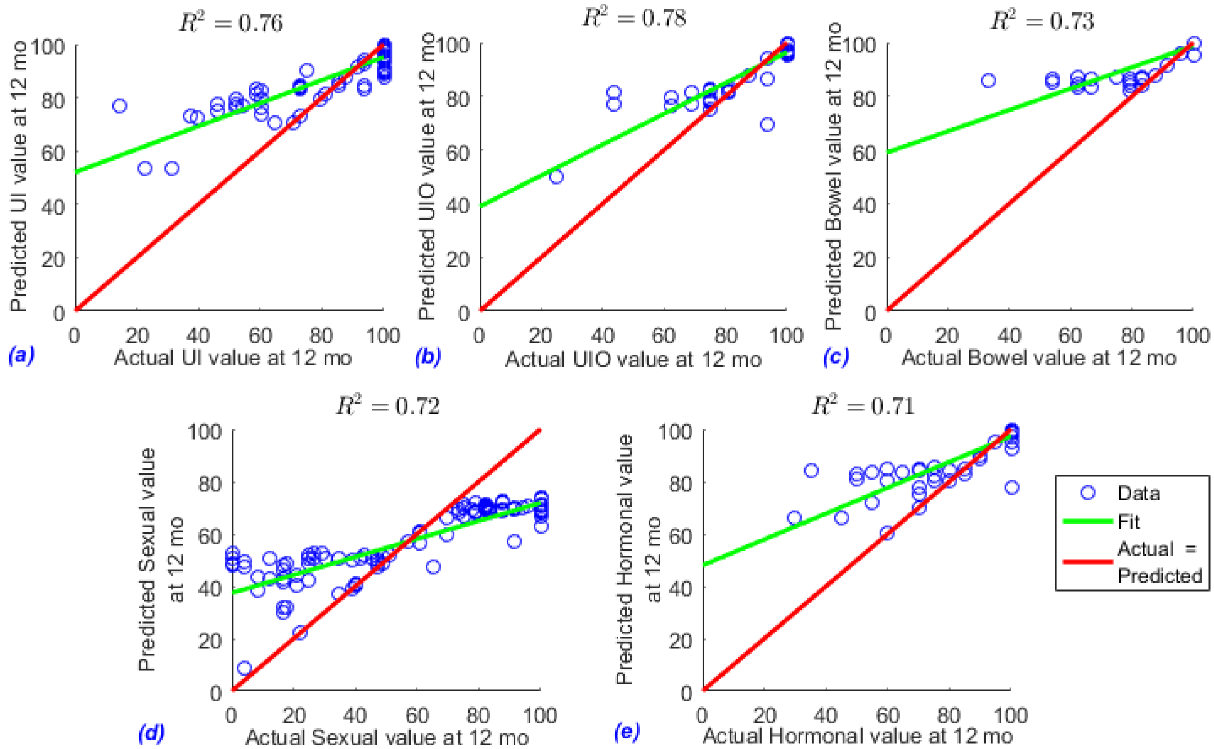


Figure 4.1: R-squared between predicted and actual values in EPIC-26 domains. EPIC-26 Domains are a) UI (urinary incontinence) b) UIO (Urinary Irritative/Obstructive) c) Bowel d) sexual and e) Hormonal.

Table 4.2: Top 10 important variables from baseline visit selected by deep neural network for predicting 1-year follow-up Urinary Incontinence along with their definitions

Variable	Definition	Type
eq_004	Usual activities	Ordinal
ep_012	How big the problem of Increased frequency of bowel movements, if any, has been for you?	Ordinal
ip_008	During the past month, how many times did you most typically get up to urinate from the time you went to bed at night until the time you got up in the morning?	Ordinal
ib_035	Number of positive cores biopsied from LA	Ordinal
in_029	Relation of the family member diagnosed (If anyone in your family has been diagnosed with prostate cancer)	Nominal
ep_020	How would you describe the FREQUENCY of your erections during the last 4 weeks?	Ordinal
ip_007	In the past month, how often have you had to push or stain to urinate?	Ordinal
rx_008	Which type of medications, vitamins or herbal supplements are you taking now?	Nominal
im_001	Bone scan?	Binary
in_002	How would you classify your race?	Nominal

Table 4.3: Top 10 important variables from baseline visit selected by deep neural network for predicting 1-year follow-up Urinary Irritative/Obstructive along with their definitions

Variable	**Definition**	**Type**
ib_064	%of first RMZ core involved	Numerical
ib_056	Number of positive cores biopsied from RA	Ordinal
ep_015	How big the problem of Abdominal/Pelvic/Rectal pain, if any, has been for you?	Ordinal
ep_020	How would you describe the FREQUENCY of your erections during the last 4 weeks?	Ordinal
in_016	Have you ever been diagnosed with chronic liver disease?	Binary
ep_021	Overall, how would you rate your ability to function sexually during the last 4 weeks?	Ordinal
ib_011	Gleason secondary grade	Ordinal
in_012	Have you ever been diagnosed with cerebrovascular disease?	Binary
ep_019	How would you describe the usual QUALITY of your erections during the last 4 weeks?	Ordinal
in_009	Have you ever been diagnosed with congestive heart failure?	Binary

Table 4.4: Top 10 important variables from baseline visit selected by deep neural network for predicting 1-year follow-up Bowel along with their definitions

Variable	**Definition**	**Type**
ib_042	Number of positive cores biopsied from LMZ	Ordinal
ib_029	Intraductal carcinoma	Binary
ib_020	High Grade PIN	Binary
ib_071	%of first RB core involved	Numerical
in_018	Have you ever been diagnosed with Hemiplegia?	Binary
ep_018	How would you describe your ability to reach orgasm (climax) during the last 4 weeks?	Ordinal
ib_013b	Total cores collected	Numerical
ep_012	How big the problem of Increased frequency of bowel movements, if any, has been for you?	Ordinal
in_029	Relation of the family member diagnosed (If anyone in your family has been diagnosed with prostate cancer)	Nominal
ib_006	DRE findings (part of biopsy-diagnostic variables)	Binary

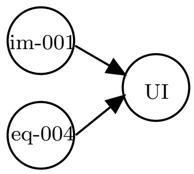
Table 4.5: Top 10 important variables from baseline visit selected by deep neural network for predicting 1-year follow-up Sexual along with their definitions

Variable	**Definition**	**Type**
ep_018	How would you describe your ability to reach orgasm (climax) during the last 4 weeks?	Ordinal
ib_008	Gland volume	Numerical
in_011	Have you ever been diagnosed with Dementia?	Binary
in_009	Have you ever been diagnosed with congestive heart failure?	Binary
ep_017	How would you describe your ability to have an erection during the last 4 weeks?	Ordinal
in_017	Have you ever been diagnosed with Diabetes?	Binary
ep_013	How big problem of Losing control of your stools, if any, has been for you?	Ordinal
ep_024	How big the problem of Breast tenderness/enlargement, if any, has been for you during the last 4 weeks?	Ordinal
ib_064	%of first RMZ core involved	Numerical
ep_023	How big the problem of Hot flashes, if any, has been for you during the last 4 weeks?	Ordinal

Table 4.6: Top 10 important variables from baseline visit selected by deep neural network for predicting 1-year follow-up Hormonal along with their definitions

Variable	**Definition**	**Type**
ib_008	Gland volume	Numerical
in_027	Have you ever been diagnosed with HIV infection?	Binary
eq_002	Mobility	Ordinal
ip_006	In the past month how often have you had a weak urinary stream?	Ordinal
ep_014	How big the problem of Bloody stools, if any, has been for you?	Ordinal
ep_003	Which of the following best describes your urinary control during the last 4 weeks?	Ordinal
ep_011	How big the problem of Urgency to have a bowel movement, if any, has been for you?	Ordinal
ip_007	In the past month how often have you had to push or stain to urinate?	Ordinal
ib_051	%of second LB core involved	Numerical
in_008	Have you ever been diagnosed with Myocardial infarction?	Binary

Figure 4.2: Bayesian networks explanation for Urinary Incontinence (UI). eq_004 (Usual activities?) has the following options:1, I have no problems doing my usual activities | 2, I have slight problems doing my usual activities | 3, I have moderate problems doing my usual activities | 4, I have severe problems doing my usual activities | 5, I am unable to doing my usual activities. im_001 (Bone scan?) has the following options: 1, yes | 2, No.



im_001[a]	**eq_004**[a]	**P[(UI**[b] **< mean) | im_001,eq_004] (unhappy)**	**P[(UI**[b] **>= mean) | im_001,eq_004] (happy)**
1	1	0.24	0.76
	2	0.42	0.58
	3	0.11	0.89
	4	0.30	0.70
	5	1.00	0.00
2	1	0.25	0.75
	2	0.35	0.65
	3	0.67	0.33
	4	1.00	0.00
	5	No Data	No Data

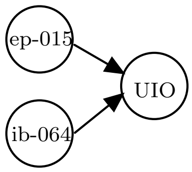
[a]Baseline visit value. [b]1-year follow-up value.

treatment options and outcomes. In this section, I first show the variable importance results and model fitting obtained from CB-DNN; then I evaluate the performance of BN method by presenting individual explanations for the patients by extracting the relationship between variables; finally I summarize the top explanations/reasons for both approaches.

4.4.1 Clustering-based Deep Neural Network (CB-DNN) Results

Initially I have 2,670 patients which have a prior PCa diagnosis data in the data set, after filtering pre-treatment factors with other required pre-processing, as explained in section **??** the data set becomes 1214 patients with 433 variables. The final neural network model includes a total of 1214 patients who have both baseline visit data and 1-year follow-up data. The predictive model, in which I use baseline visit data extracted by CB-DNN to predict 1-year follow-up data, has good performance which the average R-squared (R^2) value of 0.74 on the test data. The performance metrics of predictive model (i.e. R-squared value) on the testing data are shown on Fig. 4.1. Given the proposed solution which is regression prediction (i.e. predicting the continuous value of the output variable) not only classification prediction, and given that only pre-treatment variables are used to predict

Figure 4.3: Bayesian networks explanation for Urinary Irritative/Obstructive (UIO). ep_015 (Abdominal/Pelvic/Rectal pain?) has the following options: 1, No problem | 2, Very small problem | 3, Small problem | 4, Moderate problem | 5, Big problem. ib_064 (% of first RMZ core involved?) is a numerical variable which those less than the mean and greater than the mean are labeled with 0 and 1 respectively.



ib_064[a]	**ep_015**[a]	**P[(UIO**[b] **< mean) | ib_064,ep_015] (unhappy)**	**P[(UIO**[b] **>= mean) | ib_064,ep_015] (happy)**
0	1	0.16	0.84
	2	0.32	0.68
	3	0.57	0.43
	4	0.25	0.75
	5	0.50	0.50
1	1	0.18	0.82
	2	0.48	0.52
	3	0.67	0.33
	4	0.71	0.29
	5	0.50	0.50

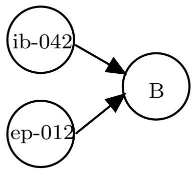
[a]Baseline visit value. [b]1-year follow-up value.

post-treatment HRQoL values, and furthermore managing to improve data reduction (in the variable extraction step), maintaining the accuracy level with average R^2 value of 0.74 would be acceptable.

The top 10 important variables selected by CB-DNN for EPIC-26 domains are reported in Table 4.2, 4.3, 4.4, 4.5 and 4.6, for UI, UIO, B, S and H, respectively. The top variable from baseline which are predictable for 1-year follow-up for each of the EPIC-26 domains are the following questions: **a) Urinary Incontinence:** Usual activities? (Answer is one of the following options: 1, I have no problems doing my usual activities | 2, I have slight problems doing my usual activities | 3, I have moderate problems doing my usual activities | 4, I have severe problems doing my usual activities | 5, I am unable to doing my usual activities) **b) Urinary Irritative/Obstructive:** %of first RMZ core involved? (Answer is a numerical value) **c) Bowel:** Number of positive cores biopsied from LMZ? (Answer is one of the following options: 1, 1 | 2, 2 | 3, Fragmented core tissue) **d) Sexual:** How would you describe your ability to reach orgasm (climax)? (Answer is one of the following options: 1, Very poor to none | 2, Poor | 3, Fair | 4, Good | 5, Very good **e) Hormonal:** How much is your Gland volume (cc)? (Answer is a numerical value)

These variables reflected by these questions point to the most important variable ex-

Figure 4.4: Bayesian networks explanation for Bowel (B). ib_042 (Number of positive cores biopsied from LMZ?) has the following options: 1, 1 | 2, 2 | 3, Fragmented core tissue. ep_012 (Increased frequency of bowel movements?) has the following options: 1, No problem | 2, Very small problem | 3, Small problem | 4, Moderate problem | 5, Big problem.



ib_042[a]	**ep_012**[a]	**P[(Bowel**[b] **< mean) | ib_042,ep_012] (unhappy)**	**P[(Bowel**[b] **>= mean) | ib_042,ep_012] (happy)**
1	1	0.01	0.99
	2	0.35	0.65
	3	0.92	0.08
	4	1.00	0.00
	5	1.00	0.00
2	1	0.02	0.98
	2	0.46	0.54
	3	1.00	0.00
	4	1.00	0.00
	5	No Data	No Data
3	1	0.00	1.00
	2	1.00	0.00
	3	No Data	No Data
	4	No Data	No Data
	5	No Data	No Data

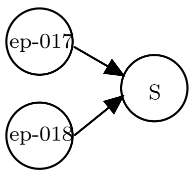
[a]Baseline visit value. [b]1-year follow-up value.

tracted by CB-DNN out of ˜400 variables which is predictable for each of the EPIC-26 domain. For example, the model discovered the ability to reach orgasm in baseline visit is the most predicable variable for 1-year follow-up in terms of Sexual quality of life. The results which are presented may vary in a different hospital and with different variables saved in patients databases. Hospital administrators may use their own database to obtain insights from the proposed method.

4.4.2 Bayesian network-based Explanation Model Results

In personalized healthcare, keeping track of patient medical history or even extracting important variables which can predict the outcome of interest are not the only objective for medical practitioners to maintain. But also they need to know the interaction between the important variables and the outcome of interest. Using the idea behind Bayesian network, practitioners can analyze the results from previous step and record patient's important variable for enhancing patient's quality of life. In addition, through necessary corrective actions, practitioners can get insights from the interaction between the variables in the past and provide better service for future patients. I applied the proposed hybrid Bayesian network

Figure 4.5: Bayesian networks explanation for Sexual (S). ep_017 (How would you describe your ability to have an erection during the last 4 weeks?) has the following options: 1, Very poor to none | 2, Poor | 3, Fair | 4, Good | 5, Very good. ep_018 (How would you describe your ability to reach orgasm (climax) during the last 4 weeks?) has the following options: 1, Very poor to none | 2, Poor | 3, Fair | 4, Good | 5, Very good.

