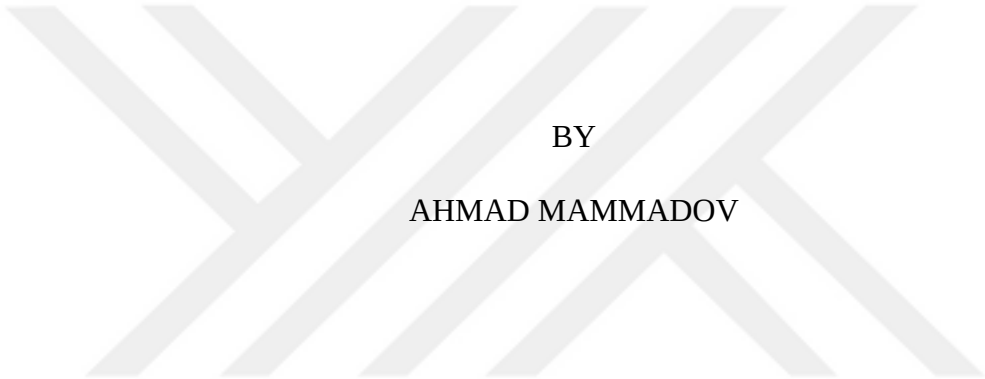


DERIVATION OF PRESCRIPTIVE ACCIDENT PREVENTION MODEL
FROM PREDICTIVE MODELS USING ML ALGORITHMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY



BY

AHMAD MAMMADOV

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
OCCUPATIONAL HEALTH AND SAFETY

DECEMBER 2021

Approval of the thesis:

**DERIVATION OF PRESCRIPTIVE ACCIDENT PREVENTION MODEL
FROM PREDICTIVE MODELS USING ML ALGORITHMS**

submitted by **AHMAD MAMMADOV** in partial fulfillment of the requirements for
the degree of **Master of Science in Occupational Health and Safety, Middle East
Technical University** by,

Prof. Dr. Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Mahmut Parlaktuna
Head of the Department, **Occupational Health and Safety**

Assist. Prof. Dr. Onur Behzat Tokdemir
Supervisor, **Civil Engineering, METU**

Examining Committee Members:

Prof. Dr. Mahmut Parlaktuna
Petroleum and Natural Gas Engineering, METU

Assist. Prof. Dr. Onur Behzat Tokdemir
Civil Engineering, METU

Prof. Dr. Gülbiye Yaşar
Department of Health Management, Ankara University

Date: 02.12.2021



I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name : Ahmad Mammadov

Signature :

ABSTRACT

DERIVATION OF PRESCRIPTIVE ACCIDENT PREVENTION MODEL FROM PREDICTIVE MODELS USING ML ALGORITHMS

Mammadov, Ahmad
Master of Science, Occupational Health and Safety
Supervisor : Assist. Prof. Dr. Onur Behzat Tokdemir

December 2021, 195 pages

The main drive upon which this study relies is to introduce a prescriptive accident prevention model to avoid work related accidents by predicting outcomes of work-related accidents in pipeline construction with using machine learning algorithms. In depth study on construction accidents is crucial due to construction being one of the most hazardous industry and being temporary in nature. To come up with a prescriptive accident prevention model, all incident reports from a pipeline project were analysed and a data set was prepared for 1,184 cases with attributes. These attributes consist of twenty-four immediate causes, eighteen root causes and three consequences that are nearmiss, asset or property damage and injury. A machine learning tool, RapidMiner, is used to predict outcomes of the cases for different data subsets by using eleven different ML algorithms. One of the machine learning algorithms, Deep Learning, was selected due to performing better in predicting outcome of complex data sets and in majority of twelve data sets. Model performance was attempted to be optimized with parameter optimization. It was concluded that predictive models with optimized parameters can predict accident outcomes better and a prescriptive accident prevention model can be presented thanks to these predictions. With a prescriptive model, it may be possible to provide a foresight

about the root cause, immediate cause, or date and time of potential accidents. The causes of accidents can be eliminated; hence accidents can be prevented with these predictions having statistical basis.

Keywords: Accident Prevention, Prediction, Prescriptive Model, Algorithms



ÖZ

MAKİNE ÖĞRENME ALGORİTMALARINI KULLANARAK TAHMİNSEL MODELLERDEN PRESKRİPTİF KAZA ÖNLEME MODELİNİN TÜRETİLMESİ

Mammadov, Ahmad
Yüksek Lisans, İş Sağlığı ve Güvenliği
Tez Yöneticisi: Dr. Öğr. Üyesi Onur Behzat Tokdemir

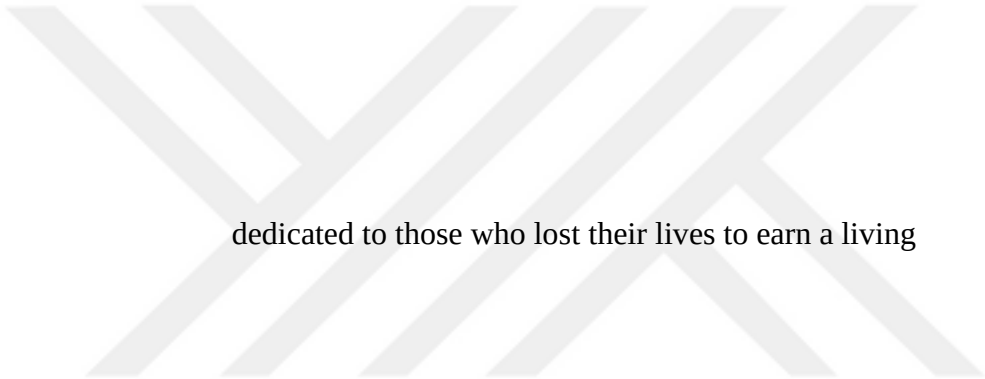
Aralık 2021, 195 sayfa

Bu çalışmanın temeli, makine öğrenimi algoritmalarını kullanılmasıyla boru hattı yapımında işle ilgili kazaların sonuçlarını tahmin etmek ve kazaları önlemek için preskriptif bir kaza önleme modeli sunmaktır. İnşaat sektörünü en tehlikeli sektörlerden biri olması ve doğası gereği geçici olması nedeniyle inşaat kazalarının derinlemesine incelenmesi çok önemlidir. Preskriptif bir kaza önleme modeli oluşturmak için bir boru hattı projesindeki tüm olay raporları analiz edilmiş ve 1,184 kaza için bir veri seti hazırlanmıştır. Yirmi dört doğrudan neden ve on sekiz kök nedenin yanı sıra ramak kala, maddi hasar ve yaralanmadan oluşan üç farklı kaza sonucu nitelikler şeklinde belirlenmiştir. On bir farklı ML algoritması, bir makine öğrenme aracı olan RapidMiner programında farklı veri alt kümeleri için kazaların sonuçlarını tahmin etmek için çalıştırılmıştır. Makine öğrenimi algoritmalarından biri olan Derin Öğrenme, karmaşık veri kümelerinin sonucunu tahmin etmede ve on iki veri kümesinin çoğunda daha iyi performans göstermesi nedeniyle seçilmiştir. Parametre optimizasyonu ile model performansı optimize edilmeye çalışılmıştır.

Optimizasyon sonrası tahminsel modellerin kaza sonuçlarını iyi şekilde tahmin edebildiği ve bu tahmin sayesinde preskriptif bir kaza önleme modelinin sunulabileceği sonucuna varılmıştır. Preskriptif bir model ile gelecekte oluşabilecek kazaların hangi kök neden, doğrudan nedenden dolayı veya hangi tarih aralığında oluşabileceğın ilişkin bir öngörü sunabilmek mümkün olabilir. Bu istatistiksel arka plana sahip öngörü ile kazaların nedenleri ortadan kaldırılarak kazaların önüne geçilebilir.

Anahtar Kelimeler: Kaza Önleme, Tahmin, Preskriptif Model, Algoritmalar





dedicated to those who lost their lives to earn a living

ACKNOWLEDGMENTS

This thesis would not have been possible with help from few very important people. Firstly, I would like to thank my advisor Assist. Prof. Dr. Onur Behzat Tokdemir. I am very grateful to him not only for his academic support but also for his guidance in my career. I consider myself lucky to know him and being a student of him. I have learnt a lot from him and would always remember the amazing talks that I had with him.

I would also like to thank Prof. Dr. Mahmut Parlaktuna for being such a great teacher and his support. He was my teacher in my bachelors' years in Petroleum and Natural Gas Engineering, he was in the admission committee for my master's degree, he was my teacher in graduate classes, and he was in the thesis examining committee. He witnessed all my education in the university and constantly mentored me throughout the time.

I want to extend my gratitude to whom I owe very much of who I am today. Prof. Dr. Nuray Demirel has always been the one I admired and respected.

I would like to express my special thanks of gratitude to Prof. Dr. Gülbiye Yaşar for her advises on the thesis and being in the examining committee.

I am indebted to my family, especially to my mother Hagigat Mammadova for her unconditional love, my father Ganimat Mammadov for continuous support, my wife Dilay Mammadov, and all my friends for creating such a loving and supportive environment.

Finally, all this would not have been possible without the support from my colleagues from beSafe "Reliable Safety Solutions".

It was a quite long journey. I will always be thankful to those who made this happen.

TABLE OF CONTENTS

ABSTRACT.....	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS.....	xi
LIST OF TABLES	xiv
LIST OF FIGURES	xvi
1 INTRODUCTION	1
1.1 Introduction.....	1
1.2 Project Background.....	2
1.3 Scope, Objectives and Research Questions	4
1.3.1 Scope.....	4
1.3.2 Objective	4
1.3.3 Research Questions	5
1.4 Thesis Framework.....	5
2 LITERATURE REVIEW	7
2.1 Pipeline Transportation.....	7
2.1.1 History of Pipeline Transportation.....	7
2.1.2 Pipeline Construction.....	8
2.1.3 Hazards and Risks in Pipeline Construction	14
2.1.4 Roles in Pipeline Construction.....	17
2.2 Predictive Modeling and Data Analytics	18

2.2.1	Types of Machine Learning Systems	23
2.2.2	Training Models	32
2.2.3	Counting the Cost	71
2.2.4	Overfitting and Underfitting.....	75
2.3	Prescriptive Data Analytics	75
3	METHODOLOGY	79
3.1	Data Collection, Variable Description and Data Processing.....	79
3.2	Descriptive Data Analysis	88
3.2.1	Distribution of total cases and type of exposure	88
3.2.2	Distribution of consequences	97
3.2.3	Distribution of injury cases	101
3.2.4	Distribution of causes	105
3.2.5	Statistics of incomplete data regarding demographic information.....	138
4	RESULTS AND DISCUSSION.....	141
4.1	Results for SET01.....	141
4.2	Results for SET02.....	143
4.3	Results for SET03.....	145
4.4	Results for SET04.....	148
4.5	Results for SET05.....	150
4.6	Results for SET06.....	152
4.7	Results for SET07.....	154
4.8	Results for SET08.....	158
4.9	Results for SET09.....	162

4.10	Results for SET10	165
4.11	Results for SET11	169
4.12	Results for SET12	173
4.13	Parameter Optimization	179
5	CONCLUSION	183
	REFERENCES	185



LIST OF TABLES

TABLES

Table 2.1 Initialization parameters for each type of activation function.....	60
Table 2.2 Default cost matrices: (a) a two-class case and (b) a three-class case. ...	73
Table 3.1 Categories of event and exposure.....	82
Table 3.2 List of immediate and root causes	83
Table 3.3 Data sets	87
Table 3.4 Distribution of immediate causes	106
Table 3.5 Immediate Causes and Type of Exposure	109
Table 3.6 Immediate Causes and Type of Consequence	111
Table 3.7 Immediate Causes and Other Immediate Causes	113
Table 3.8 Immediate Causes and Root Causes.....	115
Table 3.9 Distribution of Root Causes	122
Table 3.10 Root Causes and Type of Exposure	125
Table 3.11 Root Causes and Type of Consequence	127
Table 3.12 Root Causes and Immediate Causes.....	129
Table 3.13 Root Causes and Other Root Causes	131
Table 3.14 Age distribution of the people involved	138
Table 4.1 Results for SET01	141
Table 4.2 Results for SET02	143
Table 4.3 Results for SET03	145
Table 4.4 Results for SET04	148
Table 4.5 Results for SET05	150
Table 4.6 Results for SET06	152
Table 4.7 Results for SET07	154
Table 4.8 Results for SET08	158

Table 4.9 Results for SET09	162
Table 4.10 Results for SET10	166
Table 4.11 Results for SET11	169
Table 4.12 Results for SET12	173
Table 4.13 Overview of the Results.....	178
Table 4.14 Results for Deep Learning after parameter optimization for all data sets	179



LIST OF FIGURES

FIGURES

Figure 2.1. The conceptual illustration of pipeline construction (Wininger, 2011) ..	8
Figure 2.2 V-Groove configuration for butt welding (Balram 2019).....	12
Figure 2.3 Internal Hydraulic Pipe Clamp (WSBC, 2017).	16
Figure 2.4 A connection graph of simple neural networks (Haykin, 2005)	52
Figure 2.5 Perceptron Model (Minsky and Papert, 1969)	56
Figure 2.6 Schematic view of an autoencoder consisting of single encoder and decoder	69
Figure 2.7 Schematic view of a sparse autoencoder.....	70
Figure 2.8 Business value gained through event prediction and proactive actions (Krumeich et al., 2015).....	77
Figure 2.9 Proposed zero harm through event prediction and proactive actions	77
Figure 3.1. Flow of the study.....	80
Figure 3.2. Attribute based framework (Esmaeili and Hallowell (2012)	84
Figure 3.3. Validation procedure for predictive model	86
Figure 3.4 Distribution of the cases per year and quarter.....	89
Figure 3.5 Distribution of the cases per quarter, month, day of the week and hour	90
Figure 3.6 Distribution of the cases per type of incident	91
Figure 3.7 Distribution of the cases per type of exposure per quarter	92
Figure 3.8 Distribution of the cases per type of exposure per month.....	94
Figure 3.9 Distribution of the cases per type of exposure per day of week	95
Figure 3.10 Distribution of the cases per type of exposure per hour of day	96
Figure 3.11 Distribution of the cases per type of exposure per hour of day	97
Figure 3.12 Distribution of the cases per type of consequence per quarter	98
Figure 3.13 Distribution of the cases per type of consequence per type of exposure	99

Figure 3.14 Distribution of the cases according to type of consequences	100
Figure 3.15 Distribution of the injury cases according to type of injury	101
Figure 3.16 Distribution of injury cases according to type of exposure	102
Figure 3.17 Distribution of injury cases per quarter	103
Figure 3.18 Monthly Distribution of first aid cases	104
Figure 3.19 Distribution of first aid cases per day of the week	104
Figure 3.20 Distribution of immediate causes over other attributes	107
Figure 3.21 Quarterly distribution of immediate causes.....	116
Figure 3.22 Monthly distribution of immediate causes	118
Figure 3.23 Distribution of immediate causes over day of the week.....	120
Figure 3.24 Distribution of immediate causes over hour of the day.....	121
Figure 3.25 Distribution of root causes over other attributes	123
Figure 3.26 Quarterly distribution of root causes	132
Figure 3.27 Monthly distribution of root causes.....	134
Figure 3.28 Distribution of root causes over day of the week	135
Figure 3.29 Distribution of root causes over hour of the day	137
Figure 3.30 Age distribution of the people involved	139
Figure 3.31 Experience of the people involved	139

CHAPTER 1

INTRODUCTION

1.1 Introduction

Construction industry is one of the most hazardous industry due to being mostly temporary in nature, labor force circulation and background of the labor force available to the industry. An increase in labor force were seen in last decades as reported by the World Bank in 2017, fortunately, occupational fatalities and injuries saw a decrease in last four decades, however, level of risk remains high for construction safety (Ayhan and Tokdemir, 2019). More than 100,000 people lost their lives on worksites, which is almost 30% of all fatalities and makes construction industry three to four times more fatal than other industries in developed countries and three to 6 times in developing countries (ILO, 2015). Fatality and accident statistics from several studies and organizations in different countries such as United Kingdom, United States, Turkey and China also highlighted the importance of health and safety in construction industry (Zhang et al., 2013; Dong et al., 2013; BLS, 2021; HSE, 2014; Li and Wang, 2004; Tam et al., 2006; Liao and Perng, 2008; Tam et al., 2004; Misiurek and Misiurek, 2017; Mohammadi et al., 2018).

Construction industry accounted for almost 28.2% of all fatalities and 11.5% of all work-related accidents recorded for the year of 2020 in Turkey. Total days of temporary incapacity including outpatient and inpatient cases in construction industry constituted 14.4% of total days of temporary incapacity for all activities (SGK, 2021)

There were 1,102 fatal injuries in the construction industry in 2019 in private industry and government. These deaths represented 20.7 percent of total workplace fatalities in the United States (BLS, 2021).

Not only environmental conditions and working nature of construction industry contributes to these accident statistics, but also other factors such as human factor and insufficient training contribute to work related accidents in this industry (Kang and Ryu, 2019). Because of the reasons aforementioned and being extremely prone to occupational accidents, HSE is in the hearth of all construction projects like mining, foundry, and other sectors relying on labor intensity (Nguyen et al., 2016). Therefore, besides all classical statistic studies of construction accidents, it is crucial to study construction accidents with machine learning which has a high potential in decreasing overall fatality and accident statistics in construction works. Being a construction industry, pipeline projects share most domains with other construction works.

1.2 Project Background

This study dates to early 2018 and starting point was the result of several discussion been held in a lecture given in graduate school. Concept of Safety-I and Safety-II was the subject been discussed in these classes. This concept was first introduced by Erik Hollnagel in his technical report named “*From Safety-I to Safety-II: A White Paper*” in 2015. In Safety-I approaches, accidents are the results of identifiable failures or malfunctions of specific components. Causes and contributing factors are investigated in this approach. However, in Safety-II, Hollnagel states in his article:

Crucially, the Safety-I view does not stop to consider why human performance practically always goes right. Things do not go right because

people behave as they are supposed to, but because people can and do adjust what they do to match the conditions of work. As systems continue to develop and introduce more complexity, these adjustments become increasingly important to maintain acceptable performance. The challenge for safety improvement is therefore to understand these adjustments—in other words, to understand how performance usually goes right in spite of the uncertainties, ambiguities, and goal conflicts that pervade complex work situations. Despite the obvious importance of things going right, traditional safety management has paid little attention to this. Safety management should therefore move from ensuring that ‘as few things as possible go wrong’ to ensuring that ‘as many things as possible go right’ (Hollnagel et al., 2015).

Regardless of the resilience of the concept introduced and its potential to improve safety, I believe Safety-I concept has been always stuck in traditional data analytics, therefore, complete potential of learning from failures was not achieved. Most of the studies for Safety-I concept have mainly focused on introducing distribution of the causes and contributing factors, and some descriptive statistic. Descriptive statistics was effective in diminishing symptoms of the system but was only reactive. With technological advances, it is possible to analyze data available in depth and derive new information from this data that can be effective in avoiding traditional data analysis pitfalls and introducing new models to overcome potential system failures in the future. Data available for accidents have been studied in several research but still need more to be conducted because of the increasing computing power for machine learning and need for decreasing uncertainties and explaining attributes in accidents. To sum up, there is a lot to learn from accidents or what went wrong. Therefore, this research started in 2018 aiming at extracting more details from accident statistic and introducing new accident prevention model which is proactive or generative. In this regard, a complete accident statistic from a project was needed and it was found after putting so much effort into this.

1.3 Scope, Objectives and Research Questions

1.3.1 Scope

This thesis involves data collected from 1,184 work related accidents which consist of almost 25,000 pages of reports and aims to derive a prescriptive accident prevention model from predictive analysis of this data. Study took 3 years to complete. Almost 70% of time spent of this thesis was on reading accident reports and translating to an alphanumerical format then logical matrix was constructed using attribute-based framework. RapidMiner, a data science software platform developed by the company of the same name that provides an integrated environment for data preparation, machine learning, deep learning, text mining, and predictive analytics was used for predicting outcomes of accidents from date, time, root cause and immediate cause. It is used for business and commercial applications as well as for research, education, training, rapid prototyping, and application development and supports all steps of the machine learning process including data preparation, results visualization, model validation and optimization. RapidMiner is developed on an open core model. Results of predictive analysis of eleven ML algorithm was used to derive a prescriptive accident prevention model.

1.3.2 Objective

Based on the scope introduced, this research aims at bringing a new model, prescriptive accident prevention model with in-depth analysis of work-related accidents. For this purpose, twelve data subsets were formed from original data set, and eleven different ML algorithms were selected to predict outcomes (consequences) of this data sets. Attempt is made for finding the highest performing ML algorithms among these algorithms and increasing performance of the algorithm

with parameter optimization. After parameter optimization, performance of the model is evaluated with comparing predicted outcomes with original outcomes. Performance of the predictions are used to derive prescriptive accident prevention model assuming accidents can be avoided given that the consequences is predicted correctly.

1.3.3 Research Questions

Based on the scope and the objectives discussed above, the following research questions are formed which this thesis eventually attempts at answering.

- Will predictive models be successful in predicting outcomes of work-related accidents?
- Which predictive model is better in predicting outcome of work-related accidents from attributes selected?
- Which algorithm is better in predicting outcome for data subsets?
- Which attributes should be selected for performance improvement?
- Will parameter optimization improve performance of the selected model?
- Is it possible to introduce a prescriptive accident prevention model from predictive models?

1.4 Thesis Framework

This report is divided into five chapters, the first chapter is the introduction to the thesis where the topic is introduced and gives brief outline on how the topic was chosen, the need for this research based on which research questions were formed. In the second chapter detailed literature review is provided addressing the various research in pipeline transportation and construction, predictive modelling and data

analytics, and prescriptive data analytics. Methodology chapter is the third chapter that covers both steps taken in data collection, how variables or attributes were selected and described, stages of data processing and descriptive statistics of processed data. Forth chapter, results and discussion of the study recaps all the results from the model mentioned in the methodology chapter and these results are also discussed in this chapter. Conclusion is introduced in chapter five with future work and the limitations.



CHAPTER 2

LITERATURE REVIEW

2.1 Pipeline Transportation

2.1.1 History of Pipeline Transportation

Pipelines date back to 3000BC and used to transport water from rivers to agriculture lands and within cities. Some of these pipelines were buried underground to decrease vaporization and some were constructed from baked cylindrical clay pipes. Chinese constructed bamboo pipes to divert water to villages, Egyptians used copper pipes to supply water, Greeks used stone, lead and bronze pipes in large cities which were hammered to each other, Romans designed pipeline networks from iron, bronze, and other materials and used only stamped pipes according to water regulation in 400BC to 150 AD. The technology was accelerated with the Industrial Revolution, booming oil industry with discovery of oil in 1859 in Pennsylvania and rising natural gas and water distribution needs. Oil was first transformed successfully from wellhead to loading station via a 2-inched diameter pipeline in 1865. With the development of welding technology, pipe production growth was escalated, and standards and design of pipes were set and published in the 20th century (Kaiser, 2017 and Antaki, 2003)

Pipeline construction has become the most significant way of transportation with reaching to 2.73 million miles in 2017. The United States, Canada, and Russia own more than 75% of all pipeline networks in the world to transport oil, gas or water (CIA, 2017 and Cianciarullo, 2019). PGJ (2018) reported that those 45,412 miles of

pipeline was being constructed and 38,390 miles were planned for the year of 2018. Construction of these pipelines is complex and requires multiple execution phases especially in cross-country projects because of changing environment, geology, laws and regulation of different countries, and availability of resources (Rey and Tabucanon, 1996).

2.1.2 Pipeline Construction

The construction of pipelines starts with delivery of pipes to ROW, then stringing, pipe bending, welding, coating, positioning, installation of valves and fittings take place and finalized with trench backfilling. These steps are illustrated in the Figure 2.1 and explained in the following sections.

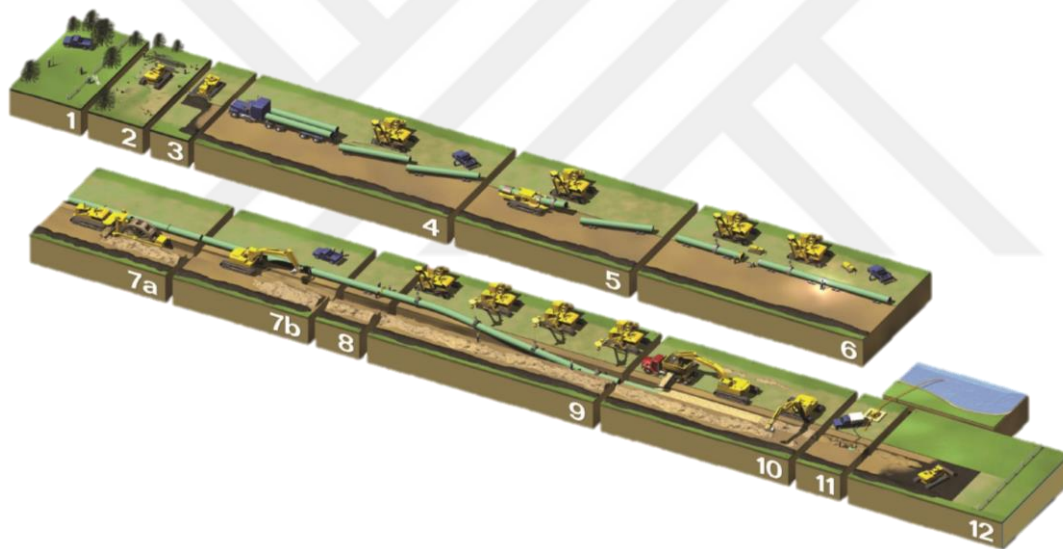


Figure 2.1. The conceptual illustration of pipeline construction (Wininger, 2011)

2.1.2.1 Surveying, Selection of Route and Stacking

Survey and stacking take place before initiation of a construction project in order to learn environmental and local issues and then the final pipeline route is decided. Selected routes are marked for upcoming stages.

Considering environmental issues are the most important step in deciding route because it aims to protect or minimize the impact on endangered species, wildlife, fertile lands, archeological or cultural heritage, nature, and water resources.

In route selection, another key factor is constructability of the route. Constructability refers to assessment of the route in terms of how the route will be constructed. Easement of right of the way, agreement between all agencies including government, property owners etc., accessibility of the route for construction equipment and vehicle, width required for activities on right-of-way, vicinity of the infrastructure, roads, railroads, and water bodies on the route, need for blasting, slope of terrain, size (diameter) of the pipeline, availability of the construction techniques and any other restrictions are considered during route selection. Inappropriate selection of the route could result in loss of time and increased budget (Ozanne, 2011 and Bauer, 2011).

2.1.2.2 Front-End Clearing

Front-end clearing aims to remove trees from pipeline route by grading ROW and workspace after a clearance certificate and permit for cutting trees on route. In this stage any existing pipeline are marked as well.

Access and service roads are constructed during this stage. These roads can be temporary or permanent and must be suitable for the vehicles without pallets (Wininger, 2011).

2.1.2.3 Right of Way Grading

ROW grading in cultivated lands takes place in order to strip off and piling of the topsoil, which will be replaced later on. Aim of the procedure is to save fertile topsoil (Bauer, 2011).

2.1.2.4 Pipe Stringing

Stringing of pipes involves pipe distribution and laying the pipes throughout the pipeline route or right-of-way for succeed welding, coating, lowering, and backfilling. These pipes are whether supplied from temporary stockpiles or directly from transport trucks. Correct allocation of the pipes from the right stockpile in this stage is critical. The distance between trench edge and pipe should at least be 100cm in stringing activity and wood skids are placed under the pipes to avoid direct contact with soil or ground. Stringing activity should be realized after any explosive job. Pipe stringing activity should be planned to minimize or avoid any interruption in water, electricity supply, road traffic and telecommunication of neighborhood (Bahadori, 2017).

2.1.2.5 Bending and Beveling

Bending of the pipes could be realized according to design requirements before welded to each other. Bending is operated if there is a need for changing direction of the pipe. Pipes are bended hydraulically to fit in the contours with track-mounted, bending machines. Hydraulic pressure and clamps series are used to avoid buckling and smooth bends.

Pipes are beveled or edges of the pipes are prepared in this stage for utilizing welding joints in the ends of each pipe. Bevels are either machined or grinded manually

(Bahadori, 2017).

2.1.2.6 Line-up, Welding and Welds Inspection

Pipes were lined up to each other with initial welding. The beveled joints of the pipes are butted to each other with automatic or manual welding machines while clamps are holding pipes together. Large diameter pipes are welded automatically with circular welding tools. Then, these welds are inspected with different nondestructive methods such as ultrasonic testing system (AUT), radiography (X-ray or Gamma-ray), magnetic particle testing etc. (Dorling and Gianetto, 2015).

SMAW (shielded metal arc welding), GMAW/GTAW (gas metal/tungsten arc welding) or combined welding techniques are used in pipeline welding with either semi/fully automated welding machines or with manual welding machines. Oxyacetylene welding should be employed in pipe welding. GMAW and GTAW activity should be run in isolated shacks to avoid wind and draught (Dorling and Gianetto, 2015).

Figure 2.2 illustrates steps of a typical V-bevel welding, which is composed of several beads. Before welding, pipes are cleaned and preheated with propane torches or electric blankets to avoid hydrogen cracking. Internal pneumatic clamps for larger diameter pipes are used to hold pipes together and then root pass is welded. Root pass or stringer pass is more critical. Root pass is followed by hot pass, fill pass and cap pass. Number of the passes varies with wall thickness of the pipes. Last step of welding is grinding and cleaning of the cap pass for coating work (Bubar, 2011 and Bahadori, 2017).

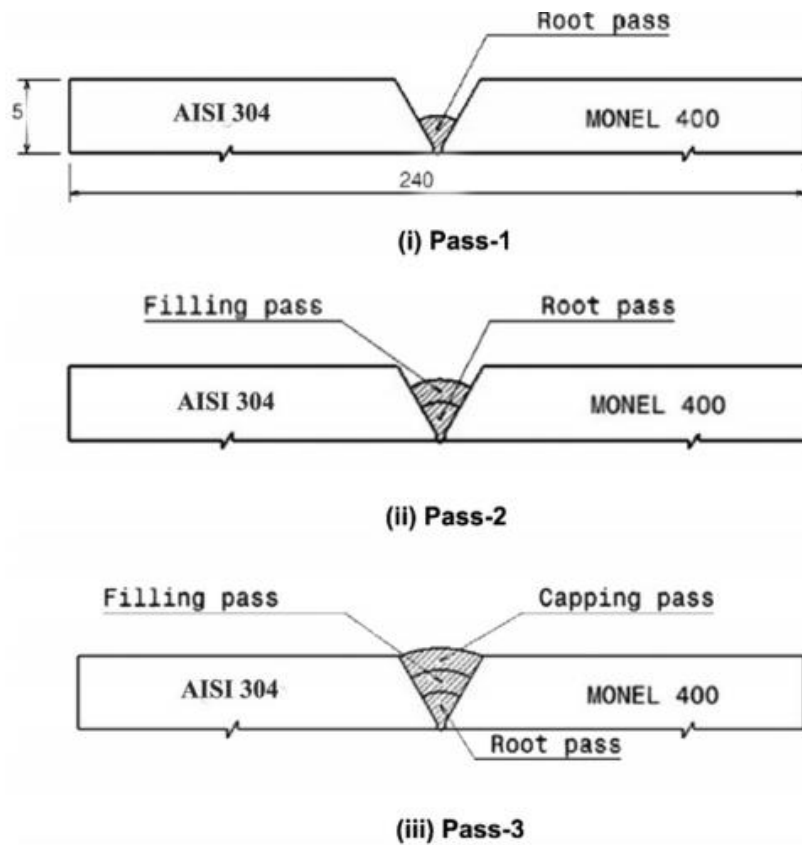


Figure 2.2 V-Groove configuration for butt welding (Balram 2019)

2.1.2.7 Trenching or Ditching

Trenches are excavated and cleared in this stage. Method of ditching is mainly dependent on the soil type. In some rock formations and crossing, blasting or horizontal drilling might be employed. Excavated material should not be stored together with topsoil.

Bell holes for welding are excavated after trenching with a suitable room for pipe coating and welding (Bahadori, 2017).

2.1.2.8 Final Coating and Inspection

Sandblasting or any mechanical method are employed to remove excessive welds for final coating. These coating are later inspected with holiday detectors before lowering-in (Zinter, 2015).

2.1.2.9 Lowering Pipe into Trench

Pipes are lowered into trench in this stage with side-booms. Side-booms are tractor like vehicles designed especially for pipe lowering. Rigging activity is a critical operation in pipe lowering activity because rigging requires proper selection of sling and its attachments according to load, determining weight to be loaded before hoisting, inspecting each component of rigging in beginning of each shifts, using load rated rig fittings, monitoring rigging personnel for qualification, training and competence, and overseeing safe work procedures during activity. Maintenance of the hoisting components, inspection of hooks for safety latches, pinning shackles for avoiding dislodgement are important for a safe lowering operation.

Side-booms must be operated within design capacity, must be equipped with rollover, falling object, operator protective structure and load rated boom extensions, must be inspected in accordance with manufacturers' specifications or any standards in place, service history and daily inspections records of side-booms must be kept in a log book on the vehicle (Wininger, 2011).

2.1.2.10 Pad, backfill, Rough Grade

Pad, backfill, rough grade involves using padding for pipes and filling trench with subsoil. Backfilling operation is immediately succeeded by pipe lowering operation to maintain correct position of the pipeline. Sides and tops of pipes are filled with

soft materials, which is free of stone and rock. Then, normal backfilling take place and trench is filled with the material with the order of being excavated initially. Final stage is grading the route to its original state (Wininger, 2011).

2.1.2.11 Testing Final Tie-In

In order to test structural integrity and welds, pressurized water is pumped inside the pipe to monitor pressure limits. Pipeline sections crossing rivers, bridges or connection points are finished after completion of mainline, which is called final tie-in (Wininger, 2011).

2.1.2.12 Final Clean-up, Full Restoration

Final clean up and restoration mean grading the land and spreading of the topsoil as close as original condition of the ROW. If any damage is given to public or private facilities such as roads and bridges, these facilities must be repaired and restored. Temporary access roads should be removed (Bahadori, 2017).

2.1.3 Hazards and Risks in Pipeline Construction

Being crushed or run over by machinery, struck by moving pipe, pipes being tensioned for welding or bending, machines, falling objects, including pipe or skids being thrown off trucks, materials on the ground that are run over and swing up are common hazards and risks in pipeline construction together with road traffic accidents (RTA) during mobilization of workers, machines, and logistics of pipes. In addition to these, machines being struck by pipe joints while moving in and out, struck by dislodged hydraulic hoses, being caught in between pipe joints and machines, rolled over vehicles are other common types of potential hazards in

construction of pipelines (WSBC, 2017; PCRA, 2018 and Oz et al., 2018).

These safety hazards can be categorized as below according to the operations, tools and equipment involved:

a) Vehicle operations involve service busses, cars, pickups, dump trucks, low-bed trucks, ambulances, tractors, side-booms, and tankers. Road, weather and vehicle conditions, excessive speeding, driver's judgment and fatigue, vicinity and parking of other vehicles, pedestrian traffics, wildlife, road and ROW design, and act of 3rd part drivers are some of the factors that may contribute to road traffic accidents.

b) Logistics operations include movement of the any heavy and light construction equipment or vehicles, containers, fences, and storage trailers during lifetime of a project. Road, ROW and weather conditions, improper loading, securing and route selection, and road and ROW design are some of the factors that may contribute to accidents during logistics operations.

c) Trips, slips and falls, worksite layout and condition, flying, dislodged and falling objects, exposure to fumes, electricity or chemicals, caught-in, contact with sharp objects and temperature extremes are common hazards in maintenance and use of air and electrical tools.

d) Grading and clearing of ROW involves several hazards such as contact with underground or overhead power lines, hazards resulting from wildlife, and other common hazard seen in other operation like falls, struck by moving equipment, slips, and trips due to terrain.

e) Blasting and rock drilling activities include all type of common hazards and hazards specific to explosives, namely, unintended explosions, dust, vibrations and

noise due to blasting operations, static electricity, unauthorized access to blasting area, and weather conditions.

f) Stringing operations involve unloading of pipe to ROW and alignment of pipes throughout the selected route. Swinging hooks, faulty lifting components, overhead power lines, pinch points, worksite conditions, and handling of skids might be source of hazards in stringing operations.

g) Struck by pipes run in/out to bending machine, pinch points, dislodged pipes and hydraulic line ruptures are very common during bending operations.

h) Trench cave-ins, fall of worker or vehicles into trenches, contact with underground power lines, telecom lines and water pipes, and moving vehicles are source of hazards in trenching activities.

i) Beveling activity involves hazards such as flying chips, exposure high noise, loose clothes and contact with temperature extremes.

j) Welding and welds inspection include electricity, radiation hazards, hazards arising from poor housekeeping and congested workplace, inhalation of fumes, sparks, gas cylinders, struck by dislodged internal pipe clamps, fire hazards and contact with hot surfaces.

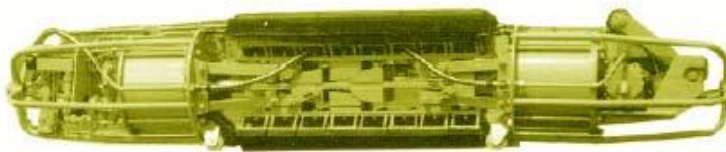


Figure 2.3 Internal Hydraulic Pipe Clamp (WSBC, 2017).

k) Exposure to chemicals, sand particles and electricity, inhalation of mist and dust, struck by dislodged pressurized equipment components are some of the hazards in coating operations.

l) Tunneling and drilling involve electricity hazards, confined space hazards, oxygen deficiency, contact with underground infrastructure, cave-ins, caught in rotating equipment and exposure to dust.

m) In pipe lowering activities, hazards of moving equipment, contact with overhead power lines, cave-ins and squeeze between moving pipes are very usual.

n) Tie-in activities have some hazards, which are arc flashing, contact with hot surfaces, ignition sources, exposure to noise, cave-ins, simultaneous vehicle operations, failing lifting elements, dust and fume hazards.

o) Contact with overhead power lines, vehicle rollover due to slope or ground condition, road traffic accidents, cave-ins, floods, debris, rock falls, and other environmental hazards can be seen not only in backfilling, full restoration, and pipeline testing stages but also almost all operations in pipeline construction.

2.1.4 Roles in Pipeline Construction

Construction manager, project manager, project engineers, superintendents, supervisors/foremen, project safety officer, engineers, surveyors, operators, sub-contractors, plant engineers, plant foreman, plant operators, welders, fitters, transport superintendent and foremen, drivers and all other unskilled personnel roles are involved in pipeline construction (WSBC, 2017).

2.2 Predictive Modeling and Data Analytics

Machine learning and data mining techniques are employed in predictive data analytics in order to predict succeeding events, trend and patterns with using existing information and calculating statistical probabilities. The difference between data mining and predictive analysis is that data mining deals with finding hidden associative patterns among variables of large data sets however predictive data analytics focuses on extracting useful information from these data sets with applying models. Predictive data analytics include techniques such as modeling of data, AI, data mining, machine learning and deep learning algorithms. Finlay (2004) states that predictive analysis is applicable to any event from past, happening presently or will happen in the future in unearthing unknowns such as suspect after a committed crime or fraud transactions. Prediction of unknown consequence take place after finding association between descriptive variables and the predicted variables from previous incidences. Siegel (2016) defines predictive data analytics as technology trained with samples to predict the forthcoming act of persons for pursuing better decision. Precision and usability of these predictions are highly dependent on the assumptions made and level of data analysis.

Data mining and machine learning aim to find patterns in data. This is not a new method because human being is used to finding patterns since the beginning of life. For example, people were observing animal behaviors to find out patterns and farmers were observing patterns in growth of crop. Storing and searching patterns electronically or data mining are not even new concept; use of automatic identification and validation of patterns in economics, communication engineering and weather forecasting for prediction were a common practice (Witten and Frank, 2005, p.4).

One of the first instance of machine learning was Skynet's application for spam mails in 1990s, which made flagging spam mails easier and automatic with learning from given sets of spam mails (Géron, 2017).