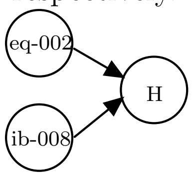


ep_017[a]	**ep_018**[a]	**P[(Sexual**[b] **< mean) | ep_017,ep_018] (unhappy)**	**P[(Sexual**[b] **>= mean) | ep_017,ep_018] (happy)**
1	1	1.00	0.00
	2	1.00	0.00
	3	No Data	No Data
	4	No Data	No Data
	5	0.33	0.67
2	1	1.00	0.00
	2	0.98	0.02
	3	1.00	0.00
	4	0.86	0.14
	5	0.75	0.25
3	1	1.00	0.00
	2	1.00	0.00
	3	0.82	0.18
	4	0.39	0.61
	5	0.25	0.75
4	1	No Data	No Data
	2	1.00	0.00
	3	0.40	0.60
	4	0.02	0.98
	5	0.00	1.00
5	1	No Data	No Data
	2	1.00	0.00
	3	0.00	1.00
	4	0.04	0.96
	5	0.00	1.00

[a]Baseline visit value. [b]1-year follow-up value.

model to all the patients but now only considering the top 10 variables from baseline visit extracted by CB-DNN and the 1-year follow-up EPIC-26 outcome variables to provide causal explanations for the relationship between the extracted variables and outcome of interest. Given that EPIC-26 outcome is a continues value between 0 and 100, I need to quantify it before applying the Bayesian network model. So I have divided the patients in to happy and unhappy groups based on their EPIC-26 outcome values. I call them unhappy because they are the ones with EPIC-26 value less than the mean value of the patients EPIC-26 values excluding those with the value of 100. Looking at test data in Fig. 4.1, which the distribution of the data is the same as whole data set, I see that the data set is imbalanced in terms of the number of patients with their EPIC-26 value is 100. Therefore, to calculate the mean value for each of the domains in EPIC-26 I have excluded those that the number

Figure 4.6: Bayesian networks explanation for Hormonal (H). eq_002 (Mobility?) has the following options: 1, I have no problems in walking about | 2, I have slight problems in walking about | 3, I have moderate problems in walking about | 4, I have severe problems in walking about | 5, I am unable to walk about. ib_008 (Gland volume?) is a numerical variable which those less than the mean and greater than the mean are labeled with 0 and 1 respectively.



eq_002[a]	**ib_008**[a]	**P[(Hormonal**[b] **< mean) | eq_002,ib_008] (unhappy)**	**P[(Hormonal**[b] **>= mean)| eq_002,ib_008] (happy)**
1	0	0.11	0.89
	1	0.09	0.91
2	0	0.28	0.72
	1	0.33	0.67
3	0	0.35	0.65
	1	0.50	0.50
4	0	0.78	0.22
	1	1.00	0.00
5	0	1.00	0.00
	1	1.00	0.00

[a]Baseline visit value. [b]1-year follow-up value.

of their EPIC-26 is 100. The happy group is defined as the ones who their EPIC-26 value is more than mean of the patients EPIC-26 values excluding those with the value of 100. For each of the 5 domains in EPIC-26, I build a Bayesian network to capture the relationship of most important variables which impact patients quality of life/EPIC-26 value after 1 year. The Bayesian network models produce two outcomes which impact the patient's EPIC-26 value. The results are shown in Figs. 4.2, 4.3, 4.4, 4.5 and 4.6, for UI, UIO, B, S and H, respectively. The Bayesian networks in these figures only show the interaction between two extracted variables and the outcome of interest, the rest of the interactions do not affecting the EPIC-26 analysis are excluded. The Bayesian network results help both medical practitioners and patients to understand the relationship between the predicted outcome in baseline visit and quality of the life for the patients after 1 year. It is worth mentioning that when the Bayesian network model is ready, I can start inferring to have answers in terms of probability for our questions or concerns. Now I am taking Hormonal domain as an example. From the CB-DNN model, the R-squared value that 10 extracted variables from baseline visit can predict the patients Hormonal severity value after 1 year is 0.71. In Fig. 4.6, the BN identified 2 locally important variables which can explain how these variables in baseline visit can affect their 1-year follow-up severity.

After inferring the Bayesian network learned for Hormonal, of these 2 variables (i.e. eq_002 (Mobility?) and ib_008 (Gland volume?)), I can see that those that their mobility (eq_002) is better have less probability to be unhappy for their hormonal dysfunction after one year. As such, if they have no problems in walking about (option 1 for eq_002) the probability to be unhappy after 1 year is only 0.1, while the probability for those who have moderate problems in walking about (option 3 for eq_002) is 0.38.

Note that although the Hormonal variable is conditioned on 2 variables, I perform the query based on the evidence which is available on only one variables (eq_002). It is straightforward to make our evidence richer by asking more specific question for individual patients. Table 4.7 depicts the results for what-if analysis to build scenarios on 5 randomly selected patients which are unhappy after 1 year in UI, UIO, B, S and H. For example, considering patient V (the fifth patient), the proposed Bayesian network method identified two locally important variables in baseline visit which can explain why patient V is unhappy in terms of hormonal quality of life after 1 year. After inspecting the current values of these 2 variables, I can see that this patient has a gland volume greater than mean (option 1 for ib_008) and have moderate problems in walking about (option 3 for eq_002). After performing What-If analysis, the BN shows that if patient V answered 1 (i.e. I have no problems in walking about) instead of 3 for question eq_002, the chances that he would be happy after 1 year for his hormonal quality of life would increase by 30%. I can better explain the impact of the extracted baseline visit variables with CB-DNN on patient' unhappiness by using this scenario-based what-if analysis that BN provides. As mentioned earlier, in this study, I am using baseline variables which are pre-treatment. However, the use of such a tool is not only useful for pre-treatment suggestion but can also be directly transferred to explain treatment related variables in a way to predict which treatment might help to improve HRQoL outcomes. The practitioners and patients can use these guidelines for future corrective measures and actions.

By comparing the top variables between two methods in Table 4.8, I figured that BN has

Table 4.7: What-if analysis scenarios for five patients.

EPIC-26 Variable	State Name	Influential Variable	Current Value	What-if	Changes in EPIC-26 Variable
Urinary Incontinence (Patient I)	A	eq_004	2	2 ->1	+ 12%
	B	im_001	2	2 ->1	+ 3%
	A and B	eq_004 im_001	2 2	2 ->1 2 ->1	+ 24%
Urinary Irritative /Obstructive (Patient II)	C	ib_064	3	3 ->1	+ 45%
	D	ep_015	1	1 ->0	+ 4%
	C and D	ib_064 ep_015	3 1	3 ->1 1 ->0	+ 56%
Bowel (Patient III)	E	ib_042	3	3 ->1	+ 3%
	F	ep_012	2	2 ->1	+ 55%
	E and F	ib_042 ep_012	3 2	3 ->1 2 ->1	+ 99%
Sexual (Patient IV)	G	ep_017	2	2 ->3	+ 21%
	H	ep_018	2	2 ->3	+ 33%
	G and H	ep_017 ep_018	2 2	2 ->1 2 ->1	+ 16%
Hormonal (Patient V)	I	eq_002	3	3 ->1	+ 30%
	J	ib_008	1	1 ->0	0%
	I and J	eq_002 ib_008	3 1	3 ->1 1 ->0	+ 39%

changed the ranking among those 10 variables extracted by CB-DNN. This suggests us that although some variables might be important in terms of predictability power they have, it cannot be guaranteed that they will have the same rank in terms of explainability power which is suggested by Bayesian network. Finding of this study provides solutions to peek inside the proposed black box and not only extracting the variable importance in a novel way for longitudinal data, but also make the results interpretable.

Table 4.8: Variable importance ranking comparisons between global algorithm CB-DNN and local Bayesian networks.

EPIC-26 Variable	**Variable Importance Ranking from Bayesian Networks Results. Rank 1 is:**	**Variable Importance Ranking by Deep Neural Network**
Urinary Incontinence	im_001	Rank 9
Urinary Irritative/Obstructive	ep_015	Rank 3
Bowel	ep_012	Rank 8
Sexual	ep_017	Rank 5
Hormonal	eq_002	Rank 3

4.5 Remarks, Limitations and On-going Work

Longitudinal data provides an excellent opportunity to study PCa patient's treatment satisfaction and HRQoL related outcomes over time. However, the analysis of such data (i.e. pre-processing and designing of predictive and interpretable models) is a challenging task due to the complex and hidden data structures, unevenly spread out visits (i.e. not all patients are evaluated at exactly the same time), missing data, and mixtures of time-varying and static measures that may affect the outcome in a non-linear complex relationships. Most of the literature focused on increasing the predictive power of the prediction models without investigating the ability of the model to explain the prediction. The contribution of this study is designing a model to predict the 1-year follow-up outcomes for the PCa patients with the highest degree of predictive power and with a decent ability to describe the predicted outcomes in terms of the most important influential input collected in the data set. This description will help practitioners and patients to share decisions about patient treatments that might affect certain aspects of their daily life routine and eventually their quality of life. Popular predictive models interpretation methods such as Local Interpretable Model-agnostic Explanations (LIME) technique, do not provide explanations of the predicted outcome based on nonlinear relationship. For example, LIME explains the predictions of a given classifier by learning a linear model locally around the prediction. The output of LIME is a list of explanations, describing the effect of each variable on the prediction of a data sample. This provides linear local interpretability, and it also allows to determine which variable changes will have the most impact on the prediction. The LIME is trained on small perturbations of the original instance and should only provide a local approximation. This presents a severe limitation of this method as it holds itself to the quality of the very nearby data points form a linear relationship.

However, in reality, the linear relationship is seldom found especially in the case of longitudinal data with such a complex structure of the data set under test (APCaRI data set). In this study, I adopt the Bayesian network algorithm to learn the hierarchical prediction explanation based on the most influential factors extracted by the CB-DNN algorithm. The CB-DNN algorithm provides a better classification per group as it performs stratified prediction that focus on each group of the patients and thus reduce the problem of highly imbalanced data.

4.6 Summary

Prostate cancer is one of the most complicated cancers that has a high level of unexplained variability such as urinary incontinence that might affect the patient's Health-Related Quality of Life (HRQoL). Being one of the most prevalent cancers worldwide, predicting and interpreting patient's treatment satisfaction is a vital step in improving HRQoL. Using 2670 patients' information with 433 measures per patient, my objective is to identify the minimal set of important variables which can predict 1-year follow-up HRQoL for prostate cancer patients. Given that both accuracy and interpretability are important criteria in healthcare for developing a reliable predictive model, we also need to add interpretability to the proposed model. Usually, complex models such as deep neural network which is part of the proposed approach can achieve high prediction accuracy but mostly black box, so they do not have enough interpretation to their prediction results. In this study, we address these problems (i.e. high accuracy and interpretable prediction model) by firstly developing deep neural networks on top of a clustering algorithm to extract minimal set of important variables of baseline visit. Secondly, we build a model to predict a 1-year follow-up of HRQoL for prostate cancer patients using the extracted important baseline variables. Thirdly, we utilize Bayesian networks method to provide insights into the proposed prediction model results to discover the relationship between patient variables and their satisfaction that can be used to explain the model prediction to a user (i.e. patient or a physician). The results support the use of the proposed machine learning technique as an essential tool in identifying potential variables for predicting 1-year HRQoL. Furthermore, the proposed approach to interpret the findings will help to establish guidelines for a better shared decision-making platform for patients with prostate cancer.

Chapter 5

Investigating NLP-based Approaches for Predicting Manual Test Case Failure

To address the problem described in section 1.1.3, this chapter is organized as follows: My proposed NLP-based method for test case prediction has been presented in Section 5.1. I have explained the experiments and results in Section 5.2 and finally, Section 5.3 concludes the chapter and present a summary.

5.1 Approach

In this section, I explain the NLP-based approach for test case failure prediction, as per Fig. 5.1, by first encoding the test cases using the mined features and then applying the prediction technique on the encoded test cases.

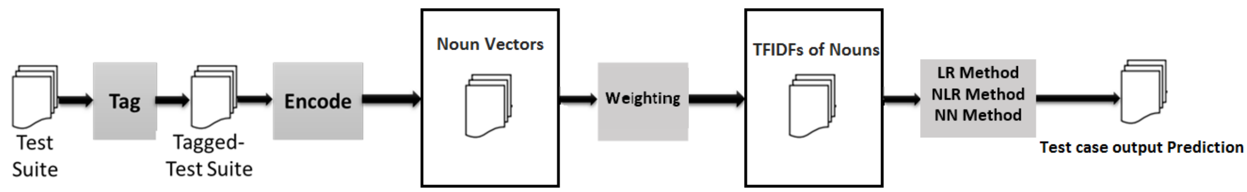


Figure 5.1: Overview of our approach

5.1.1 Test Case Encoding

A manual test case is typically a document that details the sequence of actions that testers need to take and the outputs they are expected to observe. A sample test case (only the instructions, not the expected outputs) is shown in Table 5.1 from Mozilla Firefox Mobile v16 (which also appears in v17, v18 and v27). In the absence of code and specification, my encoding's objective is to represent each test case as a set of GUI elements that are exercised/covered when the test case is executed.

The first step toward this objective is to extract the GUI elements from the test case's textual instructions. This can of course be done perfectly manually, but to do it automatically we need some assumptions and estimations. In this study, each noun is assumed to be an element of the GUI under test including the menus and sub-menus and their states that all need to be covered. For example, for the test case that is represented in Table 5.1 the list of nouns are <fennec, portrait, mode, device, landscape, mode, device, portrait, mode>

As it can be seen, all nouns except fennec, which refers to the browser itself, are valid estimations. Intuitively, we can guess that this estimation approach has a reasonable precision and very high recall, since if a test case exercises a GUI element or its state it's very likely that the test directly uses the element's or it's state name in the testing instructions (assuming that the test designers don't create very vague test cases).

In my experiment, I first preprocess all the test cases by excluding all URLs, special characters and numbers; typical preprocessing in standard text-mining [92]. Further, I split words, based on camel case and underscore, remove stop words, and stem multiple forms of the same word in one form. I use the Stanford Maximum Entropy Tagger [156] to extract nouns from the entire test suite. I ignore the sub-classifications of nouns and keep only words that are tagged as any sort of noun. Table 5.2 shows the tagging results for the sample test case of Table 5.1. Finally, each test case is represented as a set of nouns that are identified by the POS tagger.

5.1.2 NLP-based Prediction Models

My prediction models are LR, NLR and NN with two features. The first feature is based on the traditional history-based feature and the second feature is the GUI coverage of the test cases (estimated by nouns). To calculate the above features I follow the following steps:

Step 1: Calculate the history-based measures

- Simple History (SH): Assign value ONE to the test cases of this version if they passed in the immediate previous version assign value ZERO to the previously failed or new test cases.

- All History (AH): Assign the average of SH values in all previous versions. This measure considers the entire history rather than just the immediate version (SH).

- Weighted History (WH): Assign weighted average (as opposed to simple average of AH measure) of SH values in all previous versions with more importance on recent versions (decremental weights when starting from the newest release).

Step 2: Calculate the GUI coverage measure

- Tag all words in all test cases using a POS tagger and keep only the nouns.

- Represent each test case using its nouns. I call each of these nouns as the *unit* of representations.

- Traverse the entire test suite to find unique units.

- Calculate TF-IDF of each unit per test case.

- Assigned to each test case a measure which is the sum of TF-IDF of all unit's of the test case divided by the number of units in that test case.

Note that the sum of TF-IDFs basically looks at the coverage of the test as a whole without focusing on individual units.

To have a more clear idea about the process of the TF-IDF measure, let's see a sample test suite. Assume we have four test cases. The algorithm first tags all the test cases in a test suite and represents the test cases in the form of vectors of nouns, as below.

$$Test1 = <fennec, portrait, mode, device, landscape, mode, device,$$
$$portrait, mode>$$
$$Test2 = <news, article>$$
$$Test3 = <menu, settings, desktop, firefox, preferences, code, fennec,$$
$$matrix, desktop>$$
$$Test4 = <menu, settings, firefox>$$

I then traverse through the test cases and compute the TF-IDF of all unique nouns per test case as in Table 5.3. Next, I calculate the sum of the TF-IDF for all the nouns in each test case and then divide it by the number of nouns in that test case. For example, feature vector and normalized TF-IDF of the first test case described in Section III are as below:

$$Test1 = <fennec, portrait, mode, device, landscape, mode, device,$$
$$portrait, mode>$$

Test1 Normalized-TF-IDF:

$(0.03 + 0.13 + 0.2 + 0.13 + 0.06 + 0.2 + 0.13 + 0.13 + 0.2)/(9 + 1) = 0.121$

Finally, I use LR, NLR and NN algorithms to predict the PASS or FAIL for the test cases of each release using the history-based and the TF-IDF measures, as explained. To train

Table 5.1: A real test case from Mozila Firefox mobile version 16

Test Case ID:1206, Product: Mobile, Version:16
Steps to Perform
1. Launch fennec while in portrait mode.
2. Rotate the device to landscape mode.
3. Rotate the device back to portrait mode.

the models, these two measures should be calculated for each test case on a learning set (on all releases before the release under study). For each test case in the learning set, I assign a PASS or FAIL label, based on the actual test execution results in that release. The trained model then will be used in the the current release, to predict test failure using the current TF-IDF and history-based measures, per test case.

In the next section, I will apply this approach on real-world test suites.

Table 5.2: Real tagged test case from Mozila Firefox mobile version 16

Test Case ID:1206, Product: Mobile, Version:16
Tagged Steps
1. fennec/NN portrait/NN mode/NN.
2. device/NN landscape/NN mode/NN.
3. device/NN portrait/NN mode/NN.

5.2 Empirical Study

In this section, I explain the evaluation of the proposed test case prediction techniques, in the context of an experiment.

Table 5.3: Noun TF-IDF for test 1

Noun	TF-IDF
fennec	0.03
portrait	0.13
mode	0.2
device	0.13
landscape	0.06

5.2.1 Experiment Design

Objective

The objective of this experiment is to evaluate the effectiveness of the proposed NLP-based test case prediction techniques, in terms of prediction accuracy. The goal is to improve the state-of-practice (traditional history-based approach) for test case failure prediction.

Research Questions

I have formulated the objective of this study as the following research questions:

RQ1: Can a basic NLP-based test case prediction method improve the effectiveness of a traditional history-based test case prediction method? In this research question, I compare the prediction accuracy of the proposed NLP-based Linear Regression method with a simple history-based measure.