Witten and Frank define (2005) "learning" as a slippery concept because they state that machine learning is a mindless kind of learning or learning without purpose and it is not intentional, and it is more suitable to denote it as "training".

Training a neural network had been considered impossible since the 2006, most of researchers had abandoned deep learning in 1990s but this consideration has changed after Geoffrey Hinton et al., study about neural networks, which can predict handwritten digits with almost 98 percent precision. This study has redirected other researcher to vest interest on deep learning because this study showed the enormous capacity of deep learning (Géron, 2017).

With technological advance and new models, machine learning made analysis of data easier and faster in terms of dealing with multidimensional and large quantity problems. The use of machine learning for predicting the future events are seen in various areas such as marketing, retail, healthcare, aerospace, traffic, telecom, insurance, engineering, emergency preparedness, finance, tourism, banking, CRM, medicine, pharmaceutical and it provided beneficial results (Witten and Frank, 2005; Nyce, 2007; Eckerson, 2007; Conz, 2008; Fletcher, 2011; Korn, 2011; Barkin, 2011; Das and Vidyashankar, 2006; McDonald, 2010; Moreira-Matias et al., 2016; Stevenson, 2011; McKey, 2009; Olson, 2012; Wang et al., 2015; Tang et al., 2018; Farnaaz and Jabbar, 2016; Smarra et al., 2018; Masetic and Subasi, 2016; Tanaka et al., 2016; Cheng et al., 2019; Vomfell et al., 2018 and Robinson et al. 2015).

Tian et al. (2015) analyzed aircraft accidents to minimize landing accidents with prediction outcomes through SVM and RST analytical methods. Arockia and

Appavu (2013) used information about details of delays, aircraft maintenance, details of pilots and accident data in decision tree model to forecast warning level. In their later research, Arockia and Appavu (2013) were able to analyze warning level with comparing different predictive models and found out that warning level can be best predicted by decision tree model.

In order to find out the patterns in aircraft incidents and accidents with STUCCO algorithm, Nazeri et al. (2008) used NTSB accident database and causes of accidents.

Phark et al. (2018) used machine learning to predict issuance of emergency evacuation. They collected data from HSEES and ATSDR databases and extracted more than 60 thousand chemical substance accidents taken place between 1996 and 2014. Their study suggests that issuance of emergency evacuation orders in chemical accidents can be forecasted with machine learning with high accuracy within approximately a minute. The model can also be applied to new chemical substances, which are less known in terms of accidents caused by these substances.

In a study by Sánchez et al. (2011), more than 11, 000 questionnaire responses of Spanish workers for 78 variables were analyzed with SVM model after variable reduction with MARS algorithm to select which variables are more important. They concluded that model could predict approximately 98% and 100% of all workers who have suffered and have not suffered from workplace accident respectively. Their study showed that the model developed could predict work related accidents regardless of sector with high accuracy.

Oztekin et al. (2010) proposed a decision tree (DT) model for improving success of lung transplantation with using hundreds of variables about donors and patients queued for transplantation. Their model can increase success rate of transplantation and decrease the time of decision made by medical expert to allocate new potential

organ receiver.

Mistikoglu et al. (2015) used fall height and level of safety training as input variable of roof falls in a decision tree algorithm to find out whether the output is fatal or nonfatal in the United States. Their model was successful in predicting type pattern in roof falls accidents.

Even though machine learning is new for occupational accident and injury analysis, it has become a popular research area recently. Random forest and stochastic gradient boosting were used as ML in Tixier et al. study in 2016 for predicting injuries in construction works. The inputted attributes were composed of unstructured instances of injury reports and model used these reports to predict consequences of these accidents through a technique called text mining. This text mining technique was used later by Goh and Ubeynarayana in 2017 in order to classify unstructured construction accidents. They also compared several machine learning techniques such as SVM, LR, NBC, k-NN, decision trees and random forest in their study with a questionnaire collected from 80 workers. In order to learn what are the main indicators causing a construction accident, Poh et al. (2018) studied a construction database of a Singaporean company and were able to predict outcome of several input features as a near miss, major or minor accident with using same techniques as Goh and Ubeynarayana. They concluded random forest are the most precise one among these models. In another study by Kang and Ryu (2019), more than 70% of accuracy were obtained in prediction of construction accidents in Korea.

Davoudi Kakhki et al. (2019) used supervised SVM machine learning algorithm with linear, Radial Basis Function, Naïve Bayes, Boosting and quadratic kernels for training of 33,000 incidents taken between 2008 and 2016 in order to predict severity outcome of the work-related accidents in agribusiness incidents. Employees' compensation claims were used to extract and classify overall cost of accidents. Input

features include affected body part, body group, nature of injury, nature group, cause of injury, cause group, age and tenure of injured workers. Model use these inputs and give output class of accident as severity of injury with 90-98% precision. They stated that machine learning is critical in learning patterns of work-related injuries in agribusiness industry and can contribute to deep understanding of accidents nature thus assist health and safety professionals in achieving better intervention efforts.

With using Support Vector Machine (SVT), Classification Trees (CT), Bayesian Networks (BN) and Extreme Learning Machine (ELM or feedforward neural network) techniques for classification, Matías et al. (2008) analyzed 148 fall on floor level accidents taken place between 2003 and 2008, and found out that predictive capacity of Bayesian Networks are higher than other techniques in fall on floor level accidents. They suggest that Bayesian Networks is more precise in predicting falls and identifying causes of falls on floor level; therefore, this technique enables better monitoring of management system deviations and reduced work-related fall accidents.

Sanmiquel et al. (2018) analyzed approximately 56,000 accidents from both underground and surface mines with Bayesian Networks and stated that large accident data base can be interpreted with data mining techniques to find factors and root causes of the accidents. Thus, these techniques could improve corrective measures in terms of effectiveness and customization.

Neural Networks are also used in predicting outcome of occupational accidents and have several advantages such as learning from data, using incomplete data and fault tolerance. Therefore, many researchers use backward propagation algorithm of neural networks to forecast outcomes of events (Sarkar, 2018).

Ayhan and Tokdemir (2019) stated that ANN is successful in predicting the outcome

of 84% incident within 90% confidence, and 0.736 reliability.

He et al. (2010) used backward propagation algorithm of ANN and EEM to categorize deep mines according to their level of outburst (coal and gas) danger with correlating danger class and twelve factors affecting outburst. They stated that the technique is fast and precise in calculating of class of danger in coalmines.

In order to find an early-warning system for construction workers exposed to heat stress in climates with high humidity and temperatures, Yi et al. (2016) proposed a system that can warn workers against heat stress. They applied ANN method to predict heat strain level of workers who is involved in labor intensive and outdoor tasks. They proposed a system, which collects real time information from field via wearable sensors and smart phones, and collected information is used to assess the risk of heat stress. System was trained with information regarding environment, worker and task being done. Levenberg–Marquart back propagation ANN method was used and system is able to predict rating of perceived exertion (RPE) with high precision. The validity of predictive model is confirmed with other models such as R^2 and MAPE.

2.2.1 Types of Machine Learning Systems

Lots of different kind of ML are known and it is practical to make a categorical classification or regression according to type of supervision, online or batch learning, instance or model-based learning (Géron, 2017).

2.2.1.1 Supervised/Unsupervised Learning

Supervised learning has roots to early psychological studies, in which subject asked to predict a class membership and subjects' prediction were checked against correct

answer. After some trials, subjects were able to figure the proper rule or the concept out and can predict correct classification (Bruner, Goodnow, & Austin, 1956). Based on the existence of conjunctive attributes, researchers presumed class categories as negative (0) or positive (1) in most studies. In addition to conjunctive learning concepts, human can learn disjunctive concepts, in which sufficient number of features or attributes are not given for concept membership. These two concepts are not equally considered because some categories are more predictable than others. Conjunctive features are easy and fast to be spotted than disjunctive features. Response time is highly dependent on the level of category member typicality. This have led to organization of a system that gather several simple concepts in one classifier such as Concept Learning Systems (Hunt, Marin, and Stone, 1966) or probabilistic concepts assigning weight or probability to attributes (Smith and Medin, 1981).

Many machine learning concepts originated from concepts underlying human learning such as decision tress which is derived from CLS (Quinlan,1986). Conjunctive learning concepts may begin from an initial concept containing attributes of initial nonnegative instance and features are omitted according to succeeding negative instances (Mitchell, 1982; Dietterich and Michalski, 1983; Hayes-Roth and McDermott, 1978; Vere, 1980; Pazzani and Sarrett, 1990) or it may start from generalization rule without any condition and negative instances are eliminated with newly added conditions (Langley, 1987). This is known as the direction of learning or finding concept that explain something meaningful (Simon & Lea, 1974; Mitchell, 1982). Searching control strategy are various for different type of learning systems. Searching could be carried out from specific-to-general (inductive) or vice versa (Anderson & Kline, 1979; Mitchell,1982).

Supervised learning, unsupervised learning, semi-supervised learning, and Reinforcement Learning are for types of human supervision in machine learning.

a) In supervised learning, algorithm is fed with the labeled training data including desired solutions. Classification is a typical example of supervised learning. Labeling email as spam or not spam class is an example of supervised learning which facilitates classification of new emails either spam or ham.

All supervised learning algorithms aim to learn finding output variables, y with using input variables, x through a mapping function, $y = f(x)$. The challenge is to find the appropriate mapping function such that it satisfactorily predicts output variable of new instance. This is known as supervised learning because the correct answer is already known, and learning is terminated when the algorithm achieves this correct answer. Supervised learnings algorithms are grouped as regression that outputs real vales and classification that outputs category memberships (Brownlee, 2016).

k-Nearest Neighbors, Linear and Logistic Regression, Support Vector Machines (SVMs), Decision Trees (DT) and Random Forests, and Neural networks are most common supervised learning algorithms (Géron, 2017 and Brownlee, 2016).

b) In unsupervised learning data sets are unlabeled and system attempts to learn without supervision. Only input variables, x are available in this type of learning and corresponding output variables, y are missing. Underlying structure or distribution are sought for more details of the data sets with modelling and due to missing correct answer, it is known as unsupervised learning. Algorithms develop data sets' structure and present these structures. Unsupervised learning algorithms can be categorized as clustering and association problems (Brownlee, 2016).

Clustering, Visualization and Dimensionality reduction, and Association Rule learning are some of the unsupervised learning algorithm (Géron, 2017 and Brownlee, 2016).

Clustering algorithms can be useful in grouping a set of data without any supervision. Numerical taxonomy could be one of the oldest unsupervised learning research area in this category (Everitt, 1980). Clustering algorithms are effective in grouping instances without any category assigned (Witten and Frank, 2005). K-Means, Hierarchical Cluster Analysis (HCA), Expectation Maximization (EM) are some of the most significant unsupervised clustering algorithms (Witten and Frank, 2005; Géron, 2017).

Similar observations are grouped or clustered under same category and outliers are set apart from these categories in clustering. For example, dendrograms allocate objects to clusters and shows the hierarchical relationships between these groups. Similarity between groups are computed such that they can be combined. These similarities between groups can be computed with measures such as Euclidian distances, $[\sum_{k=1}^K (x_{ik} - x_{jk})^2]^{1/2}$, where x_{ik} and x_{jk} are the values of feature k for instances i and j , respectively; the closer the values of individual attributes, the lesser the denominator of the inverted distance, and the larger the similarity (Fisher and Pazzani, 1991). The strategy explained above is used by almost all clustering algorithms with variations in the determination of how similar groups are. (Everitt, 1980; Jain & Dubes, 1988). For example, k-means are known as the classic clustering method in which a parameter, k is predefined for maximum number of clusters as desired and k random class center is randomly chosen. All of the data points are assigned to cluster centroid and then similarity is calculated with Euclidian distances. Calculation take place for each new random class center for minimizing total squared distances from centroids until the stabilization of class centers (Witten and Frank, 2005).

Visualization algorithms aims to represent output of the algorithm as two- or three-dimensional plots with preserving as much structure as possible in order to understand organization of the data and patterns. On the other hand, in dimensionality reduction, aim is simplification of data with keeping as much information as possible such as merging correlated attributes in one.

Principal Component Analysis (PCA), Kernel PCA, Locally Linear Embedding (LLE), t-distributed Stochastic Neighbor Embedding (t-SNE), Linear Discriminant Analysis (LDA) are some of the unsupervised algorithms in this type.

Principal Component Analysis (PCA) is one of the most famous dimensionality reduction algorithms based on identification of closest hyperplanes to data sets, projection data on the hyperplanes and selecting the axis preserving maximum amount of variance. In other words, PCA use a specified coordinate system depending on the cluster of the instances as following: first axis is placed in the direction of the largest variance of the instances in order to maximize the variance alongside the axis and second is selected such that two axes are orthogonal. Identification of the axis with largest amount of variance between many axes orthogonal to each other are the main objective in PCA. For multidimensional representations, there are more alternatives for perpendicular axis to first chosen axis. Principal component is the unit vector defining the specified axis and all principal components are include in principal component matrix, V^T (Géron, 2017; Witten and Frank, 2005).

Locally Linear Embedding (LLE) aims to keep local representation of the training data sets in low dimensional representation so that it is less spoiled with computing the linear relation between training data points with its nearest neighbors. LLE performs well in twisted manifolds (Géron, 2017).

In Multidimensional Scaling (MDS) goal is to decrease dimensionality without interrupting the actual distance between instances in low dimensions as well. Dissimilarities between instances are illustrated as distances in MDS, which is as close as practical to high dimension representation (de Leeuw, 2001).

Isometric mapping or Isomaps are originated from spectral theory and objective is to maintain the geodesic proximities in low dimensional representations using non-Euclidian metrics (Hannachi and Turner, 2013).

t-Distributed Stochastic Neighbor Embedding (t-SNE) is mostly used for visualization of high dimensional representation and decreases number of dimensions with preserving similar points near and dissimilar points separate (Géron, 2017).

Linear Discriminant Analysis (LDA) algorithms are used to find out the most discriminatory axes between classes which later used to project data on hyperplane composed of these axes for the classification with more than two classes. It is useful in reducing dimensionality of the class representation before application of any other classifier. LDA assumes data as Gaussian and variance of each features are equal for simplification and use Bayes theorem for predicting probabilities of the output variables with using input probabilities of each class. (Géron, 2017 and Brownlee, 2016).

In Association Rule Learning, goal is to find any association between features or attributes with digging down into large data sets. A good example for this learning is finding association between any market product that sell together such as beer and chips. Apriori and Eclat algorithms are two of the Association Rule Learning algorithms (Géron, 2017).

c) Semisupervised Learning algorithms use training data with partial labels and lots of missing labels. In semisupervised learning, some of the input and output variables, x, y are labeled. Both supervised and unsupervised algorithms are used in combination in this learning. This technique can be used to predict incomplete output or output variables can contribute to figuring out the unlabeled input instances. For instance, Deep Belief Networks (DBNs) are trained on unsupervised learning, Restricted Boltzman Machines (RBMs) and then supervised learning techniques are applied for tuning (Géron, 2017 and Brownlee, 2016).

d) In Reinforcement Learning, learning system observe environment, choose an action and performs it and gets a reward or penalty in the end. System then form a policy to get highest number of reward or no reward with time by learning.

2.2.1.2 Online and Batch Learning Systems

Incremental learning capacity of the system from incoming data defines whether the learning is batch learning or online learning. Batch or offline learning is not capable of incremental learning so that all available data are fed to system for training in the beginning and system runs without any additional learning data. For new incoming data, system need to be retrained as a new version from all data sets including new data. This makes batch learning time consuming and it require large computing resources. However, in online learning, data feeding takes place in real time and in

sequence and it is faster and cheaper. This system is advantageous for limited computing resources and huge data sets not fitting main memory of the machines.

2.2.1.3 Instance and Model-Based Learning

Another categorization method in ML is generalization that means to generalize new instances according to trained instances. Instance-based learning and model-based learning are two main approaches to generalization. In instance-based learning, system generalizes new instances according to measure of similarity derived from examples such as number of words in spam mails, however, generalization take place according to a selected model built out of example sets to make prediction in model-based training. Training instances are stored as verbatim in instance-based learning and a distance function helps to identify the point within training set which is nearest to unknown test example. This nearest point is used to predict the class of the unknown test instance. The issue is determining the distance function, it is problematic especially features with values. Euclidian distance function is common in instance-based learning. The distance between an instance with feature values $a_1^{(1)}, a_2^{(1)}, \dots, a_k^{(1)}$ and one with values $a_1^{(2)}, a_2^{(2)}, \dots, a_k^{(2)}$ is determined with the Equation-1.

Equation-1. Euclidian Distance Function

$$d(a^{(1)}, a^{(2)}) = \sqrt{(a_1^{(1)} - a_1^{(2)})^2 + (a_2^{(1)} - a_2^{(2)})^2 + \dots + (a_k^{(1)} - a_k^{(2)})^2}$$

Normalization should take place in Euclidian distance function because direct use of Euclidian function can lead to fading of the weight of some attributes by other larger scale attributes. This is because of the different attributes are computed on various scales.

Equation-2. Normalization for Euclidian Distance Function

$$a_i = \frac{v_i - \min v_i}{\max v_i - \min v_i}$$

Where v_i is the actual value of feature i , and the max and min are taken over all instances in the training data set (Witten and Frank, 2005).

2.2.1.4 Parametric vs Nonparametric Machine Learning Algorithms

Parametric learning algorithms make some strong assumptions about mapping function such as normal distribution of the instances. These algorithms rely heavily on any distribution however nonparametric algorithms do not rely on any distribution. Parametric machine learning algorithms are the algorithms that simplify the function to a known form. Parametric algorithms may include nonparametric equivalents. Assumption made in parametric algorithms may restrict the learning from data. Since nonparametric machine learning algorithms doesn't make any assumption, they can learn from any training data sets

Parametric machine learning algorithms are composed of two steps, namely, selection of function form and learning of the coefficients from training data sets for the function. Linear Regression, Perceptron and Linear Discriminant Analysis are Parametric machine learning algorithms. Those are simple, fast and doesn't require more data even if data is poor, however, they have limitation such as being limited

with specified form, unsuitable for complex issues and fit poorly to mapping function.

k-means, Neural networks, decision trees and support vector machines are some of the nonparametric machine learning algorithms. They have disadvantages such as requiring large data sets for predicting mapping function, being slow due to number of parameters and overfitting risk but these algorithms are flexible and have high performance models (Brownlee, 2016).

2.2.2 Training Models

2.2.2.1 Linear Regression

Linear regression models predict outcome with weighted total of input variables and intercept term as shown in Equation-3 or in vectorized form in Equation-4.

Equation-3 Linear Regression Model Prediction

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- $\hat{y}$ is the predicted value;
- n is the number of attributes;
- x_i is the i^{th} attribute value.
- θ_j is the j^{th} model parameter (containing the bias term θ_0 and the attribute weights $\theta_1, \theta_2, \cdots \theta_n$)

Equation-4 Vectorized Form of Linear Regression Model Prediction

$$\hat{y} = h_{\theta}(\mathbf{x}) = \theta^T \cdot \mathbf{x}$$

- θ is the model's parameter vector, including the bias term θ_0 and the attribute weights $\theta_1, \theta_2, \dots, \theta_n$
- θ^T is the transpose vector of θ
- $\theta^T \cdot \mathbf{x}$ is the dot product of θ^T and $\mathbf{x}$
- $\mathbf{x}$ is the attribute vector of instances, including $x_0, x_1, x_2 \dots x_n$ with always $x_0 = 1$
- h_{θ} is the hypothesis function, using the model parameters θ

Parameters are set to measure how well training data sets fit to linear regression model and model's performance is tested with Root Mean Square Error. The value of θ or intercept is sought that minimizes RMSE or simply MSE cost function. Normal equation (Equation-5) or Gradient descent optimization algorithm for large data sets is used to find out the value of θ that minimizes MSE cost function (Equation-6).

Equation-5 Normal Equation

$$\hat{\theta} = (X^T \cdot X)^{-1} \cdot X^T \cdot \mathbf{y}$$

- $\hat{\theta}$ is the value of θ that minimizes the MSE
- $\mathbf{y}$ is the vector of target values including $y^{(1)}$ to $y^{(m)}$

Equation-6 MSE cost function for a Linear Regression model

$$\text{MSE}(\mathbf{X}, h_{\theta}) = \frac{1}{m} \sum_{i=1}^m (\theta^T \cdot \mathbf{x}^{(i)} - y^{(i)})^2$$

2.2.2.2 Gradient Descent

In gradient descent algorithm a random θ value is selected and improved gradually or step-by-step to converge to minimum value of MSE according to the slope. Size of the steps define the speed of the algorithm. if the number of steps is too many or rate is so small, the number of iterations will be many and will take a long time. On the other hand, high learning rate means divergence of algorithm and end up with a poor solution. Another pitfall in gradients descent algorithm is to end up with local minimum instead of global minimum however MSE's being convex function eliminates this. Partial derivative of the cost function is used to find out the slope and the sign of the slope is used whether to increase or the decrease the value of θ to reach the global minimum of cost function. This is called *batch gradient descent* algorithm which is solve due to use of all training data sets in every learning steps. Another method for finding θ value for MSE function to converge is *stochastic gradient descent (SGD)*. SGD uses one instance in each iteration to calculate minimum value of MSE that makes it faster and advantageous for large data sets. Another advantage of SGD over batch gradient descent is offering higher possibility to find the global minimum and to avoid local minimum (Géron, 2017 and Brownlee, 2016).

2.2.2.3 Polynomial Regression

When multiple variables exist and data don't fit to linear regression model, polynomial regression model can be helpful in finding association between variables. Polynomial models are working just like linear models, but the powers added to features (Géron, 2017).

2.2.2.4 Naïve Bayes Classifiers

NBC is a supervised learning technique based on Bayes Theorem. The Bayes predictor use observations or priori $(X_1, X_2, \dots, X_d)$ as input and predict the output class, $(Y = 1, \dots, K)$ with highest likelihood. Number of the classes is K and probability is calculated with Bayes' Theorem for the k th observation. (Lowd and Domingos, 2005).

Equation-7 Bayes' Theorem.

$$P(Y = k|X_1, X_2 \dots X_d) = \frac{P(Y = k)P(X_1, X_2 \dots X_d|Y = k)}{P(X_1, X_2 \dots X_d)}$$

Where X is discrete and where denominator is given by

$$P(X_1, X_2 \dots X_d) = \sum_{k=1}^K P(Y = k)P(X_1, X_2, \dots, X_d|Y = k)$$

The prior for Y in logistic regression classifier is given as $\beta_0 = \ln(P(Y = 1)/P(Y = 0))$ to be estimated (Clarke and Clarke, 2018; Breiman et al., 1998).

Equation-8 Bayes Classifier

$$\hat{Y}_i(x_i) = \underset{y}{\operatorname{argmax}} P(Y = y)P(x_{1i}, x_{2i}, \dots, x_{di}|Y = y)$$

In spite of the fact that situations are assumed to be independent in Naïve Bayes Classifier indeed they are dependent, NBCs are present high accuracy (Phark et al., 2018).

2.2.2.5 Regularized Liner Models

Ridge regression

Ridge regression adds a regularization term, $\alpha \sum_{i=1}^n \theta_i^2$, to linear regression model's MSE cost function during training session in order to push model to fit to the data and to keep weights as small as possible. The hypermeter, α , controls model regularization.

Equation-9 Ridge Regression cost function

$$J(\theta) = \text{MSE}(\theta) + \alpha \frac{1}{2} \sum_{i=1}^n \theta_i^2$$

Lasso Regression

Lasso regression adds a regularization term, $\alpha \sum_{i=1}^n |\theta_i|$, to linear regression model's cost function. Lasso regression algorithm eliminates weight which has less importance and set them to zero. Attribute selection is automatic in Lasso regression and sparse model is given as output. Lasso is preferred when some features are not needed or useless.

Equation-10 Lasso Regression cost function

$$J(\theta) = MSE(\theta) + \alpha \sum_{i=1}^n |\theta_i|$$

Elastic Net

Elastic net regression is the mixture of both Lasso and Ridge regression and adds regularization term to MSE cost function with a ratio parameter, r (Géron, 2017).

Equation-11 Elastic Net cost function

$$J(\theta) = MSE(\theta) + r\alpha \sum_{i=1}^n |\theta_i| + \frac{1-r}{2} \alpha \sum_{i=1}^n \theta_i^2$$

2.2.2.6 Logistic Regression

Logistic Regression (also called Logit Regression) is used to give a probabilistic estimation of the class of the instances. Logit regression doesn't output results directly as in linear regression but with logistics of this result. An estimation with

more than 50% probability are considered positive class and labeled as “1” or else considered negative class and labeled as “0”.

Equation-12 Vectorized form for Logistic Regression model estimated probability

$$\hat{p} = h_{\theta}(\mathbf{x}) = \sigma(\theta^T \cdot \mathbf{x})$$

- $\sigma, \sigma(t) = \frac{1}{1+e^{-t}}$, is a sigmoid function and outputs a number between 0 and 1
- $\hat{y} = \begin{cases} 0 & \text{if } \hat{p} < 0.5 \\ 1 & \text{if } \hat{p} \geq 0.5 \end{cases}$

The parameter vector θ are set such that the model estimates high probabilities for positive instances ($y = 1$) and low probabilities for negative instances ($y = 0$) The which is the main objective of the training. This concept is realized by the cost function shown in Equation-13 for a single training instance $\mathbf{x}$.

Equation-13 Cost function of a single training instance

$$c(\theta) = \begin{cases} -\log(\hat{p}) & \text{if } y = 1 \\ -\log(1 - \hat{p}) & \text{if } y = 0 \end{cases}$$

For all training data sets, logistics regression cost function is given below, and its derivative is used to find θ which minimizes this function:

Equation-14 Logistic Regression cost function (log loss)

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(\hat{p}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{p}^{(i)})]$$

Equation-15 Logistic cost function partial derivatives

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^m (\sigma(\theta^T \cdot \mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)}$$

Multiple Logistic Regression or Softmax Regression

Softmax regression (or multinomial logistic regression) enables handling of multiple classes, $y^{(i)} \in \{1, \dots, K\}$ instead of binary labels $y^{(i)} \in \{0, 1\}$ as in logistic regression.

Given an instance $\mathbf{x}$, the Softmax Regression model first computes a score $s_k(\mathbf{x})$ for each class k , then estimates the probability of each class (one class at a time i.e. meaning capable of mutually exclusive classes) by applying the Softmax function (also called the normalized exponential) to the scores. The equation to compute $s_k(\mathbf{x})$ is similar to the equation for Linear Regression prediction. Each class has its own dedicated parameter vector, θ_k , which are kept as rows in parameter matrix, Θ (Géron, 2017).

Equation-16 Softmax score for class k

$$s_k(\mathbf{x}) = \theta_k^T \cdot \mathbf{x}$$

For each instance $\mathbf{x}$, score of every class is calculated and normalized exponential (Equation 17) is used to reach a probabilistic estimation, $\hat{p}_k$ that training instances belongs to class k .

Equation-17 Softmax function

$$\hat{p}_k = \sigma(\mathbf{s}(\mathbf{x}))_k = \frac{\exp(s_k(\mathbf{x}))}{\sum_{j=1}^K \exp(s_j(\mathbf{x}))}$$

- K is the number of classes.
- $\mathbf{s}(\mathbf{x})$ is a vector comprising of the scores of each class for the instance $\mathbf{x}$
- $\sigma(\mathbf{s}(\mathbf{x}))_k$ is the estimated probability that the instance $\mathbf{x}$ belongs to class k given the scores of each class for that instance

Highest estimated class probability is predicted with Softmax Regression classifier (Equation 18) using the value of k which maximizes $\sigma(\mathbf{s}(\mathbf{x}))_k$ as predicted in logit regression.

Equation-18 Softmax Regression classifier

$$\hat{y} = \underset{k}{\operatorname{argmax}} \sigma(\mathbf{s}(\mathbf{x}))_k = \underset{k}{\operatorname{argmax}} s_k(\mathbf{x}) = \underset{k}{\operatorname{argmax}} (\theta_k^T \cdot \mathbf{x})$$

The aim is to train a model that predicts higher probabilities for the target class and lower probabilities for other classes. Minimizing the cost function called the cross entropy (Equation-19), should end up with the objective mentioned above because the training model is penalized in case of low probability estimations for a target class. The fitting of estimated class probabilities with target classes are measured with cross entropy.

Equation-19 Cross entropy cost function

$$J(\Theta) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \log(\hat{p}_k^{(i)})$$

- $y_k^{(i)}$ is equal to 1 if the target class for the i^{th} instance is k ; otherwise, it is equal to 0.

The gradient vector of the cost function with regards to θ_k is given by Equation 20. For every class the gradient vector is computed, then any optimization algorithm is used to find the parameter matrix, Θ that minimizes the cost function (Géron, 2017 and Brownlee, 2016).

Equation-20 Cross entropy gradient vector for class k

$$\nabla_{\theta_k} J(\Theta) = \frac{1}{m} \sum_{i=1}^m (\hat{p}_k^{(i)} - y_k^{(i)}) \mathbf{x}^{(i)}$$

2.2.2.7 Support Vector Machines

Being a robust and adaptable ML model, Support Vector Machine (SVM) is able to realize linear and nonlinear classification, regression, and detect outliers. It is suitable for classification of complex, small to medium size data sets and two type will be explained, namely, linear SVM classification and nonlinear SVM classification. Adding more training instances or support vectors do not disturb decision boundary due to being determined or supported. SVM is sensitivity to scaling of features. The powerful theoretical fundamentals of SVM, which is evolved from theory to experiments and applications contrary to neural networks, did not ensure its being appreciated initially because it was seen impracticable and irrelevant for practical applications. After successful application of SVMs in recognizing digits, categorization of digits and other fields, it was taken seriously (Kecman, 2005).

In order to apply nonlinear class boundaries, SWMs use linear models with transforming actual space into another new space for providing data representation separable with linear boundaries. (Witten and Frank, 2005 and Cervantes et al., 2015).

The maximum margin hyperplane is a linear model derived from SVM algorithms and it is the best fitting plane that training instances are linearly separable. The instances that are nearest to the maximum margin hyperplane is called support vectors. For each class at least one support vector exists, and these vectors are used to construct the maximum margin hyperplane. Algorithm of a hyperplane splitting two class instances are given below (Witten and Frank, 2005; Suárez Sánchez et al., 2011).

Equation-21 Maximum margin hyperplane function for two attribute case

$$x = w_0 + w_1 a_1 + w_2 a_2$$

- a_1 and a_2 are attribute values
- w_0 , w_1 , and w_2 , are the weight to be learned

In terms of support vectors, class values, y are written as 1 or -1 and the maximum margin hyperplane is given as:

Equation-22 Maximum margin hyperplane function for two attribute case in term of support vectors

$$x = b + \sum \alpha_i y_i \mathbf{a}(i) \cdot \mathbf{a}$$

- i is the support vector
- y_i is the class value of training instance $\mathbf{a}(i)$

- b and α_i are the numeric parameters to be determined

Soft Margin Classification

In linearly separable data sets, hard margin classification is employed, and it is very sensitive to outlier. If data set is not separable and a hard margin cannot be found, a more flexible model, soft margin classification, can be employed. The aim in soft margin classification is to choose classifier margin as large as possible and as low number of misclassifications of the instances (less outliers). The balance between this is controlled by a hyperparameter called C . Overfitting can be avoided with decreasing this hyperparameter that means enlarging classifier width and increasing margin violations (Géron, 2017).

Nonlinear SVM Classification

If data sets are linearly inseparable, nonlinear SVM classifications are more useful for these kind of data sets. Adding polynomial features to nonlinear data sets can even enable data sets linearly separable in some cases. For more complex data sets, high polynomial degrees are added which enables multiple features. High degree polynomials are good at representing multiple features but slow down the model. Therefore, applying mathematical techniques such as polynomial kernels will tackle these odds. Polynomial kernel tricks make getting the same results possible without adding many polynomial features to data sets in fact.

Another method to solve nonlinear issues in machine learning is using kernels, such as Gaussian Radial Basis Function that makes adding new features needles, therefore, speeds up the models (Witten and Frank, 2005).

SVM Regression

SVM algorithms is useful in linear and nonlinear regression. In SVM classification aim was to fit largest classifier margin with minimum misclassification between two classes, however, in SVM regression, goal is to fit as many training instances as possible inside classifier margin while keeping margin violations at minimum. Hyperparameter ε is used to control width of classier margin. Model prediction is no affected by the training instances which are in between margins because this model is known to be ε -insensitive. SVM uses a decision function to predict the class of a new instances. Result of the decision function defines whether the predicted class, $\hat{y}$, is the positive or the negative class. Training a linear SVM classifier means finding the value of $\mathbf{w}$ and b that make this margin as wide as possible while avoiding margin violations (hard margin) or limiting them (soft margin). The weight vector, $\mathbf{w}$, is inversely related with the width of margin. For larger margins, $\|\mathbf{w}\|$ should be minimized (Géron, 2017; Witten and Frank, 2005).

Equation-23 Linear SVM classifier function

$$\hat{y} = \begin{cases} 0 & \text{if } \mathbf{w}^T \cdot \mathbf{x} + b < 0, \\ 1 & \text{if } \mathbf{w}^T \cdot \mathbf{x} + b \geq 0 \end{cases}$$

Equation-24 Decision function

$$\mathbf{w}^T \cdot \mathbf{x} + b = w_1x_1 + w_1x_1 + \dots + w_1x_1 + b$$

2.2.2.8 Decision Trees

Decision trees (DT) learning algorithms are capable of regression, classification and multioutput tasks and can predict class probabilities. Basic components of one of the most robust algorithms, random forests, are decision trees. DT requires little data preparation and does not need any attribute scaling. Witten and Frank (2005) define

decision trees metaphorically as divide and conquer method. First feature is employed for splitting and this make top node. Algorithm runs recursively and finds child nodes with following features for splitting. Decision trees are composed of nodes, leaf nodes, node's sample counts, node's classes, node's *gini* (impurity, entropy is also used for impurity) and node's values. In comparison with other models, DTs make few assumptions and therefore regularization hyperparameters such as depth, samples, number of leaf nodes and features needed to be set to avoid overfitting. Pruning of the decision trees also could be employed after uninterrupted first training for stopping development of the tree. Whether prepruning (forward pruning) or postpruning could be employed according to the strategy. (backward pruning) (Kuhn and Johnson, 2013 and Géron, 2017).

A node's *gini* attribute or entropy impurity is employed to measures node's impurity: a node is "pure" (*gini*=0) if all training instances it applies to belong to the same class.

Equation-25 Gini Impurity function

$$G_i = 1 - \sum_{k=1}^n p_{i,k}^2$$

Equation-26 Entropy Impurity function

$$H_i = - \sum_{\substack{k=1 \\ p_{i,k} \neq 0}}^n p_{i,k} \log(p_{i,k})$$

- $p_{i,k}$ is the ratio of class k instances among the training instances in the i^{th} node

Classification and Regression Tree Algorithms

Decision Trees are trained with Classification and Regression Tree Algorithms, CART. First of all, algorithm separates training data sets into two subsets with using a single attribute k and a threshold, t_k . Purest subsets are sought for different pairs of (k, t_k) and the pair producing this purest subset are selected for splitting data sets into two. Same logic is used to split in next steps and terminated in the final depth with the. Algorithm attempts to minimize CART cost function for classification (Equation-27) for minimum gini value. However, algorithm tries to minimize Mean Square Error function for regression (MSE) (Equation-28) with splitting training data sets in a way to keep MSE at minimum. Being sensitive to training set rotation due to orthogonal boundaries and very sensitive to even little variations in training data sets, DTs can be controlled by PCA for better orientation of the training data sets and hyperparameters for avoiding different models from same training data sets. These instabilities also can be avoided with Random Forests Models taking average of predictions over many trees (Bevilacqua et al., 2008 and Brownlee, 2018).

Equation-27 CART cost function for classification

$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right} \quad \text{Equation-21}$$

Where;

$$\begin{cases} G_{left/right} , \text{gini score of left/right subset} \\ m_{left/right} , \text{number of instances in left/right subset} \end{cases}$$

Equation-28 Mean Square Error function for regression (MSE)

$$J(k, t_k) = \frac{m_{left}}{m} MSE_{left} + \frac{m_{right}}{m} MSE_{right}$$

Where;

$$\begin{cases} MSE_{node} = \sum_{i \in node} (\hat{y}_{node} - y^{(i)})^2 \\ \hat{y}_{node} = \frac{1}{m_{node}} \sum_{i \in node} y^{(i)} \end{cases}$$

C5.0 Classification Algorithm

C5.0, which was developed by Quinlan in 2007, is a new DT algorithm developed from C4.5 DT algorithm and use all functions of C4.5, however, it offers boosting and cost sensitive trees as a new feature. Given samples are taken as root of DT, ratio of information gain is then calculated and attribute with maximum ratio of information is selected as splitting feature. The selected split attribute is used for creating a node for being indicator of the node. Branches for each split attribute value are created and data set are divided into several subsets accordingly which corresponds to the number of the testing attribute's values. These steps are run until one of the three conditions are met, which are no instances left out the subsets, no attributes left meaning each subset attributes are executed, and each test attributes left of the instances in the subset equals to the same value, but the samples still belong to different classes. Pruning used in C4.5 is EBP algorithm (Quinlan, 2007; Pang and Gong, 2005). The difference between C5.0 algorithm and CART algorithm is the number of branches in the tree. Latter applies only a binary split, however, C5.0 applies multiple branches. Another difference is that the CART algorithms use validation data to prune overgrown decision trees by using training data while C5.0 uses training data for both growing and pruning the decision tree (Shmueli, Patel, & Bruce, 2007).

Being one of the most important advantages of C5.0 over C4.5, boosting algorithm assign weights to the instances according to the level of significance. Higher weight values affect the decision more. Initial values of the weights are equal to each other and later adjust in a way that the learner focus on the misclassified instances by the decision tree model and weight of these instance are incremented (Pang and Gong, 2005).

2.2.2.9 Ensemble Learning and Random Forests

The objective of ensemble learning is to combine several machine learning models (i.e. meta-algorithms) and thus give improved outputs with reduced noisiness, bias and variance. Ensemble learning eliminates weakness of some models and offers new perspective. Ensemble learning is based on a method called wisdom of crowd that aggregate prediction of the group predictors. For instances, predictions of all individual decision trees or different ML models are obtained class with more votes are predicted. Hard voting classifiers that aggregates the predictions of each classifiers and predict the class or soft voting classifiers that predicts with highest class probability are used (Géron, 2017).

In order to get a different set of classifiers, different algorithm can be employed. For Another way for diverse set of classifiers is using random subset sampling for training data with same algorithm. Random sampling being realized with replacement is called bootstrap aggregating (i.e. bagging) and random sampling being realized without replacement is called pasting. Aggregation of the prediction of all predictors are derived with using statistical mode function for classification or average for regression. Aggregation of individual predictors decrease bias and variance in comparison with single predictor (Witten and Frank, 2005; Clarke and Clarke, 2018 and Miner et al., 2012).

Ensemble of decision trees are known as random forests and use same hyperparameters as decision trees. Some weak learner can be combined with ensemble learning which is known as hypothesis boosting (Adaptive Boosting or AdaBoost and Gradient Boosting) which trains each predictor sequentially with instance weight updates and corrects its predecessor (Freund and Schapire, 1997).

Bagging

Easiest way of collecting all individual prediction of different models under one prediction is to take weighted vote (i.e. calculate weighted average) for each individual prediction. This approach is adopted by bagging or bootstrap aggregating.

Bagging just resamples the original training data set in place of obtaining independent datasets from the domain. Resampled datasets are non-identical to each other however they are depended because of being originated from one dataset. This combined model produced by bagging shows better performance than the single model of initial training data set. Bagging use voting for classification and weighted averaging for regression.

Boosting

Boosting use voting for classification and weighted averaging for regression similar to bagging and use same model types in order to ensemble outputs of different individual models. Performance of the initially built models affects all new models, i.e., being iterative in contrary to bagging. Boosting promotes new models for instances operated correctly and boosting weights a model's impact by its performance instead of assigning equal weight to all models as in bagging (Clarke and Clarke, 2018).