RQ2: Does looking at the entire history, when collecting the history-based measure, improve the prediction accuracy? In this research question, I upgrade the simple history-based measure (SH), which assumes the history only consists of the last version, to use the entire history (all previous versions). The two proposed new measures are All History (AH) and Weighted History (WH). AH considers an equal importance to the entire history but WH assigns more weight to more recent versions. This research question is divided into two sub-questions to evaluate SH, AH, and WH on the baseline and Linear Regression approach, separately.

RQ3: Among basic prediction approaches which one provides higher accuracy,

Table 5.4: Systems under test: Mozilla Firefox Desktop, Mobile and Tablet.

Type	Firefox Release	Release date	No. of tests	No. of faults	Failure rate
Desktop Firefox Releases (TR)	Firefox 3.0	Dec-06	580	127	21.90%
	Firefox 3.5	Jul-07	766	138	18.02%
	Firefox 3.6	8/2009	828	88	10.63%
	Firefox 4.0	2/2010	997	150	15.05%
Desktop Firefox Releases (RR)	Firefox 5.0	4/2011	1055	6	0.57%
	Firefox 6.0	4/2011	1119	4	0.36%
	Firefox 7.0	5/2011	1111	4	0.36%
	Firefox 8.0	7/2011	1119	7	0.63%
	Firefox 9.0	8/2011	1114	4	0.36%
	Firefox 10.0	9/2011	1108	12	1.08%
	Firefox 11.0	11/2011	1121	3	0.27%
	Firefox 12.0	12/2011	1121	2	0.18%
	Firefox 13.0	2/2012	1189	4	0.34%
Mobile Firefox Releases	Firefox 16	Jun-12	364	79	21.70%
	Firefox 17	Aug-12	367	72	19.62%
	Firefox 18	Aug-12	366	87	23.77%
	Firefox 19	Oct-12	388	76	19.59%
	Firefox 20	Nov-12	381	95	24.93%
	Firefox 21	Jan-13	389	72	18.51%
	Firefox 22	Feb-13	423	79	18.68%
	Firefox 23	Apr-13	426	76	17.84%
	Firefox 24	May-13	476	82	17.23%
	Firefox 25	Jul-13	514	83	16.15%
	Firefox 26	Aug-13	343	50	14.58%
	Firefox 27	Sep-13	434	84	19.35%
	Firefox 28	Nov-13	380	28	7.37%
	Firefox 29	Dec-13	293	21	7.17%
Tablet Firefox Releases	Firefox 16	Jul-12	368	77	20.92%
	Firefox 17	Aug-12	372	77	20.70%
	Firefox 18	Sep-12	377	98	25.99%
	Firefox 19	Oct-12	385	76	19.74%
	Firefox 20	Nov-12	391	62	15.86%
	Firefox 21	Jan-13	390	67	17.18%
	Firefox 22	Feb-13	396	74	18.69%
	Firefox 23	Apr-13	446	88	19.73%
	Firefox 24	May-13	448	42	9.38%
	Firefox 25	Jul-13	490	82	16.73%
	Firefox 26	Aug-13	333	50	15.06%
	Firefox 27	Sep-13	372	61	16.40%
	Firefox 28	Nov-13	345	39	11.30%
	Firefox 29	Dec-13	319	31	9.72%

when used with the NLP-based features, for test case failure prediction? In this research question, I investigate the effect of the prediction approach, on test case failure prediction. I use three basic and simple models for this comparison: Linear Regression (LR), Nonlinear Regression (NLR), and Neural Network (NN).

Dataset

In this study, I use a data set of 3 Mozilla Firefox projects (Desktop, Mobile and Tablet). The data for these projects are collected from two previously published work on Mozilla Firefox's manual test suites [62], [63]. The Desktop test suites were collected from the test cases of versions 5 to 13 of Firefox Desktop and their execution results. This project was managed by a web-based system known as Litmus. The Mobile and Tablet test suite are collected from a test management tool called Moztrap. These are test cases for versions 16 to 29 of Firefox Mobile and Tablet. Except in the first four versions of Desktop, the rest of releases follow rapid release strategy. The implication of that in this study is that the releases are small with limited changes and failure per release. There are also a lot of overlap in test suites of consecutive versions.

Table 5.4 shows the characteristic of the 13 (four traditional and nine rapid) releases of Desktop Firefox, 14 releases of Mobile Firefox and 14 releases of Tablet Firefox.

In both Litmus and Moztrap, there are records of each test execution. Each test has been executed several times by perhaps different testers. To assign a PASS or FAIL label to test cases, I first exclude unsuccessful executions of tests that are labeled as erroneous test, no PASS or no FAIL in the test report. After this cleaning step, a test case is considered as PASS only if all valid executions of the test, in that version, were passed. Therefore, a test case is considered as FAILED if there is at least one failed execution, assigned to that test case, in that version. The test execution reports also identify the bug *id* which let us identify unique faults and assign each test case to the unique fault(s) that it can detect.

In this study, the results are reported starting from the third version per project. This is due to the fact that each prediction requires a training dataset (previous version) and the history measures per test case in the training set themselves need at least one release, as history. Thus the first version to predict the result for will be the third release per project.

Comparison Baseline

The baseline in this study is a simple history-based method (called SimpleHistory-BaseLine or SH-BL, from now on). Basically, I predict the test cases that fail in the previous version to fail again and those that passed previously to pass again. I also predict the new tests (does not exist in the previous version) as fail, to give them a chance to be executed, in the new release. Note that in our dataset, percentages of new test cases from a given release to the next one is typically very low (ranges from 0.3% to 13% in Tablet – 1% to 13% in Mobile – 0.1% to 6% in Desktop Rapid Release, but 12% to 29% in Desktop Traditional release). The SH-BL prediction method is the foundation of many test case prioritization approaches. Despite its simplicity this approach works very well in regression testing, specially when the releases are frequent (the changes are limited in each iteration).

Another technique that could be considered here as a baseline of comparison was an alternative text mining-based approach for extracting features from the textual test cases. From the existing literature, the only such approach is an LDA-based test failure prediction, which has shown poor results compared to SH-BL [62][64][154]. Thus I do not consider that as a baseline.

There are also some techniques which are based on image understanding from the GUI [42], which could only be applicable if our dataset would include screen-shots and images per test cases. Therefore, I keep SH-BL as the baseline of comparison .

Normalized-TF-IDF

To calculate the TF-IDF measure, I divide the raw TF-IDFs by the sum of TF-IDFs of all nouns to normalize the measure, and I call this measure “Normalized-TF-IDF”, which will

Table 5.5: Accuracy of twelve fault prediction techniques in the Desktop, Mobile and Tablet Firefox – BL(Baseline), LR(Linear Regression), NLR(Nonlinear Regression), NN(Neural Network), SH(Simple History), AH(All History), WH(Weighted History). For NN Average(AVG) and Standard Deviation(SD) of 10 runs are reported.

Type	Version	BL			LR			NLR			NN					
		SH	AH	WH	SH	AH	WH	SH	AH	WH	SH		AH		WH	
											AVG	SD	AVG	SD	AVG	SD
Desktop (TR)	3.6	0.51	0.39	0.60	0.63	0.47	0.63	0.63	0.48	0.63	0.63	0.08	0.61	0.06	0.58	0.00
	4.0	0.45	0.54	0.50	0.69	0.67	0.66	0.69	0.67	0.67	0.68	0.00	0.64	0.01	0.66	0.00
Desktop (RR)	7.0	0.79	0.81	0.80	0.89	0.79	0.79	0.79	0.79	0.79	0.79	0.01	0.78	0.01	0.79	0.00
	8.0	0.87	0.86	0.86	0.88	0.87	0.88	0.88	0.87	0.88	0.88	0.01	0.86	0.00	0.87	0.00
	9.0	0.90	0.85	0.84	0.85	0.85	0.85	0.91	0.85	0.85	0.91	0.01	0.86	0.02	0.85	0.02
	10.0	0.85	0.86	0.85	0.87	0.85	0.85	0.86	0.85	0.85	0.86	0.01	0.85	0.00	0.85	0.00
	11.0	0.90	0.91	0.91	0.94	0.90	0.91	0.91	0.90	0.90	0.91	0.01	0.90	0.00	0.90	0.01
	12.0	0.91	0.91	0.91	0.89	0.91	0.91	0.91	0.91	0.91	0.91	0.01	0.91	0.00	0.91	0.00
	13.0	0.85	0.88	0.85	0.94	0.88	0.88	0.88	0.88	0.88	0.88	0.01	0.88	0.01	0.89	0.00
Mobile	18	0.83	0.74	0.83	0.80	0.83	0.83	0.80	0.82	0.82	0.81	0.01	0.81	0.02	0.83	0.02
	19	0.82	0.81	0.77	0.86	0.85	0.85	0.86	0.86	0.86	0.80	0.05	0.84	0.05	0.85	0.01
	20	0.85	0.76	0.83	0.82	0.83	0.84	0.83	0.82	0.84	0.83	0.01	0.84	0.00	0.85	0.00
	21	0.85	0.80	0.85	0.86	0.87	0.88	0.86	0.87	0.87	0.86	0.01	0.86	0.01	0.88	0.00
	22	0.87	0.75	0.83	0.86	0.86	0.86	0.86	0.86	0.88	0.87	0.01	0.87	0.00	0.87	0.00
	23	0.78	0.73	0.77	0.82	0.88	0.85	0.83	0.85	0.85	0.84	0.01	0.86	0.01	0.85	0.01
	24	0.80	0.69	0.76	0.88	0.88	0.88	0.88	0.88	0.88	0.87	0.02	0.88	0.00	0.87	0.00
	25	0.83	0.65	0.71	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.01	0.88	0.00	0.88	0.00
	26	0.82	0.61	0.68	0.94	0.94	0.94	0.94	0.94	0.94	0.93	0.03	0.94	0.01	0.93	0.01
	27	0.90	0.68	0.74	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.01	0.90	0.00	0.91	0.00
	28	0.91	0.63	0.77	0.96	0.97	0.97	0.96	0.97	0.97	0.93	0.01	0.96	0.00	0.96	0.00
	29	0.94	0.65	0.82	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.01	0.97	0.00	0.97	0.00
Tablet	18	0.88	0.83	0.88	0.88	0.89	0.89	0.88	0.89	0.89	0.88	0.01	0.87	0.01	0.88	0.01
	19	0.78	0.81	0.76	0.79	0.81	0.82	0.79	0.82	0.82	0.79	0.05	0.81	0.03	0.81	0.03
	20	0.88	0.77	0.80	0.89	0.89	0.89	0.89	0.89	0.89	0.88	0.01	0.86	0.01	0.85	0.01
	21	0.91	0.83	0.79	0.90	0.88	0.88	0.90	0.90	0.91	0.90	0.01	0.90	0.01	0.91	0.01
	22	0.85	0.75	0.85	0.85	0.84	0.84	0.85	0.84	0.85	0.85	0.01	0.84	0.02	0.85	0.02
	23	0.79	0.74	0.76	0.87	0.87	0.87	0.86	0.87	0.87	0.86	0.01	0.85	0.01	0.85	0.00
	24	0.81	0.66	0.71	0.94	0.94	0.94	0.93	0.94	0.94	0.93	0.02	0.92	0.01	0.92	0.01
	25	0.86	0.65	0.71	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.01	0.89	0.00	0.89	0.00
	26	0.83	0.61	0.67	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.02	0.94	0.01	0.94	0.00
	27	0.92	0.64	0.74	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.01	0.93	0.00	0.93	0.00
	28	0.93	0.62	0.81	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.01	0.95	0.00	0.95	0.00
	29	0.96	0.66	0.86	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.01	0.96	0.00	0.96	0.00

be used as one of the two inputs to the prediction models. In addition, in order to avoid a "Division by zero" in test cases without any noun, I added one to the size of the nouns.

Neural Network Set up

The NN I employ in this study is a simple multi-layer perceptron with only 2 layers containing 10 neurons in hidden layer. The network has 2 inputs, each for (Normalized-TF-IDF, History) and one output for pass or failing result. I use the logistic function as the activation function inside the neurons with backpropagation as the learning mechanism. Note that the NN model is designed to be simple and cheap, so that it is comparable with LR and NLR. Therefore, tuning NN or using higher levels of hidden layers were not an objective of this study.

Evaluation Metric

In this study, accuracy of the predictions are used to evaluate the techniques (the proposed approaches and the baseline) over different releases. The accuracy metric is defined as 5.1:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

where,

TP True Positive

TN True Negative

FP False Positive

FN False Negative.

To compare the accuracy values of the baseline and the proposed approaches, I apply a non-parametric significant test, a matched pair Wilcoxon's signed rank test [93], to determine if the difference between the accuracy results are statistically significant or not. After applying

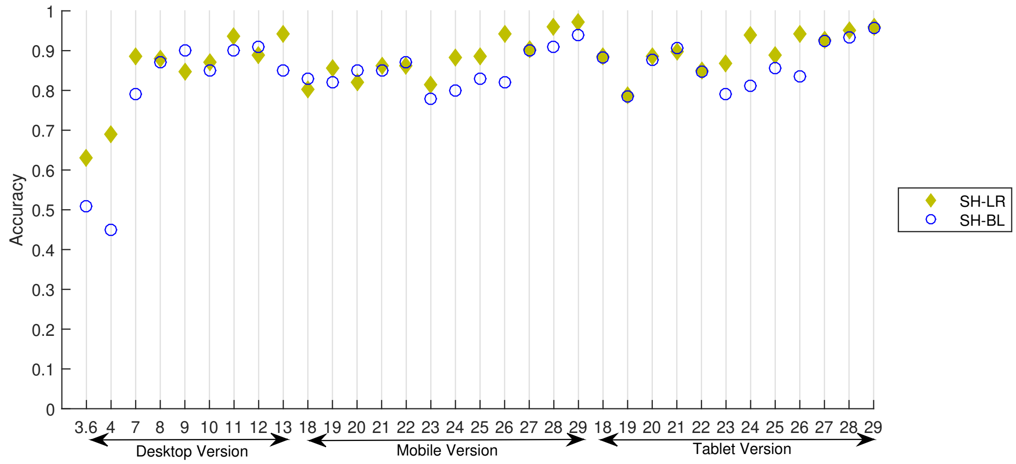


Figure 5.2: Accuracy of Firefox versions; Simple History Baseline (SH_Baseline) vs Simple History Linear Regression (SH_LR).

the test on the predicted output results, a p-value has been calculated over the releases of Desktop, Mobile, and Tablet projects, separately. The p-value threshold is set to 0.05. Note that in the case of NN, which is a randomized technique, I repeat each run 10 times and report the mean values. The significant test is then applied in the mean values over different releases, for each approach.

5.2.2 Case Study Results

In this section, I explain and discuss the results of the experiment and answer the research questions.

RQ1:

To answer this research question, I compare the simple history-based baseline method (SH-BL) with the Linear Regression as a standard statistical technique for classification, which uses both history-based metric and Normalized-TF-IDF. The idea is to see the effect of NLP without giving much advantages to the technique by using an advanced prediction model. Note that I can not use the baseline directly (i.e. witout a prediction method) with the NLP-based encoding, thus SH-LR, in this study, is the simplest NLP-based prediction

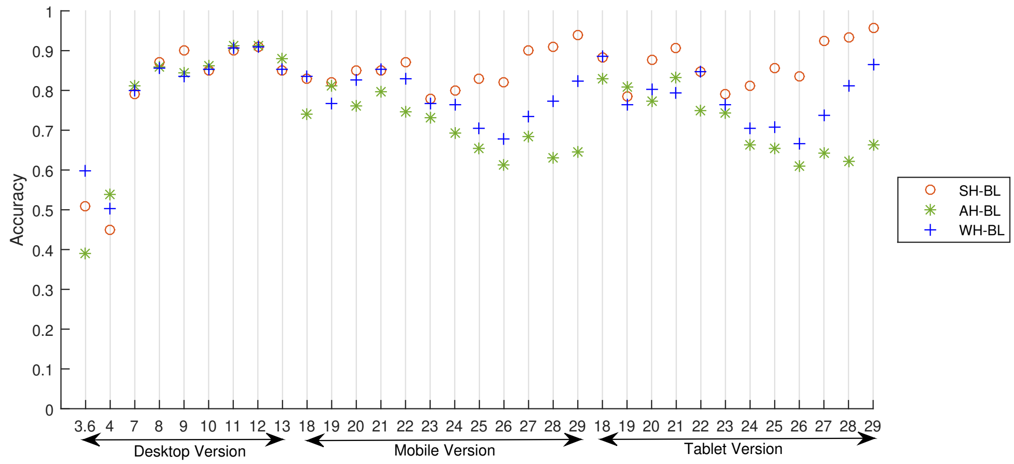


Figure 5.3: Accuracy of Baseline (BL) based on History features; Simple History (SH), All History (AH) and Weighted History (WH).

approach.