AdaBoost

Algorithm trains a first base classifier, then outputs are predicted on training data sets in AdaBoost. Weight of the inaccurately classified instances are then increased, and training takes place with these updated weights. As a result, a second classifier is trained with this way and new predictions are made. This process is continued recursively until desired number of predictors. In each step, learning rate could be whether increased or decreased for better predictors. In Adaboost, initial weights $w^{(i)}$ of all instances are set to $\frac{1}{m}$, then first predictor is trained. Weighted error rate (WER), r_1 , is calculated on the training data sets and inputted to predictor weight equation (Equation 29) for finding predictor's weight α_j (Equation 30) (Géron, 2017).

Equation-29 WER of the j^{th} predictor

$$r_j = \frac{\sum_{i=1}^m w^{(i)} \mathbb{1}_{\hat{y}_j^{(i)} \neq y^i}}{\sum_{i=1}^m w^{(i)}}$$

Where, $\hat{y}_j^{(i)}$ is the j^{th} predictor's prediction for the i^{th} instance.

Equation-30 Predictor weight equation

$$\alpha_j = \eta \log \frac{1 - r_j}{r_j}$$

Predictor's weight is used to update weight of misclassified instances according to the weight update rule equation in next step and normalization of instance weight, i.e. divided by $\sum_{i=1}^m w^{(i)}$, is realized (Equation-31).

Equation-31 Weight update rule equation

for $i = 1, 2, \dots, m$

$$w^{(i)} \leftarrow \begin{cases} w^{(i)} & \text{if } \hat{y}_j^{(i)} = y^i \\ w^{(i)} \exp(\alpha_j) & \text{if } \hat{y}_j^{(i)} \neq y^i \end{cases}$$

In the final step of AdaBoost trains new predictor with new weight and repeats all steps until the desired number of predictors reached. All predictions of predictors are calculated and weighted with predictor's weight α_j . Majority of the weighted votes define the predicted class with the Adaboost predictions equation (Equation-32).

Equation-32 Adaboost predictions equation

$$\hat{y}(\mathbf{x}) = \underset{k}{\operatorname{argmax}} \sum_{\substack{j=1 \\ \hat{y}_j(\mathbf{x})=k}}^N \alpha_j$$

where N is the number of the predictors.

Gradient Boost

Working mechanism of Gradient Boost is same as AdaBoost, sequentially adding predictors to an ensemble, each one correcting its predecessor. However, instead of adjusting the weight of the instance at every iteration like AdaBoost does, Gradient Boost fit training dataset to a regressor model initially, then residual errors by first predictor are calculated and used to train second regressor. Residual errors of second predictor is calculated and used to train third regressor. In stochastic gradient boosting

, each tree is trained with a fraction of training samples which is specified by a hyperparameter called subsample (Géron, 2017; Clarke and Clarke, 2018).

Stacking

Stacking or stacking generalization objective is to aggregate predictions and input these predictions to a final predictor known as blender or meta learner and make a final prediction (Géron, 2017).

2.2.2.10 Artificial Neural Networks (ANN)

ANN were first introduced by McCulloch and Pitts in 1943 which was presenting working mechanism of biological neurons in animal brains with using propositional logic. After a long unnoticed era, ANN has now become one of the main interest of researchers thanks to availability of huge amount of data to ANN and tremendous capacity of recent computers. ANN were inspired by biological neurons (BNN) and proposed model by McCulloch and Pitts shares same logic as working mechanism of BNN. In ANN, an output is activated with at least one or more inputs (i.e. artificial neurons), which is the basic logic (Géron, 2017 and Haykin, 2005).

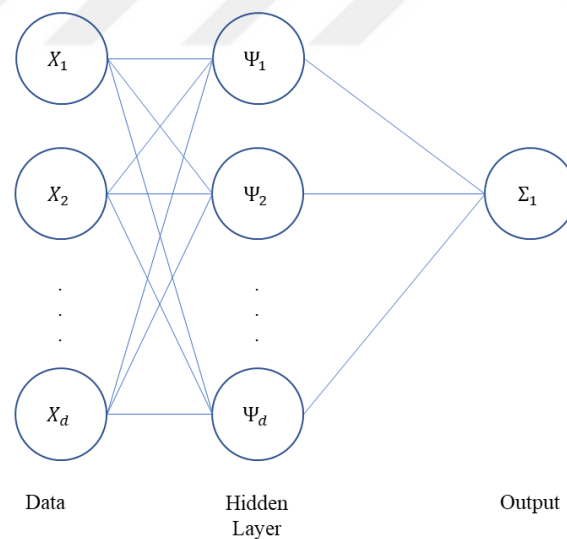


Figure 2.4 A connection graph of simple neural networks (Haykin, 2005)

Simplest form of neural networks is represented in Figure 2.4 for single hidden layered models and execution function for both regression and classification are given in Equation-33

Equation-33 A single-hidden-layer NN function

$$Y = Y(x_1, \dots, x_d) = \sum_{j=1}^r \beta_{2,j} \psi_j(\beta_{1,j,0}x_0 + \dots + \beta_{1,j,d}x_d) + \epsilon$$

- Y is output
- $x_1 = 1$
- $x_1, \dots, x_d$ are inputs
- ψ_j is transfer function at node j that such as sigmoid function, $\sigma(t)$
- $\beta_{1,j,0}x_0 + \dots + \beta_{1,j,d}x_d$ is value of linear combination (usually weights) from the hidden layer

For two hidden-layer neural networks function is given in Equation-34 and becomes more complicated.

Equation-34 A single-hidden-layer NN function

$$Y = Y(x_1, \dots, x_d) = \sum_{u=1}^S \beta_{3,u} \phi_u \left(\sum_{j_u=1}^{r_u} \beta_{2,j_u} \psi_{j,j_u}(\beta_{1,j_u,0}x_0 + \dots + \beta_{1,j_u,d}x_d) \right) + \epsilon$$

- S is the number of nodes in the second hidden layer
- ϕ_u is transfer function for second layer

Least square error function is used to find the values of weights, β .

Equation-35 SSE cost function for NN

$$SSE(w) = \sum_{i=1}^n ((y_i - Net(x_i, w))^2$$

- $Net(x, w)$ is the function represented by an NN

Dissimilar to other models, derivation of cost function $SSE(w)$ does not provide the value of the weight, w that equalize cost function to its minimum. Thus, a gradient descent technique called backpropagation is employed which takes derivative of the cost function to decide the direction of reduction (negative or positive) for new value $\hat{w}$ of w .

There are two layers of β 's, which are $\beta_{1,j,k}$ and the $\beta_{2,j}$. The derivatives of $SSE(w)$ with respect to these two classes of β 's are not same since derivatives with respect to the β 's in the hidden layer necessitate an extra use of the chain rule (Clarke and Clarke, 2018).

The results for the output layer for any $j = 1, 2, \dots, r$ are

Equation-36 Derivative of cost function with respect to β 's

$$\frac{\partial}{\partial \beta_{2,j}} SSE(w) = 2 \sum_{i=1}^n (y_i - Net(x_i, w))$$

and the results for the hidden layer for any $j = 1, 2, \dots, r$ and $k = 1, 2, \dots, d$ are

Equation-37 Derivative of cost function with respect to β 's (2)

$$\frac{\partial}{\partial \beta_{1,j,k}} SSE(w) = 2 \sum_{u=1}^r \beta_{2,u} \frac{\partial}{\partial \beta_{1,j,k}} \psi_u(\beta_{1,u,0}x_0 + \dots + \beta_{1,u,d}x_d) 2 \sum_{i=1}^n (y_i - Net(x_i, w))$$

Given an initial value value w_0 , a new value, $\hat{w}_1$ is produced by Equation-38.

Equation-38 Weight update function

$$\hat{w}_i = \hat{w}_{i-1} + \eta_i \nabla \frac{1}{2} SSE(\hat{w}_{i-1})$$

- ∇SSE is the gradient of the SSE evaluated at $\hat{w}_{i-1}$
- η_i is learning rate

The Perceptron

In 1957, Frank Rosenblatt invented one of the simplest ANN architectures, which is called, Perceptron. The perceptron is based on a linear threshold unit artificial neuron, LTU, in which inputs and output are numbers and each connection are linked with a weight. Weighted total of inputs is calculated (Equation-39), a step function (e.g. Heaviside step function and sgn, two of the most common step functions, Equation-40) is applied to the weighted total and final result is given as an output. Sum of the weights are computed with the function as shown below.

Equation-39 Weight total function

$$z = w_1x_1 + w_2x_2 + \dots + w_nx_n = \mathbf{w}^t \cdot \mathbf{x}$$

Equation-40 Output Function

$$h_w(\mathbf{x}) = \text{step}(z) = \text{step}(\mathbf{w}^t \cdot \mathbf{x})$$

Equation-41 Common step functions Heaviside and sgn

$$\text{heaviside}(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases} \quad \text{sgn} = \begin{cases} -1 & \text{if } z < 0 \\ 0 & \text{if } z = 0 \\ +1 & \text{if } z > 0 \end{cases}$$

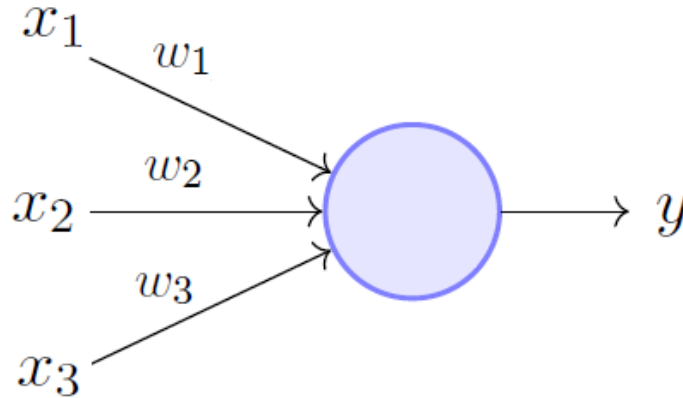


Figure 2.5 Perceptron Model (Minsky and Papert, 1969)

Training an LTU means finding out the right values of weights ($w_1 + x_2 + \dots + w_n$), which gives weather positive or negative output class according to linear combination of the inputs. Training logic in the perceptron are taking place according to a rule called Hebb's. In Hebb's rule, the connection weight between neurons is incremented when both neurons have the same output. A version of Hebb's rule used to train the perceptron; the weight of the synapses leading to right predictions are updated for every output neuron producing a wrong output. Perceptrons cannot learn from complex patterns due to having linear decision boundary and don't give class probabilities. Frank illustrated that, perceptron training algorithm would converge to a solution if the training instances were separable linearly, which is known as Perceptron convergence theorem (Géron, 2017 and Rosenblatt 1957).

Equation-42 Perceptron learning rule (weight update)

$$w_{i,j}^{(next\ step)} = w_{i,j} + \eta(\hat{y}_j - y_j)x_i$$

- $w_{i,j}$ is the connection weight between the i^{th} input neuron and the j^{th} output neuron
- x_i is the i th input value of the current training instance
- $\hat{y}_j$ is the output of the j^{th} output neuron for the current training instance
- y_j is the target output of the j^{th} output neuron for the current training instance
- η is the learning rate

Due to the limitations such as exclusive OR in perceptrons in single perceptron, an ANN known as Multi-Layer Perceptron, MLP, were introduced (Collobert and Bengio, 2004).

Multi-Layer Perceptron and Backpropagation

MLPs include an input layer, more than one hidden layer of LTU and an output layer of LTUs. Apart from output layer, each layer has bias neuron fully connected to the next layer. ANN with two or more hidden layers are known as DNN (deep neural networks). There were unsuccessful attempts in training MPLs until proposal of an algorithm called backpropagation in 1986 by Rumelhart et al. Network is fed with instances and output is computed in every consecutive layer. Network's output error is measured, and the contribution made by each last layer neuron to each output neuron's error is computed. Second step is to calculate error contribution of each neuron in the penultimate hidden layer to each output neuron in last layer. This continues until the algorithm reach the input layer. In other words, prediction is realized for each instance in backpropagation algorithm, error is measured, contribution by neurons to this error is measured and weight adjusted according to these contributions (i.e. Gradient Descent). Rumelhart et al. (1986) modified MLP's architecture with replacing step function with logistic function, $\sigma(z) = \frac{1}{1+\exp(-z)}$, unambiguous nonzero derivative, in order to avoid flat segments for gradient descent

to move in a direction, show some progress in every step. Two of the common functions used instead of logit function are S-shaped hyperbolic tangent function $\tanh(z) = 2\sigma(2z) - 1$ and ReLU function. For multiple classification, activation function of output layer of MLP is replaced with Softmax function which assigns an estimated probability to corresponding output classes. In MLP, first step of the algorithm is to calculate weighted sum (Equation-43), input the weight of sum to an activation function such as sigmoid (Equation-44) and output is calculated (Equation-45) (Tarkhaneh and Shen, 2019 ; Clarke and Clarke, 2018; Rumelhart et al., 1986).

Equation-43 Sum of weights function

$$S_j = \sum_{i=1}^n w_{ij}x_i + \beta_i$$

Where;

- x_i is the input variable
- w_{ij} is the weight between variable x_i and neuron, j
- β_i is the bias term

Equation-44 Activation function -Sigmoid

$$f_j(x) = \frac{1}{1 + e^{-S_j}}$$

Where;

- f_j is the sigmoid function for neuron j

Equation-45 Output function

$$y_j = \sum_{i=1}^k w_{ij}f_j + \beta_j$$

Where;

- y_j is the output of neuron j

Fine-Tuning Neural Network Hyperparameters

Being a flexible model neural networks require many hyperparameters to be adjusted such as topology of networks, number of neurons and layers, activation functions, logic of weight initialization etc.

Vanishing/Exploding Gradients Problems

In backpropagations, algorithm propagates error gradient from moving last layer to initial layer and updates weight according to error contribution of each neuron. As a result of this process, gradients shrink in each level and weights of the connection in lower layers remain unaffected, therefore, algorithm does not converge to a better solution. This is known as vanishing gradients problem and opposite of the vanishing gradients is exploding gradients, in which gradients grow enormously, larger weights are allocated in lower layers and algorithm diverges (Géron, 2017).

In order to avoid exploding and vanishing gradient problem, Glorot and Bengio (2010) introduced a solution, namely, Xavier or Glorot Initialization and showed that the difference between variance of output and input layers leads to this problem for weight initialization technique and logistic sigmoid activation function. They stated that variance of inputs and outputs of each layer should be equalized to each other whether in feed forward or in backpropagation. Weight initialization function is given in Equation-46 for logistic activation function and this technique is believed

to be speeding up the training. Table 2.1 illustrates other initialization techniques for different activation functions (He et al., 2015 and Géron, 2017).

Equation-46 Xavier initialization

For normal distribution with mean 0 and standard deviation:

$$\sigma = \sqrt{\frac{2}{n_{inputs} + n_{outputs}}}$$

For uniform distribution between -r and +r, with

$$r = \sqrt{\frac{6}{n_{inputs} + n_{outputs}}}$$

Table 2.1 Initialization parameters for each type of activation function

Activation function	Uniform distribution [-r, r]	Normal distribution
Logistic	$r = \sqrt{\frac{6}{n_{inputs} + n_{outputs}}}$	$\sigma = \sqrt{\frac{2}{n_{inputs} + n_{outputs}}}$
Hyperbolic tangent	$r = \sqrt{\frac{6}{n_{inputs} + n_{outputs}}}$	$\sigma = \sqrt{\frac{2}{n_{inputs} + n_{outputs}}}$
ReLU	$r = \sqrt{2} \sqrt{\frac{6}{n_{inputs} + n_{outputs}}}$	$\sigma = \sqrt{2} \sqrt{\frac{2}{n_{inputs} + n_{outputs}}}$

ReLU (Rectified Linear Unit) activation function performs better because it does not saturate for positive values and fast in computation. However, some neurons stop contributing to output in ReLU activation function, which is a weakness regarding ReLU. In case of a large learning rates, many of neurons could become passive (i.e. dying of neurons, outputting only “0”). This happens because of neurons giving output as “0” when updated weighted some of neurons’ inputs are negative. The negative outputting neurons are less likely to become active because of the gradient of the ReLU function is 0 when its input is negative. A version of ReLU function is used to tackle this issue that is called Leaky ReLU. In Leaky ReLU, a hyperparameter α , typically set to 0.01, controls the amount of the leaks (which is the slope of the function for $z < 0$) and ensures that neuron is never inactive. In some other versions of ReLU, hyperparameter α are whether learned during training (PReLU) or randomly selected (RReLU) (Géron, 2017; Dewa and Afiahayati, 2018)

Equation-47 ReLU activation function

$$RELU(z) = \max(z, 0)$$

Where, z denotes the input vector from neuron.

Equation-48 Leaky ReLU function

$$Leaky\ ReLU_{\alpha}(z) = \max(\alpha z, z)$$

Equation-49 Hyperbolic tangent (tanh) activation function

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Where, z denotes the input vector from neuron.

In 2015, Clevert et al. introduced a new activation function, exponential linear unit function (ELU), which decrease training time and ANN performs better on the test set.

Equation-50 ELU activation function

$$ELU_{\alpha}(z) = \begin{cases} \alpha(\exp(z) - 1) & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}$$

Batch Normalization or BN were proposed by 2015 by Ioffe and Szegedy, which is capable of solving vanishing/exploding gradients issues.

Equation-51 Batch Normalization algorithm

1. Mini-batch mean

$$\mu_B = \frac{1}{m_B} \sum_{i=1}^{m_B} \mathbf{x}^{(i)}$$

2. Mini-batch variance

$$\sigma_B^2 = \frac{1}{m_B} \sum_{i=1}^{m_B} (\mathbf{x}^{(i)} - \mu_B)^2$$

3. Normalize

$$\hat{\mathbf{x}}^{(i)} = \frac{\mathbf{x}^{(i)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

4. Scale and shift

$$\mathbf{z}^{(i)} = \gamma \hat{\mathbf{x}}^{(i)} + \beta$$

- μ_B is the empirical mean, evaluated over the whole mini-batch B
- σ_B is the empirical standard deviation, also evaluated over the whole mini-batch B
- m_B is the number of instances in the mini batch
- $\hat{\mathbf{x}}^{(i)}$ is the zero-centered and normalized input
- γ is the scaling parameter for the layer to be learned
- β is the shifting parameter (offset) for the layer to be learned
- ϵ is smoothing constant to avoid division by zero (is a constant added to the mini-batch variance for numerical stability)
- $\mathbf{z}^{(i)}$ is the output of the Batch Normalization operation: it is a scaled and shifted version of the inputs.

Besides initialization techniques, activation functions, batch normalization, faster optimizers are used to speed up the training model instead of regular gradient descent optimizers. Most common optimizers are Classical Momentum Optimization, Nestroy Accelerated Gradient, AdaGrad, RMSProp and Adam optimization.

Classical Momentum Optimization

Classical momentum optimization use gradient as an acceleration factor, in which momentum vector, $\mathbf{m}$ is added to the local gradient and weight is updated with subtracting moment vector, $\mathbf{m}$ from preceding weight. Algorithm introduce a hyperparameter called momentum or damping coefficient, β (changes between 0 and

1 for no friction and high friction accordingly) that controls excessive growth of the momentum vector, and helps in avoiding local minimum faster than gradient descent (Géron, 2017 and Qian, 1999).

Equation-52 Classical Momentum Optimization Algorithm

1.

$$\mathbf{m} \leftarrow \beta \mathbf{m} + \eta \nabla_{\theta} J(\theta)$$

2.

$$\theta \leftarrow \theta - \mathbf{m}$$

- η is the learning rate

Nesterov Accelerated Gradient

Nesterov (1983) introduced a change to classical momentum optimization for faster optimization, which measure gradient of cost function at $\theta + \beta \mathbf{m}$ instead of θ . In other words, stability issue is avoided in Nesterov Accelerated Gradient by adjusting the error made after moving in the path of the momentum vector, $\mathbf{m}$ (Lucas et al., 2019).

Equation-53 Nesterov Accelerated Gradient Algorithm

1.

$$\mathbf{m} \leftarrow \beta \mathbf{m} + \eta \nabla_{\theta} J(\theta + \beta \mathbf{m})$$

2.

$$\theta \leftarrow \theta - \mathbf{m}$$

AdaGrad

AdaGrad is used to speed algorithm up to reach to global minimum with adopting learning rate to parameters meaning to assign low learning rates to frequent attributes and high learning rates for rare attributes (Duchi et al., 2011). In other words, learning rate is decayed fast to reach global optima. It contributed to the improvement of Stochastic Gradient Descent and enabled to process large-scale data sets and requires little tweaking of the learning parameter, η (Dean et al., 2012).

Equation-54 AdaGrad Algorithm

1.

$$\mathbf{s} \leftarrow \mathbf{s} + \nabla_{\theta} J(\theta) \otimes \nabla_{\theta} J(\theta)$$

- $\otimes$ is the element-wise multiplication

2.

$$\theta \leftarrow \theta - \eta \nabla_{\theta} J(\theta) \oslash \sqrt{\mathbf{s} + \epsilon}$$

- $\oslash$ is the element-wise division
- ϵ is smoothing constant to avoid division by zero

The squared gradients in the initial step are collected under the vector $\mathbf{s}$, and s_i of the vector adds square of partial derivative of cost function with respect to θ_i for each element s_i of the vector $\mathbf{s}$, $s_i \leftarrow s_i + (\frac{\partial}{\partial \theta_i} J(\theta))^2$. At each iteration, s_i grows substantially given that cost function is steep along the i^{th} dimension. The second step is similar to gradient decent and gradient vector is scaled down by $\sqrt{s + \epsilon}$. This vectorized form corresponds to $\theta_i \leftarrow \theta_i - \frac{\eta \frac{\partial}{\partial \theta_i} J(\theta)}{\sqrt{s_i + \epsilon}}$ for all parameters θ_i .

RMSProp

RMSProp algorithm was introduced to resolve the shrinking learning rate issue in AdaGrad due to accumulated square of partial derivative of cost function in the denominator. RMSProp avoids this issue by accumulating only square of partial derivative of cost function in the denominator from the most recent iterations with the help of exponential decay in the first step (Géron, 2017).

Equation-55 RMSProp algorithm

1.

$$\mathbf{s} \leftarrow \beta \mathbf{s} + (1 - \beta) \nabla_{\theta} J(\theta) \otimes \nabla_{\theta} J(\theta)$$

- $\otimes$ is the element-wise multiplication
- β is the learning rate

2.

$$\theta \leftarrow \theta - \eta \nabla_{\theta} J(\theta) \oslash \sqrt{\mathbf{s} + \epsilon}$$

- $\oslash$ is the element-wise division
- ϵ is smoothing constant to avoid division by zero

Adaptive Moment (Adam) Optimization

Adam optimization merges two advantages of RMSProp and Momentum optimization, namely maintaining track of an exponentially decaying average of past gradients and an exponentially decaying average of past square of gradients (Kingma and Ba, 2015).

Equation-56 Adam algorithm

1.

$$\mathbf{m} \leftarrow \beta_1 \mathbf{m} + (1 - \beta_1) \nabla_{\theta} J(\theta)$$

2.

$$\mathbf{s} \leftarrow \beta_2 \mathbf{s} + (1 - \beta_2) \nabla_{\theta} J(\theta) \otimes \nabla_{\theta} J(\theta)$$

3.

$$\mathbf{m} \leftarrow \frac{\mathbf{m}}{1 - \beta_1^T}$$

4.

$$\mathbf{s} \leftarrow \frac{\mathbf{s}}{1 - \beta_2^T}$$

5.

$$\theta \leftarrow \theta - \eta \mathbf{m} \oslash \sqrt{\mathbf{s} + \epsilon}$$

- T is the number of iterations

AMSGrad, Nadam (Dozat, 2016), AdaMax (Kingma and Ba, 2015) and Adadelta (Zeiler, 2012) are other optimization techniques in ANN.

.

Avoiding Overfitting Through Regularization

Early stopping, ℓ_1 and ℓ_2 regularization, dropout, max-norm regularization, and data augmentation are used to avoid overfitting in training data. In early stopping, training is interrupted as performance on the validation data sets worsen. ℓ_1 and ℓ_2

regularizations are used to restrain the weight of the ANN connections. Another technique used for overfitting is the use of dropout technique that assigns dropout probability to input neurons at every training step known as dropout hyperparameter and ignores these neurons for that specific training step. Dropouts improve performance of neural networks from almost all domains and efficient in assembling different neural networks architecture, however, increase training time (Srivastava, 2014).

Max-Norm regularization technique restricts the weight $\mathbf{w}$ of the incoming connections for each neuron such that $\|\mathbf{w}\|_2 \leq r$, where r is the max-norm hyperparameter and $\|\cdot\|_2$ is the ℓ_2 norm. $\|\mathbf{w}\|_2$ is computed after each training step and $\mathbf{w}$ is restricted $\mathbf{w} \leftarrow \mathbf{w} \frac{r}{\|\mathbf{w}\|_2}$. For decreased overfitting and increased regularization, r need to be reduced.

2.2.2.11 Autoencoders

Autoencoders are unsupervised ANN that can learn from unlabeled input data sets and represent efficiently. In other words, autoencoders try to replicate original input by encoding (recognizing network) and decoding (generating network), i.e., generating novel data sets which are very similar to original data sets with copying inputs to outputs. Some constraints are added to the network to avoid carbon-copying of inputs to outputs such as size restriction of internal representation and noisiness to input instances, thus, make system to learn efficient representation. Figure 2.6 represents schematic view of an autoencoder consisting of single encoder and decoder.

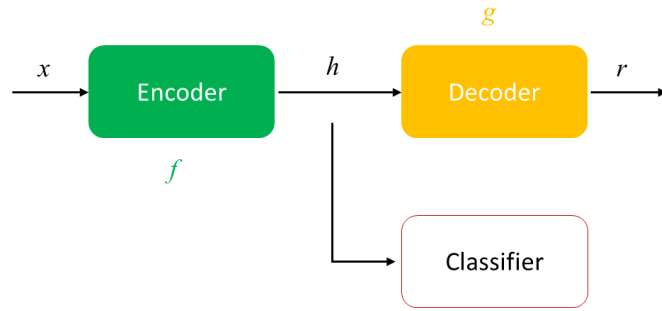


Figure 2.6 Schematic view of an autoencoder consisting of single encoder and decoder

Architecture of MLP and autoencoders are same apart from that autoencoders requires equal number of neurons in input and output layer. Autoencoders with multiple hidden layers are called stacked or deep autoencoders that are good at solving complex issues. Encoders transform original input data into a hidden interpretation, $h = f(x)$, and decoder transforms this hidden interpretation back to original input, $r = g(x)$. The goal here is to keep reconstruction error, $L(x, r)$ at minimum while replicating input instances, that means obtaining as many attribute transformations as possible (Jiang et al., 2017).

Equation-57 Autoencoder Activation Functions

1.

$$h = f(x) = a_f(b_f + W_f x)$$

- a_f is the activation function for the encoder
- b_f is the bias term
- W_f is corresponding weight matrix

2.

$$g(h) = a_g(b_g + W_g x)$$

- a_g is the activation function for the decoder
- b_g is the bias term for the decoder
- W_g is corresponding weight matrix for the decoder

Sparse Autoencoders

For better attribute extraction, sparsity can be added to autoencoders, in which cost function is adjusted with a term called sparsity penalty. In sparse autoencoders, encoders are designed for dimensionality reduction in input sets (i.e., reduced active neurons in the coding layer), thus, increase number of output neurons and chance of extracting more information from the original data set. This kind of autoencoders are called sparse autoencoders (Jiang et al., 2017).

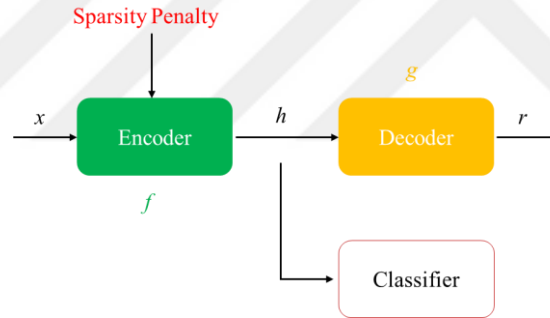


Figure 2.7 Schematic view of a sparse autoencoder.

Actual sparsity is measured at each iteration with calculating mean activation of each neuron in encoder per whole training data set. Neurons having higher mean activity is penalized with sparsity loss added to cost function. In other words, neuron with higher activities are reduced according to their mean activation weight. MSE or Kullback–Leibler divergence approaches are used for sparsity loss (Géron, 2017 and Jiang et al., 2017).

Equation-58 Kullback–Leibler divergence

$$D_{KL}(P\|Q) = \sum_i P(i) \log \frac{P_i}{Q_i}$$

- P (target) and Q (actual) are distributions of discrete probability

2.2.2.12 Reinforcement Learning

Reinforced learning is machine learning technique commonly used with Markov Decision Process. Learning system observe environment, choose an action and performs it and gets a reward or penalty in the return. System then form a policy to get highest number of reward or no reward with time by learning (Woergoetter and Porr, 2008).

2.2.3 Counting the Cost

Counting the cost means taking the cost of decision or classification made incorrectly. The cost of predicted desired class is higher than predicted undesired class but the probability of predicting undesired class is way higher desired class for classification problems. For example, cost of failing to detect a fraud alert in transaction is higher than the cost of detecting a safe transaction as fraud however the probability of correctly detected fraud transaction is far smaller than the probability of safe transaction. In this context, precision and sensitive (recall) become important (Witten and Frank, 2005).

Equation-59 Recall (sensitivity) or true positive rate (TPR) formula

$$Recall = P(+|+) = \frac{TP}{TP + FN}$$

Equation-60 False negative rate (FNR)

$$FNR = \frac{TN}{TP + FN}$$

Equation-61 Precision formula

$$Precision = \frac{TP}{TP + FP}$$

Equation-62 Specificity formula

$$Specificity = P(-|-) = \frac{TN}{TN + FP}$$

Equation-63 False Positive Rate (FPR)

$$FPR = 1 - Specificity = \frac{FP}{TN + FP}$$

Equation-64 Accuracy formula

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

Where,

- TP is true positive (correct prediction)
- TN is true negative (correct prediction)

- *FP* is false positive (incorrect prediction)
- *FN* is false negative (incorrect prediction)

Table 2.2 Default cost matrices: (a) a two-class case and (b) a three-class case.

a)		Predicted		b)		Predicted		
		Class				Class		
		+	-			a	b	c
Actual class	+	TP	FN	Actual class	a	0	1	1
	-	FP	TN		b	1	0	1
					c	1	1	0

Table 2.2 represents confusion matrix for binary and three class case. Greater numbers on the main diagonal and lower small number represents ideal matrices (Clarke and Clarke, 2018; Davoudi Kakhki et al., 2019).

Recall/Precision Tradeoff

Recall and precision corresponding to 1.0 value are ideal for a perfect separable data set, however, this is not possible for data having nosiness meaning that data sets are not divisible ideally in practice. False negative and false positive instances obstruct

perfect recall and precision. Noisiness of data pushes boundaries and this result in decrease of whether recall or precision. Recall (sensitivity) and precision tradeoff takes place due to increase of one term affects other to decrease. A threshold value is selected to shift the decision boundaries for optimum precision and recall together, however, in some case precision could be preferred over recall or vice versa (Zhu et al., 2010). For convenience, harmonic average of recall and precision are combined under one metric, which eases the comparison of different classifiers.

Equation-65 F_1 score

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

Lift Charts

Lift is a measure of the effectiveness of a predictive model computed as the proportion of results obtained with and without the predictive model or compared to random estimate. It consists of a baseline and a curve. The size of the area between bell curve and baseline defines the robustness of the model (Zhu et al., 2010).

ROC curves

Receiver Operating Characteristic (ROC) is used to identify the quality of diagnostic classification with plotting TPR ($y = TPRate(t)$) against FPR ($x = FPRate(t)$). All combination of true positive rate and false positive rate makes ROC space. Tradeoff is compared with random classification as well in ROC. Perfect tradeoff is

sensitivity corresponding to 1.0 and FPR corresponding to 0.0 that means 1.0 specificity. Performance measurement for classification issues for different threshold settings are realized with AUC, area under curve-ROC curve. Higher AUC means higher TP (true positive) and TR (true negative) (Zhu et al., 2010).

Equation-66 AUC

$$AUC = \int_0^1 ROC(t)dt$$

2.2.4 Overfitting and Underfitting

While modelling an algorithm to learn from data, one of the main objectives to avoid both overfitting and underfitting. Overfitting means fitting of data to algorithm too well or algorithm shows high variance and low bias, however, underfitting means algorithm lacking in fitting to data well. Both should be avoided because they are the one of the key factors affecting performance of a model in machine learning (Brownlee, 2016).

2.3 Prescriptive Data Analytics

Prescriptive data analytics uses results of predictive analytics to put forward any recommendation. These recommendations include control measures or actions thanks to background data, ML algorithms, and any external data sets. Prescriptive analytics has been used in business domain commonly, namely manufacturing, sales and marketing (Lepenioti et al., 2020). It is immature compared with other statistical methods and research interests are mainly on predictive analytics however it

progressively attracts researchers from different domains (Larson and Chang, 2016). Šikšnys and Pedersen (2016) stated that predictive analytics could add value to business and main objective is to give a decision based on extracted information through predictive analysis. The quality of the prescriptive analytics is highly dependent on stages of predictive analysis. Mathematical programming, Evolutionary computation, Simulation, Machine learning/data mining and Logic-based models are used in prescriptive analytics in different research fields (Lepenioti et al., 2020). Model illustrated in Figure 2.8 can be used as a proactive control measure against occupational accidents. The time gap between realization of a complex event pattern and subsequent action is potential loss of business value in terms of information disadvantage. Early detection of these complex events and patterns can be crucial in minimizing this time gap, hence, loss of business. The time gap between event prediction and proactive action is called inevitable loss, and the minimization of this gap is of outmost importance. Gained business value is the distance between implementation of proactive action and reactive action (Krumeich et al., 2015). Timely prediction of future events can help in increasing business value through decreased decision latency. From the same point of view, timely prediction of unwanted events or accidents can contribute to faster decision (smaller gap between proactive decision and reactive decision) regarding actions or countermeasures that will be taken against. From the perspective of prescriptive analytics of work-related accidents, objective should be loss avoidance meaning decreasing injuries, near misses, road traffic accident etc. Figure 2.9 illustrates proposed model. Available historical data sets can be used in predictive models for training, which can be useful in predicting new occurrences of specific complex events during runtime (Engel and Etzion 2011; Engel et al. 2012; Fülöp et al. 2012).

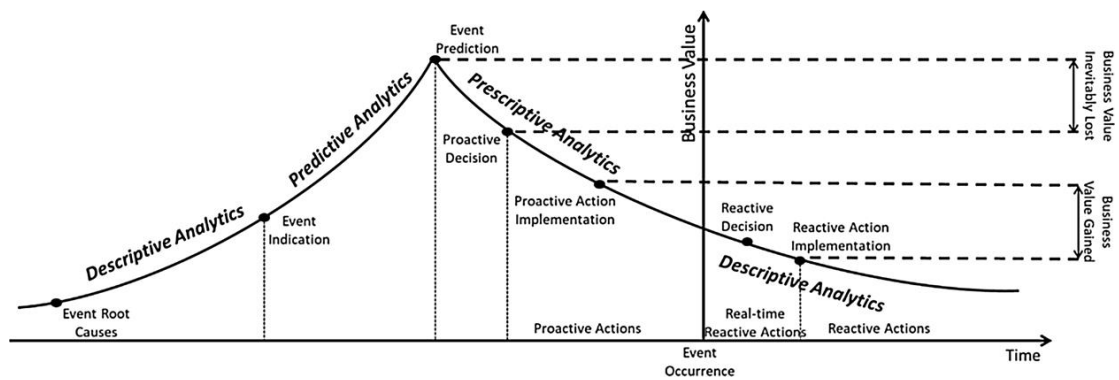


Figure 2.8 Business value gained through event prediction and proactive actions (Krumeich et al., 2015)

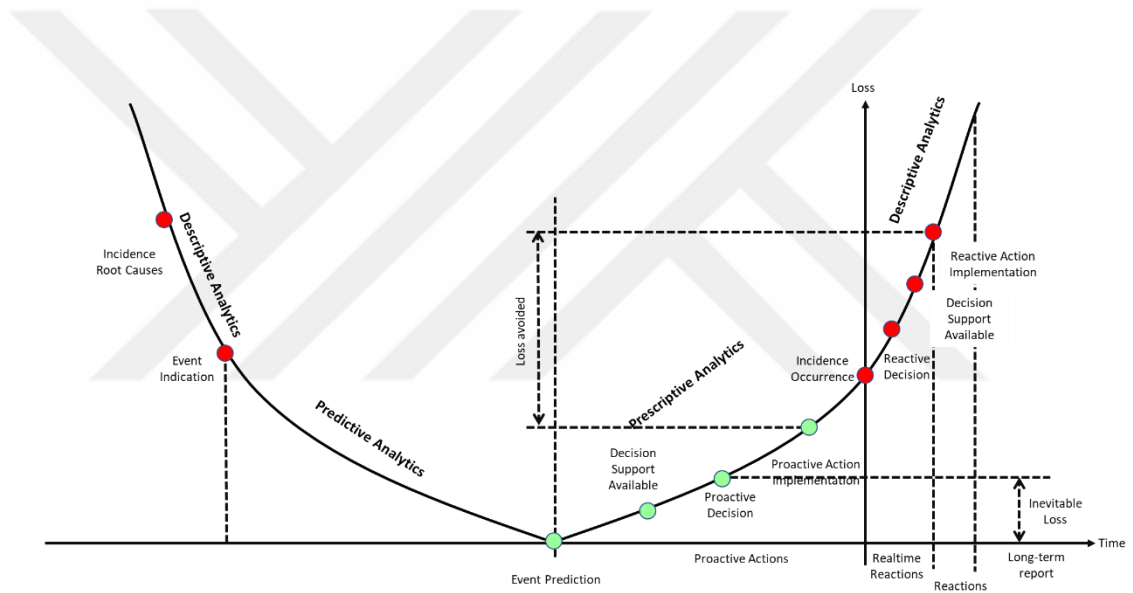


Figure 2.9 Proposed zero harm through event prediction and proactive actions



CHAPTER 3

METHODOLOGY

Methodology section covers both steps taken in data collection, how variables or attributes were selected and described, stages of data processing and descriptive statistics of processed data.

3.1 Data Collection, Variable Description and Data Processing

This chapter aims to discuss variables, their explanations, source of information, and all the steps taken during the course of this study. The study consists of five steps which was illustrated in Figure 3.1.

The research begun with data collection from a megaproject in which several countries and people from various nationalities were involved. The collected raw data were unstructured and composed of accident/incident investigations, initial incident reports, photographs, drawings, videos, lessons learned and statements. Due to the documents from various companies and report formats, it was challenging and troublesome to extract useful information from these reports. After data collection, reports including 5,500 files and approximately 25,000 pages were organized according to type of report. The next stage was choosing a standard incident reporting format that can be fit for all or majority of the report types. Therefore, format of the operating company was chosen and some parts were improved according to the known practices in the field. This made overall process easier

because of reports coming from operating company composes of more than 75% of all reports.

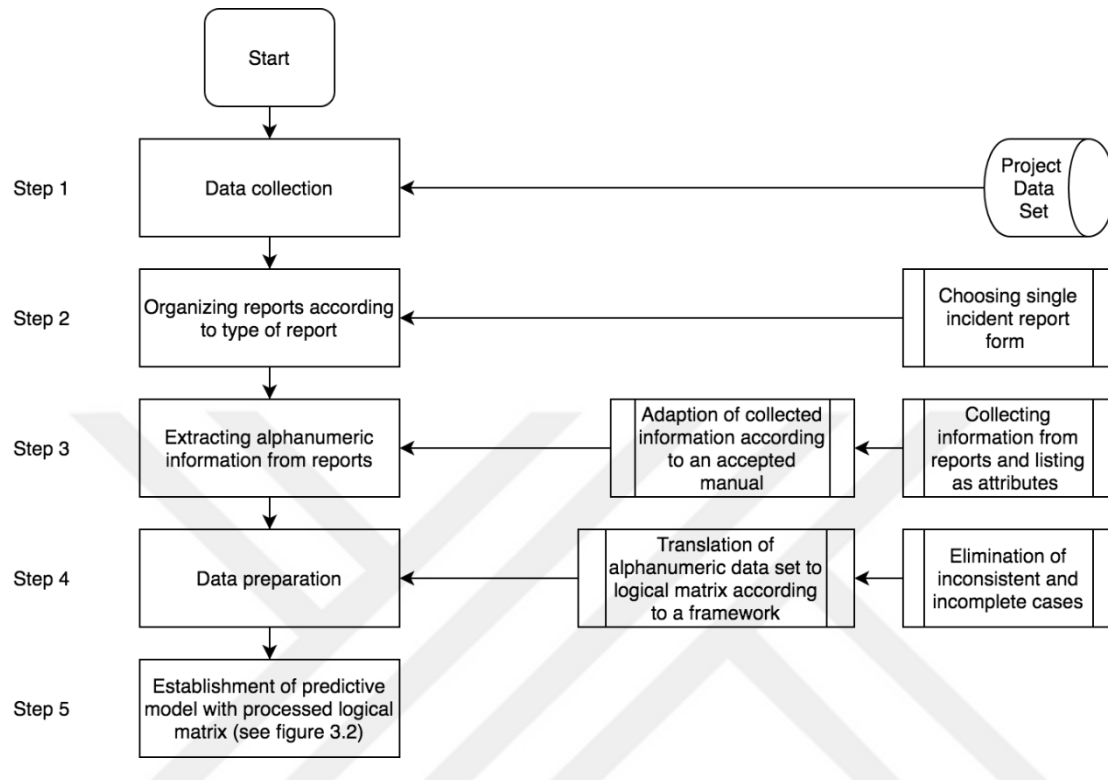


Figure 3.1. Flow of the study

Extraction of information from reports covers detailed reading of all reports and finding out information regarding every single case. This information includes date and time of the cases, demographic information about injured person, root and immediate causes, part of body injured, countermeasures, operation at the time of incident, outcomes, contributing factors etc. For root and immediate cause selection, a guidance is used with 24 immediate causes and 18 root causes which can be found in Table 3.1. This guidance is widely used by all sectors.

Health and Safety Executive gives definitions of causes as immediate cause being the most obvious reason why an adverse event takes place and root cause is an initiating event or failing from which all other causes or failings spring.

Four types of injury were listed which are First aid case, Medical treatment case, Lost workday case and Restricted work case and explained below.

First aid case refers to injuries that can be treated at the workplace without the need for medical treatment.

Medical treatment case refers to any injury where medical treatment is given by a physician or other registered medical practitioner. Medical treatment usually involves treatment above and beyond that provided at the workplace by a first aid attendant.

Lost workday case or Lost-time injury (LTI) refers to any injury that prevents a worker from coming to work beyond the date of the injury for a given period.

Restricted Work Case is when a person is so injured that they cannot perform their normal duties.

In the following stage, three type of category was found which are injury, asset or production damage and near miss. These categories hereinafter referred to as consequences. Besides categorization of incident type, type of the event and exposure are defined according to “Occupational Injury and Illness Classification Manual” of U.S. Department of Labor Bureau of Labor Statistics, which was published in 1992. The main category and subcategory of the classification is given in the Table 3.2. Some of the incomplete and inconsistent information is eliminated for predictive model.

Table 3.1 Categories of event and exposure

CONTACT WITH OBJECTS AND EQUIPMENT	FALLS
Struck against object	Fall to lower level
Struck by object	Jump to lower level
Struck by swinging or slipping object	Fall on same level
Caught in or compressed by equipment or objects	BODILY REACTION AND EXERTION
Caught in or crushed in collapsing materials	Bodily reaction
Rubbed or abraded by friction or pressure	Overexertion
Rubbed, abraded, or jarred by vibration	Repetitive motion
Contact with objects and equipment, n.e.c.	Sustained viewing
EXPOSURE TO HARMFUL SUBSTANCES OR ENVIRONMENTS	Bodily conditions, n.e.c.
Contact with electric current	Bodily reaction and exertion, n.e.c.
Contact with temperature extremes	TRANSPORTATION ACCIDENTS
Exposure to air pressure changes	Highway accident
Exposure to caustic, noxious, or allergenic substances	Nonhighway accident, except rail, air, water
Exposure to noise	Pedestrian, nonpassenger struck by vehicle, mobile equipment
Exposure to radiation	Railway accident
Exposure to traumatic or stressful event, n.e.c.	Water vehicle accident
Oxygen deficiency, n.e.c.	Aircraft accident
Exposure to harmful substances or environments, n.e.c.	Transportation accident, n.e.c.
FIRES AND EXPLOSIONS	OTHER EVENTS OR EXPOSURES
Fire--unintended or uncontrolled	Assaults and violent acts by person(s)
Explosion	Self-inflicted injury
	Assaults by animals
	Nonclassifiable

Table 3.2 List of immediate and root causes

Immediate Causes	Root Causes
1.Design of Plant Facilities, or Equipment (IC1)	1.Procedures & Safe Work Practices (RC1)
2.Job Planning, Poor Supervision or Instruction Inadequate (IC2)	2.Communications (RC2)
3.Rules, Procedures or JHA Not Followed (IC3)	3.Design/Road Design/Vehicle Design (RC3)
4.Rules, Procedures or JHA Inadequate (IC4)	4.People & Behavior (RC4)
5.Incorrect Body Position in Relation to Work (IC5)	5.Monitoring & Inspection (RC5)
6.Guarding or Protective Devices Not Provided or Ineffective/ Lack of Preventive Action (IC6)	6.Management of Change (RC6)
7.Plant Equipment Operated Incorrectly (IC7)	7.Materials Control/ Procurement (RC7)
8.Housekeeping Congested, Incorrect Storage (IC8)	8.Leadership Accountability (RC8)
9.Incorrect Tools or Mechanical Aids Used (IC9)	9.Planning & Workload (RC9)
10.Incorrect or Lack of Personal Protective Equipment (IC10)	10.Hazard Prevention/ Investigation (RC10)
11.Inadequate Knowledge or Skill (IC11)	11.Risk Recognition & Assessment & Situational Awareness (RC11)
12.Chemical Exposure, Personal Hygiene (IC12)	12.Supervision/ Oversight (RC12)
13.Improper Vehicle Operation (IC13)	13.Contractor Operations (RC13)
14.Environmental Factors, Weather (IC14)	14.Training/ Competency (RC14)
15.Inattention to Details and Hazards of Job or Not Controlling Surrounding (IC15)	15.Work Environment/ Workplace Layout/Road Condition (RC15)
16.Action of Fellow Employee (IC16)	16.Emergency Response (RC16)
17.Maintenance, Inspection Not Adequate (IC17)	17. Conflicting roles and responsibilities (RC17)
18.Other Mechanical Failure (IC18)	18. No control (RC18)
19.Action of 3rd party (IC19)	
20.Using defective tools/equipment or materials (IC20)	
21. Poor loading/lifting technique or improper placement (IC21)	
22. Limited Control Over (IC22)	
23. Poor Coordination/Reaction Time /Mechanical Aptitude (IC23)	
24. Pre-existing health condition (IC24)	

After completion of the alphanumeric data set, logical matrix was constructed using attribute-based framework, that was introduced by Esmaeili and Hallowell (2012). The framework is a comprehensive and systematic method of extracting structured information from unstructured reports. The final data set is a logical matrix composed of 59 columns (attributes) and 1,184 rows (cases). The framework is illustrated in Figure 3.2. There are 24 immediate causes, 18 root causes, 7 type of exposure, 3 type of incident, 5 type of injury, and 2 time attributes.

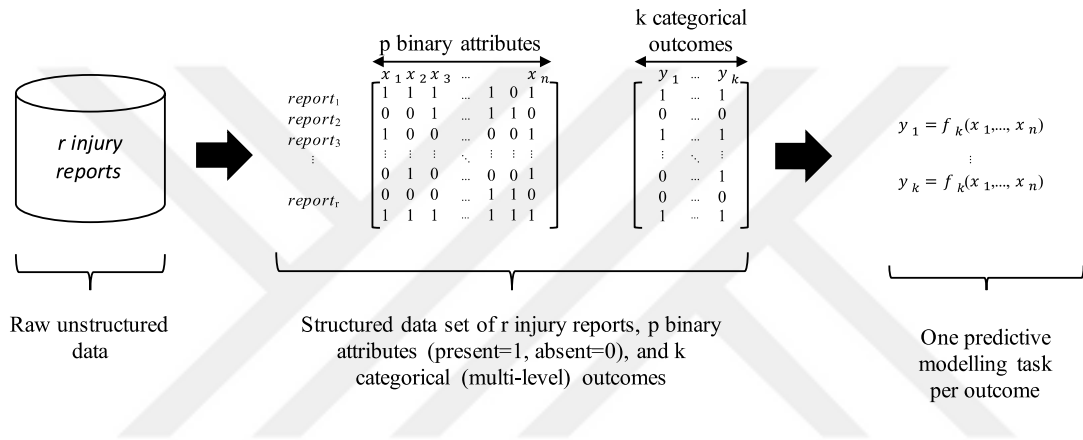


Figure 3.2. Attribute based framework (Esmaeili and Hallowell (2012))

The final stage of the study is to feed ML algorithms with the data set and find out the performance of the models. In the first run all data was used in the model to predict outcome of the causes as injury, asset/production damage or near miss. However, in the second and following runs near miss cases is excluded to increase performance of model because of the number of the near miss cases being few in number. In each run 10% of all cases excluded for testing set and rest of the date is used for training set. The model was expected to predict outcome of incident from variables (attributes). Testing set was later used in trained model from training set to predict its outcomes. These outcomes are compared with true outcomes and this was

given as performance of the model. Number of iterations and parameter optimization are realized to increase performance of the model in addition to model selection. This procedure is represented in Figure 3.3.

In this supervised, nonparametric and model based-training, algorithm is fed with the labeled training data set including desired solutions aiming to learn finding output variables, type of incident in this case. Results can be found in results and discussion chapter.



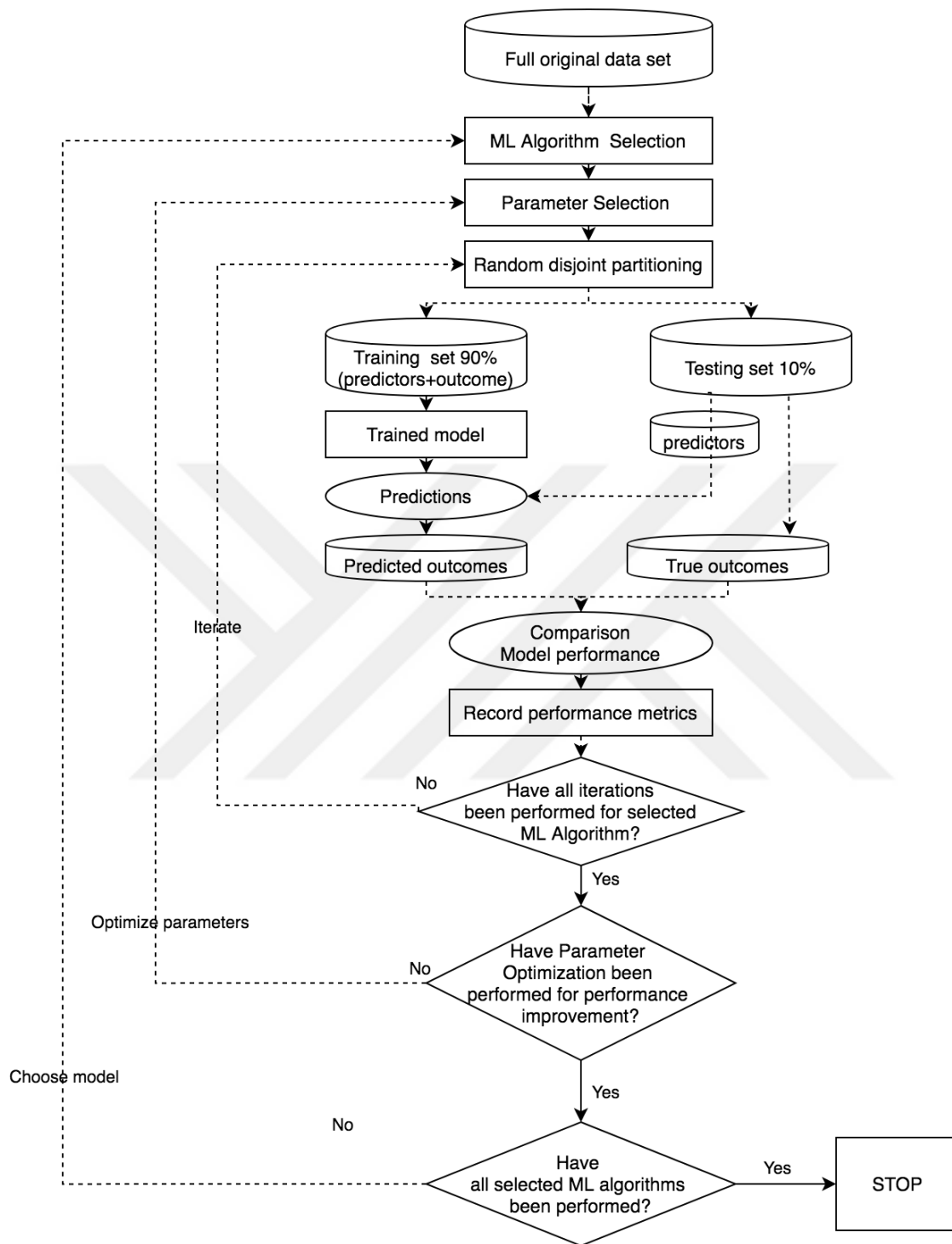


Figure 3.3. Validation procedure for predictive model

Different sets of data will be selected as per the table below; twelve sets of data will be used in analysis. For instance, in SET01, consequences will be predicted with keeping nearmiss, date and time, ICs and RCs, however, in SET11, consequences will be predicted form ICs with excluding nearmiss, date and time.

Table 3.3 Data sets

with Nearmiss	with Date and Time	from ICs and RCs	SET01
		From ICs	SET02
		From RCs	SET03
	without Date and Time	from ICs and RCs	SET04
		From ICs	SET05
		From RCs	SET06
without Nearmiss	with Date and Time	from ICs and RCs	SET07
		From ICs	SET08
		From RCs	SET09
	without Date and Time	from ICs and RCs	SET10
		From ICs	SET11
		From RCs	SET12

3.2 Descriptive Data Analysis

Preceding chapter involves stages of data collection, introduction of attributes selection procedure, and preparation of raw data. This chapter not only serve as descriptive statistics of those attributes in order to uncover any hidden details and patterns but also illustrates pre-processing of data set for predictive analysis.

3.2.1 Distribution of total cases and type of exposure

Distribution of total cases and type of exposure over date and time is given in this section in order to better introduce and represent data set. The project took three consecutive years.

In the Figure 3.4, it can be seen that most of the cases took place in the last quarters with approximately 35% and followed by third and first quarters with 27% and 20%. Note that first quarter of the first year is missing because of the project initiated in the second quarter. It can be concluded from the figure that there was an increasing trend over the three years and upward trend in the number of cases from first to last quarter.

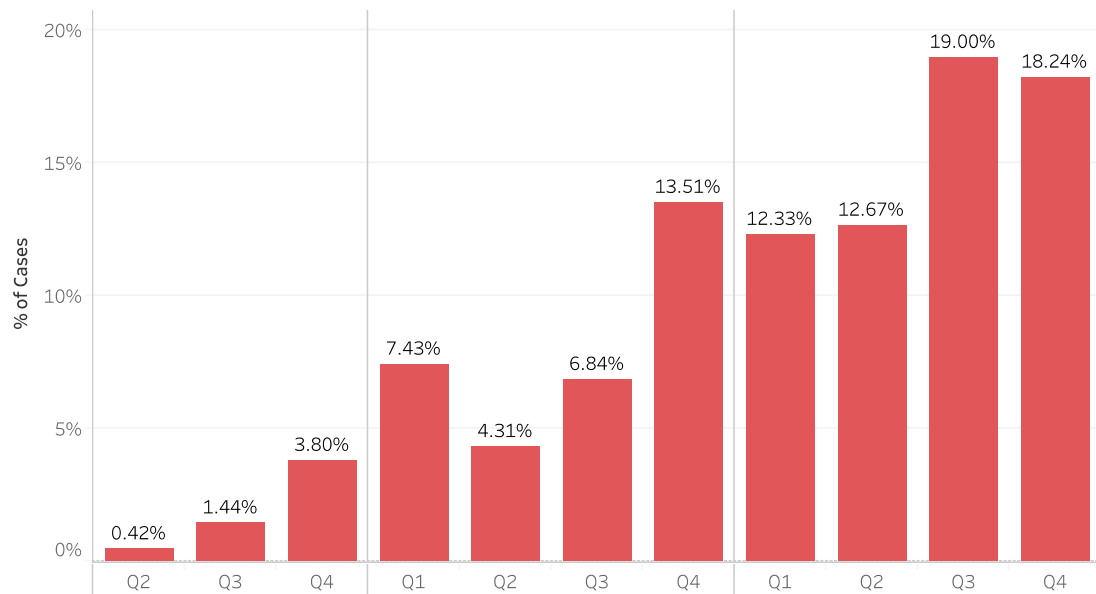


Figure 3.4 Distribution of the cases per year and quarter

In Figure 3.5.a, pareto chart illustrates percentage of cases per quarter, month, day of the week and hour of the day that incident took place. October, December and November saw the highest number of cases with approximately 50% more than monthly average. August and September were also slightly higher than the average number of cases per month. Month of May marks the month in which the fewest number of cases were occurred with roughly 50% percent below the monthly average.

There was not a significant change in the number of cases per day of the week apart from Sundays which accounted for only 7.2%.

The heights figure per hour distribution were for hours of 11, 14, 9, 10 and 16 with almost 57% of all cases.

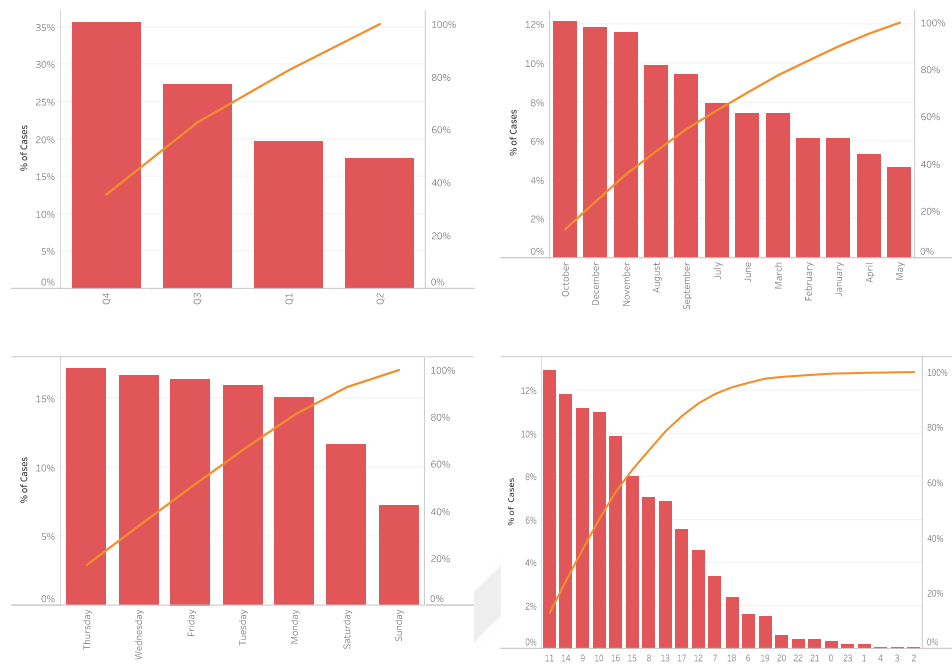


Figure 3.5 Distribution of the cases per quarter, month, day of the week and hour

It could be concluded from the Figure 3.6 that Contact with objects and Transportations accidents were accounted almost three-quarter of all seven type of accidents with 41% and 33% accordingly. Falls were the third highest figure which was accounted 11% of all cases.

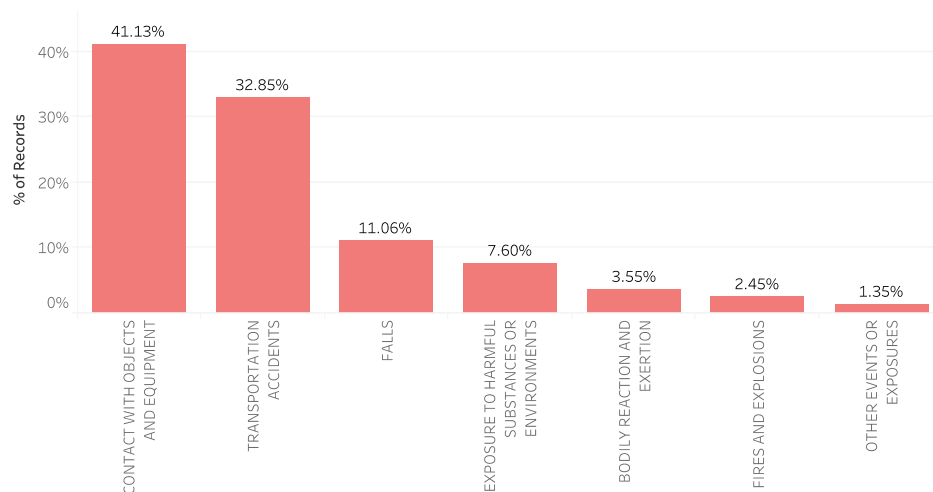


Figure 3.6 Distribution of the cases per type of incident

In Figure 3.7, distribution of event exposure type per quarter was introduced. It could be concluded from the figure that fourth quarter was the highest in the number of cases for all exposure type excluding Other exposure types which was second quarter. Third quarter were the second highest in Bodily reaction and exertions, Contact with object and equipment, Other and Transportation accidents, however, second quarter was second highest for only Exposure to harmful substances or environments.

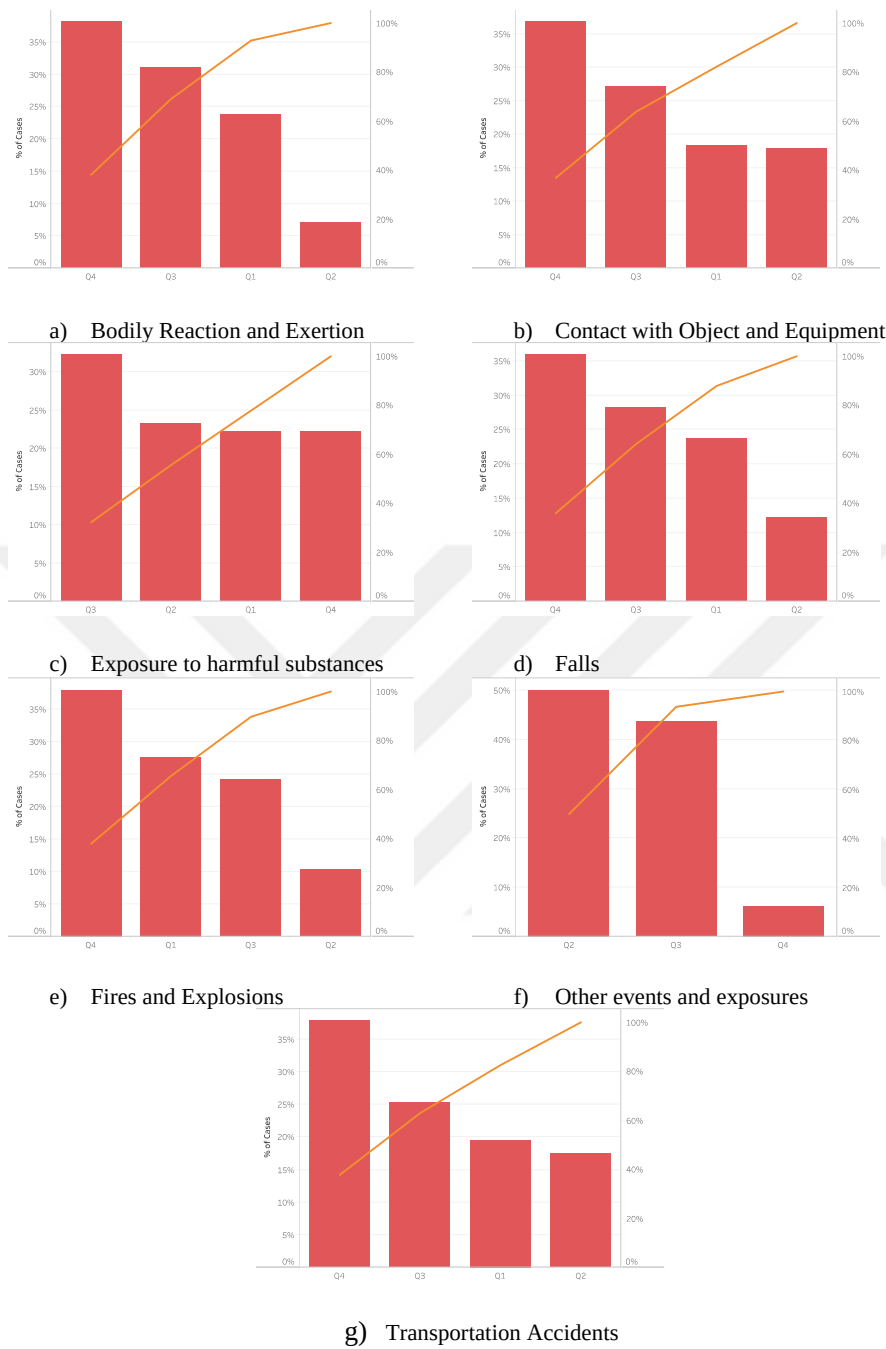


Figure 3.7 Distribution of the cases per type of exposure per quarter

From Figure 3.8, 3.9 and 3.10, It was clear that Bodily reaction and exertions were highest in Octobers, on Thursdays and at around 11 a.m. Distribution of this type of event per month, day of the week and hour were similar to total distribution which was given in Figure 3.5.

Highest number of cases were seen at around 11 a.m. for three exposure types, which are Bodily reaction and exertions, and Contact with object and equipment. It could be concluded that hourly distribution of the most exposure types was in line with total distribution of the cases. August accounted for most of the cases in Falls and Other exposure types, however, December was highest for Fire and explosion, and Transportation accidents. In addition to this, Contact with objects and equipment was very common in November and on Fridays.

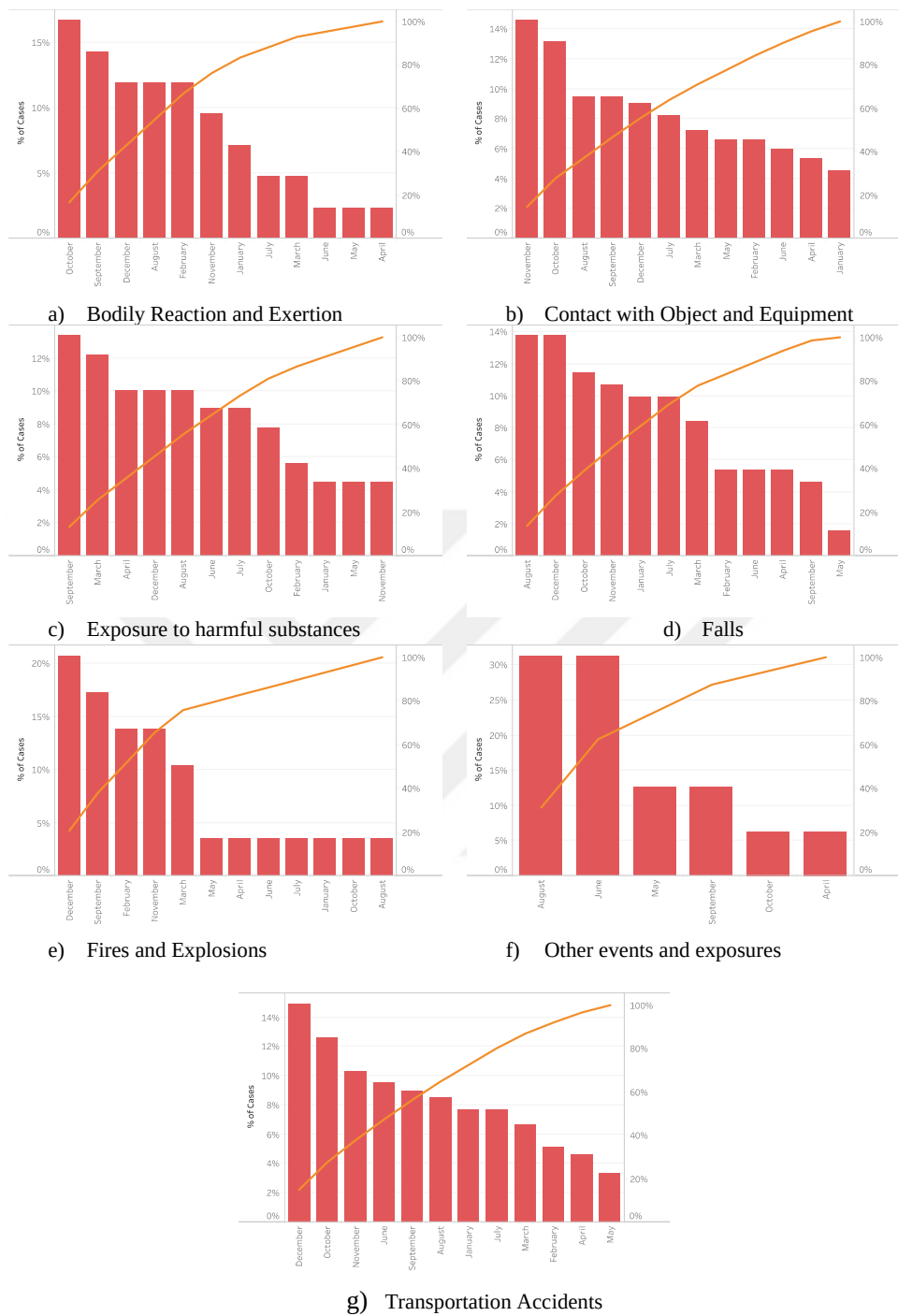


Figure 3.8 Distribution of the cases per type of exposure per month

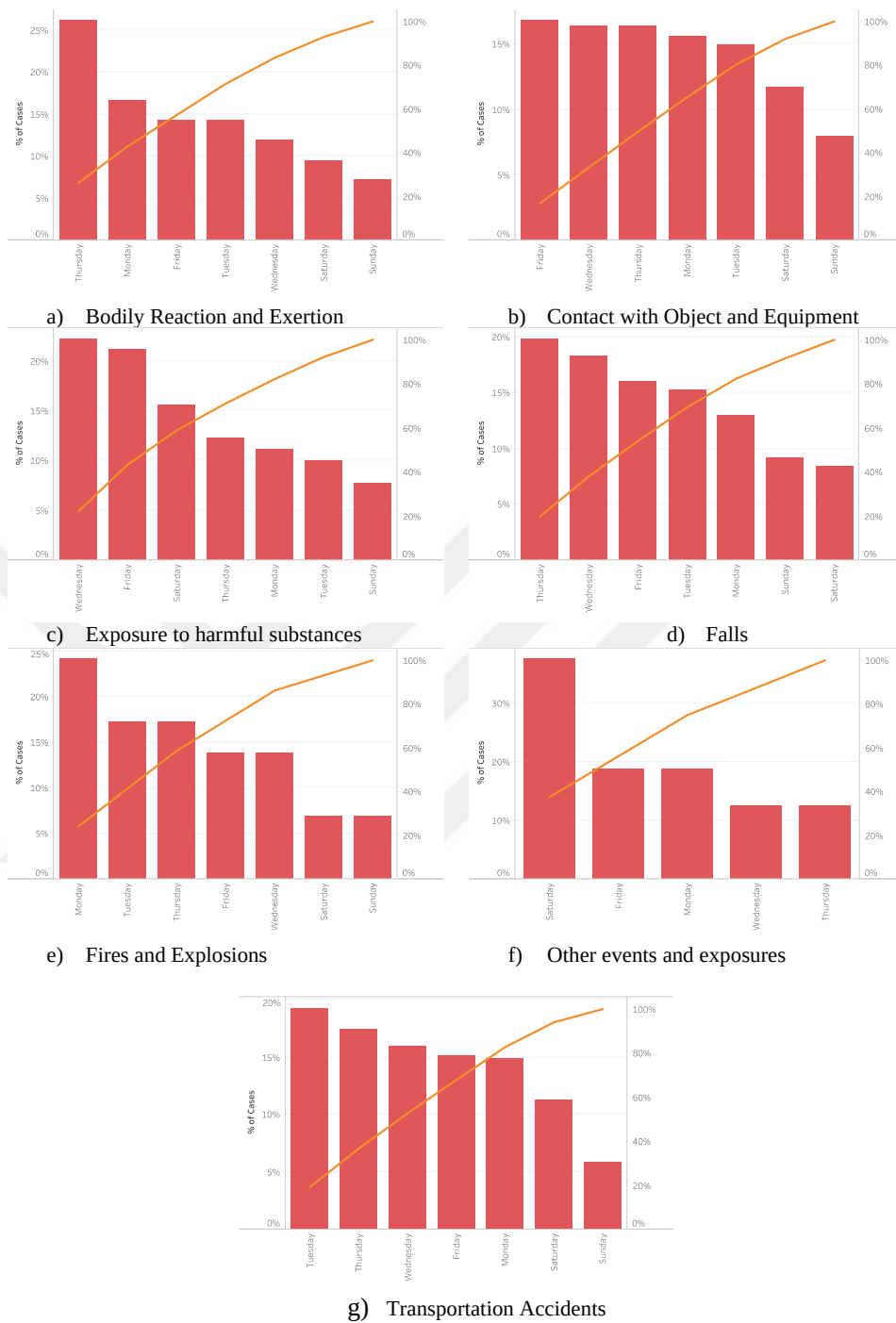


Figure 3.9 Distribution of the cases per type of exposure per day of week

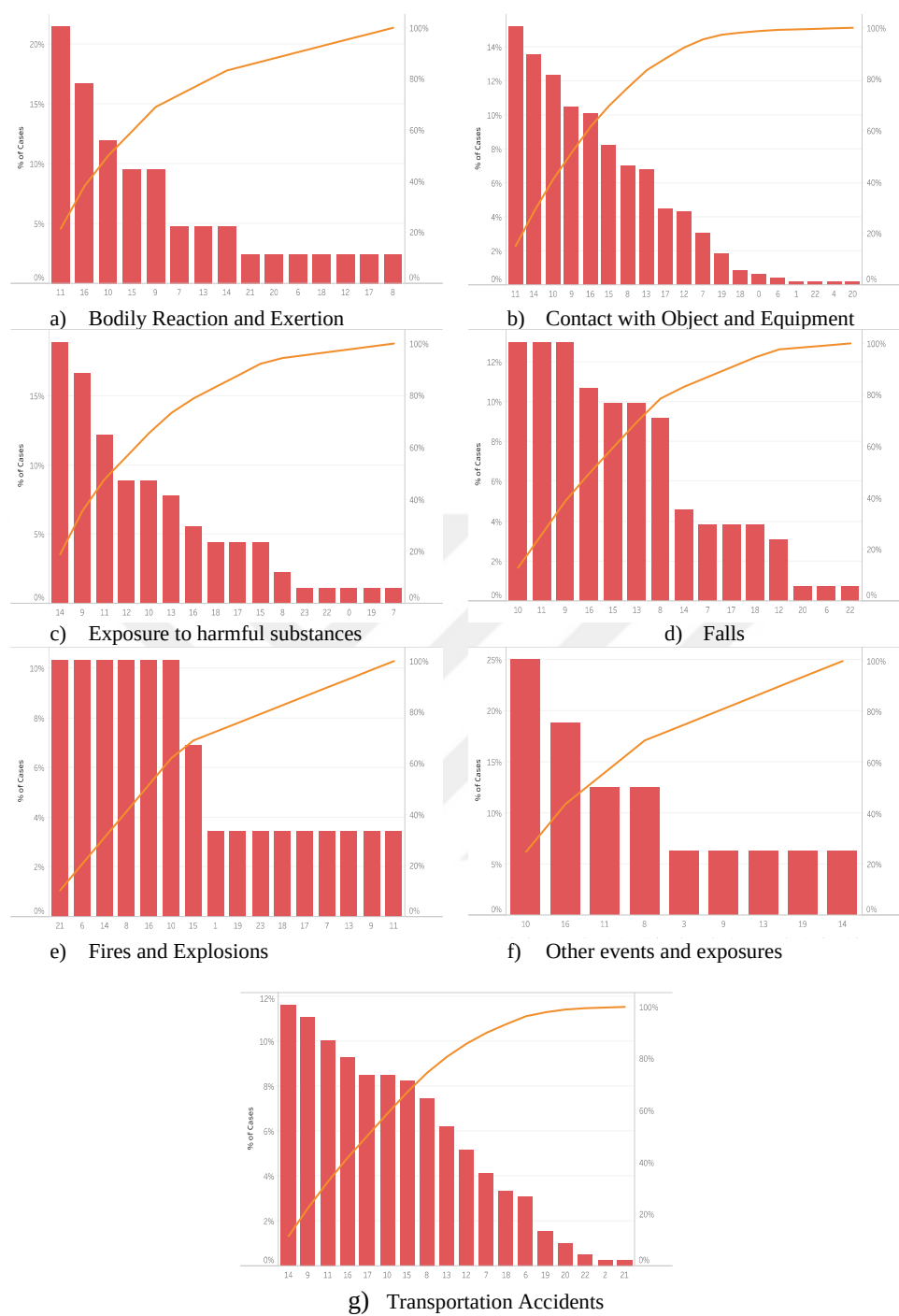


Figure 3.10 Distribution of the cases per type of exposure per hour of day

3.2.2 Distribution of consequences

This section aims to introduce distribution of cases according to their consequences, which are Asset or production damage, Injury and Nearmiss.

It can be inferred from figure 3.11 that almost 48% of all incident were resulted in asset or production damage and injury involved cases accounted almost 43% of all cases. The lowest figure was for nearmiss cases with around 9%.