Looking at Fig. 5.2, our first observation is that overall the accuracies are high for both techniques. This shows that the history-based approach is quite effective in these releases (mostly rapid releases). However, the SH-LR shows even higher accuracy. We can see accuracy improvements up to 24% (e.g., for version 4.0 Firefox Desktop). In addition, the low p-values reported in Table 5.6 , 0.0004, show that the improvements are not by chance.

Another observation is that given that in many versions SH-BL already provides high accuracies, the average improvements provided by SH-LR are not that high (on average 3%, 3%, and 6% improvements, on Tablet, Mobile, and Desktop projects, respectively). However, the maximum improvements are more considerable: up to 12% (in version 24), 12% (in version 26), and 24% (in version 4) in Tablet, Mobile, and Desktop projects, respectively. It also worth mentioning that SH-BL outperforms SH-LR only in 6 cases out of 33 releases.

Therefore, the answer to the RQ1 is Yes! I can improve the basic history-based test case prediction using a simple NLP-base LR test case prediction model, even when it has already a high accuracy. This is quite interesting since it shows that the mined textual data brings extra knowledge that was not in the execution history. This knowledge is basically the coverage information which can not be seen in the history metric. However, the history

Table 5.6: P-value of baseline and modeling techniques

Method		BL			LR			NLR			NN		
		SH	AH	WH	SH	AH	WH	SH	AH	WH	SH	AH	WH
BL	SH	1.0000	0.0000	0.0003	0.0004	0.0025	0.0002	0.0001	0.0019	0.0000	0.0002	0.0049	0.0000
	AH	0.0000	1.0000	0.0004	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	WH	0.0003	0.0004	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
LR	SH	0.0004	0.0000	0.0000	1.0000	0.2627	0.6389	0.4074	0.4781	0.5034	0.0811	0.0589	0.3498
	AH	0.0025	0.0000	0.0000	0.2627	1.0000	0.0627	0.5633	0.4080	0.0099	0.5698	0.0641	0.7935
	WH	0.0002	0.0000	0.0000	0.6389	0.0627	1.0000	0.4237	0.3061	0.0424	0.0656	0.0028	0.4106
NLR	SH	0.0001	0.0000	0.0000	0.4074	0.5633	0.4237	1.0000	0.8484	0.0363	0.0687	0.0261	0.5372
	AH	0.0019	0.0000	0.0000	0.4781	0.4080	0.3061	0.8484	1.0000	0.0010	0.5178	0.0548	0.8370
	WH	0.0000	0.0000	0.0000	0.5034	0.0099	0.0424	0.0363	0.0010	1.0000	0.0032	0.0001	0.0589
NN	SH	0.0002	0.0000	0.0000	0.0811	0.5698	0.0656	0.0687	0.5178	0.0032	1.0000	0.3595	0.4774
	AH	0.0049	0.0000	0.0000	0.0589	0.0641	0.0028	0.0261	0.0548	0.0001	0.3595	1.0000	0.0342
	WH	0.0000	0.0000	0.0000	0.3498	0.7935	0.4106	0.5372	0.8370	0.0589	0.4774	0.0342	1.0000

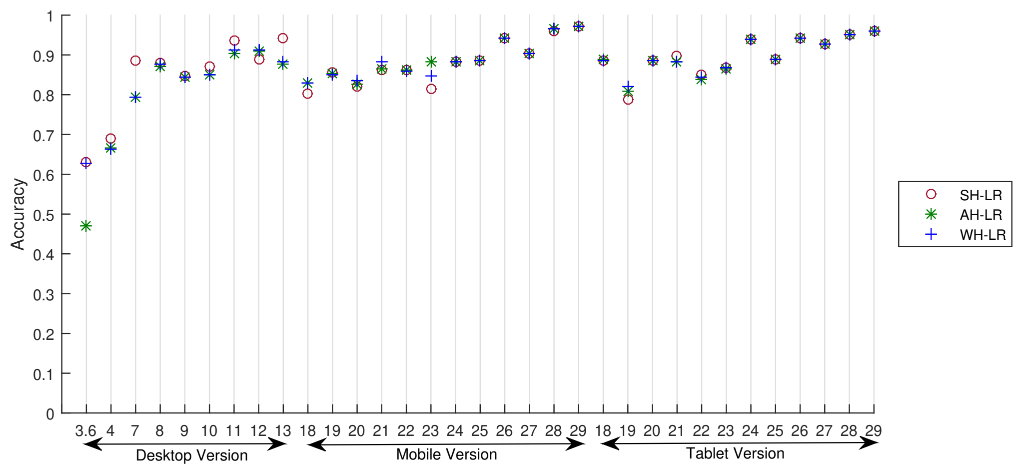


Figure 5.4: Accuracy of Linear Regression (LR) based on History features; Simple History (SH), All History (AH) and Weighted History (WH).

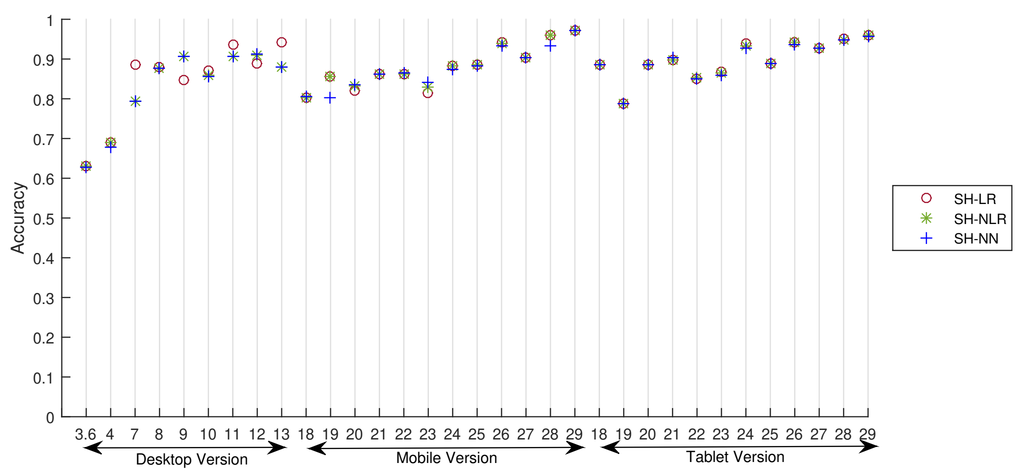


Figure 5.5: Accuracy of different learning methods; SH-LR, SH-NLR and SH-NN.

measure that was used in RQ1 was quite naive and simple. So the next question will be whether a more complex history measure can help in prediction or not, which is studied in RQ2.

RQ2:

The objective of RQ2 is to study the history-based measure in more details. In RQ2.1, I want to know whether SH-BL can be improved if I consider the entire history rather than only previous venison, or not. This basically tries to improve the baseline without using the NLP-based feature. In RQ2.2, I study the same question but in the context of the proposed NLP-based LR method. In other words, I try to improve the SH-LR by a better history-based feature. To answer RQ2.1 and 2.2, I replace SH with AH and WH, as defined in Section 5.1.2. Fig. 5.3 and 5.4 show the accuracy achieved by the baseline and LR models using the different historical attributes. Focusing on RQ2.1, (Fig. 5.3) shows that while all three modifications of history are almost the same in Desktop versions, Simple History works the best on Mobile and Tablet, meaning that the most significant information, in terms of simple baseline method, is in the last version and then in the more recent versions, explaining why Weighted History is outperforming All History. Note that the reason behind poor performance of SH in the first versions of Desktop is that those versions do not follow rapid release and each version lasts for almost a year. Therefore, the previous version's test execution results are outdated.

Looking at Fig. 5.4 to answer RQ2.2, I see that all three modifications of history-based features, in the context of the NLP-based LR model, have the same level of accuracy, indicating that expanding history from one version to all previous versions, does not influence the learning accuracy of the NLP-based model. The above conclusion can also be drawn by looking at high p-values between SH-LR with WH-LR and AH-LR which are 0.63 and 0.26.

Therefore, given that the cost of collecting AH and WH features are higher than SH and they do not provide better accuracies, in most cases, the answer to RQ2 is: NO! Looking at the richer history does not improve the prediction accuracies, at least when using BL or LR, as the prediction model. Now the follow up question is whether improving the prediction model helps in improving the accuracy or not, which is studied in RQ3.

RQ3:

In this RQ, I focus on the learning side of the approach. So I keep the SH-LR as the default history measure and replace LR with NLR and NN models.

Fig. 5.5 compares these three techniques, in terms of accuracy. As a general observation, I can see that replacing the learning algorithm does not provide any significant changes compared to SH-LR, which means that the output of the model, i.e., the failure of the test cases, can be learned without a complex model as well. Table 5.5 shows the results for all 12 combinations of the three history-based features and four prediction methods. The conclusions drawn from those parts of the table that are related to Fig. 5.5 is that replacing LR with an NLR or NN-based approaches does not help improving the accuracies of the predictions. This is also clear from the p-value results in Table 5.6, where comparing SH-LR with none of the *-NLR and *-NN approaches result in lower than threshold p-values.

Given that the more complex prediction models also come with more cost and tuning needs, conclusion is that extracting the GUI coverage information from the textual data in test cases is possible through a simple NLP-based approach (POS+TF-IDF). This feature combined with a basic history-based feature (historical test execution data from previous release) can potentially provide a highly accurate test case failure prediction approach, using a linear regression classifier.

5.3 Summary

Manual acceptance testing is one of the most expensive testing activities. In manual testing, typically, a human tester is given an instruction to follow, mostly based on the GUI. Then the tester records the results as "passed" or "failed", according to the instructions. Therefore, any attempt in reducing the amount of this type of expensive testing is essential in practice. Unfortunately, most of the existing heuristics for test selection, prioritization, and reduction are either based on source code or specification of the software under test, which are typically not being accessed by the manual acceptance testers.

In this study, I propose a feature that extracts some GUI-related information per test case using Natural Language Processing (NLP) and Information Retrieval (IR) methods. The feature is based on the frequency of covering different elements of the GUI in the manual test scripts. I show that a simple linear regression model using the extracted NLP/IR-based feature and a typical history-based feature (previous test execution results) can accurately predict the test cases' failure in new releases. This information is useful for test managers to effectively assign their testing effort. I have conducted an extensive empirical study on manual test suites of 41 releases of Mozilla Firefox over three projects (Mobile, Tablet, Desktop). The comparison of several proposed approaches for predicting failure shows that a) I can accurately predict the test case failure and b) the NLP-based feature can improve the prediction models by up to 24%.

Chapter 6

Combining Geophysical Models with Convolutional Neural Network for Automatic Velocity Picking in Seismic Data

This chapter is presented to address the deficiencies in current literature in implementing the convolutional neural network for seismic velocity picking as explained in Section 1.1.4. To do this, this chapter is organized as follows: Section 6.1 introduces the methodology of the work and describes CNN semblance problem formulation and hybrid geophysical approach with CNN in Sections 6.1.3 and 6.1.4 respectively. Section 6.2 explains the fields data used for evaluation and compares the proposed approach with state-of-the-art approach. Section 6.3 presents a summary of this chapter.

6.1 Approach

The objective of the algorithm is to get a semblance panel input image S and automatically determine stacking velocities V_Γ (henceforth "velocities") on that semblance where traveltime Set Γ is detected in reference file. For this purpose, in this section, I explain our hybrid CNN approach combined with geophysical models. The main process of our method is shown in Fig. 6.1.

I use the following notation throughout the study:

τ : interval zero-offset two-way traveltime of the reflection hyperbola (henceforth "traveltime").

t_m : traveltime at index m in Set Γ.

P_m : semblance height Pixel associated with traveltime at index m in Set Γ .

W, H : Width and Height of semblance panel input image S respectively .

V_{min}, V_{max} : Minimum and Maximum velocities in the semblance panel which are 1500m/s and 600m/s respectively . V_{t_i}

6.1.1 Semblance Encoding

Since its introduction by Taner and Koehler (1969) [153], the semblance as an energy measure has been an indispensable tool for optimally determine stacking velocities [104]. A sample semblance panel input image with its true reference velocities is shown in Fig. 6.2 from Vermilion field (Canada, 3D). The traveltime $m, m \in \{1, 2, ..., M_j\}$ in the text file, denoted by t_m corresponds to the input height pixel (P_m) with the following equation:

$$P_m = \lfloor \frac{t_m}{Fs} \rfloor \tag{6.1}$$

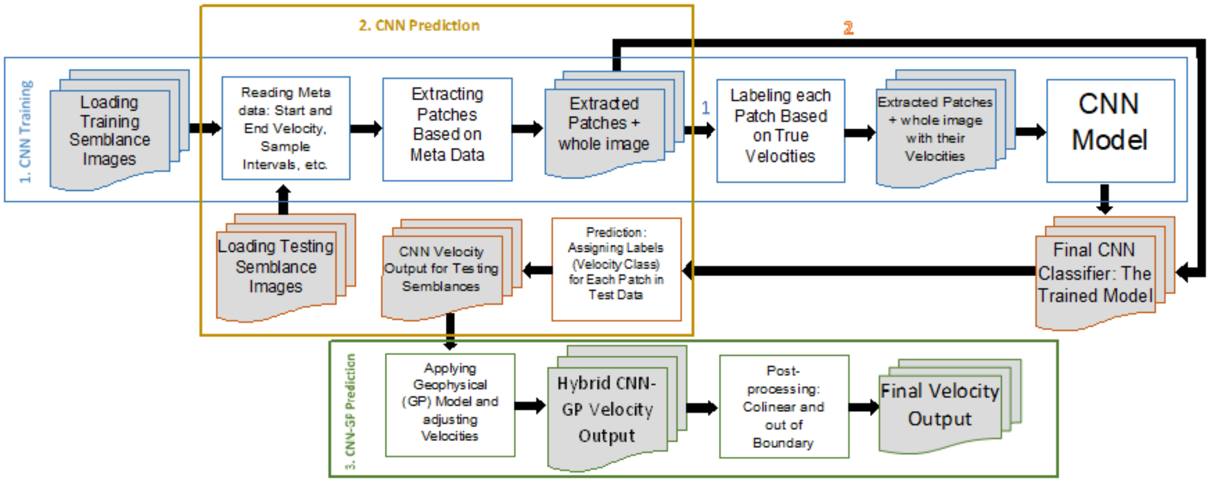


Figure 6.1: Overview of our approach which includes three phases: 1. CNN Training 2. CNN Prediction 3. CNN-GP Prediction.

where $Fs(ms)$ is the sampling frequency (henceforth I use the terms "sampling frequency" and "sample interval" interchangeably) and M_j is the number of velocity picks in the text file for semblance input number j. For example, sample interval in Fig. 6.2 is $2(ms)$. Therefore, the last velocity pick which is in time 2914.7, corresponds to Height pixel of 1457 ($\lfloor \frac{2914.7}{2} \rfloor$) in semblance panel input image.

The traveltime shown in reference velocities are selected by expert. Therefore, in the absence of the traveltime in CNN prediction phase, our semblance encoding objective is to extract patches from input image that can simulate the traveltime selected by expert. For this purpose, I have proposed the following two methods to extract the patches from the semblances.

Velocity-based Patch Size: For each input image, a moving time window is used to obtain subset patches of the input image for determining traveltime set from the input image and their corresponding velocities from the reference velocity Set in the text file. The moving time window needs 2 pixel value couples to be specified on the semblance image which are $(top, left)$ and $(bottom, right)$. As the Width of patches remains unchanged, $left$ and $right$ values are always constant values of 0 and W respectively. As shown in Fig. 6.3 each patch

Reference Velocity
(Picked by Expert)

Time(ms)	Vrms(m/s)
0.0	2532.35
13.4	2532.35
241.2	2714.72
377.8	2905.78
634.4	3183.67
804.6	3357.36
1013.3	3487.62
1249.8	3602.21
1383.3	3645.45
1463.3	3740.57
1663.4	4008.65
1947.0	4276.72
2914.7	4746.85

Semblance Detailed Info

Project Name: 1933-vermilion
Line Name: ratwolf20083d
BIN_line: 52
BIN_trace: 100
Sample_Interval (ms): 2.0

Figure 6.2: Real Semblance Panel Input (left) and the its reference velocity file picked by expert (right top). The Semblance is for Vermilion project (2008) with the detailed info shown in right bottom

represents the velocity at t_i that falls into the following τ interval:

$$\tau_{top} = \lfloor \frac{t_(i-1)}{Fs} \rfloor)$$

$$\tau_{left} = 0$$

$$\tau_{bottom} = \lfloor \frac{t_(i+1)}{Fs} \rfloor)$$

$\tau_{Right} = W$

where $i \in \{1, 2, ..., N_j\}$ and N_j is the total Number of Patches in the semblance panel input image number j. It should be noted that in this approach, N_j is equal to M_j (The number of velocity picks in text file), because I am considering the traveltime associated with velocity picks to extract the patches. i (index for patch number in input image) is also exactly the same as m (index for the traveltime in Reference Velocity text file).