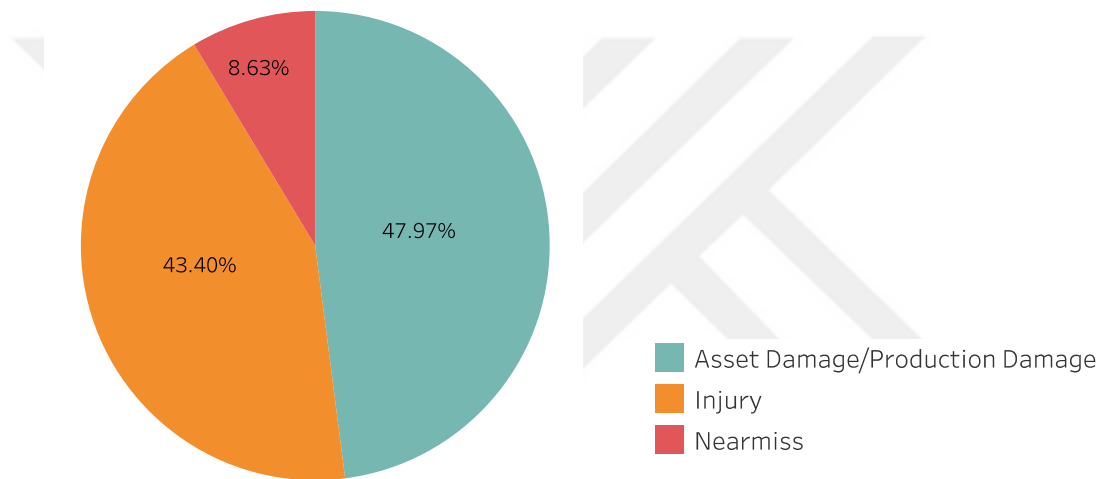
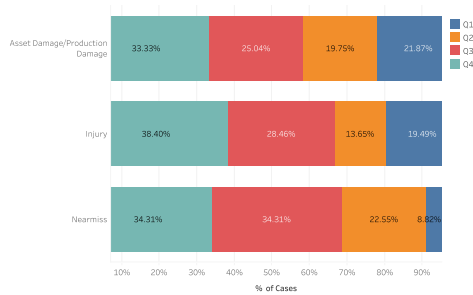
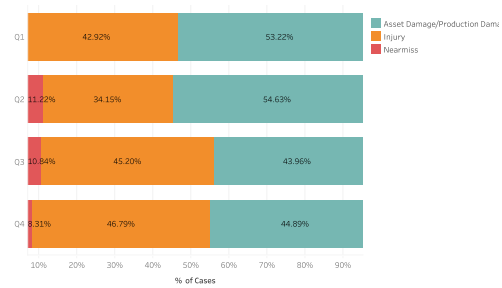


Figure 3.11 Distribution of the cases per type of exposure per hour of day

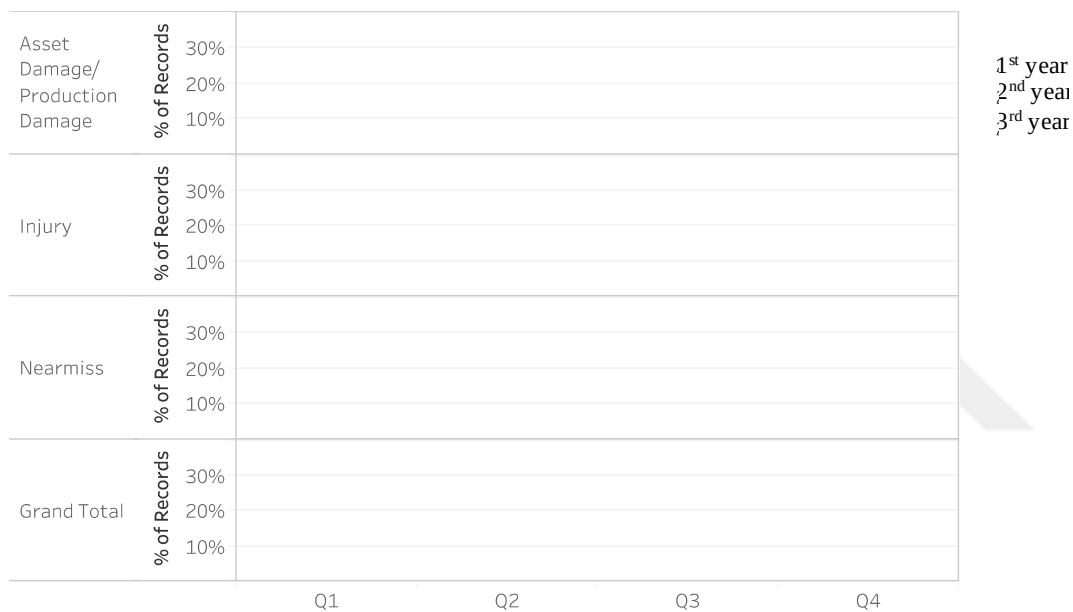
From figure 3.12, it is clear that majority of the all three type of consequence took place in fourth quarter and followed by third quarter. Asset and production damage and injury cases saw a decline in the second quarter compared with first quarter, however, from second to fourth quarter an increasing trend was observed for all three type of consequences. Asset and production damage were higher than injury cases in first and second quarters but in third and fourth quarters injury cases overtook asset or production damage cases.



a)



b)



c)

Figure 3.12 Distribution of the cases per type of consequence per quarter

As seen from figure 3.13, 92% of all falls resulted in injury and 89% of all transportation accident resulted in asset or production damage. Another significant

figure was the result of the contact with objects and equipment with 61% of injury rate.

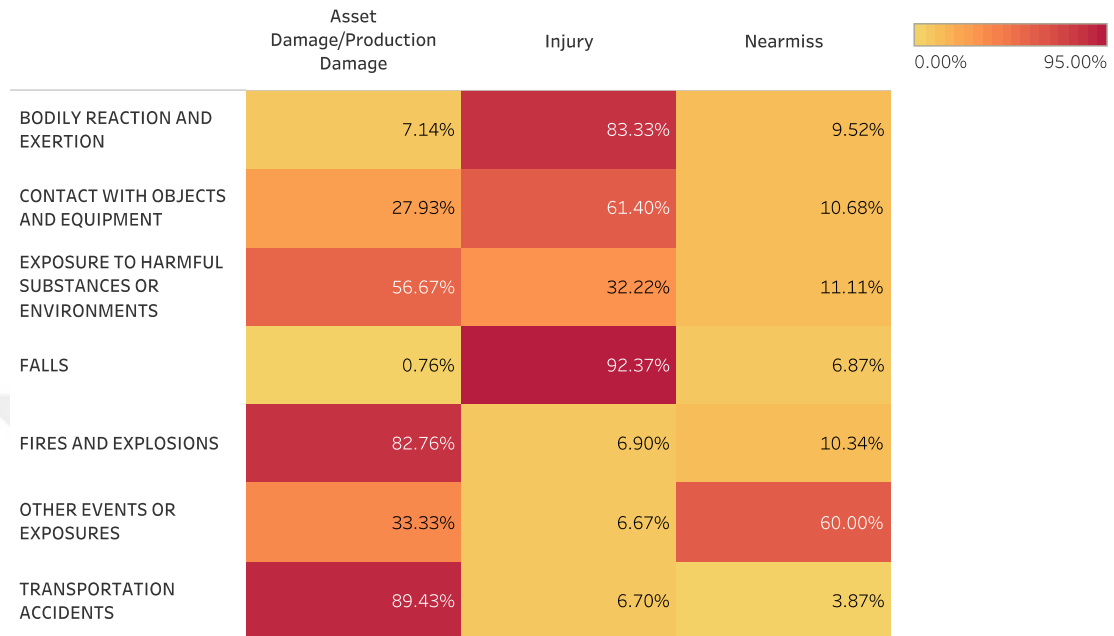


Figure 3.13 Distribution of the cases per type of consequence per type of exposure

Figure 3.14 introduces distribution over month, day of the week, day of the month and hour of the day according to type of consequences. December and November mark the months, Thursday and Friday mark the day of the week on which highest number of asset or production damage and injury took place accordingly. From the same figure it can also be concluded that peak hours of the day were 14 a.m. for asset or production damage and 11 a.m. for injury with 13% and 15% accordingly.

3.2.3 Distribution of injury cases

This section shows the distribution of injury cases over, type of injury time and type of exposure. In figure 3.15, it is clear that first aid cases formed almost 80% of injury cases and medical treatment cases were approximately 14% of injury cases.

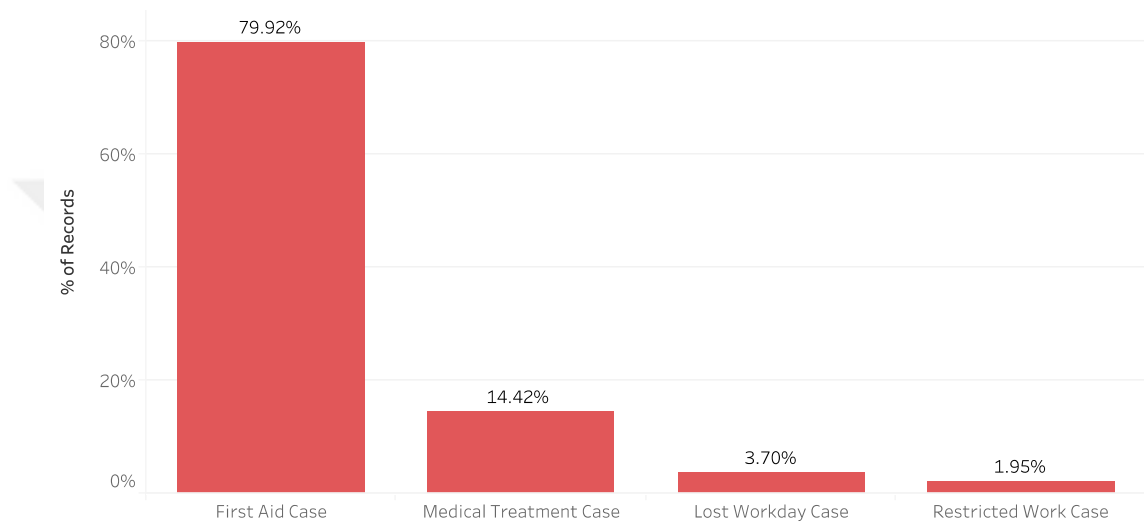


Figure 3.15 Distribution of the injury cases according to type of injury

Figure 3.16 highlights the distribution of the injury cases according to exposure type. Apart from other events and exposures which constituted only 1.35% of all cases, all exposure types led to first aid case. The most significant figure among all exposure types are contact with objects and equipment, and falls because they accounted for more than half of the events with 41% and 11% accordingly. Moreover, 61% of contact with objects and equipment and 92% of falls resulted in injuries, which makes them significant from the point of analysis of type of injury. Transportation accidents were forming almost 33% of all events however only 6.7% resulted in

injury and rest resulted in asset damage or near miss. Because of this, transportation accidents were not significant from the aspect of type of injury analysis.

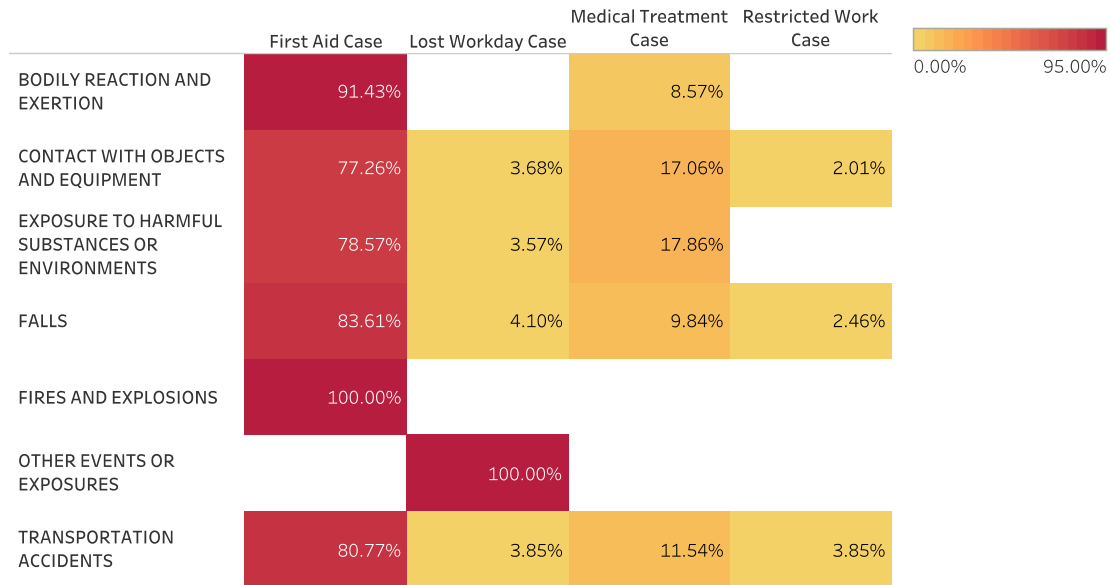


Figure 3.16 Distribution of injury cases according to type of exposure

Only figures for first aid cases is introduced and other type of injuries were eliminated due to being low in number hence giving less information except quarterly distribution. From figure 3.17, first aid and medical treatment cases saw a decrease in the second quarter and saw an increase in the third and fourth quarter with reaching its maximum in the forth quarter.

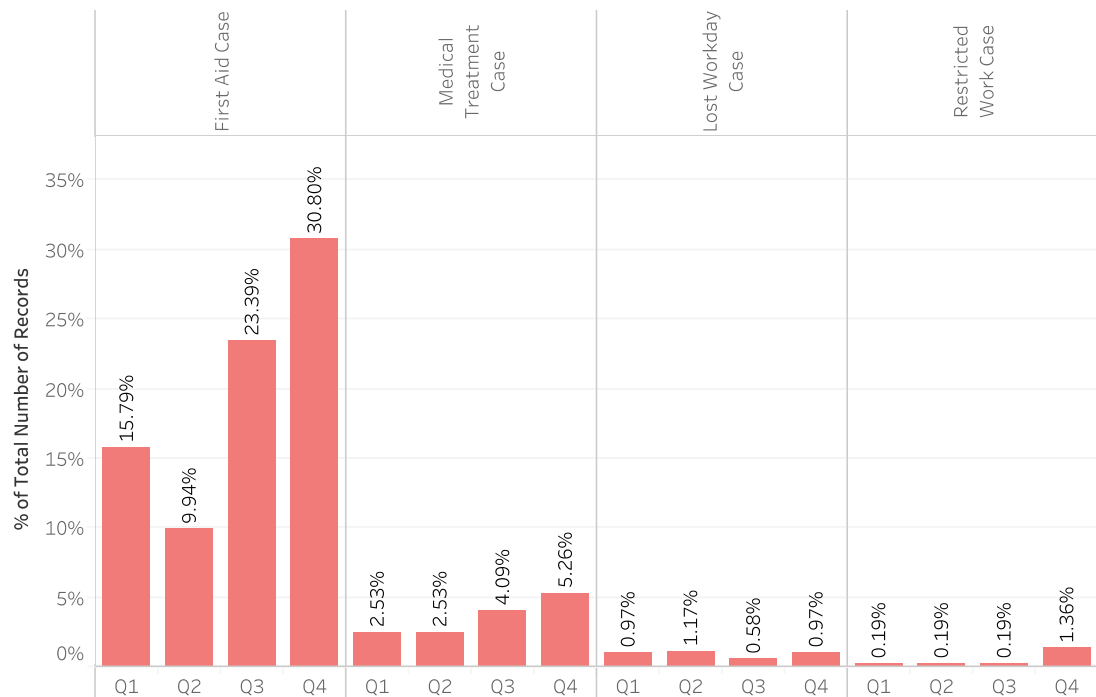


Figure 3.17 Distribution of injury cases per quarter

Distribution of injury cases over month and day of the week, figure 3.18 and 3.19 shows that Novembers were very active in terms of first aid cases and Friday, Thursdays, Wednesdays, and Tuesdays were above average with being around 17%. Sundays and May were the day of the week and months a on which least number of first aid cases were seen accordingly.

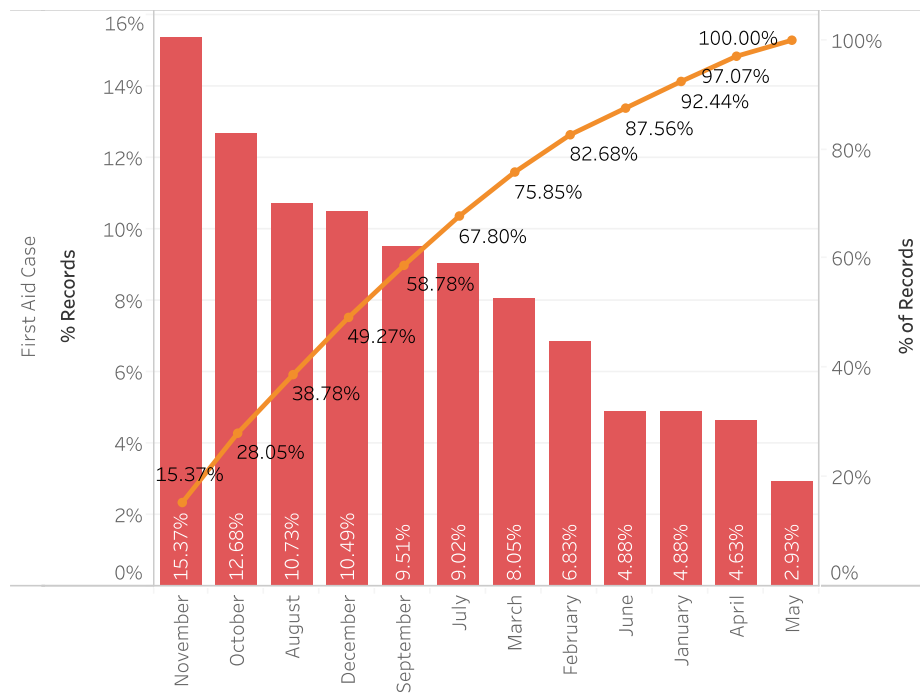


Figure 3.18 Monthly Distribution of first aid cases

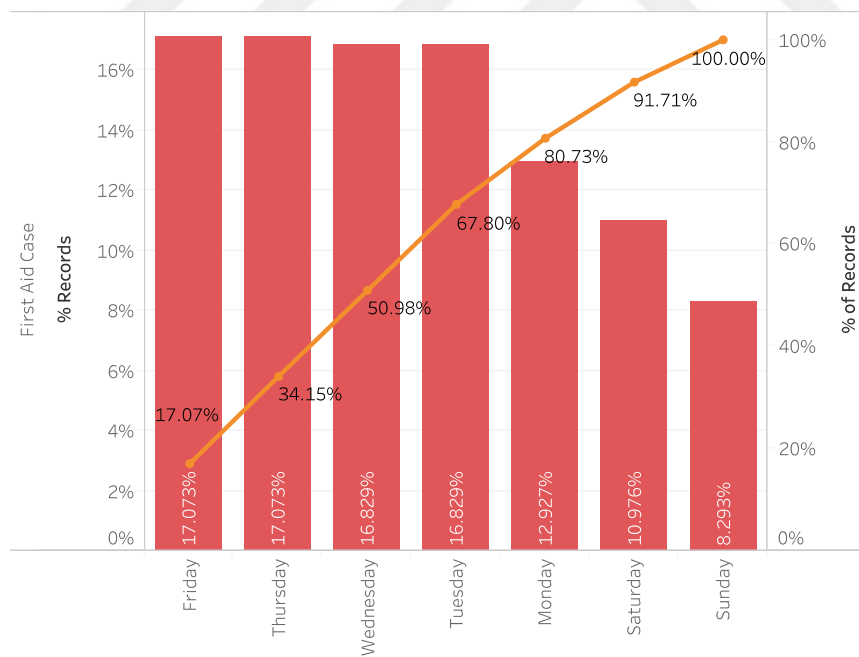


Figure 3.19 Distribution of first aid cases per day of the week

3.2.4 Distribution of causes

This section introduces distribution immediate and root cause over date and time, and over other dimensions. Distribution of immediate causes are listed first in the section.

Figure 3.20 and Table 3.4 show that most common immediate causes are IC15, IC3, IC13 and IC14 with 33%, 30%, 23%, and 19% accordingly. IC5, IC2, IC17, IC10, IC6, IC21, IC18, IC8, IC7, IC11, IC23 and IC9 change between 5% to 12%. All other immediate causes have values between 0% to 3% which were very rare.

Table 3.4 Distribution of immediate causes

IMMEDIATE CAUSE	%	IMMEDIATE CAUSE (SORTED)	%
IC1	0.59%	IC15	33.28%
IC2	11.82%	IC3	30.32%
IC3	30.32%	IC13	23.23%
IC4	1.27%	IC14	18.92%
IC5	12.16%	IC5	12.16%
IC6	9.12%	IC2	11.82%
IC7	6.08%	IC17	10.05%
IC8	7.69%	IC10	9.71%
IC9	4.73%	IC6	9.12%
IC10	9.71%	IC21	8.95%
IC11	5.41%	IC18	8.02%
IC12	0.34%	IC8	7.69%
IC13	23.23%	IC7	6.08%
IC14	18.92%	IC11	5.41%
IC15	33.28%	IC23	5.15%
IC16	2.79%	IC9	4.73%
IC17	10.05%	IC19	3.38%
IC18	8.02%	IC16	2.79%
IC19	3.38%	IC4	1.27%
IC20	0.93%	IC20	0.93%
IC21	8.95%	IC24	0.68%
IC22	0.42%	IC1	0.59%
IC23	5.15%	IC22	0.42%
IC24	0.68%	IC12	0.34%

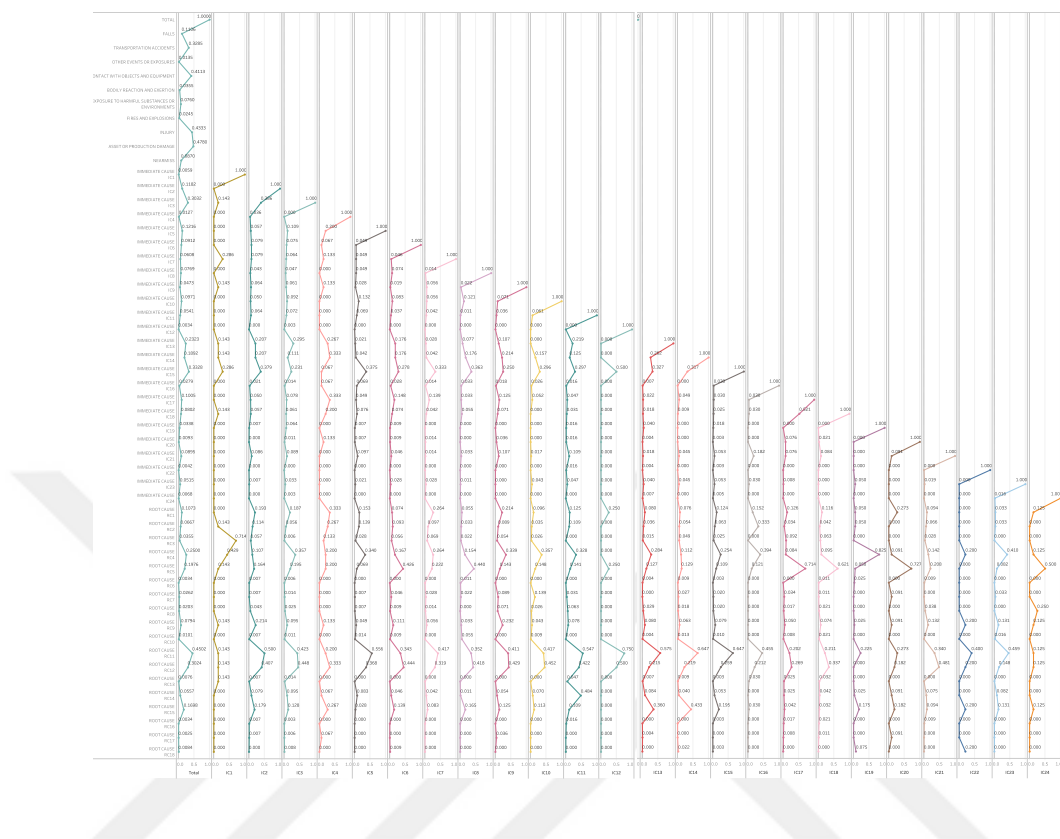


Figure 3.20 Distribution of immediate causes over other attributes

Table 3.5 shows the percentage share of immediate causes for each type of exposure. Since the number of cases are not significant for some of the exposure types, the distribution of the cause for falls, transportation, contact with objects and equipment, and exposure to harmful substances or environments cases are selected to be introduced. Out of 24 immediate causes, IC15, IC3, IC13, IC14, IC5, IC2, and IC17 were the top seven causes underlying fall accidents, with above 10% each. However, in transportation accidents, lineup changes slightly and top five causes become IC13, IC3, IC15, IC14 and IC2 with each being more than 10%, which are also shared

causes in falls. IC15, IC5, IC3, IC10, IC21, IC17, IC14, IC18, and IC2 are the causes which played important role in contact with objects and equipment and IC3, IC2, IC15, IC6, IC17, IC7, and IC13 for exposure to harmful substances or environments cases with each having shares more than 10%.



Table 3.5 Immediate Causes and Type of Exposure

IMMEDIATE CAUSE															
IC15	TOTAL (%)	IC15	FALLS (%)	IC13	TRANSPORTATION ACCIDENTS (%)	IC3	OTHER EVENTS OR EXPOSURES (%)	IC15	CONTACT WITH OBJECTS AND EQUIPMENT (%)	IC15	BODILY REACTION AND EXERTION (%)	IC3	EXPOSURE TO HARMFUL SUBSTANCES OR ENVIRONMENTS (%)	IC18	FIRES AND EXPLOSIONS (%)
	33.3%		43.8%		63.8%		56.3%		31.4%		50.0%		40.0%		65.5%
IC3	30.3%	IC14	28.9%	IC3	38.8%	IC21	12.5%	IC5	25.7%	IC14	26.2%	IC2	36.7%	IC17	58.6%
IC13	23.2%	IC3	26.4%	IC15	34.7%	IC24	12.5%	IC3	24.8%	IC8	21.4%	IC15	30.0%	IC3	24.1%
IC14	18.9%	IC8	24.0%	IC14	27.0%	IC1	6.3%	IC10	16.8%	IC6	16.7%	IC6	16.7%	IC15	17.2%
IC5	12.2%	IC23	17.4%	IC2	11.8%	IC2	6.3%	IC21	16.0%	IC10	14.3%	IC17	14.4%	IC2	10.3%
IC2	11.8%	IC6	14.9%	IC19	IC13	IC6	6.3%	IC17	14.0%	IC5	11.9%	IC7	13.3%	IC8	10.3%
IC17	10.1%	IC10	14.9%	IC6	IC3	IC7	6.3%	IC14	12.7%	IC23	11.9%	IC13	10.0%	IC14	10.3%
IC10	9.7%	IC7	7.4%	IC11	4.9%	IC8	6.3%	IC18	10.9%	IC2	7.1%	IC10	8.9%	IC6	6.9%
IC6	9.1%	IC9	7.4%	IC21	4.4%	IC9	6.3%	IC2	10.3%	IC3	7.1%	IC14	8.9%	IC21	6.9%
IC21	9.0%	IC5	5.0%	IC23	4.1%	IC17	6.3%	IC6	8.4%	IC11	4.8%	IC18	7.8%	IC7	3.4%
IC18	8.0%	IC11	5.0%	IC17	3.9%	IC20	6.3%	IC7	7.4%	IC16	4.8%	IC11	5.6%	IC9	3.4%
IC8	7.7%	IC2	3.3%	IC8	3.6%	IC4	0.0%	IC8	7.2%	IC24	4.8%	IC5	4.4%	IC10	3.4%
IC7	6.1%	IC17	3.3%	IC9	3.6%	IC5	0.0%	IC11	6.4%	IC1	2.4%	IC23	3.3%	IC11	3.4%
IC11	5.4%	IC21	3.3%	IC18	3.6%	IC10	0.0%	IC9	6.0%	IC7	2.4%	IC9	2.2%	IC16	3.4%
IC23	5.2%	IC16	2.5%	IC7	3.1%	IC11	0.0%	IC16	4.3%	IC17	2.4%	IC12	2.2%	IC1	0.0%
IC9	4.7%	IC18	1.7%	IC16	1.3%	IC12	0.0%	IC13	3.7%	IC21	2.4%	IC20	2.2%	IC4	0.0%
IC19	3.4%	IC24	1.7%	IC4	1.0%	IC13	0.0%	IC23	3.3%	IC4	0.0%	IC21	2.2%	IC5	0.0%
IC16	2.8%	IC1	0.0%	IC5	1.0%	IC14	0.0%	IC4	2.3%	IC9	0.0%	IC22	2.2%	IC12	0.0%
IC4	1.3%	IC4	0.0%	IC12	0.5%	IC15	0.0%	IC20	1.4%	IC12	0.0%	IC16	1.1%	IC13	0.0%
IC20	0.9%	IC12	0.0%	IC22	0.5%	IC16	0.0%	IC1	0.8%	IC13	0.0%	IC19	1.1%	IC19	0.0%
IC24	0.7%	IC13	0.0%	IC24	0.5%	IC18	0.0%	IC19	0.2%	IC18	0.0%	IC1	0.0%	IC20	0.0%
IC1	0.6%	IC19	0.0%	IC1	0.3%	IC19	0.0%	IC22	0.2%	IC19	0.0%	IC4	0.0%	IC22	0.0%

IC15, IC5, IC3, IC10, IC14, and IC8 play important role in injury cases that can be derived from Table 3.6. From the same table, top two causes for asset and production damage cases are different in injury cases. IC13, IC2, IC17, and IC18 are 4 causes out of seven highest causes in asset and production damage cases that are different than top six causes of injury cases. IC17, IC21, and IC6 are the three of top five causes, and for nearmiss cases that are unshared by injury cases. Compared with asset and production damage cases, only IC21 and IC6 are different for nearmiss cases.

Table 3.6 Immediate Causes and Type of Consequence

<u>IMMEDIATE</u> <u>CAUSE</u> <u>(SORTED)</u>	<u>INJURY</u>	<u>IC</u> <u>(SORTED)</u>	<u>ASSET OR</u> <u>PRODUCTION</u> <u>DAMAGE</u>	<u>IC</u> <u>(SORTED)</u>	<u>NEARMISS</u>
IC15	40.2%	IC13	42.6%	IC3	39.8%
IC5	26.5%	IC3	34.3%	IC15	19.4%
IC3	24.2%	IC15	29.7%	IC17	16.5%
IC10	21.4%	IC14	23.5%	IC21	16.5%
IC14	16.2%	IC2	18.4%	IC6	15.5%
IC8	12.1%	IC17	12.9%	IC18	9.7%
IC6	8.8%	IC18	11.8%	IC8	8.7%
IC23	8.2%	IC21	8.5%	IC13	7.8%
IC21	8.0%	IC6	8.3%	IC11	6.8%
IC7	6.4%	IC7	6.0%	IC14	6.8%
IC11	6.0%	IC19	6.0%	IC2	5.8%
IC2	5.7%	IC9	4.6%	IC5	3.9%
IC9	5.7%	IC11	4.6%	IC7	3.9%
IC17	5.7%	IC8	3.5%	IC10	3.9%
IC13	4.9%	IC23	3.2%	IC19	1.9%
IC16	4.5%	IC16	1.6%	IC9	1.0%
IC18	3.5%	IC4	1.4%	IC16	1.0%
IC4	1.2%	IC20	1.1%	IC23	1.0%
IC24	1.2%	IC1	0.7%	IC1	0.0%
IC20	1.0%	IC5	0.7%	IC4	0.0%
IC19	0.8%	IC12	0.4%	IC12	0.0%
IC1	0.6%	IC22	0.4%	IC20	0.0%
IC22	0.6%	IC10	0.2%	IC22	0.0%
IC12	0.4%	IC24	0.2%	IC24	0.0%

The relations between immediate cause and other immediate causes are introduced in Table 3.7. Other immediate causes underlying IC15 were IC13, IC3, IC14, IC5, and IC2 that are above 10% individually. For IC3, IC13, IC15, IC2, IC14, and IC5 are the top five contributing immediate cause. IC3, IC15, IC14 and IC2 are top four causes seen in cases with IC13.



Table 3.7 Immediate Causes and Other Immediate Causes

IC1	IC2	IC3	IC4	IC5	IC6	IC7	IC8	IC9	IC10	IC11	IC12
IC1(100.0%)	IC2(100.0%)	IC3(100.0%)	IC4(100.0%)	IC5(100.0%)	IC6(100.0%)	IC7(100.0%)	IC8(100.0%)	IC9(100.0%)	IC10(100.0%)	IC11(100.0%)	IC12(100.0%)
IC7(28.6%)	IC3(38.6%)	IC13(29.5%)	IC2(33.3%)	IC15(37.5%)	IC15(27.8%)	IC15(33.3%)	IC15(36.3%)	IC3(39.3%)	IC15(29.6%)	IC3(40.6%)	IC15(50.0%)
IC15(28.6%)	IC15(37.9%)	IC15(23.1%)	IC14(33.3%)	IC3(27.1%)	IC3(25.0%)	IC3(31.9%)	IC3(18.7%)	IC15(25.0%)	IC3(28.7%)	IC15(29.7%)	IC3(25.0%)
IC3(14.3%)	IC13(20.7%)	IC2(15.0%)	IC17(33.3%)	IC10(13.2%)	IC13(17.6%)	IC2(15.3%)	IC14(17.6%)	IC14(21.4%)	IC5(16.5%)	IC13(21.9%)	IC1(0.0%)
IC9(14.3%)	IC14(20.7%)	IC14(11.1%)	IC13(26.7%)	IC21(9.7%)	IC14(17.6%)	IC17(13.9%)	IC10(12.1%)	IC2(16.1%)	IC14(15.7%)	IC5(15.6%)	IC2(0.0%)
IC13(14.3%)	IC21(8.6%)	IC5(10.9%)	IC5(20.0%)	IC18(7.6%)	IC17(14.8%)	IC5(9.7%)	IC6(8.8%)	IC17(12.5%)	IC8(9.6%)	IC2(14.1%)	IC4(0.0%)
IC14(14.3%)	IC6(7.9%)	IC10(9.2%)	IC18(20.0%)	IC11(6.9%)	IC2(10.2%)	IC6(6.9%)	IC5(7.7%)	IC13(10.7%)	IC6(7.8%)	IC14(12.5%)	IC5(0.0%)
IC18(14.3%)	IC7(7.9%)	IC21(8.9%)	IC7(13.3%)	IC16(6.9%)	IC10(8.3%)	IC9(5.6%)	IC13(7.7%)	IC21(10.7%)	IC2(6.1%)	IC10(10.9%)	IC6(0.0%)
IC2(0.0%)	IC9(6.4%)	IC17(7.8%)	IC9(13.3%)	IC2(5.6%)	IC8(7.4%)	IC10(5.6%)	IC2(6.6%)	IC5(7.1%)	IC11(6.1%)	IC21(10.9%)	IC7(0.0%)
IC4(0.0%)	IC11(6.4%)	IC6(7.5%)	IC20(13.3%)	IC6(4.9%)	IC18(7.4%)	IC11(4.2%)	IC18(5.5%)	IC7(7.1%)	IC17(5.2%)	IC6(6.3%)	IC8(0.0%)
IC5(0.0%)	IC5(5.7%)	IC11(7.2%)	IC6(6.7%)	IC7(4.9%)	IC5(6.5%)	IC14(4.2%)	IC16(3.3%)	IC10(7.1%)	IC23(4.3%)	IC7(4.7%)	IC9(0.0%)
IC6(0.0%)	IC18(5.7%)	IC7(6.4%)	IC15(6.7%)	IC8(4.9%)	IC7(4.6%)	IC18(4.2%)	IC17(3.3%)	IC18(7.1%)	IC7(3.5%)	IC17(4.7%)	IC10(0.0%)
IC8(0.0%)	IC10(5.0%)	IC19(6.4%)	IC16(6.7%)	IC17(4.9%)	IC21(4.6%)	IC1(2.8%)	IC21(3.3%)	IC4(3.6%)	IC9(3.5%)	IC23(4.7%)	IC11(0.0%)
IC10(0.0%)	IC17(5.0%)	IC9(6.1%)	IC1(0.0%)	IC14(4.2%)	IC11(3.7%)	IC4(2.8%)	IC9(2.2%)	IC6(3.6%)	IC16(2.6%)	IC9(3.1%)	IC13(0.0%)
IC11(0.0%)	IC8(4.3%)	IC18(6.1%)	IC3(0.0%)	IC9(2.8%)	IC16(2.8%)	IC13(2.8%)	IC7(1.1%)	IC8(3.6%)	IC21(1.7%)	IC18(3.1%)	IC14(0.0%)
IC12(0.0%)	IC4(3.6%)	IC8(4.7%)	IC8(0.0%)	IC4(2.1%)	IC23(2.8%)	IC23(2.8%)	IC11(1.1%)	IC11(3.6%)	IC1(0.0%)	IC8(1.6%)	IC16(0.0%)
IC16(0.0%)	IC16(2.1%)	IC23(3.3%)	IC10(0.0%)	IC13(2.1%)	IC9(1.9%)	IC8(1.4%)	IC23(1.1%)	IC20(3.6%)	IC4(0.0%)	IC16(1.6%)	IC17(0.0%)
IC17(0.0%)	IC19(0.7%)	IC16(1.4%)	IC11(0.0%)	IC23(2.1%)	IC4(0.9%)	IC16(1.4%)	IC1(0.0%)	IC1(1.8%)	IC12(0.0%)	IC19(1.6%)	IC18(0.0%)
IC19(0.0%)	IC23(0.7%)	IC20(1.1%)	IC12(0.0%)	IC19(0.7%)	IC19(0.9%)	IC20(1.4%)	IC4(0.0%)	IC16(1.8%)	IC13(0.0%)	IC20(1.6%)	IC19(0.0%)
IC20(0.0%)	IC1(0.0%)	IC1(0.3%)	IC19(0.0%)	IC20(0.7%)	IC20(0.9%)	IC21(1.4%)	IC12(0.0%)	IC12(0.0%)	IC18(0.0%)	IC12(0.0%)	IC20(0.0%)
IC21(0.0%)	IC12(0.0%)	IC12(0.3%)	IC21(0.0%)	IC1(0.0%)	IC1(0.0%)	IC12(0.0%)	IC19(0.0%)	IC19(0.0%)	IC19(0.0%)	IC1(0.0%)	IC21(0.0%)
IC22(0.0%)	IC20(0.0%)	IC24(0.3%)	IC22(0.0%)	IC12(0.0%)	IC12(0.0%)	IC19(0.0%)	IC20(0.0%)	IC22(0.0%)	IC20(0.0%)	IC4(0.0%)	IC22(0.0%)
IC23(0.0%)	IC22(0.0%)	IC4(0.0%)	IC23(0.0%)	IC22(0.0%)	IC22(0.0%)	IC22(0.0%)	IC22(0.0%)	IC23(0.0%)	IC22(0.0%)	IC12(0.0%)	IC23(0.0%)
IC24(0.0%)	IC24(0.0%)	IC22(0.0%)	IC24(0.0%)	IC24(0.0%)	IC24(0.0%)	IC24(0.0%)	IC24(0.0%)	IC24(0.0%)	IC24(0.0%)	IC24(0.0%)	IC24(0.0%)
IC13	IC14	IC15	IC16	IC17	IC18	IC19	IC20	IC21	IC22	IC23	IC24
IC13(100.0%)	IC14(100.0%)	IC15(100.0%)	IC16(100.0%)	IC17(100.0%)	IC18(100.0%)	IC19(100.0%)	IC20(100.0%)	IC21(100.0%)	IC22(100.0%)	IC23(100.0%)	IC24(100.0%)
IC3(38.5%)	IC13(32.1%)	IC13(22.8%)	IC15(36.4%)	IC18(52.1%)	IC17(65.3%)	IC3(57.5%)	IC17(81.8%)	IC3(30.2%)	IC11(20.0%)	IC15(34.4%)	IC13(25.0%)
IC15(32.7%)	IC15(31.7%)	IC3(21.1%)	IC5(30.3%)	IC3(23.2%)	IC13(27.5%)	IC3(36.4%)	IC15(19.8%)	IC13(20.0%)	IC19(19.7%)	IC15(25.0%)	
IC14(26.2%)	IC3(17.9%)	IC14(18.0%)	IC21(18.2%)	IC6(13.4%)	IC5(11.6%)	IC15(17.5%)	IC4(18.2%)	IC5(13.2%)	IC15(20.0%)	IC13(18.0%)	IC3(12.5%)
IC2(10.5%)	IC2(12.9%)	IC5(13.7%)	IC3(15.2%)	IC15(10.1%)	IC15(10.5%)	IC23(5.0%)	IC9(18.2%)	IC2(11.3%)	IC1(0.0%)	IC14(16.4%)	IC23(12.5%)
IC6(6.9%)	IC6(8.5%)	IC2(13.5%)	IC2(9.1%)	IC14(9.2%)	IC2(8.4%)	IC2(2.5%)	IC18(18.2%)	IC14(9.4%)	IC2(0.0%)	IC1(8.2%)	IC1(0.0%)
IC11(5.1%)	IC10(8.0%)	IC10(8.6%)	IC6(9.1%)	IC7(8.4%)	IC6(8.4%)	IC5(2.5%)	IC5(9.1%)	IC17(8.5%)	IC3(0.0%)	IC5(4.9%)	IC2(0.0%)
IC19(4.0%)	IC8(7.1%)	IC8(8.4%)	IC8(9.1%)	IC20(7.6%)	IC21(8.4%)	IC6(2.5%)	IC6(9.1%)	IC18(7.5%)	IC4(0.0%)	IC6(4.9%)	IC4(0.0%)
IC23(4.0%)	IC9(5.4%)	IC6(7.6%)	IC10(9.1%)	IC21(7.6%)	IC8(5.3%)	IC11(2.5%)	IC7(9.1%)	IC11(6.6%)	IC5(0.0%)	IC11(4.9%)	IC5(0.0%)
IC8(2.5%)	IC17(4.9%)	IC7(6.1%)	IC13(6.1%)	IC2(5.9%)	IC13(5.3%)	IC1(0.0%)	IC11(9.1%)	IC9(5.7%)	IC6(0.0%)	IC7(3.3%)	IC6(0.0%)
IC9(2.2%)	IC21(4.5%)	IC21(5.3%)	IC4(3.0%)	IC5(5.9%)	IC9(4.2%)	IC4(0.0%)	IC13(9.1%)	IC16(5.7%)	IC7(0.0%)	IC19(3.3%)	IC7(0.0%)
IC17(2.2%)	IC23(4.5%)	IC23(5.3%)	IC7(3.0%)	IC9(5.9%)	IC4(3.2%)	IC7(0.0%)	IC15(9.1%)	IC6(4.7%)	IC8(0.0%)	IC21(3.3%)	IC8(0.0%)
IC18(1.8%)	IC11(3.6%)	IC11(4.8%)	IC9(3.0%)	IC10(5.0%)	IC7(3.2%)	IC8(0.0%)	IC21(9.1%)	IC13(4.7%)	IC9(0.0%)	IC2(1.6%)	IC9(0.0%)
IC21(1.8%)	IC5(2.7%)	IC9(3.6%)	IC11(3.0%)	IC13(5.0%)	IC11(2.1%)	IC9(0.0%)	IC1(0.0%)	IC8(2.8%)	IC10(0.0%)	IC8(1.6%)	IC10(0.0%)
IC4(1.5%)	IC4(2.2%)	IC16(3.0%)	IC17(3.0%)	IC4(4.2%)	IC14(2.1%)	IC10(0.0%)	IC2(0.0%)	IC10(1.9%)	IC12(0.0%)	IC16(1.6%)	IC11(0.0%)
IC5(1.1%)	IC7(1.3%)	IC17(3.0%)	IC18(3.0%)	IC8(2.5%)	IC20(2.1%)	IC12(0.0%)	IC8(0.0%)	IC23(1.9%)	IC14(0.0%)	IC17(1.6%)	IC12(0.0%)
IC7(0.7%)	IC18(0.9%)	IC18(2.5%)	IC23(3.0%)	IC11(2.5%)	IC1(1.1%)	IC14(0.0%)	IC10(0.0%)	IC7(0.9%)	IC16(0.0%)	IC24(1.6%)	IC14(0.0%)
IC16(0.7%)	IC1(0.4%)	IC19(1.8%)	IC1(0.0%)	IC16(0.8%)	IC16(1.1%)	IC16(0.0%)	IC12(0.0%)	IC20(0.9%)	IC17(0.0%)	IC1(0.0%)	IC16(0.0%)
IC24(0.7%)	IC12(0.0%)	IC1(0.5%)	IC12(0.0%)	IC23(0.8%)	IC10(0.0%)	IC17(0.0%)	IC14(0.0%)	IC1(0.0%)	IC18(0.0%)	IC4(0.0%)	IC17(0.0%)
IC1(0.4%)	IC16(0.0%)	IC12(0.5%)	IC14(0.0%)	IC1(0.0%)	IC12(0.0%)	IC18(0.0%)	IC16(0.0%)	IC4(0.0%)	IC19(0.0%)	IC9(0.0%)	IC18(0.0%)
IC20(0.4%)	IC19(0.0%)	IC24(0.5%)	IC19(0.0%)	IC12(0.0%)	IC19(0.0%)	IC20(0.0%)	IC19(0.0%)	IC12(0.0%)	IC20(0.0%)	IC12(0.0%)	IC19(0.0%)
IC22(0.4%)	IC20(0.0%)	IC4(0.3%)	IC20(0.0%)	IC19(0.0%)	IC22(0.0%)	IC21(0.0%)	IC22(0.0%)	IC19(0.0%)	IC21(0.0%)	IC18(0.0%)	IC20(0.0%)
IC10(0.0%)	IC22(0.0%)	IC20(0.3%)	IC22(0.0%)	IC22(0.0%)	IC23(0.0%)	IC22(0.0%)	IC23(0.0%)	IC22(0.0%)	IC23(0.0%)	IC20(0.0%)	IC21(0.0%)
IC12(0.0%)	IC24(0.0%)	IC22(0.3%)	IC24(0.0%)	IC24(0.0%)	IC24(0.0%)	IC24(0.0%)	IC24(0.0%)	IC24(0.0%)	IC24(0.0%)	IC22(0.0%)	IC22(0.0%)

The relations between immediate cause and root causes are introduced in Table 3.8. Root causes underlying IC15 were RC11, RC12, RC4, RC15, RC1, and RC5 that are above 10% individually. For IC3, RC12, RC11, RC4, RC5, RC1 and RC15 are the top six causes. In cases with IC13, IC14, IC5, IC2, and IC17 immediate causes, the root causes are similar that of IC15 and IC3 as seen in table 2.10.



Table 3.8 Immediate Causes and Root Causes

IC1	IC2	IC3	IC4	IC5	IC6	IC7	IC8	IC9	IC10	IC11	IC12
RC3(71.4%)	RC11(50.0%)	RC12(44.8%)	RC1(33.3%)	RC11(55.6%)	RC12(44.4%)	RC11(41.7%)	RC5(44.0%)	RC12(42.9%)	RC12(45.2%)	RC11(54.7%)	RC11(75.0%)
RC4(42.9%)	RC12(40.7%)	RC11(42.3%)	RC12(33.3%)	RC12(36.8%)	RC5(42.6%)	RC12(31.9%)	RC12(41.8%)	RC11(41.1%)	RC11(41.7%)	RC14(48.4%)	RC12(50.0%)
RC2(14.3%)	RC9(21.4%)	RC4(35.7%)	RC2(26.7%)	RC4(34.0%)	RC11(34.3%)	RC1(26.4%)	RC11(35.2%)	RC4(33.9%)	RC4(35.7%)	RC12(42.2%)	RC1(25.0%)
RC5(14.3%)	RC1(19.3%)	RC5(19.5%)	RC15(26.7%)	RC1(15.3%)	RC4(16.7%)	RC4(26.4%)	RC15(16.5%)	RC9(23.2%)	RC5(14.8%)	RC4(32.8%)	RC5(25.0%)
RC9(14.3%)	RC15(17.9%)	RC1(18.7%)	RC4(20.0%)	RC2(13.9%)	RC15(13.9%)	RC5(22.2%)	RC4(15.4%)	RC1(21.4%)	RC7(13.9%)	RC5(14.1%)	RC2(0.0%)
RC11(14.3%)	RC5(16.4%)	RC15(12.8%)	RC5(20.0%)	RC14(8.3%)	RC9(11.1%)	RC2(9.7%)	RC1(5.5%)	RC5(14.3%)	RC15(11.3%)	RC1(12.5%)	RC3(0.0%)
RC12(14.3%)	RC2(11.4%)	RC9(9.5%)	RC11(20.0%)	RC5(6.9%)	RC2(9.3%)	RC15(8.3%)	RC10(5.5%)	RC15(12.5%)	RC1(9.6%)	RC2(10.9%)	RC4(0.0%)
RC13(14.3%)	RC4(10.7%)	RC14(9.5%)	RC3(13.3%)	RC9(4.9%)	RC1(7.4%)	RC3(6.9%)	RC2(3.3%)	RC2(8.9%)	RC14(7.0%)	RC15(10.9%)	RC6(0.0%)
RC1(0.0%)	RC14(7.9%)	RC2(5.6%)	RC9(13.3%)	RC3(2.8%)	RC3(5.6%)	RC9(5.6%)	RC9(3.3%)	RC7(8.9%)	RC9(4.3%)	RC9(7.8%)	RC7(0.0%)
RC6(0.0%)	RC3(5.7%)	RC8(2.5%)	RC14(6.7%)	RC15(2.8%)	RC7(4.6%)	RC14(4.2%)	RC3(2.2%)	RC8(7.1%)	RC2(3.5%)	RC8(6.3%)	RC8(0.0%)
RC7(0.0%)	RC8(4.3%)	RC7(1.4%)	RC17(6.7%)	RC10(1.4%)	RC14(4.6%)	RC7(2.8%)	RC7(2.2%)	RC3(5.4%)	RC3(2.6%)	RC13(4.7%)	RC9(0.0%)
RC8(0.0%)	RC6(0.7%)	RC13(1.4%)	RC6(0.0%)	RC7(0.7%)	RC8(0.9%)	RC8(1.4%)	RC6(1.1%)	RC14(5.4%)	RC8(2.6%)	RC7(3.1%)	RC10(0.0%)
RC10(0.0%)	RC7(0.7%)	RC10(1.1%)	RC7(0.0%)	RC8(0.7%)	RC10(0.9%)	RC6(0.0%)	RC14(1.1%)	RC17(3.6%)	RC10(0.9%)	RC16(1.6%)	RC13(0.0%)
RC14(0.0%)	RC10(0.7%)	RC18(0.8%)	RC8(0.0%)	RC6(0.0%)	RC18(0.9%)	RC10(0.0%)	RC8(0.0%)	RC6(0.0%)	RC6(0.0%)	RC3(0.0%)	RC14(0.0%)
RC15(0.0%)	RC13(0.7%)	RC3(0.6%)	RC10(0.0%)	RC13(0.0%)	RC6(0.0%)	RC13(0.0%)	RC13(0.0%)	RC10(0.0%)	RC13(0.0%)	RC6(0.0%)	RC15(0.0%)
RC16(0.0%)	RC16(0.7%)	RC6(0.6%)	RC13(0.0%)	RC16(0.0%)	RC13(0.0%)	RC16(0.0%)	RC16(0.0%)	RC13(0.0%)	RC16(0.0%)	RC10(0.0%)	RC16(0.0%)
RC17(0.0%)	RC17(0.7%)	RC17(0.6%)	RC16(0.0%)	RC17(0.0%)	RC16(0.0%)	RC17(0.0%)	RC17(0.0%)	RC16(0.0%)	RC17(0.0%)	RC17(0.0%)	RC17(0.0%)
RC18(0.0%)	RC18(0.0%)	RC16(0.3%)	RC18(0.0%)	RC18(0.0%)	RC17(0.0%)	RC18(0.0%)	RC18(0.0%)	RC18(0.0%)	RC18(0.0%)	RC18(0.0%)	RC18(0.0%)
IC13	IC14	IC15	IC16	IC17	IC18	IC19	IC20	IC21	IC22	IC23	IC24
RC11(57.5%)	RC11(64.7%)	RC11(64.7%)	RC11(45.5%)	RC5(71.4%)	RC5(62.1%)	RC4(82.5%)	RC5(72.7%)	RC12(48.1%)	RC11(40.0%)	RC11(45.9%)	RC5(50.0%)
RC15(36.0%)	RC15(43.3%)	RC12(25.9%)	RC4(39.4%)	RC12(26.9%)	RC12(33.7%)	RC11(22.5%)	RC1(27.3%)	RC11(34.0%)	RC4(20.0%)	RC4(41.0%)	RC8(25.0%)
RC4(28.4%)	RC12(21.9%)	RC4(25.4%)	RC2(33.3%)	RC11(20.2%)	RC11(21.1%)	RC15(17.5%)	RC11(27.3%)	RC5(20.8%)	RC9(20.0%)	RC12(14.8%)	RC1(12.5%)
RC12(21.5%)	RC5(12.9%)	RC15(19.5%)	RC12(21.2%)	RC1(12.6%)	RC1(11.6%)	RC18(7.5%)	RC12(18.2%)	RC4(14.2%)	RC12(20.0%)	RC9(13.1%)	RC4(12.5%)
RC5(12.7%)	RC4(11.2%)	RC1(12.4%)	RC1(15.2%)	RC3(9.2%)	RC4(9.5%)	RC1(5.0%)	RC15(18.2%)	RC9(13.2%)	RC15(20.0%)	RC15(13.1%)	RC9(12.5%)
RC14(8.4%)	RC1(7.6%)	RC5(10.9%)	RC5(12.1%)	RC4(8.4%)	RC9(7.4%)	RC2(5.0%)	RC4(9.1%)	RC1(9.4%)	RC18(20.0%)	RC5(8.2%)	RC11(12.5%)
RC1(8.0%)	RC9(6.3%)	RC9(7.9%)	RC13(3.0%)	RC9(5.0%)	RC3(6.3%)	RC5(5.0%)	RC7(9.1%)	RC15(9.4%)	RC1(0.0%)	RC14(8.2%)	RC12(12.5%)
RC9(8.0%)	RC2(5.4%)	RC2(6.3%)	RC15(3.0%)	RC15(4.2%)	RC2(4.2%)	RC6(2.5%)	RC9(9.1%)	RC14(7.5%)	RC2(0.0%)	RC1(3.3%)	RC15(12.5%)
RC2(3.6%)	RC3(4.9%)	RC14(5.3%)	RC3(0.0%)	RC2(3.4%)	RC14(4.2%)	RC9(2.5%)	RC14(9.1%)	RC2(6.6%)	RC3(0.0%)	RC2(3.3%)	RC2(0.0%)
RC8(2.9%)	RC14(4.0%)	RC3(2.5%)	RC6(0.0%)	RC7(3.4%)	RC13(3.2%)	RC14(2.5%)	RC17(9.1%)	RC8(3.8%)	RC5(0.0%)	RC7(3.3%)	RC3(0.0%)
RC3(1.5%)	RC7(2.7%)	RC7(2.0%)	RC7(0.0%)	RC13(2.5%)	RC15(3.2%)	RC3(0.0%)	RC2(0.0%)	RC3(2.8%)	RC6(0.0%)	RC10(1.6%)	RC6(0.0%)
RC13(0.7%)	RC18(2.2%)	RC8(2.0%)	RC8(0.0%)	RC14(2.5%)	RC8(2.1%)	RC7(0.0%)	RC3(0.0%)	RC6(0.9%)	RC7(0.0%)	RC3(0.0%)	RC7(0.0%)
RC6(0.4%)	RC8(1.8%)	RC10(1.0%)	RC9(0.0%)	RC8(1.7%)	RC10(2.1%)	RC8(0.0%)	RC6(0.0%)	RC16(0.9%)	RC8(0.0%)	RC6(0.0%)	RC10(0.0%)
RC10(0.4%)	RC10(1.3%)	RC6(0.3%)	RC10(0.0%)	RC16(1.7%)	RC16(2.1%)	RC10(0.0%)	RC8(0.0%)	RC7(0.0%)	RC10(0.0%)	RC8(0.0%)	RC13(0.0%)
RC17(0.4%)	RC6(0.9%)	RC13(0.3%)	RC14(0.0%)	RC10(0.8%)	RC6(1.1%)	RC12(0.0%)	RC10(0.0%)	RC10(0.0%)	RC13(0.0%)	RC13(0.0%)	RC14(0.0%)
RC7(0.0%)	RC13(0.9%)	RC16(0.3%)	RC16(0.0%)	RC17(0.8%)	RC7(1.1%)	RC13(0.0%)	RC13(0.0%)	RC13(0.0%)	RC14(0.0%)	RC16(0.0%)	RC16(0.0%)
RC16(0.0%)	RC16(0.0%)	RC18(0.3%)	RC17(0.0%)	RC6(0.0%)	RC17(1.1%)	RC16(0.0%)	RC16(0.0%)	RC17(0.0%)	RC16(0.0%)	RC17(0.0%)	RC17(0.0%)
RC18(0.0%)	RC17(0.0%)	RC17(0.0%)	RC18(0.0%)	RC18(0.0%)	RC18(0.0%)	RC17(0.0%)	RC18(0.0%)	RC18(0.0%)	RC17(0.0%)	RC18(0.0%)	RC18(0.0%)

From figure 3.21 it can be concluded that the 4th quarter marks the quarter, in which immediate causes IC15, IC3, IC13 IC5, IC14, and IC17 were highest compared with other three quarters. Of top seven immediate cause, only IC2 was highest in third quarter.

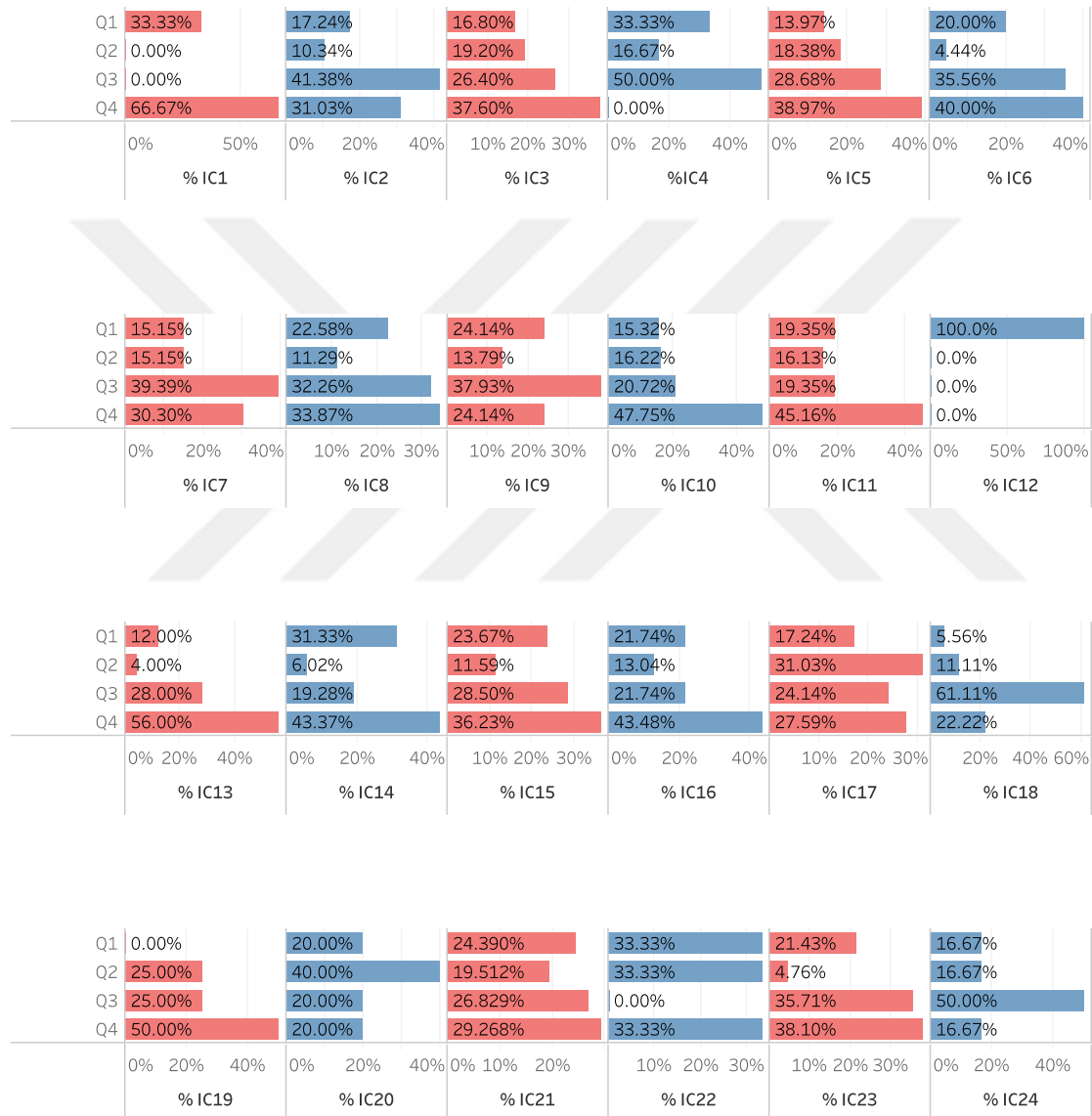


Figure 3.21 Quarterly distribution of immediate causes

Figure 3.22 compares the monthly distribution of the immediate causes. From the figure, for selected top seven immediate causes, it can be concluded that IC15 was very frequent in October and November, IC3 in November and August, IC13 in November and October, IC14 in December and January, IC5 in October and November, IC2 in August and November, and IC17 in April and October.



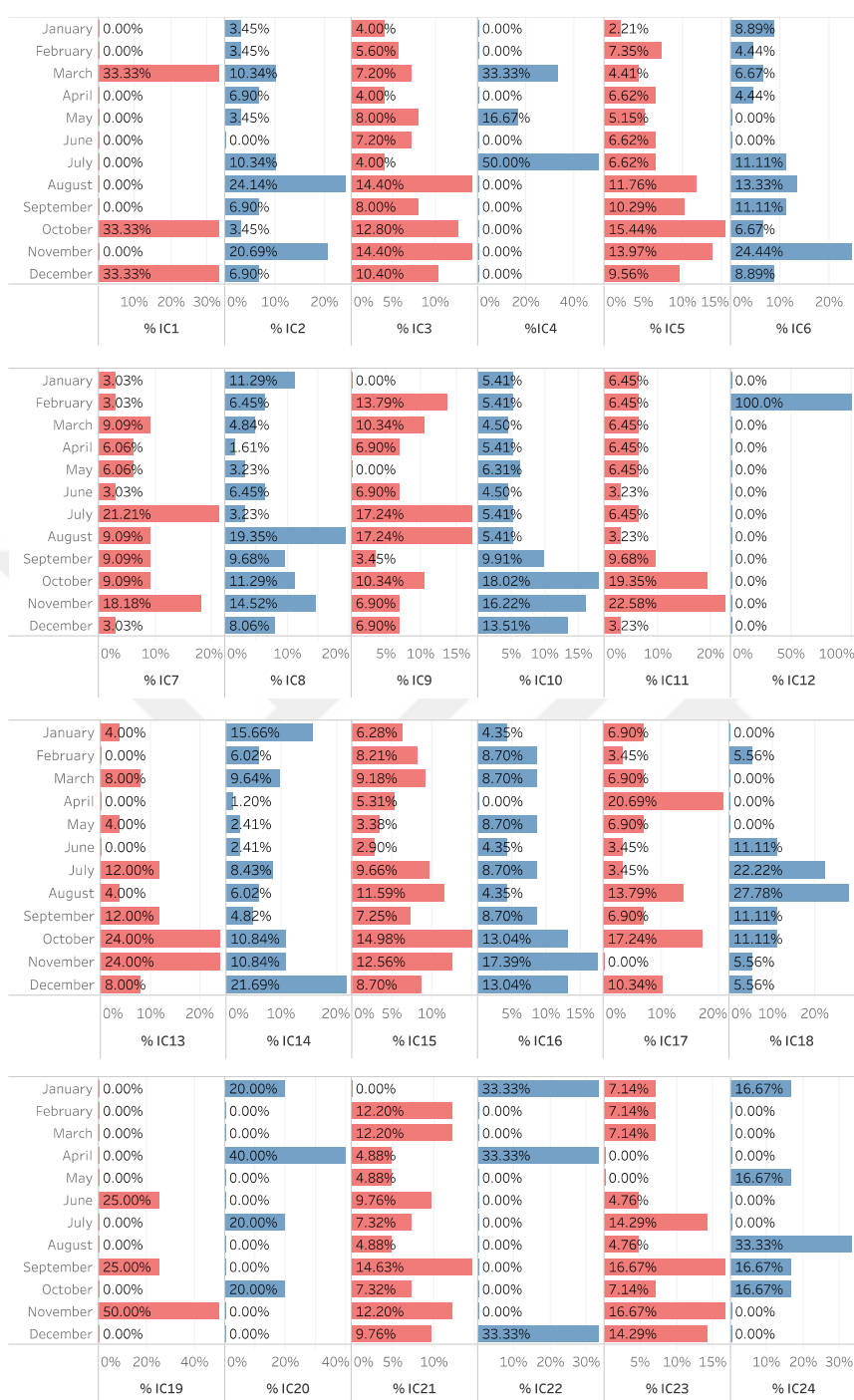


Figure 3.22 Monthly distribution of immediate causes

Figure 3.23 compares the distribution of the immediate causes over day of the week. From the figure, for selected top seven immediate causes, it can be concluded that IC15 was very frequent on Wednesday and Friday, IC3 on Monday and Thursday, IC13, IC14 and IC5 on Friday and Tuesday, IC2 on Thursday and Saturday, and IC17 on Saturday and Monday.



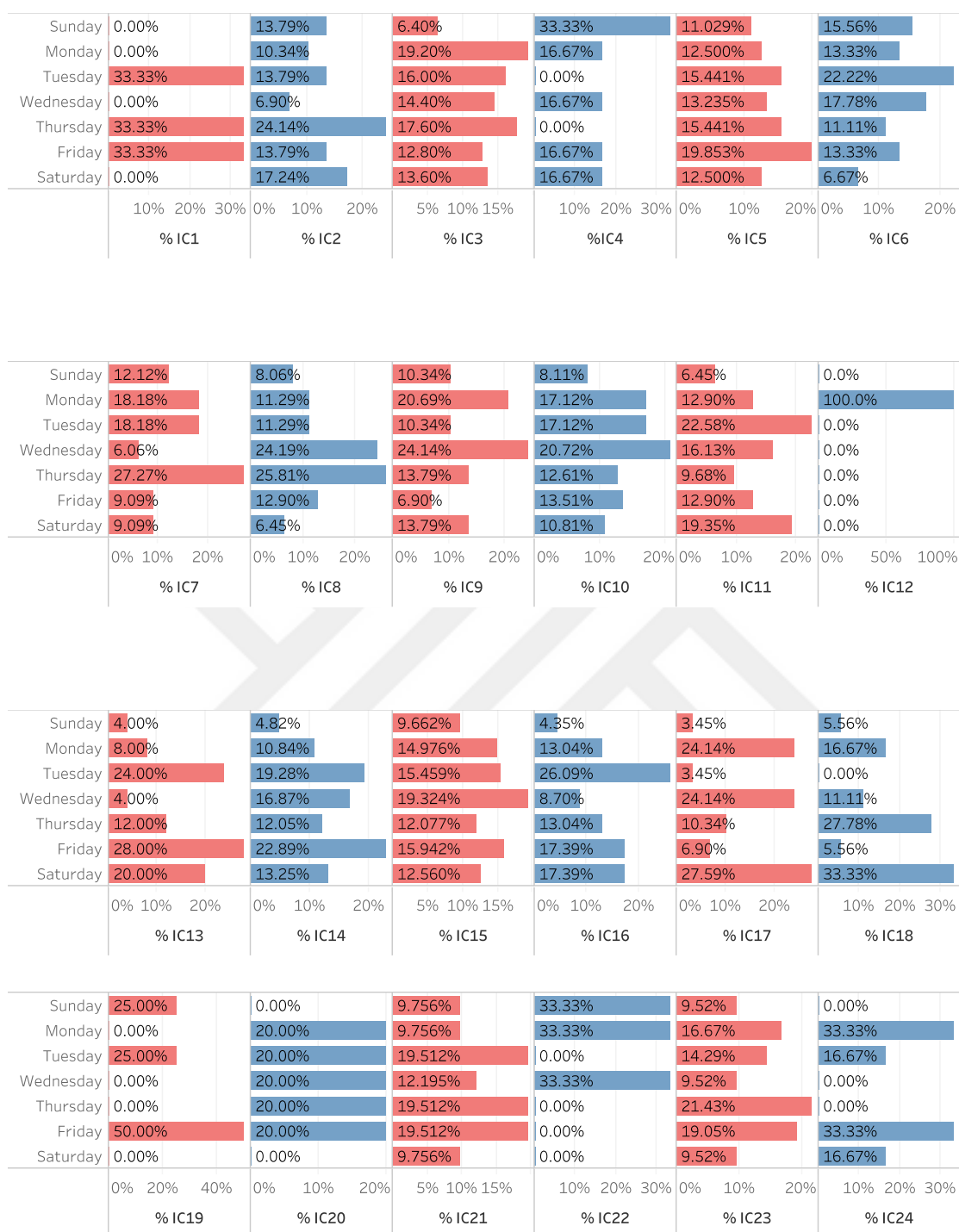


Figure 3.23 Distribution of immediate causes over day of the week

Figure 3.24 relates immediate causes and hour of the day. From the figure, for selected top seven immediate causes, it can be concluded that most active hours for IC15 are around 9:00 to 11:00 and 13:00 to 16:00, for IC3 are around 11:00 and 14:00 to 16:00, for IC13 are around 11:00, 14:00, 15:00 and 17:00, for IC14 are around 10:00 and 11:00, for IC5 are around 9:00 to 11:00, 14:00 and 15:00, for IC2 around 10:00, 11:00, 13:00 and 16:00, and for IC17 are around 10:00, 11:00, 14:00 and 16:00.

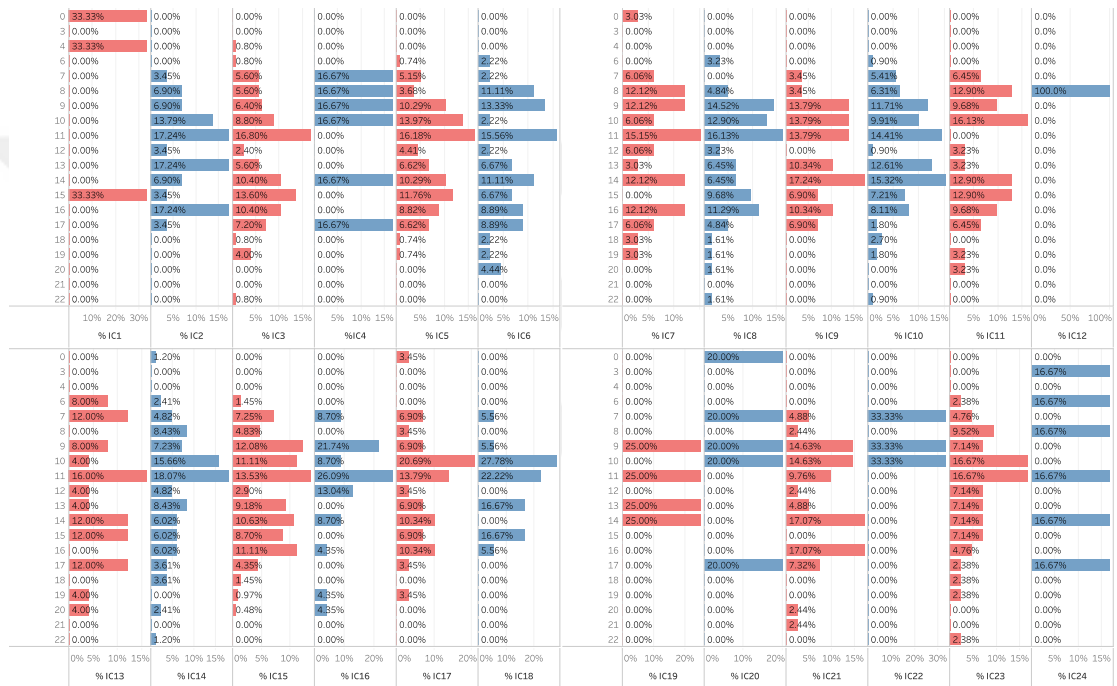


Figure 3.24 Distribution of immediate causes over hour of the day

Figure 3.25 and Table 3.9 show that most common root causes are RC11, RC12, RC4, RC5, RC15 and RC1 with 45%, 30%, 25%, 20%, 17% and 11% accordingly. Other root causes change between 0% to 8%.

Table 3.9 Distribution of Root Causes

ROOT CAUSE	%	ROOT CAUSES (SORTED)	%
RC1	0.11	RC11	0.45
RC2	0.07	RC12	0.30
RC3	0.04	RC4	0.25
RC4	0.25	RC5	0.20
RC5	0.20	RC15	0.17
RC6	0.00	RC1	0.11
RC7	0.03	RC9	0.08
RC8	0.02	RC2	0.07
RC9	0.08	RC14	0.06
RC10	0.01	RC3	0.04
RC11	0.45	RC7	0.03
RC12	0.30	RC8	0.02
RC13	0.01	RC10	0.01
RC14	0.06	RC18	0.01
RC15	0.17	RC13	0.01
RC16	0.00	RC6	0.00
RC17	0.00	RC16	0.00
RC18	0.01	RC17	0.00

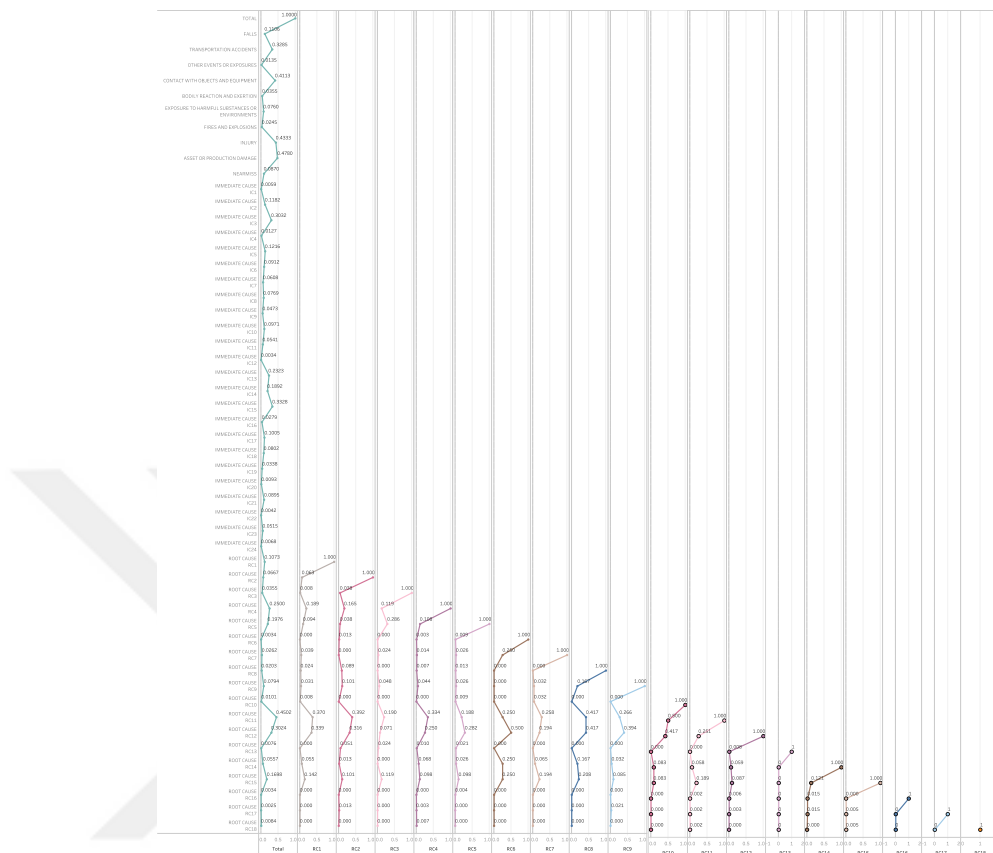


Figure 3.25 Distribution of root causes over other attributes

Table 3.10 shows the percentage share of root causes for each type of exposure. Since the number of cases are not significant for some of the exposure types, the distribution of the cause for falls, transportation, contact with objects and equipment, and exposure to harmful substances or environments cases are selected to be introduced. Out of 18 immediate causes, RC11, RC12, RC4, RC5, and RC15 were the top five causes underlying fall accidents, with above 10% each. However, in transportation accidents, lineup changes slightly and top five causes become RC11, RC15, RC4, RC12 and RC5 with each being more than 10%, which are also shared

causes in falls. RC11, RC12, RC4, RC5, RC1 and RC2 are the causes which played important role in contact with objects and equipment and RC11, RC12, RC5, RC1, RC15, RC9, and RC4 for exposure to harmful substances or environments cases with each having shares more than 10%.



Table 3.10 Root Causes and Type of Exposure

TOTAL	FALLS	TRANSPORTATION ACCIDENTS	OTHER EVENTS OR EXPOSURES	CONTACT WITH OBJECTS AND EQUIPMENT	BODILY REACTION AND EXERTION	EXPOSURE TO HARMFUL SUBSTANCES OR ENVIRONMENTS	FIRES AND EXPLOSIONS
RC11(45.0%)	RC11(52.1%)	RC11(51.9%)	RC12(56.3%)	RC11(40.5%)	RC11(35.7%)	RC11(51.1%)	RC5(69.0%)
RC12(30.2%)	RC12(34.7%)	RC15(32.4%)	RC5(31.3%)	RC12(38.2%)	RC4(23.8%)	RC12(32.2%)	RC11(31.0%)
RC4(25.0%)	RC4(33.1%)	RC4(29.3%)	RC1(25.0%)	RC4(24.4%)	RC15(21.4%)	RC5(20.0%)	RC12(17.2%)
RC5(19.8%)	RC5(26.4%)	RC12(20.8%)	RC4(18.8%)	RC5(19.5%)	RC5(14.3%)	RC1(17.8%)	RC1(6.9%)
RC15(17.0%)	RC15(19.8%)	RC5(14.9%)	RC3(6.3%)	RC1(11.5%)	RC12(14.3%)	RC15(13.3%)	RC7(6.9%)
RC1(10.7%)	RC1(8.3%)	RC1(9.0%)	RC8(6.3%)	RC2(10.7%)	RC1(9.5%)	RC9(12.2%)	RC10(6.9%)
RC9(7.9%)	RC9(6.6%)	RC9(8.5%)	RC9(6.3%)	RC9(7.6%)	RC3(9.5%)	RC4(10.0%)	RC14(6.9%)
RC2(6.7%)	RC2(5.0%)	RC14(6.4%)	RC11(6.3%)	RC14(6.2%)	RC9(7.1%)	RC2(7.8%)	RC16(6.9%)
RC14(5.6%)	RC3(3.3%)	RC2(2.8%)	RC2(0.0%)	RC15(6.2%)	RC2(4.8%)	RC14(4.4%)	RC2(3.4%)
RC3(3.5%)	RC10(3.3%)	RC8(1.5%)	RC6(0.0%)	RC3(5.1%)	RC7(4.8%)	RC3(3.3%)	RC3(3.4%)
RC7(2.6%)	RC14(3.3%)	RC18(1.5%)	RC7(0.0%)	RC7(3.9%)	RC8(4.8%)	RC7(3.3%)	RC4(3.4%)
RC8(2.0%)	RC7(2.5%)	RC13(1.3%)	RC10(0.0%)	RC8(2.9%)	RC10(2.4%)	RC18(3.3%)	RC6(3.4%)
RC10(1.0%)	RC8(0.8%)	RC3(1.0%)	RC13(0.0%)	RC10(0.6%)	RC14(2.4%)	RC10(1.1%)	RC9(3.4%)
RC18(0.8%)	RC6(0.0%)	RC7(0.5%)	RC14(0.0%)	RC13(0.6%)	RC6(0.0%)	RC13(1.1%)	RC8(0.0%)
RC13(0.8%)	RC13(0.0%)	RC6(0.3%)	RC15(0.0%)	RC6(0.4%)	RC13(0.0%)	RC17(1.1%)	RC13(0.0%)
RC6(0.3%)	RC16(0.0%)	RC10(0.3%)	RC16(0.0%)	RC16(0.2%)	RC16(0.0%)	RC6(0.0%)	RC15(0.0%)
RC16(0.3%)	RC17(0.0%)	RC16(0.3%)	RC17(0.0%)	RC17(0.2%)	RC17(0.0%)	RC8(0.0%)	RC17(0.0%)
RC17(0.3%)	RC18(0.0%)	RC17(0.3%)	RC18(0.0%)	RC18(0.2%)	RC18(0.0%)	RC16(0.0%)	RC18(0.0%)

RC11, RC4, RC12, and RC5 play important role in all type of consequences with different lineup that can be derived from Table 3.11. Compared with nearmiss cases, only RC15 and RC1 has more than 10% share in asset and production damage. For injury cases, RC15 is a root cause in comparison with nearmiss cases.