The extracted Patch is denoted by S_{j,t_i}, where j indicate the semblance panel input image number and the index $i = \{1, 2, ..., N_j\}$ corresponds to travel time t_i in semblance panel input image j.

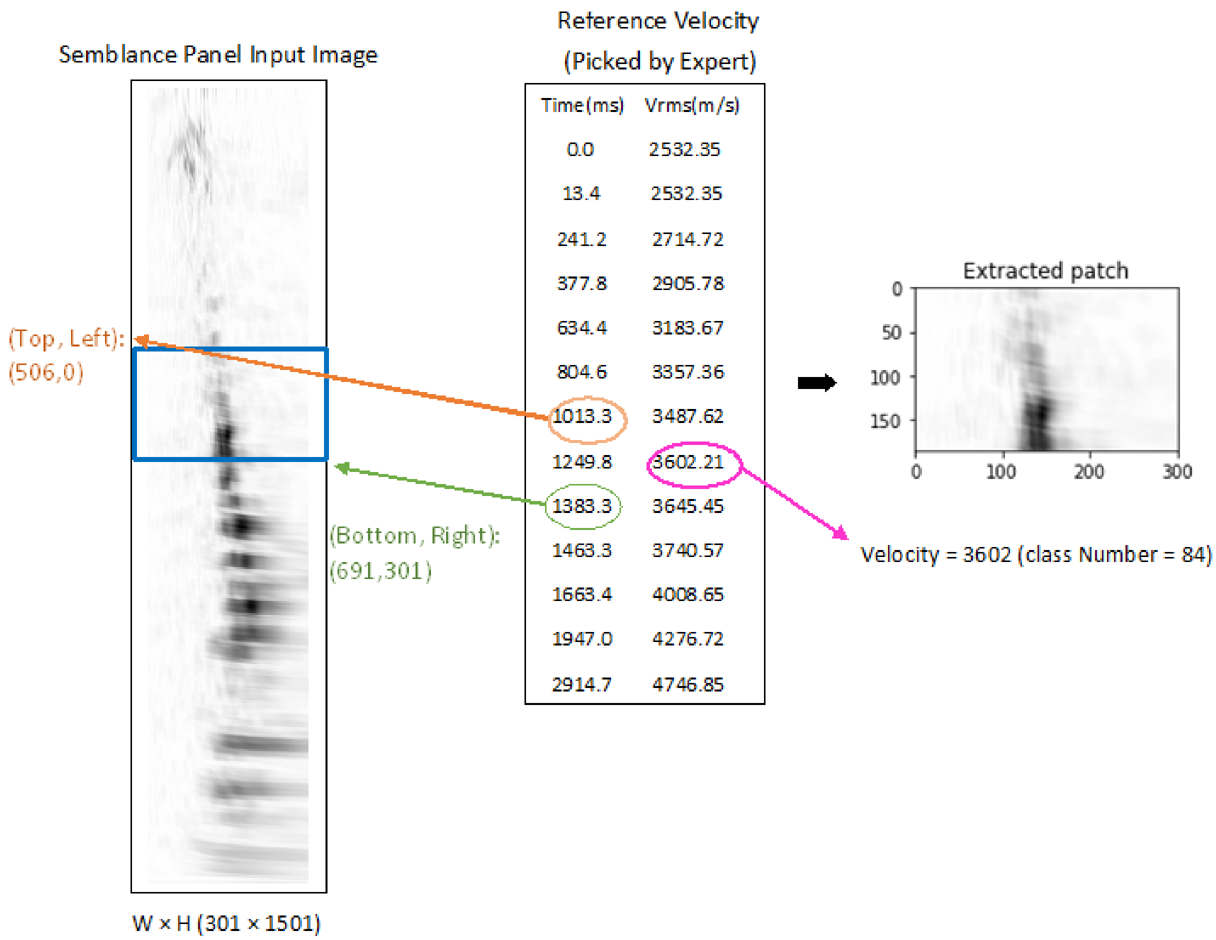


Figure 6.3: A velocity-based moving time window (τ interval) shown in blue square is used to obtain subset patches of the input image for determining traveltime from the Semblance image and the corresponding velocities from the reference velocity Set is assigned.

Constant Patch Size: In this study, I am also considering a fixed size for moving time window, to obtain patches with constant size throughout the semblance panel input image. The reason behind considering fixed window size is that the traveltime selected by expert are not always available. Such that when a new dataset for testing (i.e. CNN prediction phase in Fig. 6.1) is used, obviously there is no reference velocity file to get the required traveltime. To Minimize the effect of resizing the patches which will be applied later for CNN training phase, I consider the Height of the moving window time to be constant value of 50 pixels without overlapping between patches. Therefore, N_j (total Number of Patches in the semblance panel input image j) is $N_j = \lceil \frac{H}{50} \rceil$. In other word, although the patch size is constant, the number of patch per semblance might be different.

Regarding the $(top, left)$ and $(bottom, right)$ for moving time window (τ interval) in this approach, $left$ and $right$ values are again constant values of 0 and W respectively. top and $bottom$ values start from 0 and 50 and increase by 50 for each next patch.

In CNN Training phase, I need to assign the reference velocities to extracted patches as well. To achieve this goal, considering patch S_{j,t_i}: a) if there is any t_m in the reference velocity text file that falls into the τ interval of this patch, I assign V_{t_m} as the velocity for patch S_{j,t_i}, where V_{t_m} is the velocity picked by expert at time t_m. b) Otherwise, if there is no t_m that falls into the τ interval of this patch, it means τ interval (which t_i is in the middle) is between two traveltime t_m and t_{m+1} in the reference velocity text file. With expert suggestion, I consider a linear line between V_{t_m} and $V_{t_{m+1}}$ and calculate V_{t_i} as follows:

$$V_{t_i} = \frac{V_{t_{m+1}} - V_{t_m}}{t_{m+1} - t_m} \cdot (t_i - t_m) + V_{t_m} \tag{6.2}$$

where, t_i and V_{t_i} are the traveltime and velocity for patch index i denoted as S_{j,t_i}.

To have a more clear idea about the process of the assigning velocity to a patch that does not have one, let's see a sample patch from Fig. 6.3. Assume we have a patch with the following τ interval:

$$\tau_{top} = 550, \tau_{left} = 0$$

$$\tau_{bottom} = 600, \tau_{Right} = 301$$

Considering the Fs of the semblance which is $2(ms)$ and using equation 6.1, the t_i located at the middle of the patch is:

$$P_i = \frac{(550 + 600)}{2} = 575$$

$$\rightarrow t_i = 575 * 2 = 1150(ms)$$

Therefore, t_i is between t_m and t_{m+1} with the values 1013.3 and 1249.8 respectively. Using the equation 6.2, V_{t_i} for this patch is calculated as below:

$$V_{t_i=1150} = \frac{3602.21 - 3487.62}{1249.8 - 1013.3} \cdot (1150 - 1013.3) + 3487.62 = 3553.24$$

6.1.2 Velocity Labelling

labelling the velocities to fit into our classification workflow process while keeping its value precisely is an important step. Based on expert suggestion, I expect velocity piked for the semblances to fall within an certain range of rock velocities obtained from decades of observational and empirical evidence. That range is between 1500m/s and 6000m/s. These are considered as the min and max (respectively) values of the velocity in this study. The range of the velocity in each class is considered to be 25m/s and I have 180 classes (labels). For example, all the velocities between 1500m/s to 1525 are considered as class 0, from 1525m/s to 1550 are considered as class 1 and so on. The class number associated with velocity V_{t_i} is denoted as Γ_{t_i} and is calculated as below:

$$\Gamma_{t_i} = \lfloor \frac{V_{t_i} - V_{min}}{25} \rfloor \tag{6.3}$$

For example, the velocity calculated for the patch extracted in Fig. 6.3 is 3602.21. The class number associated with this patch is $\lfloor \frac{3602.21-1500}{25} \rfloor = 84$.

6.1.3 CNN Semblance Problem Formulation

In order to obtain the objective of the work, I develop an CNN network. CNN classification uses training samples formed by image-label pairs to determine classification rules that assign labels to images. In what follows, I denote by X and Y the sets of possible images and labels, respectively; This network takes features produced by the proposed representation as input and outputs a vector which includes weighting over labels.

Local Feature Representation: I treat the patch extracted in specific τ interval as local feature representation for the traveltime i. Although the goal is to predict the velocity for that traveltime, the local feature is not adequate. Given said that, I combine the whole semblance image to have coverage on overall trend of semblance as the global feature with the local feature shown in extracted patch as illustrated in Fig. 6.4. This has been shown to be effective for seismic image local feature representation [114]. By doing so, each CNN input data first captures the local information by using the patch image that explicitly looks into the $50 * W$ spatial window around traveltime t_i; 50 is the height of patch if extracted by Constant Patch Size method as illustrated in Fig. 6.4. Second, to obtain the overall trend of the image that this patch is part of, CNN input data keeps the image of the whole semblance as well.

The model proposed in this study is not restricted by semblance panel input image size. For the sake of consistency, I scaled the spatial range of the pair images used as CNN input data to a fixed scale (e.g. 50×50) before applying the filters of the CNN.

Training Data Label Generation: In order to generate the label for training data, I consider a binary vector with length $K = 180$ as the label of the input. Thus for each CNN input image X_{j,t_i}, I have a vector Y_{j,t_i} as the label output. For instance, in section 6.1.2, the velocity class calculated for the patch extracted in Fig. 6.3 is 84. Thus the label vector

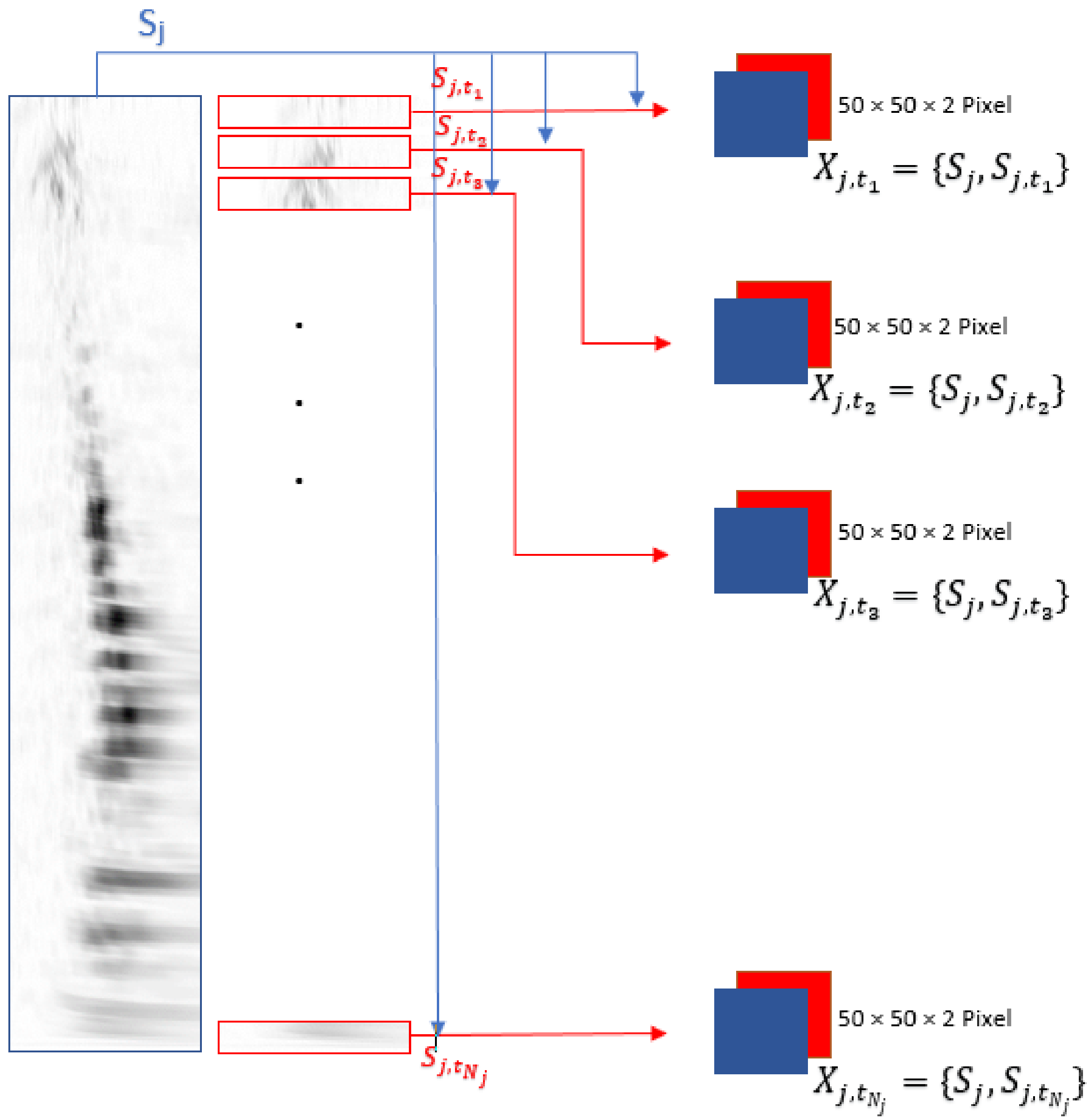


Figure 6.4: Process of defining input X data for CNN. Blue and red arrows indicate resizing of whole image and patch to the size 50 * 50 pixel respectively. Each input image X_{j,t_i} is produced by pairing two resized images: 1) S_j,the image of whole semblance and j is the semblance number 2) S_{j,t_i}, the i^{th} patch extracted at time t_i from semblance S_j, where $i \in \{1, 2, ..., N_j\}$ and N_j is the total Number of Patches in the semblance panel input image S_j.

for this input is $[0, 0, ..., 0, 1, 0, ..., 0]$ in which all the elements are 0 except index number 84 which is 1.

Data Label Prediction

In previous two sections I match all input data with the corresponding label to construct the data set for training the CNN model. Afterwards, in CNN prediction phase, our model demonstrate the output for input X_{j,t_i} as vector $\hat{Y}[k]_{j,t_i}, k = 1...K$; where k is the class number which is a number between 1 and 180. Given that applying Softmax loss in CNN has achieved tremendous successes in image classification tasks [60] [70] [177], I adopt a Softmax classifier at the last layer of the CNN. This outputs a weighted vector of probabilities of the classes, denoted as $\tilde{Y}[k]_{j,t_i}, k = 1...K$ calculated as below:

$$\tilde{Y}[k]_{j,t_i} = \frac{exp(\hat{Y}[k]_{j,t_i})}{\sum_{i=1}^{K} exp(\hat{Y}[k]_{j,t_i})} \tag{6.4}$$

where each element in vector $\tilde{Y}[k]_{j,t_i}$ is a value between 0 and 1, representing a predicted probability for velocity class k. Conventionally the index of maximum probability in vector $\tilde{Y}[k]_{j,t_i}$ is considered as predicted velocity. However the advantage of our hybrid (Geophysical model with CNN) formulation is that I have a ranking system to consider the top P (i.e. 5 is the optimum value achieved by try-and-error) probabilities predicted by model. Then I use geophysical model to have quality assurance on top of these classes and select the best one. In what follows, the top P values are Denoted by $\tilde{Y}[max_1]_{j,t_i}...\tilde{Y}[max_P]_{j,t_i}$, where $max_p, p \in 1, 2, ..., P$ is the index of the class with p^{th} best probability.

CNN Objective Function: The loss function is the objective function which technically is an indicator of the directions of weight changes. In our setting, the velocity prediction for CNN input image X_{j,t_i} is predicted by vector $\tilde{Y}_{j,t_i}$. Thus, the objective for learning our velocity prediction is to minimize a loss function that measures the similarity between $\tilde{Y}_{j,t_i}$ and Y_{j,t_i}. This study use the categorical cross-entropy (CE) [129] function as the loss function. The reason is that Softmax with CE loss is widely investigated and adopted by many classification approaches due to its ability for probabilistic interpretation, simplicity and good performance.