Table 3.11 Root Causes and Type of Consequence

TOTAL	INJURY	ASSET OR PRODUCTION DAMAGE	NEARMISS
RC11(45.0%)	RC11(47.6%)	RC11(44.3%)	RC12(53.4%)
RC12(30.2%)	RC4(32.4%)	RC15(24.4%)	RC11(36.9%)
RC4(25.0%)	RC12(32.4%)	RC12(24.0%)	RC5(24.3%)
RC5(19.8%)	RC5(15.6%)	RC5(22.4%)	RC4(17.5%)
RC15(17.0%)	RC15(10.9%)	RC4(19.6%)	RC2(9.7%)
RC1(10.7%)	RC1(9.0%)	RC1(12.7%)	RC1(8.7%)
RC9(7.9%)	RC2(8.0%)	RC9(9.5%)	RC15(6.8%)
RC2(6.7%)	RC9(7.0%)	RC14(5.5%)	RC3(3.9%)
RC14(5.6%)	RC14(6.4%)	RC2(4.9%)	RC7(3.9%)
RC3(3.5%)	RC7(3.9%)	RC3(3.9%)	RC9(3.9%)
RC7(2.6%)	RC3(2.9%)	RC8(1.6%)	RC6(1.9%)
RC8(2.0%)	RC8(2.3%)	RC7(1.2%)	RC8(1.9%)
RC10(1.0%)	RC10(1.4%)	RC13(1.2%)	RC14(1.9%)
RC18(0.8%)	RC18(0.6%)	RC18(1.2%)	RC10(1.0%)
RC13(0.8%)	RC13(0.2%)	RC10(0.7%)	RC13(1.0%)
RC6(0.3%)	RC16(0.2%)	RC16(0.5%)	RC16(0.0%)
RC16(0.3%)	RC17(0.2%)	RC6(0.4%)	RC17(0.0%)
RC17(0.3%)	RC6(0.0%)	RC17(0.4%)	RC18(0.0%)

The relations between root cause and immediate causes is introduced in Table 3.12. Other root causes underlying RC11 were IC15, IC13, IC3, IC14, IC5 and IC2 that are above 10% individually. For RC12 cases, IC3, IC15, IC13, IC2, IC5, IC10, IC21,

IC14 and IC5 are the top nine contributing causes. IC3, IC15, IC13, IC5, IC10, and IC19 are top six causes seen in cases with RC4.



Table 3.12 Root Causes and Immediate Causes

RC1	RC2	RC3	RC4	RC5	RC6	RC7	RC8	RC9
IC3(52.8%)	IC15(31.6%)	IC14(26.2%)	IC3(43.2%)	IC17(36.3%)	IC3(50.0%)	IC10(51.6%)	IC3(37.5%)	IC3(36.2%)
IC15(38.6%)	IC3(25.3%)	IC17(26.2%)	IC15(33.8%)	IC3(29.9%)	IC14(50.0%)	IC15(25.8%)	IC13(33.3%)	IC15(33.0%)
IC2(21.3%)	IC5(25.3%)	IC15(23.8%)	IC13(26.4%)	IC18(25.2%)	IC2(25.0%)	IC14(19.4%)	IC15(33.3%)	IC2(31.9%)
IC5(17.3%)	IC2(20.3%)	IC2(19.0%)	IC5(16.6%)	IC6(19.7%)	IC8(25.0%)	IC3(16.1%)	IC2(25.0%)	IC13(23.4%)
IC13(17.3%)	IC14(15.2%)	IC6(14.3%)	IC10(13.9%)	IC15(18.4%)	IC13(25.0%)	IC6(16.1%)	IC9(16.7%)	IC14(14.9%)
IC7(15.0%)	IC16(13.9%)	IC18(14.3%)	IC19(11.1%)	IC8(17.1%)	IC15(25.0%)	IC9(16.1%)	IC11(16.7%)	IC21(14.9%)
IC14(13.4%)	IC6(12.7%)	IC1(11.9%)	IC14(8.4%)	IC13(15.0%)	IC18(25.0%)	IC17(12.9%)	IC14(16.7%)	IC9(13.8%)
IC17(11.8%)	IC13(12.7%)	IC7(11.9%)	IC23(8.4%)	IC14(12.4%)	IC19(25.0%)	IC7(6.5%)	IC21(16.7%)	IC6(12.8%)
IC9(9.4%)	IC7(8.9%)	IC5(9.5%)	IC11(7.1%)	IC2(9.8%)	IC21(25.0%)	IC8(6.5%)	IC10(12.5%)	IC23(8.5%)
IC18(8.7%)	IC11(8.9%)	IC13(9.5%)	IC7(6.4%)	IC21(9.4%)	IC1(0.0%)	IC11(6.5%)	IC17(8.3%)	IC5(7.4%)
IC18(8.7%)	IC21(8.9%)	IC9(7.1%)	IC9(6.4%)	IC10(7.3%)	IC4(0.0%)	IC23(6.5%)	IC18(8.3%)	IC18(7.4%)
IC21(7.9%)	IC9(6.3%)	IC10(7.1%)	IC6(6.1%)	IC7(6.8%)	IC5(0.0%)	IC2(3.2%)	IC24(8.3%)	IC17(6.4%)
IC6(6.3%)	IC4(5.1%)	IC21(7.1%)	IC2(5.1%)	IC5(4.3%)	IC6(0.0%)	IC5(3.2%)	IC5(4.2%)	IC10(5.3%)
IC11(6.3%)	IC10(5.1%)	IC3(4.8%)	IC21(5.1%)	IC11(3.8%)	IC7(0.0%)	IC18(3.2%)	IC6(4.2%)	IC11(5.3%)
IC4(3.9%)	IC17(5.1%)	IC4(4.8%)	IC8(4.7%)	IC9(3.4%)	IC9(0.0%)	IC20(3.2%)	IC7(4.2%)	IC7(4.3%)
IC8(3.9%)	IC18(5.1%)	IC8(4.8%)	IC16(4.4%)	IC20(3.4%)	IC10(0.0%)	IC1(0.0%)	IC1(0.0%)	IC8(3.2%)
IC16(3.9%)	IC8(3.8%)	IC11(0.0%)	IC17(3.4%)	IC23(2.1%)	IC11(0.0%)	IC4(0.0%)	IC4(0.0%)	IC4(2.1%)
IC20(2.4%)	IC19(2.5%)	IC12(0.0%)	IC18(3.0%)	IC16(1.7%)	IC12(0.0%)	IC12(0.0%)	IC8(0.0%)	IC1(1.1%)
IC19(1.6%)	IC23(2.5%)	IC16(0.0%)	IC1(1.0%)	IC24(1.7%)	IC16(0.0%)	IC13(0.0%)	IC12(0.0%)	IC19(1.1%)
IC23(1.6%)	IC1(1.3%)	IC19(0.0%)	IC4(1.0%)	IC4(1.3%)	IC17(0.0%)	IC16(0.0%)	IC16(0.0%)	IC20(1.1%)
IC12(0.8%)	IC12(0.0%)	IC20(0.0%)	IC20(0.3%)	IC19(0.9%)	IC20(0.0%)	IC19(0.0%)	IC19(0.0%)	IC22(1.1%)
IC24(0.8%)	IC20(0.0%)	IC22(0.0%)	IC22(0.3%)	IC1(0.4%)	IC22(0.0%)	IC21(0.0%)	IC20(0.0%)	IC24(1.1%)
IC1(0.0%)	IC22(0.0%)	IC23(0.0%)	IC24(0.3%)	IC12(0.4%)	IC23(0.0%)	IC22(0.0%)	IC22(0.0%)	IC12(0.0%)
IC22(0.0%)	IC24(0.0%)	IC24(0.0%)	IC12(0.0%)	IC22(0.0%)	IC24(0.0%)	IC24(0.0%)	IC23(0.0%)	IC16(0.0%)
RC10	RC11	RC12	RC13	RC14	RC15	RC16	RC17	RC18
IC8(41.7%)	IC15(47.8%)	IC3(45.0%)	IC3(55.6%)	IC3(51.5%)	IC13(49.3%)	IC17(50.0%)	IC3(66.7%)	IC14(50.0%)
IC3(33.3%)	IC13(29.6%)	IC15(28.5%)	IC11(33.3%)	IC11(47.0%)	IC14(48.3%)	IC18(50.0%)	IC9(66.7%)	IC3(30.0%)
IC15(33.3%)	IC3(28.5%)	IC13(16.5%)	IC17(33.3%)	IC13(34.8%)	IC15(38.3%)	IC2(25.0%)	IC2(33.3%)	IC19(30.0%)
IC14(25.0%)	IC14(27.2%)	IC2(15.9%)	IC18(33.3%)	IC15(31.8%)	IC3(22.9%)	IC3(25.0%)	IC4(33.3%)	IC6(10.0%)
IC5(16.7%)	IC5(15.0%)	IC5(14.8%)	IC13(22.2%)	IC5(18.2%)	IC2(12.4%)	IC11(25.0%)	IC13(33.3%)	IC15(10.0%)
IC18(16.7%)	IC2(13.1%)	IC10(14.5%)	IC14(22.2%)	IC2(16.7%)	IC6(7.5%)	IC15(25.0%)	IC17(33.3%)	IC22(10.0%)
IC2(8.3%)	IC10(9.0%)	IC21(14.2%)	IC1(11.1%)	IC14(13.6%)	IC8(7.5%)	IC21(25.0%)	IC18(33.3%)	IC1(0.0%)
IC6(8.3%)	IC6(6.9%)	IC14(13.7%)	IC2(11.1%)	IC10(12.1%)	IC10(6.5%)	IC1(0.0%)	IC20(33.3%)	IC2(0.0%)
IC10(8.3%)	IC21(6.8%)	IC6(13.4%)	IC15(11.1%)	IC21(12.1%)	IC21(5.0%)	IC4(0.0%)	IC1(0.0%)	IC4(0.0%)
IC13(8.3%)	IC11(6.6%)	IC8(10.6%)	IC16(11.1%)	IC6(7.6%)	IC23(4.0%)	IC5(0.0%)	IC5(0.0%)	IC5(0.0%)
IC17(8.3%)	IC8(6.0%)	IC17(8.9%)	IC4(0.0%)	IC23(7.6%)	IC9(3.5%)	IC6(0.0%)	IC6(0.0%)	IC7(0.0%)
IC23(8.3%)	IC7(5.6%)	IC18(8.9%)	IC5(0.0%)	IC18(6.1%)	IC11(3.5%)	IC7(0.0%)	IC7(0.0%)	IC8(0.0%)
IC1(0.0%)	IC23(5.3%)	IC11(7.5%)	IC6(0.0%)	IC7(4.5%)	IC19(3.5%)	IC8(0.0%)	IC8(0.0%)	IC9(0.0%)
IC4(0.0%)	IC17(4.5%)	IC9(6.7%)	IC7(0.0%)	IC9(4.5%)	IC7(3.0%)	IC9(0.0%)	IC10(0.0%)	IC10(0.0%)
IC7(0.0%)	IC9(4.3%)	IC7(6.4%)	IC8(0.0%)	IC17(4.5%)	IC17(2.5%)	IC10(0.0%)	IC11(0.0%)	IC11(0.0%)
IC9(0.0%)	IC18(3.8%)	IC23(2.5%)	IC9(0.0%)	IC4(1.5%)	IC4(2.0%)	IC12(0.0%)	IC12(0.0%)	IC12(0.0%)
IC11(0.0%)	IC16(2.8%)	IC16(2.0%)	IC10(0.0%)	IC8(1.5%)	IC5(2.0%)	IC13(0.0%)	IC14(0.0%)	IC13(0.0%)
IC12(0.0%)	IC19(1.7%)	IC4(1.4%)	IC12(0.0%)	IC19(1.5%)	IC18(1.5%)	IC14(0.0%)	IC15(0.0%)	IC16(0.0%)
IC16(0.0%)	IC4(0.6%)	IC12(0.6%)	IC19(0.0%)	IC20(1.5%)	IC20(1.0%)	IC16(0.0%)	IC16(0.0%)	IC17(0.0%)
IC19(0.0%)	IC12(0.6%)	IC20(0.6%)	IC20(0.0%)	IC1(0.0%)	IC16(0.5%)	IC19(0.0%)	IC19(0.0%)	IC18(0.0%)
IC20(0.0%)	IC20(0.6%)	IC1(0.3%)	IC21(0.0%)	IC12(0.0%)	IC22(0.5%)	IC20(0.0%)	IC21(0.0%)	IC20(0.0%)
IC21(0.0%)	IC22(0.4%)	IC22(0.3%)	IC22(0.0%)	IC16(0.0%)	IC24(0.5%)	IC22(0.0%)	IC22(0.0%)	IC21(0.0%)
IC22(0.0%)	IC1(0.2%)	IC24(0.3%)	IC23(0.0%)	IC22(0.0%)	IC1(0.0%)	IC23(0.0%)	IC23(0.0%)	IC23(0.0%)
IC24(0.0%)	IC24(0.2%)	IC19(0.0%)	IC24(0.0%)	IC24(0.0%)	IC12(0.0%)	IC24(0.0%)	IC24(0.0%)	IC24(0.0%)

The relations between root cause and other root causes is introduced in table 3.13. Root causes underlying RC11 were RC12, RC15 and RC4 that are above 10% individually. RC11, RC4, RC5, RC1, and RC9 are the five causes seen in RC12. In cases with RC4, the immediate causes are RC11, RC12 and RC5 as seen in Table 3.8.



Table 3.13 Root Causes and Other Root Causes

RC1	RC2	RC3	RC4	RC5	RC6	RC7	RC8	RC9
RC1(100.0%)	RC2(100.0%)	RC3(100.0%)	RC4(100.0%)	RC5(100.0%)	RC6(100.0%)	RC7(100.0%)	RC8(100.0%)	RC9(100.0%)
RC11(37.0%)	RC11(39.2%)	RC5(28.6%)	RC11(33.4%)	RC12(28.2%)	RC5(50.0%)	RC11(25.8%)	RC11(41.7%)	RC12(39.4%)
RC12(33.9%)	RC12(31.6%)	RC11(19.0%)	RC12(25.0%)	RC11(18.8%)	RC12(50.0%)	RC5(19.4%)	RC12(41.7%)	RC11(26.6%)
RC4(18.9%)	RC4(16.5%)	RC4(11.9%)	RC5(10.8%)	RC4(13.7%)	RC2(25.0%)	RC12(19.4%)	RC2(29.2%)	RC4(13.8%)
RC15(14.2%)	RC1(10.1%)	RC15(11.9%)	RC15(9.8%)	RC15(9.8%)	RC4(25.0%)	RC15(19.4%)	RC15(20.8%)	RC2(8.5%)
RC5(9.4%)	RC9(10.1%)	RC2(7.1%)	RC1(8.1%)	RC1(5.1%)	RC7(25.0%)	RC1(16.1%)	RC9(16.7%)	RC15(8.5%)
RC2(6.3%)	RC15(10.1%)	RC12(7.1%)	RC14(6.8%)	RC3(5.1%)	RC11(25.0%)	RC4(12.9%)	RC14(16.7%)	RC5(6.4%)
RC14(5.5%)	RC8(8.9%)	RC9(4.8%)	RC2(4.4%)	RC7(2.6%)	RC14(25.0%)	RC14(6.5%)	RC1(12.5%)	RC1(4.3%)
RC7(3.9%)	RC13(5.1%)	RC1(2.4%)	RC9(4.4%)	RC9(2.6%)	RC15(25.0%)	RC2(3.2%)	RC5(12.5%)	RC8(4.3%)
RC9(3.1%)	RC3(3.8%)	RC7(2.4%)	RC3(1.7%)	RC14(2.6%)	RC1(0.0%)	RC6(3.2%)	RC4(8.3%)	RC14(3.2%)
RC8(2.4%)	RC5(3.8%)	RC13(2.4%)	RC7(1.4%)	RC13(2.1%)	RC3(0.0%)	RC9(3.2%)	RC3(0.0%)	RC3(2.1%)
RC3(0.8%)	RC6(1.3%)	RC6(0.0%)	RC13(1.0%)	RC2(1.3%)	RC8(0.0%)	RC10(3.2%)	RC6(0.0%)	RC17(2.1%)
RC10(0.8%)	RC14(1.3%)	RC8(0.0%)	RC8(0.7%)	RC8(1.3%)	RC9(0.0%)	RC2(0.0%)	RC7(0.0%)	RC7(1.1%)
RC6(0.0%)	RC17(1.3%)	RC10(0.0%)	RC18(0.7%)	RC6(0.9%)	RC10(0.0%)	RC8(0.0%)	RC10(0.0%)	RC6(0.0%)
RC13(0.0%)	RC7(0.0%)	RC14(0.0%)	RC6(0.3%)	RC10(0.9%)	RC13(0.0%)	RC13(0.0%)	RC13(0.0%)	RC10(0.0%)
RC16(0.0%)	RC10(0.0%)	RC16(0.0%)	RC17(0.3%)	RC16(0.4%)	RC16(0.0%)	RC16(0.0%)	RC16(0.0%)	RC13(0.0%)
RC17(0.0%)	RC16(0.0%)	RC17(0.0%)	RC10(0.0%)	RC17(0.0%)	RC17(0.0%)	RC17(0.0%)	RC17(0.0%)	RC16(0.0%)
RC18(0.0%)	RC18(0.0%)	RC18(0.0%)	RC16(0.0%)	RC18(0.0%)	RC18(0.0%)	RC18(0.0%)	RC18(0.0%)	RC18(0.0%)
RC10	RC11	RC12	RC13	RC14	RC15	RC16	RC17	RC18
RC10(100.0%)	RC11(100.0%)	RC12(100.0%)	RC13(100.0%)	RC14(100.0%)	RC15(100.0%)	RC16(100.0%)	RC17(100.0%)	RC18(100.0%)
RC11(50.0%)	RC12(25.1%)	RC11(37.4%)	RC5(55.6%)	RC11(47.0%)	RC11(50.2%)	RC12(50.0%)	RC9(66.7%)	RC4(20.0%)
RC12(41.7%)	RC15(18.9%)	RC4(20.7%)	RC2(44.4%)	RC12(31.8%)	RC12(15.4%)	RC5(25.0%)	RC2(33.3%)	RC11(10.0%)
RC5(16.7%)	RC4(18.6%)	RC5(18.4%)	RC4(33.3%)	RC4(30.3%)	RC4(14.4%)	RC11(25.0%)	RC4(33.3%)	RC15(10.0%)
RC1(8.3%)	RC1(8.8%)	RC1(12.0%)	RC12(33.3%)	RC15(12.1%)	RC5(11.4%)	RC14(25.0%)	RC11(33.3%)	RC1(0.0%)
RC7(8.3%)	RC5(8.3%)	RC9(10.3%)	RC3(11.1%)	RC1(10.6%)	RC1(9.0%)	RC1(0.0%)	RC12(33.3%)	RC2(0.0%)
RC14(8.3%)	RC2(5.8%)	RC15(8.7%)	RC1(0.0%)	RC5(9.1%)	RC2(4.0%)	RC2(0.0%)	RC14(33.3%)	RC3(0.0%)
RC15(8.3%)	RC14(5.8%)	RC2(7.0%)	RC6(0.0%)	RC8(6.1%)	RC9(4.0%)	RC3(0.0%)	RC15(33.3%)	RC5(0.0%)
RC2(0.0%)	RC9(4.7%)	RC14(5.9%)	RC7(0.0%)	RC9(4.5%)	RC14(4.0%)	RC4(0.0%)	RC1(0.0%)	RC6(0.0%)
RC3(0.0%)	RC8(1.9%)	RC8(2.8%)	RC8(0.0%)	RC7(3.0%)	RC7(3.0%)	RC6(0.0%)	RC3(0.0%)	RC7(0.0%)
RC4(0.0%)	RC3(1.5%)	RC7(1.7%)	RC9(0.0%)	RC2(1.5%)	RC3(2.5%)	RC7(0.0%)	RC5(0.0%)	RC8(0.0%)
RC6(0.0%)	RC7(1.5%)	RC10(1.4%)	RC10(0.0%)	RC6(1.5%)	RC8(2.5%)	RC8(0.0%)	RC6(0.0%)	RC9(0.0%)
RC8(0.0%)	RC10(1.1%)	RC3(0.8%)	RC11(0.0%)	RC10(1.5%)	RC6(0.5%)	RC9(0.0%)	RC7(0.0%)	RC10(0.0%)
RC9(0.0%)	RC6(0.2%)	RC13(0.8%)	RC14(0.0%)	RC16(1.5%)	RC10(0.5%)	RC10(0.0%)	RC8(0.0%)	RC12(0.0%)
RC13(0.0%)	RC16(0.2%)	RC6(0.6%)	RC15(0.0%)	RC17(1.5%)	RC17(0.5%)	RC13(0.0%)	RC10(0.0%)	RC13(0.0%)
RC16(0.0%)	RC17(0.2%)	RC16(0.6%)	RC16(0.0%)	RC3(0.0%)	RC18(0.5%)	RC15(0.0%)	RC13(0.0%)	RC14(0.0%)

From figure 3.26, it can be concluded that the 4th quarter marks the quarter, in which root causes RC11, RC12, RC4 RC5, and RC15 were highest compared with other three quarters. Of top six immediate cause, only RC1 was highest in third quarter.

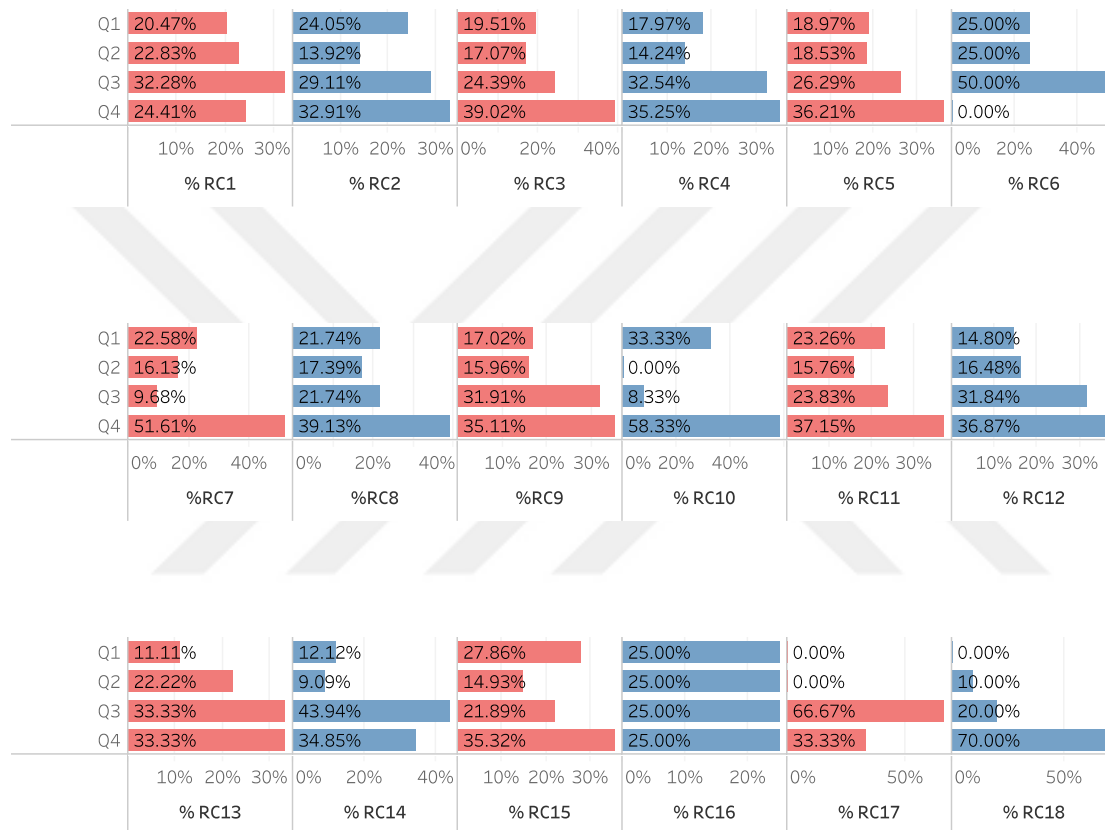


Figure 3.26 Quarterly distribution of root causes

Monthly distribution of the immediate causes is represented in figure 3.27. From the figure, for selected top six root causes, it can be concluded that RC11 was very

frequent in December and November, RC12 in August and November, RC4 in December and August, RC5 in October and November, RC15 in December and January, and RC1 in July and October.



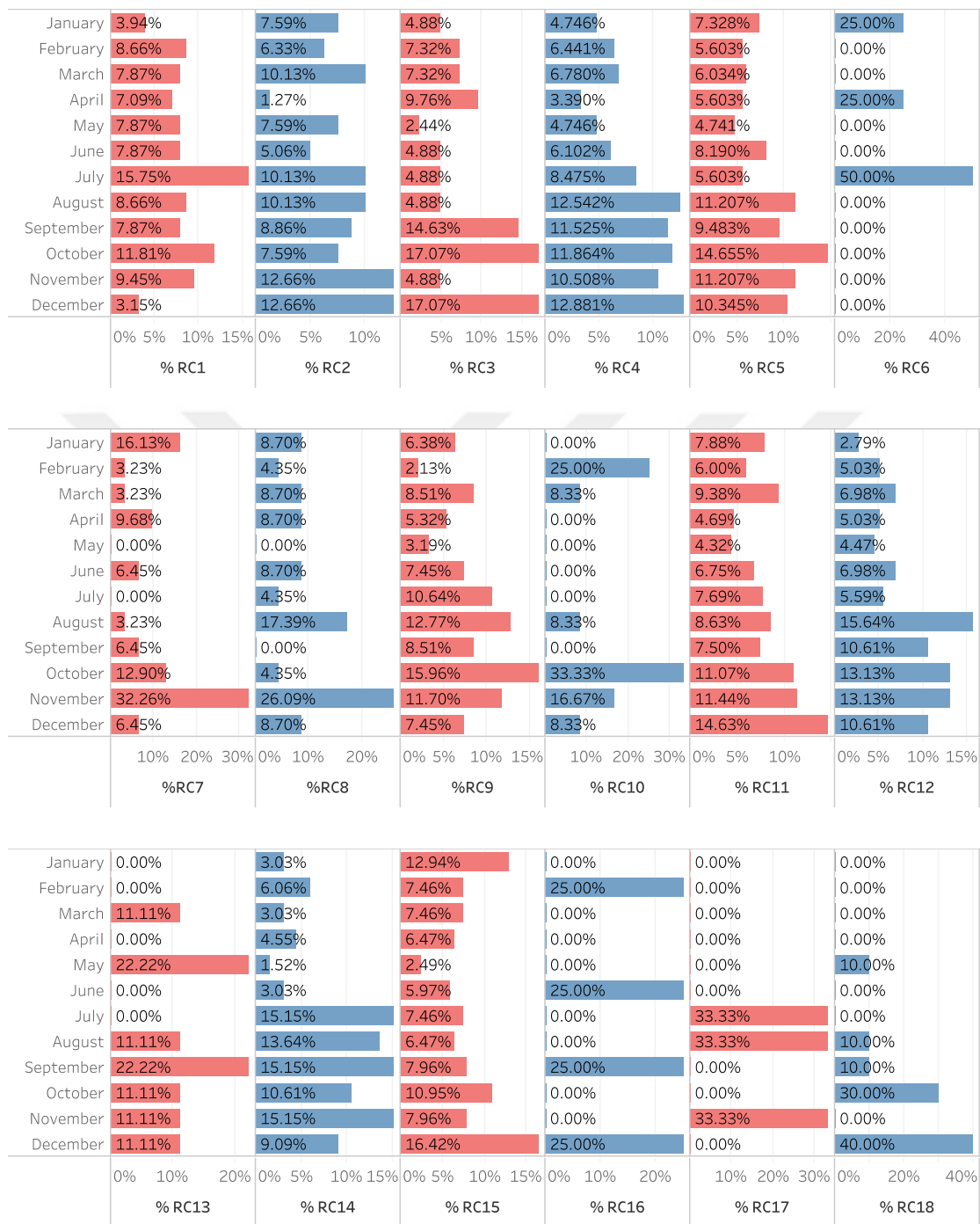


Figure 3.27 Monthly distribution of root causes

Figure 3.28 compares the distribution of the immediate causes over day of the week. From the figure, for selected top six root causes, it can be concluded that RC11 was very frequent on Thursday and Friday, RC12 on Thursday and Monday, RC4 was on Tuesday and Monday, RC5 was on Wednesday and Monday, RC15 was on Thursday, Tuesday, and Friday, and Rc1 was on Thursday and Friday.

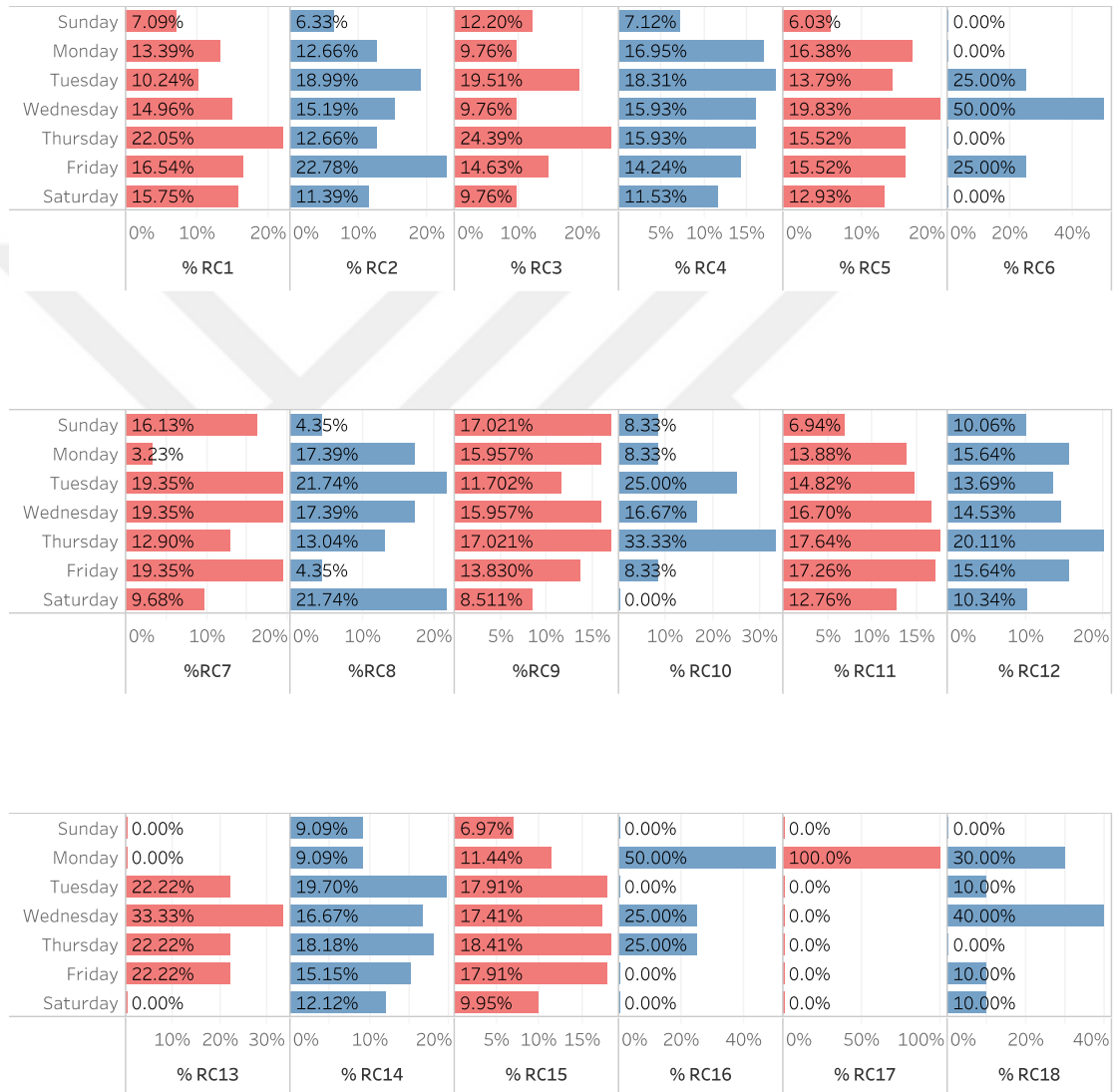


Figure 3.28 Distribution of root causes over day of the week

Figure 3.24 relates root causes and hour of the day. From the figure, for selected top six root causes, it can be concluded that most active hours for RC11 are around 9:00 to 11:00, 14:00 and 16:00, for RC12 are around 9:00 to 11:00 and 14:00, for RC4 are around 9:00 to 11:00, for RC5 are around 10:00 to 11:00, 14:00 and 16:00, for RC15 are around 9:00 to 11:00, and 16:00, and for RC1 around 09:00, and 14:00 to 16:00.



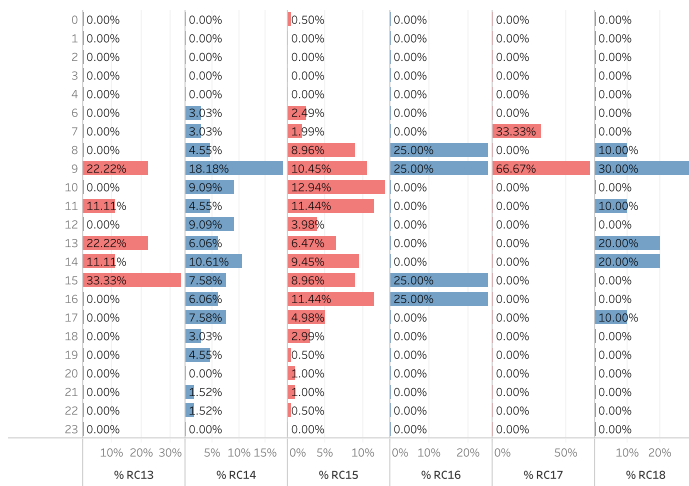
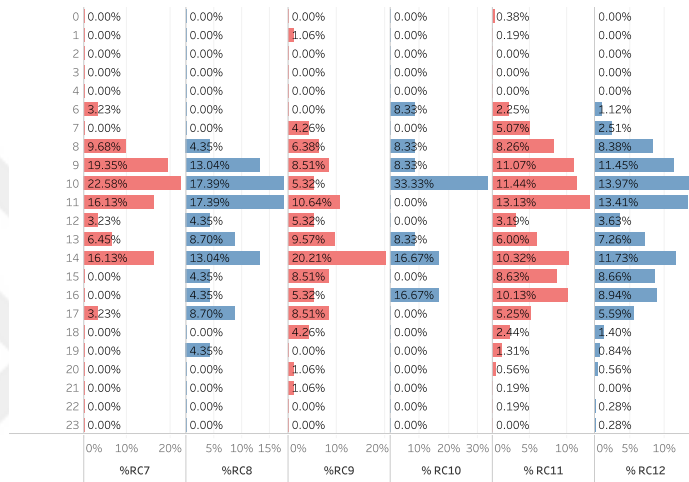
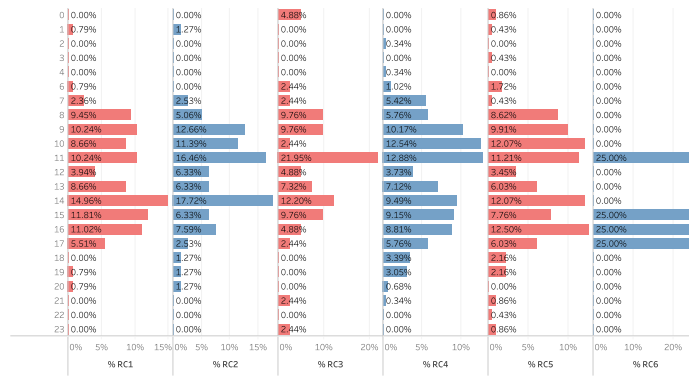


Figure 3.29 Distribution of root causes over hour of the day

3.2.5 Statistics of incomplete data regarding demographic information

In this section, statistics regarding age and experience of the people involved or injured in the accidents is introduced. Gender information are not introduced because there are only 3 accidents out of 1184 cases in which females were involved.

Figure 3.30 and Table 3.14 show information regarding age of 166 injured people or people involved in the accidents out of 1184 cases, which constitutes only 14% of all cases. The age range was between 18 and 66 ($M_{male}=34.0$, $SD=10.97$)

Table 3.14 Age distribution of the people involved

Group	Age(years)		
	Cases (n=166*)		
	Age Range	Mean	SD
1	18-22 (n=23, 13.9%)	19.4	0.94
2	22-26 (n=20, 12.0%)	23.3	1.07
3	26-30 (n=20, 12.0%)	27.6	1.14
4	30-34 (n=25, 15.1%)	31.9	1.05
5	34-38 (n=25, 15.1%)	35.6	1.25
6	38-42 (n=14, 8.4%)	39.9	1.00
7	42-46 (n=13, 7.8%)	42.8	0.99
8	46-50 (n=9, 5.4%)	46.9	0.78
9	50-54 (n=7, 4.2%)	51.4	1.27
10	54-58 (n=4, 2.4%)	55.3	1.50
11	58-62 (n=4, 2.4%)	60.0	1.15
12	62-66 (n=2, 1.2%)	64.0	2.83
		34.0	10.97

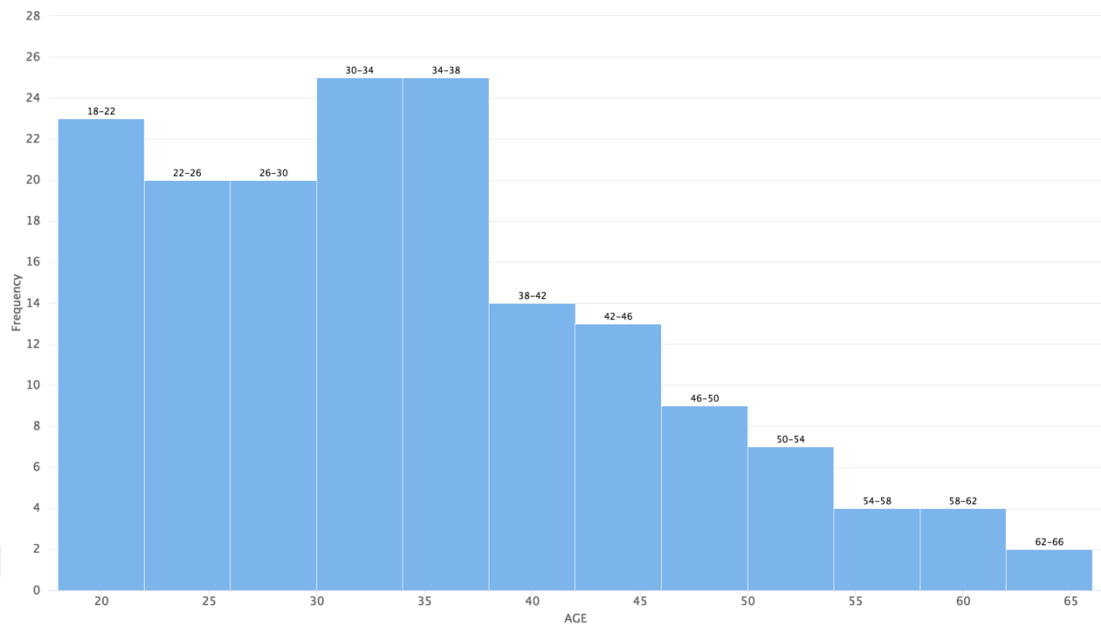


Figure 3.30 Age distribution of the people involved

The duration of experience changed from 3 months to more than sixty months. Only roughly 7.2% of all cases are known in terms of the length of the experience in the current project. The experience of the people involved in incidents is presented in figure 3.31.

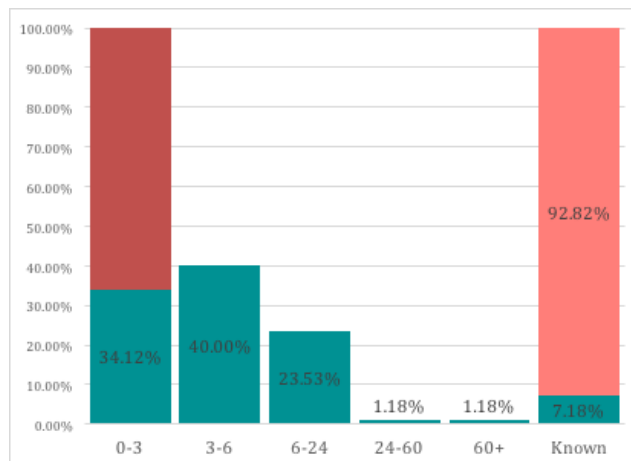


Figure 3.31 Experience of the people involved



CHAPTER 4

RESULTS AND DISCUSSION

This chapter of the study recaps all the results from the model discussed in the methodology chapter. SET01 to SET12 will undergo 11 predefined ML model and results will be listed in relevant section. In section 4.13 a comparison is introduced.