6.1.4 Hybrid Geophysical Approach with CNN

Having spatiotemporal semblance panel images, with some geophysical infrastructure basis within it, it is necessary to consider those basis in any solution proposed for automatic velocity picking. In this section, I explain how we incorporate these knowledge in our proposed method.

Interval Velocity Aggregation

The hybrid geophysical step is designed to have quality control on initial velocities predicted by CNN. To perform quality control and assess the validity of the velocities predicted by CNN, I devised Dix equation [33] as explained in 2.11.

Using the Dix equation, the interval velocities are approximated. The interval velocity is the velocity between upper reflector field and the lower one. In this study, the extracted patches are considered as the reflector field. Since Dix velocities are approximation to the interval velocities, I expect them to fall within certain range of rock velocities (i.e. $[1500, 6000]$) explained in Section 6.1.2. For instance, in the highlighted example (the pink one) in Fig. 6.3, the Dix velocity is calculated as below:

$$\nu = \sqrt{\frac{(3602.21)^2(1249.8) - (3487.62)^2(1013.3)}{1249.8 - 1013.3}} = 4056.69$$

As the calculated interval velocity is in range $[1500, 6000]$, the stacking velocity 3602.21 is an acceptable velocity for the corresponding patch. The above mentioned example is calculated using the velocities selected by expert, however the process is the same on the CNN predicted velocity.

Quality Control using Interval Velocities

To incorporate the above mentioned Dix velocity in quality control step, the following steps are performed:

1) The CNN testing phase outputs a class number for each patch. class number is associated with the highest probability in the output vector (See Section 6.1.3). In order to convert the class number to a numeric velocity value, the number in the middle of the correspondence class is considered. For example, if the predicted class is 1, its velocity value is considered to be $\frac{1500+1525}{2} = 1512.5$. In this step, for each semblance, I create an output file same as what we have in Fig. 6.2. The first column is filled with the traveltime at the middle of each patch, the second column will be the predicted velocity value which is calculated from the predicted class number. In particular, it should be noted that now the class with the highest probability, i.e. $\tilde{Y}[max_1]_{j,t_i}$ is the accepted output (Section 6.1.3).

2) For each semblance, now we have the output file extracted in the previous step. This auxiliary step takes this output file for each semblance and computes the interval velocity using Dix equation for each traveltime reported in the file one by one. In other words, for each pair of the (traveltime, predicted velocity) the interval velocity is calculated and if is out of the authorized range, $\tilde{Y}[max_1]_{j,t_i}$ is replaced by $\tilde{Y}[max_2]_{j,t_i}$. The interval velocity again is calculated and this process with be repeated to Maximum 5 times (i.e. using $\tilde{Y}[max_5]_{j,t_i}$). Please refer to 6.1.3 for details about notations.

One can observe that this hybrid auxiliary step has ability to remove the predicted velocities leading to unacceptable interval velocities, and try to replace them with a more accurate prediction.

6.2 Results

6.2.1 Data Sets

In this study, I use data set of 9 fields (seismic fields) mostly located in Canada. Table 6.1 shows the characteristics of the 9 fields data sets. It consists of 980 semblances collected

Table 6.1: Description of the data sets selected for our experiments.DS:Data Set

Field Name (Abbreviation)	# semblances	sampling frequency (ms)
DS1	148	1
DS2	251	1
DS3	125	1
DS4	170	1
DS5	67	2
DS6	16	1
DS7	11	1
DS8	152	2
DS9	40	1
Total	**980**	

from various locations which their sampling frequency is either 1 or 2. The true velocity values selected by expert for each semblance are given in a text file format as shown in Fig. 6.2.

6.2.2 Evaluation Metrics

I follow the evaluation protocol of Ferreira *et al.* [44] named Velocity Picking Distance to Reference (VPDR), where performance is measured by the average velocity displacement between the true velocity and predicted velocity. To apply VPDR on the results I first convert the classification outputs to continues velocity values as explained in Section 6.1.4. I also perform Mean-Square Error (MSE) and Mean Absolute Error (MAE) for evaluating the classification results. In our experiments, I did not use accuracy because our problem is an ordinal multi-class (180 classes) problem. It means if a true velocity for an input data in class 2, if one approach predicts class 1 should be more acceptable than predicting class 100; whereas, accuracy measure considers same penalty for both cases. Furthermore, loss function explained in section 6.1.3 is also used for evaluation in this study.

Table 6.2: Evaluation results for three CNN models.

CNN Model	Train Loss	Validation Performance			Time (s)
		Loss	MSE	MAE	
AlexNet	0.05	4.2	0.0053	0.00	900
VGG16	0.07	4.6	0.0054	0.01	2800
LeNet-5	3.2	3.6	0.0051	0.01	400

6.2.3 Comparison of the CNN Architectures

To find the CNN architecture suitalbe for first part of the proposed approach (i.e. Section 6.1.3), 3 different deep learning models AlexNet [80], VGG16 [142] and LeNet-5 [83] are quantitatively compared. Table 6.2 summarize the average loss on train set, Loss, MSE and on validation set and the total training time for three CNN models used during this experiment. All of the results in this section are from the same validation dataset and training dataset with no overlap after 100 training epochs. From the results, we see that the LeNet model performs very badly on the training dataset in comparison with the other two methods. This is because the layers parameters in the LeNet are very few (around 4 million) and are not enough to learn the tasks. Even showing a good results on validation data means that knowledge transfer does happen from our base module that has been trained. So it can not be a suitable option for this specific task training purpose. The results also indicate that AlexNet and VGG16 have almost the same performance on both validation and training dataset, however, we can clearly observe that the duration time for training VGG16 model is around 3 times more than AlexNet. Thus, I select AleXNet architecture to be used in firs part of our approach. Hyper-parameters are optimized by balancing the time and performance of the results for different "learning rate" and "batch size" values. The rates that "learning rate" and "batch size" are searched over are $(0.0001, 0.001, 0.01)$ and $(64, 256, 512, 1024)$ respectively. We observed that the train model with learning rate 0.0001 and batch size 256 has the best balance between time and performance.

6.2.4 Comparison to State-of-the-art

The objective of this section is to study the results of the proposed hybrid CNN approach combined with geophysical model in more details. To do this, I compare our performance with state-of-the-art method [114]. The goal of our experiments, in particular, is to computationally demonstrate the performance superiority of our proposed algorithm in two respects: (1) ability to find the optimal velocity and (2) the time required to do so.

All experiments including the training and testing are performed on a Google Colab Pro Platform 25 GB GPU. Google Colab is a GPU based framework offered by Google for which can speed up training time of the deep learning. Python 3.6.9 is used for implementation of all the proposed method. Two different field datasets are specifically selected for testing purpose without being used in training or validation. For the sake of fairness in comparison, Park and Sacchi's state-of-the-art method [114] is also re-implemented and re-trained using exactly the same platform and datasets respectively. I only need to adjust the number of classes from 40 in their work to 180 in our work which does not affect the effectiveness of their algorithm.

A summary of computational results is presented in Table 6.3. Performance of the proposed method and Park and Sacchi's method [114] is evaluated in terms of (1) VPDR: to evaluate the regression prediction ability of the methods. Such that, the class numbers are converted to continuous velocity values and then using VPDR the performance of methods is evaluated. (2) MSE and MAE in which I quantify the classification performance for our ordinal multi-class problem (3) the training and testing run time. The reported run time does not include patch extraction time for neither of the methods.

For all performance metrics, as can be seen in Table 6.3, our method consistently outperforms the state-of-the-art method [114] on both field datasets with the margin of at least 23% for all the metrics in all the datasets. Park and Sacchi's state-of-the-art method [114] is purely based on CNN and trasfer learning without leveraging the domain knowledge. Given said that, the experiments presented in this study illustrate that aggregating geophysical

Table 6.3: Results on our method and state-of-the-art method (Park&Sacchi) using two separate field data for test.

Dataset	Method	VPDR	MSE	MAE	Time (s)
D1	Ours	280.89	233.2	11.2	930
	Park&Sacchi	457.15	835.82	18.26	2815
D2	Ours	400.76	445.68	16.01	923
	Park&Sacchi	548.125	977.55	21.9	2810

model with CNN has a dramatic impact on velocity picking. For example, I can improve the state-of-the-art method [114] by 39% and 27% in VPDR for first and second field dataset. Regarding the run time, our method utilize only 33% of the previous method.

Moreover, because in our study the number of classes are 180 with range of 25 in each class (the state-of-the-art method has only 40 classes with range of 50), the metrics reported in this study are more accurate than those reported in the state-of-the-art method. For example, the MSE value reported in that study (550) is only between 40 classes. However, our study has 180 classes to be considered which makes the MSE values between the prediction and the true label velocity fields to have potentially higher values. To the best of our knowledge, extensive experiments on multiple datasets show that our model exhibits state-of-the-art performance on automatic velocity picking in seismic data.

6.3 Summary

Velocity analysis process takes a significant amount of time and mostly performed manually which ultimately impacting the cost and duration of a typical seismic interpretation procedure. This chapter presents a novel data-driven hybrid strategy for combining geophysical models with Convolutional Neural Network (CNN) with two key contributions: (i) I use a CNN to generate robust velocity picks on spatiotemporally varying image data and (ii) I design a geophysical-based model to check gradually increasing features of the picked velocities to improve their prediction. Extensive experiments performed on four field data sets show that the proposed methodology can not only produce near human-level velocity pickings but also outperform the current state-of-the-art method in most scenarios.

Chapter 7

Conclusions and Future Work

This chapter summarizes the results of the research studies presented in this thesis and provides directions for future research. Section 7.1 summarizes the research findings and Section 7.2 provides future research directions.

7.1 Thesis Summary

This thesis presents four studies related to leveraging feature exploitation to automate practical machine learning with text, Image and Tabular Data. These studies had four distinct but inter-related objectives that can be summarized as follows

- With the goal of automating the practical machine learning, translating raw data into accurate predictions\estimation with minimal software requirement. In particular, replacing human time\effort and manual experimentation with models that can be improved by simply setting up in appropriate environment to learn.
- Design, implement and evaluate a feature extraction pipeline that enables automating the practical machine learning based on the context of the data. While focusing on outcome of interest, identify possible hidden features in the datasets that can help in removing bottlenecks on the correctness of test results.
- With the goal of reducing the number of features needed in a machine learning train-

ing study, identify an intelligent technique to select, extract or even propose related features.

- provide broader perspective that also encompasses tabular, text and image data and other equally important aspects of the ML pipeline like: data processing, model ensembling practical automation of deep learning technique. Then perform an in-depth analysis of the proposed method with real-world data sets.

To achieve these objectives, I designed and implemented applied machine learning tools and conducted multiple experimental evaluation studies all using real world data sets. The outcome was several findings related to the effect of data related feature exploitation on the performance of machine learning applications. The contribution of this thesis can be summarized as follows:

Regarding the first study, extracting important features is an expensive and tedious task for big data mining projects, which is essential for developing fast and high accuracy prediction models. In this study I develop a CB-RF model to identify the importance of features from highly dimensional dataset. I then proposed a model to provide additional service to patients by predicting the desired output for the EPIC-26 domains. Finally, I used three different feature extraction models to find the important features for the desired output, separately. I demonstrated the application of this approach using diagnostic, clinical, and PRO data extracted from patients diagnosed with prostate cancer. I use CB-RF to identify the most important features for predicting patients' HRQOL. Based on the empirical study results, I showed that a RF combined with patient clustering in a neural network model can accurately predict all variables of HRQOL with impressive improvement on the predicted results performance in comparison with three other models. In addition, the proposed model was completely on H2O, which has the advantage of both run time and code complexity for both clustering and feature extraction.

In the second study, I construct a framework to extract important influential variables for prostate cancer EPIC-26 prediction, learn and interpret prostate cancer patient data in a novel way. I demonstrated how to fit the patient data with proposed CB-DNN algorithm and make accurate predictions of patients' HRQoL outcomes. Furthermore, our framework uses an Bayesian network based approach to provide explanations for CB-DNN, which is a black box algorithm, result to represents the probability that the patient will feel a particular outcome in 1-year follow-up based on the baseline visit variables. To the best of our knowledge, this work is the first one to combine machine learning model and explanation method for analyzing prostate cancer data regarding their HRQoL measure and considering all of the EPIC-26 domains outcomes. The CB-DNN allows stakeholders to extract important variables out of the big data set that might affect the outcome of interest in longitudinal data. The implementation of our methods help identify the key drivers of patient 1-year follow-up severity, in baseline visit, which the patient is newly diagnosed with prostate cancer. Besides that, the proposed method is also the first that is predicting the continuous value for 1-year follow-up EPIC-26 outcomes, not only general classifications. Therefore, the framework generally can provide guidance for medical practitioners to navigate through the baseline variables to complex future outcomes. Despite the aforementioned strengths of this study and the successful implementation with real world data, it has its own limitations which can be improved as new directions of future work. To start with, the proposed method is validated in a real world big data set, but it can be externally validated as well to improve the trust for the end users. Secondly, the choice of the proposed hybrid model for the Bayesian network needs theoretical evidences, and needs more investigation. Thirdly, I want to use this framework for improvement in the feedback for their HRQoL outcome even considering their treatment choice. With considering the treatment related variables, I can advice the future patients to which treatment they should select regarding their HRQoL preferences, this part is still under investigation. Overall, despite these limitations, this work can contribute in ongoing effort to adopt good predictability and interpretability practices when considering HRQoL outcomes in patients with prostate cancer.

The focus of the third study is developing feature extraction in using text data in the context of manual testing. Manual testing is an expensive and tedious software testing activity, which is essential for getting users perspective on the released features. In this study, I proposed an NLP-based approach (POS) for extracting test case's coverage on the GUI by estimating the GUI elements names and states (as nouns) from the textual data of test cases. I then proposed an IR-based measure (TF-IDF) to be assigned to each test case. Finally, I use three different prediction models to predict failure of test cases using this measure and three different variations of history-based measures, separately. Our results based on an empirical study of 41 releases of Mozilla Firefox's manual test suites showed that a simple history-based feature (only previous release data) combined with POS/TF-IDF data in a linear regression model can accurately predict test cases' failure in new releases. In addition, the NLP-based approach can provide up to 24% improvement on accuracy of prediction over the baseline approach (only using history and not the mined GUI coverage data). To the best of our knowledge, this work is the first use of NLP on manual test case scripts for test failure prediction and has shown promising results, which I plan to replicate on different systems and expand on different NLP-based features to more accurately extract the GUI-related information.

With the focus on exploiting feature extraction techniques in image date, in the last study, I propose using a hybrid geophysical model with CNN to predict the velocities which is time-consuming step in seismic imaging. I utilize AlexNet in first phase the approach for image classification. The geophysical part of the model leverages interval velocity in selected time horizons to improve the performance of CNN output for both image classification and regression. I define the large number of classes with relatively small range over the continuous velocity values which potentially yields to adding regression capability to the model as the velocities can be estimated within each class. The proposed method significantly outperforms the existing state-of-the-art model with requiring less training and testing time. Extensive experiments on the largest number of datasets existing in literature show that our model exhibits state-of-the-art performance on automatic velocity picking in seismic data. Further, I show that our model has near-expert performance if all the settings are set up properly.

7.2 Future Direction

The studies presented in this thesis can be expanded in multiple directions including the following perspectives.

To the best of our knowledge, the work done in the first two studies, is the first use of H2O on feature extraction and predicted models in medical domain and has shown promising results, which I am planning to replicate on different data sets and expand on different cluster-based feature extracting models to more accurately extract features from big datasets. In particular, the limitation of the second study is that I adhere to the assumption behind the Naive Bayes rule which is there is no interaction between the input variables given the outcome. I assume that to be able to interpret the prediction results in the variable domains. This limitation is the special focus of the future work as I believe that the nature of the interaction between the input variables toward the outcome is not linear and also may not be independent given the output. To address this limitation one may adapt several orthonormal transformations of the variables and study the effect of these transformations on the predictive and interpretability power of the model.

The experiments in third study show that certain properties of a manual test case which is written in natural language, such as nouns, are correlated with higher probability of failure rate in testing. This work can motivate other researchers to explore these tasks and develop better models that outperform the ones used in this work. More importantly, we hope that other researchers will identify novel more deep network opportunities in NLP like BERT, which we have not explored to analyze the peer reviews in this real world dataset.

Regarding the image analysis work to predict the velocity picking, one main direction for future study is to change the CNN loss function such that comply with geophysical part of our method. This way, they could have a trained CNN model that is completely capable of predicting the velocities by minimizing the geophysical-based loss function. Moreover, in the context of combining geophysical knowledge with machine learning, automating the first velocity pick has the potential to improve the velocity picking in the whole semblance.

The last but not least, is the idea of converting the tabular data investigated in first two studies and convert it to image data which is used in the last study. In tabular data the information from its features or elements is spread over arbitrarily, making it challenging to extract relevant details for identification using the image-related deep-learning techniques. However, an arrangement of similar features into clusters makes these differences more accessible and allows for robust extraction of hidden patterns between features than dealing with elements individually. The future direction can converts tabular data into a well-organized image-form. Thereby, the power of convolution neural network (CNN), including GPU utilization which is used in the last study of this thesis, can be utilize for tabular data used in first two studies.

Bibliography

[1] Institute for health technology transformation transforming health care through big data: Strategies for leveraging big data in the health care industry, New York (2013).

[2] S. L. Abebe and P. Tonella. Natural language parsing of program element names for concept extraction. In *2010 IEEE 18th International Conference on Program Comprehension*, pages 156–159, June 2010.

[3] Webpage: APCARI. Home - apcari.

[4] Philip Aspden, Janet M Corrigan, Julie Wolcott, Shari M Erickson, et al. Key capabilities of an electronic health record system: Letter report. In *Patient Safety: Achieving a New Standard for Care*. National Academies Press (US), 2004.

[5] Adarsh Barik and Jean Honorio. Learning bayesian networks with low rank conditional probability tables. In *Advances in Neural Information Processing Systems*, pages 8962–8971, 2019.

[6] Michael J Barry. The prostate cancer treatment bazaar: comment on "physician visits prior to treatment for clinically localized prostate cancer". *Archives of internal medicine*, 170(5):450–452, 2010.

[7] Michael J Barry. Helping patients make better personal health decisions: the promise of patient-centered outcomes research. *JAMA*, 306(11):1258–1259, 2011.

[8] Ethan Basch, Xiaoyu Jia, Glenn Heller, Allison Barz, Laura Sit, Michael Fruscione, Mark Appawu, Alexia Iasonos, Thomas Atkinson, Shari Goldfarb, et al. Adverse symptom event reporting by patients vs clinicians: relationships with clinical outcomes. *JNCI: Journal of the National Cancer Institute*, 101(23):1624–1632, 2009.

[9] Rafael Alves Batista, Andrej Dundovic, Martin Erdmann, Karl-Heinz Kampert, Daniel Kuempel, Gero Müller, Guenter Sigl, Arjen van Vliet, David Walz, and Tobias Winchen. Crpropa 3—a public astrophysical simulation framework for propagating extraterrestrial ultra-high energy particles. *Journal of Cosmology and Astroparticle Physics*, 2016(05):038, 2016.

[10] Richard E Bellman. *Adaptive control processes: a guided tour*, volume 2045. Princeton university press, 2015.

[11] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *International conference on database theory*, pages 217–235. Springer, 1999.

[12] Umair bin Waheed, Saleh Al-Zahrani, and Sherif M Hanafy. Machine learning algorithms for automatic velocity picking: K-means vs. dbscan. In *SEG Technical Program Expanded Abstracts 2019*, pages 5110–5114. Society of Exploration Geophysicists, 2019.

[13] Reetam Biswas, Anthony Vassiliou, Rodney Stromberg, and Mrinal K Sen. Estimating normal moveout velocity using the recurrent neural network. *Interpretation*, 7(4):T819–T827, 2019.

[14] Dhruba Borthakur. The hadoop distributed file system: Architecture and design. *Hadoop Project Website*, 11(2007):21, 2007.

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

[16] Leo Breiman, Jerome Friedman, and Charles J Stone. Ra olshen classification and regression trees chapman and hall, 1984.

[17] Randal Bryant, Randy H Katz, and Edward D Lazowska. Big-data computing: creating revolutionary breakthroughs in commerce, science and society, 2008.

[18] Carlos Calderón-Mac ı´ as, Mrinal K Sen, and Paul L Stoffa. Automatic nmo correction and velocity estimation by a feedforward neural network. *Geophysics*, 63(5):1696–1707, 1998.

[19] Webpage: Canadian Cancer Society. Prostate cancer statistics - canadian cancer society.

[20] Mauro Carrara, Eleonora Massari, Alessandro Cicchetti, Tommaso Giandini, Barbara Avuzzi, Federica Palorini, Claudio Stucchi, Giovanni Fellin, Pietro Gabriele, Vittorio Vavassori, et al. Development of a ready-to-use graphical tool based on artificial neural network classification: application for the prediction of late fecal incontinence after prostate cancer radiation therapy. *International Journal of Radiation Oncology* Biology* Physics*, 102(5):1533–1542, 2018.

[21] Jonathan Cheung-Wai Chan and Desiré Paelinckx. Evaluation of random forest and adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery. *Remote Sensing of Environment*, 112(6):2999–3011, 2008.

[22] Elizabeth S Chen and Indra Neil Sarkar. Mining the electronic health record for disease knowledge. In *Biomedical Literature Mining*, pages 269–286. Springer, 2014.

[23] Yuqing Chen. Automatic semblance picking by a bottom-up clustering method. In *SEG 2018 Workshop: SEG Maximizing Asset Value Through Artificial Intelligence and Machine Learning, Beijing, China, 17-19 September 2018*, pages 44–48. Society of Exploration Geophysicists and the Chinese Geophysical Society, 2018.

[24] Gregory F Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.

[25] Michael Cox and David Ellsworth. Application-controlled demand paging for out-of-core visualization. In *Proceedings. Visualization'97 (Cat. No. 97CB36155)*, pages 235–244. IEEE, 1997.

[26] Dipanjan Das and Slav Petrov. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 600–609, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[27] Jean-Bernard Delobel, Khemara Gnep, Juan David Ospina, Véronique Beckendorf, Ciprian Chira, Jian Zhu, Alberto Bossi, Taha Messai, Oscar Acosta, Joël Castelli, et al. Nomogram to predict rectal toxicity following prostate cancer radiotherapy. *PloS one*, 12(6):e0179845, 2017.

[28] Giovanni Denaro and Mauro Pezzè. An empirical evaluation of fault-proneness models. In *Proceedings of the 24th International Conference on Software Engineering*, ICSE '02, pages 241–251, New York, NY, USA, 2002. ACM.

[29] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[30] Darren Desantis, Richard J Baverstock, Andrea Civitarese, R Trafford Crump, and Kevin V Carlson. A clinical perspective on electronically collecting patient-reported outcomes at the point-of-care for overactive bladder. *Canadian Urological Association Journal*, 10(11-12):E359, 2016.

[31] Robert T Dess, Holly E Hartman, Nima Aghdam, William C Jackson, Payal D Soni, Ahmed E Abugharib, Simeng Suy, Neil B Desai, Zachary S Zumsteg, Rohit Mehra, et al. Erectile function after stereotactic body radiotherapy for localized prostate cancer. *BJU international*, 121(1):61–68, 2018.

[32] Ivo D Dinov. Methodological challenges and analytic opportunities for modeling and interpreting big healthcare data. *Gigascience*, 5(1):12, 2016.

[33] C Hewitt Dix. Seismic velocities from surface measurements. *Geophysics*, 20(1):68–86, 1955.

[34] David L Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(32):375, 2000.

[35] James A Eastham, Peter T Scardino, and Michael W Kattan. Predicting an optimal outcome after radical prostatectomy: the trifecta nomogram. *The Journal of urology*, 179(6):2207–2211, 2008.

[36] Anna Ehrenberg and Christina Birgersson. Nursing documentation of leg ulcers: adherence to clinical guidelines in a swedish primary health care district. *Scandinavian journal of caring sciences*, 17(3):278–284, 2003.

[37] Sebastian Elbaum, Gregg Rothermel, and John Penix. Techniques for improving regression testing in continuous integration development environments. In *Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FSE 2014, pages 235–245, New York, NY, USA, 2014. ACM.

[38] Carmelo Evoli, Daniele Gaggero, Andrea Vittino, Giuseppe Di Bernardo, Mattia Di Mauro, Arianna Ligorini, Piero Ullio, and Dario Grasso. Cosmic-ray propagation with dragon2: I. numerical solver and astrophysical ingredients. *Journal of Cosmology and Astroparticle Physics*, 2017(02):015, 2017.

[39] Ruogu Fang, Samira Pouyanfar, Yimin Yang, Shu-Ching Chen, and SS Iyengar. Computational health informatics in the big data age: a survey. *ACM Computing Surveys (CSUR)*, 49(1):12, 2016.

[40] KM Faraoun and Aoued Boukelif. Neural networks learning improvement using the k-means clustering algorithm to detect network intrusions. *INFOCOMP*, 5(3):28–36, 2006.

[41] Bonnie Feldman, Ellen M Martin, and Tobi Skotnes. Big data in healthcare hype and hope. *Dr. Bonnie*, 360:122–125, 2012.

[42] Yang Feng, James A. Jones, Zhenyu Chen, and Chunrong Fang. Multi-objective test report prioritization using image understanding. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, ASE 2016, pages 202–213, New York, NY, USA, 2016. ACM.

[43] Norman E Fenton and Martin Neil. A critique of software defect prediction models. volume 25, pages 675–689. IEEE, 1999.

[44] Rodrigo S Ferreira, Dário AB Oliveira, Daniil G Semin, and Semen Zaytsev. Automatic velocity analysis using a hybrid regression approach with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[45] Barry C Fish and Tony Kusuma. A neural network approach to automate velocity picking. In *SEG Technical Program Expanded Abstracts 1994*, pages 185–188. Society of Exploration Geophysicists, 1994.

[46] Nir Friedman et al. Learning belief networks in the presence of missing values and hidden variables. In *ICML*, volume 97, pages 125–133, 1997.

[47] G David Garson. Interpreting neural-network connection weights. *AI expert*, 6(4):46–51, 1991.

[48] Gartner. It glossary: Big data, 2014.

[49] Tamás D Gedeon. Data mining of inputs: analysing magnitude and functional measures. *International Journal of Neural Systems*, 8(02):209–218, 1997.

[50] Aindrila Ghosh, Mona Nashaat, James Miller, Shaikh Quader, and Chad Marston. A comprehensive review of tools for exploratory analysis of tabular industrial datasets. *Visual Informatics*, 2(4):235–253, 2018.

[51] Richard E Gliklich, Nancy A Dreyer, and Michelle B Leavy. Assessing quality. In *Registries for Evaluating Patient Outcomes: A User's Guide [Internet]. 3rd edition.* Agency for Healthcare Research and Quality (US), 2014.

[52] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[53] Fred Glover. Tabu search and adaptive memory programming—advances, applications and challenges. In *Interfaces in computer science and operations research*, pages 1–75. Springer, 1997.

[54] Ian J Goodfellow, David Warde-Farley, and Mehdi Mirza Aaron Courville. Bengio, yoshua. maxout networks. In *International Conference on Machine Learning (ICML)*, pages 1319–1327, 2013.

[55] Nikolaos Grivas, Rosanne van der Roest, Daan Schouten, Francesca Cavicchioli, Corine Tillier, Axel Bex, Ivo Schoots, Walter Artibani, Stijn Heijmink, and Henk Van Der Poel. Quantitative assessment of fascia preservation improves the prediction of membranous urethral length and inner levator distance on continence outcome after robot-assisted radical prostatectomy. *Neurourology and urodynamics*, 37(1):417–425, 2018.

[56] Philip J. Guo, Thomas Zimmermann, Nachiappan Nagappan, and Brendan Murphy. "not my bug!" and other reasons for software bug report reassignments. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, CSCW '11, pages 395–404, New York, NY, USA, 2011. ACM.

[57] Webpage: H2O.ai. Home - h2o.ai.

[58] M. Harman. Making the case for morto: Multi objective regression test optimization. In *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops*, pages 111–114, March 2011.

[59] Paul A Harris, Robert Taylor, Robert Thielke, Jonathon Payne, Nathaniel Gonzalez, and Jose G Conde. Research electronic data capture (redcap)—a metadata-driven methodology and workflow process for providing translational research informatics support. *Journal of biomedical informatics*, 42(2):377–381, 2009.

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[61] Yang-Bo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(Nov):2523–2547, 2008.

[62] H. Hemmati, Z. Fang, and M. V. Mantyla. Prioritizing manual test cases in traditional and rapid release environments. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, pages 1–10, apr 2015.

[63] Hadi Hemmati, Zhihan Fang, Mika V Mäntylä, and Bram Adams. Prioritizing manual test cases in rapid release environments. *Software Testing, Verification and Reliability*, 27(6):e1609, 2017.

[64] Hadi Hemmati, Zhihan Fang, Mika V. Mäntylä, and Bram Adams. Prioritizing manual test cases in rapid release environments. *Software Testing, Verification and Reliability*, pages n/a–n/a, 2016. stvr.1609.

[65] J Henry, Yuriy Pylypchuk, Talisha Searcy, and Vaishali Patel. Adoption of electronic health record systems among us non-federal acute care hospitals: 2008-2015. *ONC Data Brief*, 35:1–9, 2016.

[66] Matthew Herland, Taghi M Khoshgoftaar, and Randall Wald. Survey of clinical data mining applications on big data in health informatics. In *2013 12th International Conference on Machine Learning and Applications*, volume 2, pages 465–472. IEEE, 2013.

[67] Matthew Herland, Taghi M Khoshgoftaar, and Randall Wald. A review of data mining using big data in health informatics. *Journal of Big data*, 1(1):2, 2014.

[68] Sender Herschorn, Jerzy Gajewski, Jane Schulz, and Jacques Corcos. A population-based study of urinary symptoms and incontinence: the canadian urinary bladder survey. *BJU international*, 101(1):52–58, 2008.

[69] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[70] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[71] Kou-Yuan Huang, Kai-Ju Chen, and Jia-Rong Yang. Seismic velocity picking by simulated annealing. In *2012 IEEE International Geoscience and Remote Sensing Symposium*, pages 6079–6082. IEEE, 2012.

[72] Kou-Yuan Huang, Kai-Ju Chen, and Jia-Rong Yang. Seismic velocity picking by genetic algorithm. In *2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS*, pages 1548–1551. IEEE, 2013.

[73] Amjad Hudaib, Bassam Hammo, and Yara Alkhader. Towards software requirements extraction using natural language approach. In *Proceedings of the 6th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems*, SEPADS'07, pages 155–160, Stevens Point, Wisconsin, USA, 2007. World Scientific and Engineering Academy and Society (WSEAS).

[74] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[75] Michael W Kattan, Changhong Yu, Andrew J Stephenson, Oliver Sartor, and Bertrand Tombal. Clinicians versus nomogram: predicting future technetium-99m bone scan positivity in patients with rising prostate-specific antigen after radical prostatectomy for prostate cancer. *Urology*, 81(5):956–961, 2013.

[76] Lawrence HC Kim, Amit Patel, Netty Kinsella, Mansour TA Sharabiani, Derfel Ap Dafydd, and Declan Cahill. Association between preoperative magnetic resonance imaging–based urethral parameters and continence recovery following robot-assisted radical prostatectomy. *European urology focus*, 2019.

[77] Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. Experimental design for learning causal graphs with latent variables. In *Advances in Neural Information Processing Systems*, pages 7018–7028, 2017.

[78] K Koga. Software reliability design method in hitachi. In *Proc. Third European Conf. Software Quality*, 1992.

[79] IJ Korfage, Marie-Louise Essink-Bot, ACJW Janssens, FH Schröder, and HJ De Koning. Anxiety and depression after prostate cancer diagnosis and treatment: 5-year follow-up. *British journal of cancer*, 94(8):1093, 2006.

[80] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[81] Mark Landry and B Angela. Machine learning with r and h2o. *Mountain View, CA*, 2018.

[82] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[83] Yann LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20(5):14, 2015.

[84] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*, 2018.

[85] A Liaw and M Wiener. Classification and regression by randomforest. r news, 2 (3): 18–22. r package version 4.6. 10, 2002.

[86] Dawei Liu, Wei Wang, Wenchao Chen, Xiaokai Wang, Yanhui Zhou, and Zhensheng Shi. Random noise suppression in seismic data: What can deep learning do? In *SEG Technical Program Expanded Abstracts 2018*, pages 2016–2020. Society of Exploration Geophysicists, 2018.

[87] Ning Liu, Soundar Kumara, and Eric Reich. Explainable data-driven modeling of patient satisfaction survey data. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 3869–3876. IEEE, 2017.

[88] Hilary A Llewellyn-Thomas and R Trafford Crump. Decision support for patients: values clarification and preference elicitation. *Medical care research and review*, 70(1_suppl):50S–79S, 2013.

[89] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[90] Sabina Luszczak, Christopher Kumar, Vignesh Krishna Sathyadevan, Benjamin S Simpson, Kathy A Gately, Hayley C Whitaker, and Susan Heavey. Pim kinase inhibition: co-targeted therapeutic approaches in prostate cancer. *Signal Transduction and Targeted Therapy*, 5(1):1–10, 2020.

[91] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[92] A. Marcus. Semantic driven program analysis. In *20th IEEE International Conference on Software Maintenance, 2004. Proceedings.*, pages 469–473, Sept 2004.

[93] Robert L. McCornack. Extended tables of the wilcoxon matched pair signed rank statistic. *J. Am. Stat. Assoc.*, 60(311):864–871, 1965.

[94] Dalton McLean*, Katelin Cunningham, and William Ricke. Mp68-20 establishing a negative feedback loop between fibroblast growth factor 5 and androgen receptor in prostate cancer. *The Journal of Urology*, 201(Supplement 4):e983–e983, 2019.

[95] David McLean. Kevin gurney, an introduction to neural networks, university college london (ucl) press, 1997. isbn 1-85728-673-1 hb. £14.95. xi+234 pages. *Nat. Lang. Eng.*, 6(2):203–204, June 2000.

[96] M Dror Michaelson, Shane E Cotter, Patricio C Gargollo, Anthony L Zietman, Douglas M Dahl, and Matthew R Smith. Management of complications of prostate cancer treatment. *CA: a cancer journal for clinicians*, 58(4):196–213, 2008.

[97] Uwe HG Michl, Martin G Friedrich, Markus Graefen, Alexander Haese, Hans Heinzer, and Hartwig Huland. Prediction of postoperative sexual function after nerve sparing radical retropubic prostatectomy. *The Journal of urology*, 176(1):227–231, 2006.

[98] Gustav Mikkelsen and Jan Aasly. Concordance of information in parallel electronic and paper based patient records. *International journal of medical informatics*, 63(3):123–131, 2001.

[99] D. Mondal, H. Hemmati, and S. Durocher. Exploring test suite diversification and code coverage in multi-objective test case selection. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, pages 1–10, April 2015.

[100] Eugenia Mylona, Oscar Acosta, Thibaut Lizee, Caroline Lafond, Gilles Crehange, Nicolas Magné, Sophie Chiavassa, Stéphane Supiot, Juan David Ospina Arango, Borris Campillo-Gimenez, et al. Voxel-based analysis for identification of urethrovesical subregions predicting urinary toxicity after prostate cancer radiation therapy. *International Journal of Radiation Oncology* Biology* Physics*, 104(2):343–354, 2019.

[101] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *J. Am. Med. Inform. Assoc.*, 18(5):544, 2011.

[102] Nachiappan Nagappan, Laurie Williams, Mladen Vouk, and Jason Osborne. Early estimation of software quality using in-process testing metrics: A controlled case study. *SIGSOFT Softw. Eng. Notes*, 30(4):1–7, May 2005.

[103] Andrew Ng. Clustering with the k-means algorithm. *Machine Learning*, 2012.

[104] N Niedell and M Taner. Semblance and other coherency measures for multichanel data. *Geophysics*, 36:482–497, 1971.

[105] Gunnar Nilsson, Håkan Petersson, Hans Åhlfeldt, and L-E Strender. Evaluation of three swedish icd-10 primary care versions: reliability and ease of use in diagnostic coding. *Methods of information in medicine*, 39(04/05):325–331, 2000.

[106] T. B. Noor and H. Hemmati. A similarity-based approach for test case prioritization using historical failure data. In *2015 IEEE 26th International Symposium on Software Reliability Engineering (ISSRE)*, pages 58–68, Nov 2015.

[107] T. B. Noor and H. Hemmati. Test case analytics: Mining test case traces to improve risk-driven testing. In *2015 IEEE 1st International Workshop on Software Analytics (SWAN)*, pages 13–16, March 2015.

[108] ME O'Callaghan, Elspeth Raymond, J Campbell, AD Vincent, Kerri Beckmann, D Roder, S Evans, J McNeil, J Millar, J Zalcberg, et al. Tools for predicting patient-reported outcomes in prostate cancer patients undergoing radical prostatectomy: a systematic review of prognostic accuracy and validity. *Prostate cancer and prostatic diseases*, 20(4):378–388, 2017.

[109] Webpage: Office of National Coordinator. Office of the national coordinator for health information technology, 2016.

[110] IM Okkes, A Groen, SK Oskam, and H Lamberts. Advantages of long observation in episode-oriented electronic patient records in family practice. *Methods of information in medicine*, 40(03):229–235, 2001.

[111] Daniel E O'Leary. Artificial intelligence and big data. *IEEE intelligent systems*, 28(2):96–99, 2013.

[112] World Health Organization. Cancer.

[113] Emily Oshima Lee and Ezekiel J Emanuel. Shared decision making to improve care and reduce costs. *New England Journal of Medicine*, 368(1):6–8, 2013.

[114] Min Jun Park and Mauricio D Sacchi. Automatic velocity analysis using convolutional neural network and transfer learning. *Geophysics*, 85(1):V33–V43, 2020.

[115] Razvan Pascanu, Guido Montufar, and Yoshua Bengio. On the number of response regions of deep feed forward networks with piece-wise linear activations. *arXiv preprint arXiv:1312.6098*, 2013.

[116] J Pearl. Morgan kaufmann series in representation and reasoning. *Probabilistic reasoning in intelligent systems: Networks of plausible inference. San Mateo, CA, US: Morgan Kaufmann*, 1988.

[117] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[118] Lars-Erik Strender Hans åhlfeldt Håkan Petersson, Gunnar Nilsson. The connection between terms used in medical records and coding system: a study on swedish primary health care data. *Medical informatics and the Internet in medicine*, 26(2):87–99, 2001.

[119] A. Podgurski, D. Leon, P. Francis, W. Masri, M. Minch, Jiayang Sun, and Bin Wang. Automated support for classifying software failure reports. In *25th International Conference on Software Engineering, 2003. Proceedings.*, pages 465–475, May 2003.

[120] Cecilia Pyper, Justin Amery, Marion Watson, and Claire Crook. Patients' experiences when accessing their on-line electronic patient records in primary care. *Br J Gen Pract*, 54(498):38–43, 2004.

[121] Wullianallur Raghupathi and Viju Raghupathi. Big data analytics in healthcare: promise and potential. *Health information science and systems*, 2(1):3, 2014.

[122] Dimitris Rizopoulos. Max kuhn and kjell johnson. applied predictive modeling. new york, springer. *Biometrics*, 74(1):383–383, 2018.

[123] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[124] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1961.

[125] Timothy J Ross. *Fuzzy logic with engineering applications*. John Wiley & Sons, 2009.

[126] Linda Rossi, Rik Bijman, Wilco Schillemans, Shafak Aluwini, Carlo Cavedon, Marnix Witte, Luca Incrocci, and Ben Heijmen. Texture analysis of 3d dose distributions for predictive modelling of toxicity rates in radiotherapy. *Radiotherapy and Oncology*, 129(3):548–553, 2018.

[127] Gregg Rothermel, Roland H. Untch, Chengyun Chu, and Mary Jean Harrold. Test case prioritization: An empirical study. In *Proceedings of the IEEE International Conference on Software Maintenance*, ICSM '99, pages 179–, Washington, DC, USA, 1999. IEEE Computer Society.

[128] D Rumerlhart, G Hinton, and R Williams. Learning internal representations by back-propagation errors. *Nature*, 323(99):533–536, 1986.

[129] A Rusiecki. Trimmed categorical cross-entropy for deep learning with label noise. *Electronics Letters*, 55(6):319–320, 2019.

[130] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[131] J Russell Stuart and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall, 2009.

[132] Takuya Sadahira, Yosuke Mitsui, Motoo Araki, Yuki Maruyama, Koichiro Wada, Kohei Edamura, Yasuyuki Kobayashi, Masami Watanabe, Toyohiko Watanabe, and Yasutomo Nasu. Pelvic magnetic resonance imaging parameters predict urinary incontinence after robot-assisted radical prostatectomy. *LUTS: Lower Urinary Tract Symptoms*, 11(3):122–126, 2019.

[133] M. Saeki, H. Horai, and H. Enomoto. Software development process from natural language specification. In *11th International Conference on Software Engineering*, pages 64–73, May 1989.

[134] Marc Wallace Salisbury, Chris and Alan A. Montgomery. Patients' experience and satisfaction in primary care: secondary analysis using multilevel modelling. *BMJ: British Medical Journal*, 2010.

[135] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.

[136] Sandhya Samarasinghe. *Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition*. CRC Press, 2016.

[137] MG Sanda, JT Wei, and MS Litwin. Scoring instructions for the expanded prostate cancer index composite short form (epic-26). 2019.

[138] Wouter Schaake, Arjen van der Schaaf, Lisanne V van Dijk, Alfons CM van den Bergh, and Johannes A Langendijk. Development of a prediction model for late urinary incontinence, hematuria, pain and voiding frequency among irradiated prostate cancer patients. *PloS one*, 13(7):e0197757, 2018.

[139] Jurandyr Schmidt and Frank A Hadsell. Neural network stacking velocity picking. In *SEG Technical Program Expanded Abstracts 1992*, pages 18–21. Society of Exploration Geophysicists, 1992.

[140] David W Scott. *Multivariate density estimation: theory, practice, and visualization.* John Wiley & Sons, 2015.

[141] Shahrokh F Shariat, Pierre I Karakiewicz, Claus G Roehrborn, and Michael W Kattan. An updated catalog of prostate cancer predictive tools. *Cancer: Interdisciplinary International Journal of the American Cancer Society*, 113(11):3075–3099, 2008.

[142] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[143] Anshu Sinha, George Hripcsak, and Marianthi Markatou. Large datasets in biomedicine: a discussion of salient analytic issues. *Journal of the American Medical Informatics Association*, 16(6):759–767, 2009.

[144] Ted A Skolarus, Rodney L Dunn, Martin G Sanda, Peter Chang, Thomas K Greenfield, Mark S Litwin, John T Wei, Meredith Regan, Larry Hembroff, Dan Hamstra, et al. Minimally important difference for the expanded prostate cancer index composite short form. *Urology*, 85(1):101–106, 2015.

[145] Anders Søgaard. Simple semi-supervised training of part-of-speech taggers. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 205–208, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[146] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search.* MIT press, 2000.

[147] Dawn Stacey, Carol L Bennett, Michael J Barry, Nananda F Col, Karen B Eden, Margaret Holmes-Rovner, Hilary Llewellyn-Thomas, Anne Lyddiatt, France Légaré,

and Richard Thomson. Decision aids for people facing health treatment or screening decisions. *Cochrane Database Syst Rev*, 10(10), 2011.

[148] Jürgen Stausberg, Dietrich Koch, Josef Ingenerf, and Michael Betzler. Comparing paper-based with electronic patient records: lessons learned during a study on diagnosis and procedure codes. *Journal of the American Medical Informatics Association*, 10(5):470–477, 2003.

[149] ME Stokes, L Black, A Benedict, CG Roehrborn, and P Albertsen. Long-term medical-care costs related to prostate cancer: estimates from linked seer-medicare data. *Prostate cancer and prostatic diseases*, 13(3):278, 2010.

[150] Vladimir Svetnik, Andy Liaw, Christopher Tong, J Christopher Culberson, Robert P Sheridan, and Bradley P Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of Chemical Information and Computer Sciences*, 43(6):1947–1958, 2003.

[151] Wei Liu Yangqing Jia Pierre Sermanet Scott Reed Dragomir Anguelov Dumitru Erhan Vincent Vanhoucke Szegedy, Christian and Andrew Rabinovich. Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. 2015.*, 2015.

[152] Konrad M Szymanski, John T Wei, Rodney L Dunn, and Martin G Sanda. Development and validation of an abbreviated version of the expanded prostate cancer index composite instrument for measuring health-related quality of life among prostate cancer survivors. *Urology*, 76(5):1245–1250, 2010.

[153] MT Taner and F Koehler. Velocity spectra—digital computer derivation and applications of velocity functions: Geophysics. *X59-88*, 1, 1969.

[154] Stephen W. Thomas, Hadi Hemmati, Ahmed E. Hassan, and Dorothea Blostein. Static test case prioritization using topic models. *Empirical Software Engineering*, 19(1):182–212, 2014.

[155] Antonio Tienza, Jose E Robles, Mateo Hevia, Ruben Algarra, Fernando Diez-Caballero, and Juan I Pascual. Prevalence analysis of urinary incontinence after radical prostatectomy and influential preoperative factors in a single institution. *The Aging Male*, 21(1):24–30, 2018.

[156] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[157] Hastie Trevor, Tibshirani Robert, and Friedman JH. The elements of statistical learning: data mining, inference, and prediction, 2009.

[158] Kuo Chun Tsai, Wenyi Hu, Xuqing Wu, Jiefu Chen, and Zhu Han. Automatic first arrival picking via deep learning with human interactive learning. *IEEE Transactions on Geoscience and Remote Sensing*, 58(2):1380–1391, 2019.

[159] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

[160] Galina Velikova, Laura Booth, Adam B Smith, Paul M Brown, Pamela Lynch, Julia M Brown, and Peter J Selby. Measuring quality of life in routine oncology practice improves communication and patient well-being: a randomized controlled trial. *Journal of Clinical Oncology*, 22(4):714–724, 2004.

[161] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[162] Jochen Walz, Andrea Gallina, Paul Perrotte, Claudio Jeldres, Quoc-Dien Trinh, Georg C Hutterer, Miriam Traumann, Alvaro Ramirez, Shahrokh F Shariat, Michael McCormack, et al. Clinicians are poor raters of life-expectancy before radical prostatectomy or definitive radiotherapy for localized prostate cancer. *BJU international*, 100(6):1254–1258, 2007.

[163] Weiyu Wang and Keng Siau. Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: a review and research agenda. *Journal of Database Management (JDM)*, 30(1):61–79, 2019.

[164] Lindsay Ward and Michael Innes. Electronic medical summaries in general practice–considering the patient's contribution. *Br J Gen Pract*, 53(489):293–297, 2003.

[165] John T Wei, Rodney L Dunn, Mark S Litwin, Howard M Sandler, and Martin G Sanda. Development and validation of the expanded prostate cancer index composite (epic) for comprehensive assessment of health-related quality of life in men with prostate cancer. *Urology*, 56(6):899–905, 2000.

[166] S Wei, O Yonglin, Z Qingcai, H Jiaqiang, and S Yaying. Unsupervised machine learning: K-means clustering velocity semblance auto-picking. In *80th EAGE Conference and Exhibition 2018*, volume 2018, pages 1–5. European Association of Geoscientists & Engineers, 2018.

[167] Weiderpass and Stewart BW. World cancer report: Cancer research for cancer prevention.

[168] Jeffery C Wheat, Ryan C Hedgepeth, Chang He, LingLing Zhang, and David P Wood Jr. Clinical interpretation of the expanded prostate cancer index composite-short form sexual summary score. *The Journal of urology*, 182(6):2844–2849, 2009.

[169] Susan White. A review of big data in healthcare: challenges and opportunities. *Open Access Bioinformatics*, 2014:13–18, 10 2014.

[170] A Wolfe. Institute of medicine report: crossing the quality chasm: a new health care system for the 21st century. *Policy, Politics, & Nursing Practice*, 2(3):233–235, 2001.

[171] Xinming Wu, Yunzhi Shi, Sergey Fomel, and Luming Liang. Convolutional neural networks for fault interpretation in seismic images. In *SEG Technical Program Expanded Abstracts 2018*, pages 1946–1950. Society of Exploration Geophysicists, 2018.

[172] Illhoi Yoo, Patricia Alafaireet, Miroslav Marinov, Keila Pena-Hernandez, Rajitha Gopidi, Jia-Fu Chang, and Lei Hua. Data mining in healthcare and biomedicine: a survey of the literature. *Journal of medical systems*, 36(4):2431–2448, 2012.

[173] J Yoo, RV Borselen, MS Mubarak, and C Tsingas. Automated first break picking method using a random sample consensus (ransac). In *81st EAGE Conference and Exhibition 2019*, volume 2019, pages 1–5. European Association of Geoscientists & Engineers, 2019.

[174] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. *HotCloud*, 10(10-10):95, 2010.

[175] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[176] Hao Zhang, Peimin Zhu, Yuan Gu, and Xiaozhang Li. Automatic velocity picking based on deep learning. In *SEG Technical Program Expanded Abstracts 2019*, pages 2604–2608. Society of Exploration Geophysicists, 2019.

[177] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.