4.1 Results for SET01

Models were fed with data set SET01 with splitting data as 90% being training set and 10% being test data. Parameters and results can be found in Table 4.1. From the table Deep Learning performed best in terms of accuracy and kappa which are 83.05% and 0.691 respectively. Four models were eliminated for this data set due to models being incapable of handling polynomial labels, namely INJ, ADPD and NM.

Table 4.1 Results for SET01

Model	Parameters	Results
1	Neural Nets	PerformanceVector: accuracy: 72.03% ConfusionMatrix: True: INJ ADPD NM INJ: 31 1 4 ADPD: 16 53 5 NM: 4 3 1 kappa: 0.500
		ConfusionMatrix: True: INJ ADPD NM INJ: 31 1 4 ADPD: 16 53 5 NM: 4 3 1
2	Deep Learning	PerformanceVector: accuracy: 83.05% ConfusionMatrix: True: INJ ADPD NM INJ: 44 4 3 ADPD: 7 52 5

Model		Parameters	Results
			NM: 0 1 2 kappa: 0.691 ConfusionMatrix: True: INJ ADPD NM INJ: 44 4 3 ADPD: 7 52 5 NM: 0 1 2
3	Fast Large Margin	Solver: L2 SVM Dual C: 1.0 Epsilon: 0.01	*Cannot handle polynomial label
4	Decision Tree	Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply preprunning: Yes Minimal gain: 0.01	PerformanceVector: accuracy: 67.80% ConfusionMatrix: True: INJ ADPD NM INJ: 24 1 1 ADPD: 27 56 9 NM: 0 0 0 kappa: 0.390 ConfusionMatrix: True: INJ ADPD NM INJ: 24 1 1 ADPD: 27 56 9 NM: 0 0 0
5	Gradient Boosted Trees	Number of trees: 50 Reproducible: no Maximal depth: 5 Min rows:10.0 Min split improvements: 1.0E-5 Number of bins:20 Learning rate: 0.01 Sample rate: 1.0 Distribution: AUTO	PerformanceVector: accuracy: 76.27% ConfusionMatrix: True: INJ ADPD NM INJ: 36 3 3 ADPD: 13 54 7 NM: 2 0 0 kappa: 0.562 ConfusionMatrix: True: INJ ADPD NM INJ: 36 3 3 ADPD: 13 54 7 NM: 2 0 0
6	Random Forest	Number of trees: 100 Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply preprunning: Yes Minimal gain: 0.01 Minimal leaf size: 2 Guess subset ratio: Yes Voting strategy: confidence vote	PerformanceVector: accuracy: 71.19% ConfusionMatrix: True: INJ ADPD NM INJ: 27 0 1 ADPD: 24 57 9 NM: 0 0 0 kappa: 0.455 ConfusionMatrix: True: INJ ADPD NM INJ: 27 0 1 ADPD: 24 57 9 NM: 0 0 0
7	Linear Regression	Min tolerance: 0.05 Ridge: 1.0E-8	*Cannot handle polynomial label
8	Generalized Linear Model	Family: AUTO Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No	PerformanceVector: accuracy: 78.81% ConfusionMatrix: True: INJ ADPD NM INJ: 39 2 3 ADPD: 12 53 6 NM: 0 2 1 kappa: 0.612 ConfusionMatrix: True: INJ ADPD NM INJ: 39 2 3

Model	Parameters	Results
		ADPD: 12 53 6 NM: 0 2 1
9	Naive Bayes	PerformanceVector: accuracy: 21.19% ConfusionMatrix: True: INJ ADPD NM INJ: 9 3 0 ADPD: 0 6 0 NM: 42 48 10 kappa: 0.083 ConfusionMatrix: True: INJ ADPD NM INJ: 9 3 0 ADPD: 0 6 0 NM: 42 48 10
10	Logistic Regression	Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No *Cannot handle polynomial label
11	Support Vector Machine	Kernel type: dot C: 1.0 Convergence epsilon: 0.001 *Cannot handle polynomial label

4.2 Results for SET02

Models were fed with data set SET02 with splitting data as 90% being training set and 10% being test data. Parameters and results can be found in Table 4.2. From the table Deep Learning performed best in terms of accuracy and kappa which are 80.51% and 0.637 respectively. Four models were eliminated for this data set due to models being incapable of handling polynomial labels, namely INJ, ADPD and NM.

Table 4.2 Results for SET02

Model	Parameters	Results
1	Neural Nets	PerformanceVector: accuracy: 77.97% ConfusionMatrix: True: INJ ADPD NM INJ: 37 2 4 ADPD: 14 55 6 NM: 0 0 0 kappa: 0.589 ConfusionMatrix: True: INJ ADPD NM
	Training cycle: 200 Learning rate: 0.01 Momentum: 0.9 Error epsilon: 1.0E-4	

Model	Parameters	Results
		INJ: 37 2 4 ADPD: 14 55 6 NM: 0 0 0
2	Deep Learning Activation: Rectifier Epochs:10.0 Compute variable importance: yes	PerformanceVector: accuracy: 80.51% ConfusionMatrix: True: INJ ADPD NM INJ: 40 2 3 ADPD: 11 55 7 NM: 0 0 0 kappa: 0.637 ConfusionMatrix: True: INJ ADPD NM INJ: 40 2 3 ADPD: 11 55 7 NM: 0 0 0
3	Fast Large Margin Solver: L2 SVM Dual C: 1.0 Epsilon: 0.01	*Cannot handle polynomial label
4	Decision Tree Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01	PerformanceVector: accuracy: 68.64% ConfusionMatrix: True: INJ ADPD NM INJ: 24 0 1 ADPD: 27 57 9 NM: 0 0 0 kappa: 0.406 ConfusionMatrix: True: INJ ADPD NM INJ: 24 0 1 ADPD: 27 57 9 NM: 0 0 0
5	Gradient Boosted Trees Number of trees: 50 Reproducible: no Maximal depth: 5 Min rows:10.0 Min split improvements: 1.0E-5 Number of bins:20 Learning rate: 0.01 Sample rate: 1.0 Distribution: AUTO	PerformanceVector: accuracy: 77.12% ConfusionMatrix: True: INJ ADPD NM INJ: 36 3 3 ADPD: 15 54 6 NM: 0 0 1 kappa: 0.575 ConfusionMatrix: True: INJ ADPD NM INJ: 36 3 3 ADPD: 15 54 6 NM: 0 0 1
6	Random Forest Number of trees: 100 Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01 Minimal leaf size: 2 Guess subset ratio: Yes Voting strategy: confidence vote	PerformanceVector: accuracy: 71.19% ConfusionMatrix: True: INJ ADPD NM INJ: 27 0 1 ADPD: 24 57 9 NM: 0 0 0 kappa: 0.455 ConfusionMatrix: True: INJ ADPD NM INJ: 27 0 1 ADPD: 24 57 9 NM: 0 0 0
7	Linear Regression Min tolerance: 0.05 Ridge: 1.0E-8	*Cannot handle polynomial label
8	Generalized Linear Model Family: AUTO Solver: AUTO	PerformanceVector: accuracy: 79.66%

Model		Parameters	Results			
9	Naive Bayes	Reproducible: no	ConfusionMatrix:			
		Use regularization: No	True:	INJ	ADPD	NM
		Standardize: Yes	INJ:	38	2	2
		Non-negative coefficients:	ADPD:	12	54	6
		No	NM:	1	1	2
		kappa: 0.629				
		ConfusionMatrix:				
		True:	INJ	ADPD	NM	
		INJ:	38	2	2	
		ADPD:	12	54	6	
		NM:	1	1	2	
		PerformanceVector:				
		accuracy: 22.03%				
		ConfusionMatrix:				
True:	INJ	ADPD	NM			
INJ:	8	0	0			
ADPD:	2	9	1			
NM:	41	48	9			
kappa: 0.084						
ConfusionMatrix:						
True:	INJ	ADPD	NM			
INJ:	8	0	0			
ADPD:	2	9	1			
NM:	41	48	9			
10	Logistic Regression	Solver: AUTO	*Cannot handle polynomial label			
		Reproducible: no				
		Use regularization: No				
		Standardize: Yes				
		Non-negative coefficients: No				
11	Support Vector Machine	Kernel type: dot	*Cannot handle polynomial label			
		C: 1.0				
		Convergence epsilon:				
		0.001				

4.3 Results for SET03

Models were fed with data set SET03 with splitting data as 90% being training set and 10% being test data. Parameters and results can be found in Table 4.3. From the table Deep Learning performed best in terms of accuracy and kappa which are 55.93% and 0.204 respectively. Four models were eliminated for this data set due to models being incapable of handling polynomial labels, namely INJ, ADPD and NM.

Table 4.3 Results for SET03

Model	Parameters	Results
1	Neural Nets Training cycle: 200 Learning rate: 0.01 Momentum: 0.9 Error epsilon: 1.0E-4	PerformanceVector: accuracy: 56.78% ConfusionMatrix: True: INJ ADPD NM INJ: 26 16 6 ADPD: 25 41 4 NM: 0 0 0 kappa: 0.196 ConfusionMatrix: True: INJ ADPD NM INJ: 26 16 6 ADPD: 25 41 4 NM: 0 0 0
2	Deep Learning Activation: Rectifier Epochs:10.0 Compute variable importance: yes	PerformanceVector: accuracy: 55.93% ConfusionMatrix: True: INJ ADPD NM INJ: 44 35 6 ADPD: 7 22 4 NM: 0 0 0 kappa: 0.204 ConfusionMatrix: True: INJ ADPD NM INJ: 44 35 6 ADPD: 7 22 4 NM: 0 0 0
3	Fast Large Margin Solver: L2 SVM Dual C: 1.0 Epsilon: 0.01	*Cannot handle polynomial label
4	Decision Tree Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01	PerformanceVector: accuracy: 55.93% ConfusionMatrix: True: INJ ADPD NM INJ: 45 36 8 ADPD: 6 21 2 NM: 0 0 0 kappa: 0.206 ConfusionMatrix: True: INJ ADPD NM INJ: 45 36 8 ADPD: 6 21 2 NM: 0 0 0
5	Gradient Boosted Trees Number of trees: 50 Reproducible: no Maximal depth: 5 Min rows:10.0 Min split improvements: 1.0E-5 Number of bins:20 Learning rate: 0.01 Sample rate: 1.0 Distribution: AUTO	PerformanceVector: accuracy: 51.69% ConfusionMatrix: True: INJ ADPD NM INJ: 22 18 4 ADPD: 28 39 6 NM: 1 0 0 kappa: 0.104 ConfusionMatrix: True: INJ ADPD NM INJ: 22 18 4 ADPD: 28 39 6 NM: 1 0 0
6	Random Forest Number of trees: 100 Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01	PerformanceVector: accuracy: 56.78% ConfusionMatrix: True: INJ ADPD NM INJ: 24 14 5 ADPD: 27 43 5 NM: 0 0 0

Model		Parameters	Results
		Minimal leaf size: 2 Guess subset ratio: Yes Voting strategy: confidence vote	kappa: 0.193 ConfusionMatrix: True: INJ ADPD NM INJ: 24 14 5 ADPD: 27 43 5 NM: 0 0 0
7	Linear Regression	Min tolerance: 0.05 Ridge: 1.0E-8	*Cannot handle polynomial label
			PerformanceVector: accuracy: 55.08% ConfusionMatrix: True: INJ ADPD NM INJ: 27 18 6 ADPD: 24 38 4 NM: 0 1 0
8	Generalized Linear Model	Family: AUTO Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No	kappa: 0.172 ConfusionMatrix: True: INJ ADPD NM INJ: 27 18 6 ADPD: 24 38 4 NM: 0 1 0
			PerformanceVector: accuracy: 15.25% ConfusionMatrix: True: INJ ADPD NM INJ: 8 15 1 ADPD: 0 1 0 NM: 43 41 9
9	Naive Bayes		kappa: -0.007 ConfusionMatrix: True: INJ ADPD NM INJ: 8 15 1 ADPD: 0 1 0 NM: 43 41 9
10	Logistic Regression	Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No	*Cannot handle polynomial label
11	Support Vector Machine	Kernel type: dot C: 1.0 Convergence epsilon: 0.001	*Cannot handle polynomial label

4.4 Results for SET04

Models were fed with data set SET04 with splitting data as 90% being training set and 10% being test data. Parameters and results can be found in Table 4.4. From the table Deep Learning performed best in terms of accuracy and kappa which are 81.36% and 0.663 respectively. Four models were eliminated for this data set due to models being incapable of handling polynomial labels, namely INJ, ADPD and NM.

Table 4.4 Results for SET04

Model	Parameters	Results
1	Neural Nets Training cycle: 200 Learning rate: 0.01 Momentum: 0.9 Error epsilon: 1.0E-4	PerformanceVector: accuracy: 78.81% ConfusionMatrix: True: INJ ADPD NM INJ: 39 1 4 ADPD: 10 54 6 NM: 2 2 0 kappa: 0.614 ConfusionMatrix: True: INJ ADPD NM INJ: 39 1 4 ADPD: 10 54 6 NM: 2 2 0
2	Deep Learning Activation: Rectifier Epochs:10.0 Compute variable importance: yes	PerformanceVector: accuracy: 81.36% ConfusionMatrix: True: INJ ADPD NM INJ: 41 2 2 ADPD: 9 53 6 NM: 1 2 2 kappa: 0.663 ConfusionMatrix: True: INJ ADPD NM INJ: 41 2 2 ADPD: 9 53 6 NM: 1 2 2
3	Fast Large Margin Solver: L2 SVM Dual C: 1.0 Epsilon: 0.01	*Cannot handle polynomial label
4	Decision Tree Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01	PerformanceVector: accuracy: 71.19% ConfusionMatrix: True: INJ ADPD NM INJ: 27 0 2 ADPD: 24 57 8 NM: 0 0 0 kappa: 0.456 ConfusionMatrix: True: INJ ADPD NM INJ: 27 0 2 ADPD: 24 57 8

	Model	Parameters	Results			
			NM:	0	0	0
5	Gradient Boosted Trees	Number of trees: 50	PerformanceVector:			
		Reproducible: no	accuracy: 75.42%			
		Maximal depth: 5	ConfusionMatrix:			
		Min rows:10.0	True:	INJ	ADPD	NM
		Min split improvements: 1.0E-5	INJ:	36	4	3
		Number of bins:20	ADPD:	15	53	7
		Learning rate: 0.01	NM:	0	0	0
		Sample rate: 1.0	kappa: 0.541			
		Distribution: AUTO	ConfusionMatrix:			
			True:	INJ	ADPD	NM
			INJ:	36	4	3
			ADPD:	15	53	7
			NM:	0	0	0
6	Random Forest	Number of trees: 100	PerformanceVector:			
		Criterion: gain_ratio	accuracy: 79.66%			
		Maximal depth: 10	ConfusionMatrix:			
		Apply pruning: Yes	True:	INJ	ADPD	NM
		Confidence: 0.1	INJ:	37	0	2
		Apply prepruning: Yes	ADPD:	14	57	8
		Minimal gain: 0.01	NM:	0	0	0
		Minimal leaf size: 2	kappa: 0.619			
		Guess subset ratio: Yes	ConfusionMatrix:			
		Voting strategy: confidence vote	True:	INJ	ADPD	NM
			INJ:	37	0	2
			ADPD:	14	57	8
			NM:	0	0	0
7	Linear Regression	Min tolerance: 0.05	*Cannot handle polynomial label			
		Ridge: 1.0E-8				
8	Generalized Linear Model	Family: AUTO	PerformanceVector:			
		Solver: AUTO	accuracy: 78.81%			
		Reproducible: no	ConfusionMatrix:			
		Use regularization: No	True:	INJ	ADPD	NM
		Standardize: Yes	INJ:	39	2	4
		Non-negative coefficients: No	ADPD:	12	53	5
			NM:	0	2	1
			kappa: 0.612			
			ConfusionMatrix:			
			True:	INJ	ADPD	NM
			INJ:	39	2	4
			ADPD:	12	53	5
			NM:	0	2	1
9	Naive Bayes		PerformanceVector:			
			accuracy: 20.34%			
			ConfusionMatrix:			
			True:	INJ	ADPD	NM
			INJ:	9	3	0
			ADPD:	0	5	0
			NM:	42	49	10
			kappa: 0.077			
			ConfusionMatrix:			
			True:	INJ	ADPD	NM
			INJ:	9	3	0
			ADPD:	0	5	0
			NM:	42	49	10
10	Logistic Regression	Solver: AUTO	*Cannot handle polynomial label			
		Reproducible: no				
		Use regularization: No				
		Standardize: Yes				
		Non-negative coefficients: No				

	Model	Parameters	Results
11	Support Vector Machine	Kernel type: dot C: 1.0 Convergence epsilon: 0.001	*Cannot handle polynomial label

4.5 Results for SET05

Models were fed with data set SET05 with splitting data as 90% being training set and 10% being test data. Parameters and results can be found in Table 4.5 From the table Deep Learning performed best in terms of accuracy and kappa which are 77.97% and 0.592 respectively. Four models were eliminated for this data set due to models being incapable of handling polynomial labels, namely INJ, ADPD and NM.

Table 4.5 Results for SET05

	Model	Parameters	Results
1	Neural Nets	Training cycle: 200 Learning rate: 0.01 Momentum: 0.9 Error epsilon: 1.0E-4	PerformanceVector: accuracy: 77.12% ConfusionMatrix: True: INJ ADPD NM INJ: 37 3 4 ADPD: 14 54 6 NM: 0 0 0 kappa: 0.573 ConfusionMatrix: True: INJ ADPD NM INJ: 37 3 4 ADPD: 14 54 6 NM: 0 0 0
2	Deep Learning	Activation: Rectifier Epochs:10.0 Compute variable importance: yes	PerformanceVector: accuracy: 77.97% ConfusionMatrix: True: INJ ADPD NM INJ: 41 6 7 ADPD: 10 51 3 NM: 0 0 0 kappa: 0.592 ConfusionMatrix: True: INJ ADPD NM INJ: 41 6 7 ADPD: 10 51 3 NM: 0 0 0
3	Fast Large Margin	Solver: L2 SVM Dual C: 1.0 Epsilon: 0.01	*Cannot handle polynomial label
4	Decision Tree	Criterion: gain_ratio Maximal depth: 10	PerformanceVector: accuracy: 68.64%

Model	Parameters	Results
	Apply pruning: Yes Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01	ConfusionMatrix: True: INJ ADPD NM INJ: 27 3 2 ADPD: 24 54 8 NM: 0 0 0 kappa: 0.409 ConfusionMatrix: True: INJ ADPD NM INJ: 27 3 2 ADPD: 24 54 8 NM: 0 0 0
5	Gradient Boosted Trees Number of trees: 50 Reproducible: no Maximal depth: 5 Min rows:10.0 Min split improvements: 1.0E-5 Number of bins:20 Learning rate: 0.01 Sample rate: 1.0 Distribution: AUTO	PerformanceVector: accuracy: 75.42% ConfusionMatrix: True: INJ ADPD NM INJ: 36 4 3 ADPD: 15 53 7 NM: 0 0 0 kappa: 0.541 ConfusionMatrix: True: INJ ADPD NM INJ: 36 4 3 ADPD: 15 53 7 NM: 0 0 0
6	Random Forest Number of trees: 100 Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01 Minimal leaf size: 2 Guess subset ratio: Yes Voting strategy: confidence vote	PerformanceVector: accuracy: 71.19% ConfusionMatrix: True: INJ ADPD NM INJ: 27 0 1 ADPD: 24 57 9 NM: 0 0 0 kappa: 0.455 ConfusionMatrix: True: INJ ADPD NM INJ: 27 0 1 ADPD: 24 57 9 NM: 0 0 0
7	Linear Regression Min tolerance: 0.05 Ridge: 1.0E-8	*Cannot handle polynomial label
8	Generalized Linear Model Family: AUTO Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No	PerformanceVector: accuracy: 76.27% ConfusionMatrix: True: INJ ADPD NM INJ: 37 3 4 ADPD: 14 53 6 NM: 0 1 0 kappa: 0.560 ConfusionMatrix: True: INJ ADPD NM INJ: 37 3 4 ADPD: 14 53 6 NM: 0 1 0
9	Naive Bayes	PerformanceVector: accuracy: 21.19% ConfusionMatrix: True: INJ ADPD NM INJ: 7 0 0 ADPD: 2 9 1 NM: 42 48 9 kappa: 0.077 ConfusionMatrix: True: INJ ADPD NM

Model	Parameters	Results
		INJ: 7 0 0 ADPD: 2 9 1 NM: 42 48 9
10	Logistic Regression	Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No *Cannot handle polynomial label
11	Support Vector Machine	Kernel type: dot C: 1.0 Convergence epsilon: 0.001 *Cannot handle polynomial label

4.6 Results for SET06

Models were fed with data set SET06 with splitting data as 90% being training set and 10% being test data. Parameters and results can be found in Table 4.6. From the table Neural Nets performed best in terms of accuracy and kappa which are 60.17% and 0.273 respectively. Four models were eliminated for this data set due to models being incapable of handling polynomial labels, namely INJ, ADPD and NM.

Table 4.6 Results for SET06

Model	Parameters	Results
1	Neural Nets	PerformanceVector: accuracy: 60.17% ConfusionMatrix: True: INJ ADPD NM INJ: 40 26 6 ADPD: 11 31 4 NM: 0 0 0 kappa: 0.273 ConfusionMatrix: True: INJ ADPD NM INJ: 40 26 6 ADPD: 11 31 4 NM: 0 0 0
2	Deep Learning	PerformanceVector: accuracy: 55.08% ConfusionMatrix: True: INJ ADPD NM INJ: 12 3 0 ADPD: 39 53 10 NM: 0 1 0 kappa: 0.147 ConfusionMatrix: True: INJ ADPD NM INJ: 12 3 0

Model		Parameters	Results
			ADPD: 39 53 10 NM: 0 1 0
3	Fast Large Margin	Solver: L2 SVM Dual C: 1.0 Epsilon: 0.01	*Cannot handle polynomial label
			PerformanceVector: accuracy: 57.63% ConfusionMatrix: True: INJ ADPD NM INJ: 40 29 6 ADPD: 11 28 4 NM: 0 0 0 kappa: 0.229
4	Decision Tree	Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01	ConfusionMatrix: True: INJ ADPD NM INJ: 40 29 6 ADPD: 11 28 4 NM: 0 0 0
			PerformanceVector: accuracy: 50.85% ConfusionMatrix: True: INJ ADPD NM INJ: 22 19 4 ADPD: 29 38 6 NM: 0 0 0 kappa: 0.084
5	Gradient Boosted Trees	Number of trees: 50 Reproducible: no Maximal depth: 5 Min rows:10.0 Min split improvements: 1.0E-5 Number of bins:20 Learning rate: 0.01 Sample rate: 1.0 Distribution: AUTO	ConfusionMatrix: True: INJ ADPD NM INJ: 22 19 4 ADPD: 29 38 6 NM: 0 0 0
			PerformanceVector: accuracy: 56.78% ConfusionMatrix: True: INJ ADPD NM INJ: 25 15 5 ADPD: 26 42 5 NM: 0 0 0 kappa: 0.194
6	Random Forest	Number of trees: 100 Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01 Minimal leaf size: 2 Guess subset ratio: Yes Voting strategy: confidence vote	ConfusionMatrix: True: INJ ADPD NM INJ: 25 15 5 ADPD: 26 42 5 NM: 0 0 0
7	Linear Regression	Min tolerance: 0.05 Ridge: 1.0E-8	*Cannot handle polynomial label
			PerformanceVector: accuracy: 52.54% ConfusionMatrix: True: INJ ADPD NM INJ: 23 18 5 ADPD: 28 39 5 NM: 0 0 0 kappa: 0.116
8	Generalized Linear Model	Family: AUTO Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No	ConfusionMatrix: True: INJ ADPD NM INJ: 23 18 5 ADPD: 28 39 5 NM: 0 0 0
9	Naive Bayes		PerformanceVector: accuracy: 12.71% ConfusionMatrix:

Model	Parameters	Results
		True: INJ ADPD NM INJ: 4 7 0 ADPD: 0 1 0 NM: 47 49 10 kappa: 0.008 ConfusionMatrix: True: INJ ADPD NM INJ: 4 7 0 ADPD: 0 1 0 NM: 47 49 10
10	Logistic Regression	Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No *Cannot handle polynomial label
11	Support Vector Machine	Kernel type: dot C: 1.0 Convergence epsilon: 0.001 *Cannot handle polynomial label

4.7 Results for SET07

Models were fed with data set SET07 with splitting data as 90% being training set and 10% being test data. Parameters and results can be found in Table 4.7. From the table Random Forest resulted in highest AUC with 0.949 (accuracy: 82.41%, precision:75.00% and recall 100.00%)

Table 4.7 Results for SET07

Model	Parameters	Results
1	Neural Nets	PerformanceVector: accuracy: 78.70% ConfusionMatrix: True: INJ ADPD INJ: 34 6 ADPD: 17 51 precision: 75.00% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 34 6 ADPD: 17 51 recall: 89.47% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 34 6 ADPD: 17 51 AUC (optimistic): 0.906 (positive class: ADPD) AUC: 0.906 (positive class: ADPD) AUC (pessimistic): 0.906 (positive class: ADPD)

Model	Parameters	Results
2	Deep Learning	Activation: Rectifier
		Epochs:10.0
		Compute variable importance: yes
		PerformanceVector: accuracy: 84.26% ConfusionMatrix: True: INJ ADPD INJ: 40 6 ADPD: 11 51 precision: 82.26% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 40 6 ADPD: 11 51 recall: 89.47% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 40 6 ADPD: 11 51 AUC (optimistic): 0.913 (positive class: ADPD) AUC: 0.913 (positive class: ADPD) AUC (pessimistic): 0.913 (positive class: ADPD)
3	Fast Large Margin	Solver: L2 SVM Dual
		C: 1.0
		Epsilon: 0.01
		PerformanceVector: accuracy: 52.78% ConfusionMatrix: True: INJ ADPD INJ: 0 0 ADPD: 51 57 precision: 52.78% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 0 0 ADPD: 51 57 recall: 100.00% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 0 0 ADPD: 51 57 AUC (optimistic): 0.503 (positive class: ADPD) AUC: 0.503 (positive class: ADPD) AUC (pessimistic): 0.503 (positive class: ADPD)
4	Decision Tree	Criterion: gain_ratio
		Maximal depth: 10
		Apply pruning: Yes
		Confidence: 0.1
		Apply prepruning: Yes Minimal gain: 0.01 PerformanceVector: accuracy: 74.07% ConfusionMatrix: True: INJ ADPD INJ: 25 2 ADPD: 26 55 precision: 67.90% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 25 2 ADPD: 26 55 recall: 96.49% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 25 2 ADPD: 26 55 AUC (optimistic): 0.963 (positive class: ADPD) AUC: 0.871 (positive class: ADPD) AUC (pessimistic): 0.779 (positive class: ADPD)
5	Gradient Boosted Trees	Number of trees: 50
		Reproducible: no
		Maximal depth: 5
		Min rows:10.0
		PerformanceVector: accuracy: 77.78% ConfusionMatrix: True: INJ ADPD INJ: 34 7

Model	Parameters	Results
6	Min split improvements: 1.0E-5	ADPD: 17 50 precision: 74.63% (positive class: ADPD)
	Number of bins:20	ConfusionMatrix:
	Learning rate: 0.01	True: INJ ADPD
	Sample rate: 1.0	INJ: 34 7
	Distribution: AUTO	ADPD: 17 50
		recall: 87.72% (positive class: ADPD)
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 34 7
		ADPD: 17 50
		AUC (optimistic): 0.912 (positive class: ADPD)
		AUC: 0.902 (positive class: ADPD)
		AUC (pessimistic): 0.892 (positive class: ADPD)
		PerformanceVector:
		accuracy: 82.41%
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 32 0
	Number of trees: 100	ADPD: 19 57
	Criterion: gain_ratio	precision: 75.00% (positive class: ADPD)
	Maximal depth: 10	ConfusionMatrix:
	Apply pruning: Yes	True: INJ ADPD
	Confidence: 0.1	INJ: 32 0
	Apply prepruning: Yes	ADPD: 19 57
	Minimal gain: 0.01	recall: 100.00% (positive class: ADPD)
	Minimal leaf size: 2	ConfusionMatrix:
	Guess subset ratio: Yes	True: INJ ADPD
	Voting strategy: confidence vote	INJ: 32 0
		ADPD: 19 57
		AUC (optimistic): 0.949 (positive class: ADPD)
		AUC: 0.949 (positive class: ADPD)
		AUC (pessimistic): 0.949 (positive class: ADPD)
		PerformanceVector:
		accuracy: 82.41%
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 34 2
		ADPD: 17 55
		precision: 76.39% (positive class: ADPD)
		ConfusionMatrix:
		True: INJ ADPD
	Min tolerance: 0.05	INJ: 34 2
	Ridge: 1.0E-8	ADPD: 17 55
		recall: 96.49% (positive class: ADPD)
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 34 2
		ADPD: 17 55
		AUC (optimistic): 0.939 (positive class: ADPD)
		AUC: 0.939 (positive class: ADPD)
		AUC (pessimistic): 0.939 (positive class: ADPD)
		PerformanceVector:
		accuracy: 84.26%
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 39 5
	Family: AUTO	ADPD: 12 52
	Solver: AUTO	precision: 81.25% (positive class: ADPD)
	Reproducible: no	ConfusionMatrix:
	Use regularization: No	True: INJ ADPD
	Standardize: Yes	INJ: 39 5
	Non-negative coefficients: No	ADPD: 12 52
		precision: 81.25% (positive class: ADPD)
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 39 5
		ADPD: 12 52
		precision: 81.25% (positive class: ADPD)
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 39 5
		ADPD: 12 52

Model	Parameters	Results
9	Naive Bayes	ADPD: 12 52 recall: 91.23% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 39 5 ADPD: 12 52 AUC (optimistic): 0.939 (positive class: ADPD) AUC: 0.939 (positive class: ADPD) AUC (pessimistic): 0.939 (positive class: ADPD)
		PerformanceVector: accuracy: 84.26% ConfusionMatrix: True: INJ ADPD INJ: 45 11 ADPD: 6 46 precision: 88.46% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 45 11 ADPD: 6 46 recall: 80.70% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 45 11 ADPD: 6 46 AUC (optimistic): 0.903 (positive class: ADPD) AUC: 0.903 (positive class: ADPD) AUC (pessimistic): 0.903 (positive class: ADPD)
		PerformanceVector: accuracy: 84.26% ConfusionMatrix: True: INJ ADPD INJ: 39 5 ADPD: 12 52 precision: 81.25% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 39 5 ADPD: 12 52 recall: 91.23% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 39 5 ADPD: 12 52 AUC (optimistic): 0.939 (positive class: ADPD) AUC: 0.939 (positive class: ADPD) AUC (pessimistic): 0.939 (positive class: ADPD)
10	Logistic Regression	Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No ConfusionMatrix: True: INJ ADPD INJ: 39 5 ADPD: 12 52 precision: 84.38% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 41 3 ADPD: 10 54 recall: 94.74% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 41 3
11	Support Vector Machine	Kernel type: dot C: 1.0 Convergence epsilon: 0.001 ConfusionMatrix: True: INJ ADPD INJ: 41 3 ADPD: 10 54 precision: 84.38% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 41 3 ADPD: 10 54 recall: 94.74% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 41 3

Model	Parameters	Results
		ADPD: 10 54
		AUC (optimistic): 0.942 (positive class: ADPD)
		AUC: 0.942 (positive class: ADPD)
		AUC (pessimistic): 0.942 (positive class: ADPD)

4.8 Results for SET08

Models were fed with data set SET08 with splitting data as 90% being training set and 10% being test data. Parameters and results can be found in Table 4.8. From the table Random Forest resulted in highest AUC with 0.943 (accuracy: 77.78%, precision: 70.37% and recall 100.00%). However, Neural Nets, Generalized Linear Model, Logistic Regression and Support Vector Machine returned with highest accuracy at 86.11%.

Table 4.8 Results for SET08

Model	Parameters	Results
1	Neural Nets	PerformanceVector:
		accuracy: 86.11%
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 43 7
		ADPD: 8 50
		precision: 86.21% (positive class: ADPD)
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 43 7
2	Deep Learning	ADPD: 8 50
		recall: 87.72% (positive class: ADPD)
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 43 7
		ADPD: 8 50
		AUC (optimistic): 0.915 (positive class: ADPD)
		AUC: 0.915 (positive class: ADPD)
		AUC (pessimistic): 0.915 (positive class: ADPD)
		PerformanceVector:
	Activation: Rectifier Epochs:10.0 Compute variable importance: yes	accuracy: 84.26%
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 37 3
		ADPD: 14 54
		precision: 79.41% (positive class: ADPD)
		ConfusionMatrix:
		True: INJ ADPD

Model	Parameters	Results
3	Fast Large Margin	Solver: L2 SVM Dual C: 1.0 Epsilon: 0.01
		INJ: 37 3 ADPD: 14 54 recall: 94.74% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 37 3 ADPD: 14 54 AUC (optimistic): 0.927 (positive class: ADPD) AUC: 0.927 (positive class: ADPD) AUC (pessimistic): 0.927 (positive class: ADPD)
		PerformanceVector: accuracy: 52.78% ConfusionMatrix: True: INJ ADPD INJ: 0 0 ADPD: 51 57 precision: 52.78% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 0 0 ADPD: 51 57 recall: 100.00% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 0 0 ADPD: 51 57 AUC (optimistic): 0.503 (positive class: ADPD) AUC: 0.503 (positive class: ADPD) AUC (pessimistic): 0.503 (positive class: ADPD)
4	Decision Tree	Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01
		INJ: 25 0 ADPD: 26 57 precision: 68.67% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 25 0 ADPD: 26 57 recall: 100.00% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 25 0 ADPD: 26 57 AUC (optimistic): 0.979 (positive class: ADPD) AUC: 0.884 (positive class: ADPD) AUC (pessimistic): 0.789 (positive class: ADPD)
		PerformanceVector: accuracy: 75.93% ConfusionMatrix: True: INJ ADPD INJ: 25 0 ADPD: 26 57 precision: 68.67% (positive class: ADPD)
5	Gradient Boosted Trees	Number of trees: 50 Reproducible: no Maximal depth: 5 Min rows:10.0 Min split improvements: 1.0E-5 Number of bins:20 Learning rate: 0.01 Sample rate: 1.0 Distribution: AUTO
		INJ: 36 6 ADPD: 15 51 precision: 77.27% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 36 6 ADPD: 15 51 recall: 89.47% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD
		PerformanceVector: accuracy: 80.56% ConfusionMatrix: True: INJ ADPD INJ: 36 6 ADPD: 15 51 precision: 77.27% (positive class: ADPD)

Model	Parameters	Results
		INJ: 36 6 ADPD: 15 51 AUC (optimistic): 0.908 (positive class: ADPD) AUC: 0.893 (positive class: ADPD) AUC (pessimistic): 0.877 (positive class: ADPD)
6	Random Forest	PerformanceVector: accuracy: 77.78% ConfusionMatrix: True: INJ ADPD INJ: 27 0 ADPD: 24 57 precision: 70.37% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 27 0 ADPD: 24 57 recall: 100.00% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 27 0 ADPD: 24 57 AUC (optimistic): 0.944 (positive class: ADPD) AUC: 0.943 (positive class: ADPD) AUC (pessimistic): 0.942 (positive class: ADPD)
7	Linear Regression	PerformanceVector: accuracy: 83.33% ConfusionMatrix: True: INJ ADPD INJ: 36 3 ADPD: 15 54 precision: 78.26% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 36 3 ADPD: 15 54 recall: 94.74% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 36 3 ADPD: 15 54 AUC (optimistic): 0.937 (positive class: ADPD) AUC: 0.936 (positive class: ADPD) AUC (pessimistic): 0.934 (positive class: ADPD)
8	Generalized Linear Model	PerformanceVector: accuracy: 86.11% ConfusionMatrix: True: INJ ADPD INJ: 38 2 ADPD: 13 55 precision: 80.88% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 38 2 ADPD: 13 55 recall: 96.49% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 38 2 ADPD: 13 55 AUC (optimistic): 0.941 (positive class: ADPD) AUC: 0.941 (positive class: ADPD)

Model	Parameters	Results
9	Naive Bayes	AUC (pessimistic): 0.941 (positive class: ADPD)
		PerformanceVector: accuracy: 79.63% ConfusionMatrix: True: INJ ADPD INJ: 29 0 ADPD: 22 57 precision: 72.15% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 29 0 ADPD: 22 57 recall: 100.00% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 29 0 ADPD: 22 57 AUC (optimistic): 0.916 (positive class: ADPD) AUC: 0.916 (positive class: ADPD) AUC (pessimistic): 0.916 (positive class: ADPD)
10	Logistic Regression	Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No AUC (optimistic): 0.941 (positive class: ADPD) AUC: 0.941 (positive class: ADPD) AUC (pessimistic): 0.941 (positive class: ADPD)
		PerformanceVector: accuracy: 86.11% ConfusionMatrix: True: INJ ADPD INJ: 38 2 ADPD: 13 55 precision: 80.88% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 38 2 ADPD: 13 55 recall: 96.49% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 38 2 ADPD: 13 55 AUC (optimistic): 0.941 (positive class: ADPD) AUC: 0.941 (positive class: ADPD) AUC (pessimistic): 0.941 (positive class: ADPD)
11	Support Vector Machine	Kernel type: dot C: 1.0 Convergence epsilon: 0.001 AUC (optimistic): 0.941 (positive class: ADPD) AUC: 0.941 (positive class: ADPD) AUC (pessimistic): 0.941 (positive class: ADPD)
		PerformanceVector: accuracy: 86.11% ConfusionMatrix: True: INJ ADPD INJ: 42 6 ADPD: 9 51 precision: 85.00% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 42 6 ADPD: 9 51 recall: 89.47% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 42 6 ADPD: 9 51 AUC (optimistic): 0.941 (positive class: ADPD) AUC: 0.941 (positive class: ADPD) AUC (pessimistic): 0.941 (positive class: ADPD)

4.9 Results for SET09

Models were fed with data set SET09 with splitting data as 90% being training set and 10% being test data. Parameters and results can be found in Table 4.9. From the table Neural Nets resulted in highest AUC with 0.697 (accuracy: 64.81%, precision: 63.77% and recall 77.19%)

Table 4.9 Results for SET09

Model	Parameters	Results
1	Neural Nets	PerformanceVector: accuracy: 64.81% ConfusionMatrix: True: INJ ADPD INJ: 26 13 ADPD: 25 44 precision: 63.77% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 26 13 ADPD: 25 44 recall: 77.19% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 26 13 ADPD: 25 44 AUC (optimistic): 0.697 (positive class: ADPD) AUC: 0.697 (positive class: ADPD) AUC (pessimistic): 0.697 (positive class: ADPD)
		Training cycle: 200 Learning rate: 0.01 Momentum: 0.9 Error epsilon: 1.0E-4
2	Deep Learning	PerformanceVector: accuracy: 54.63% ConfusionMatrix: True: INJ ADPD INJ: 14 12 ADPD: 37 45 precision: 54.88% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 14 12 ADPD: 37 45 recall: 78.95% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 14 12 ADPD: 37 45 AUC (optimistic): 0.685 (positive class: ADPD) AUC: 0.685 (positive class: ADPD)
		Activation: Rectifier Epochs:10.0 Compute variable importance: yes

Model	Parameters	Results
3	Fast Large Margin	AUC (pessimistic): 0.685 (positive class: ADPD)
		PerformanceVector: accuracy: 52.78% ConfusionMatrix: True: INJ ADPD INJ: 0 0 ADPD: 51 57 precision: 52.78% (positive class: ADPD)
		ConfusionMatrix: True: INJ ADPD INJ: 0 0 ADPD: 51 57 recall: 100.00% (positive class: ADPD)
4	Decision Tree	ConfusionMatrix: True: INJ ADPD INJ: 0 0 ADPD: 51 57 AUC (optimistic): 0.503 (positive class: ADPD) AUC: 0.503 (positive class: ADPD) AUC (pessimistic): 0.503 (positive class: ADPD)
		PerformanceVector: accuracy: 61.11% ConfusionMatrix: True: INJ ADPD INJ: 45 36 ADPD: 6 21 precision: 77.78% (positive class: ADPD)
		ConfusionMatrix: True: INJ ADPD INJ: 45 36 ADPD: 6 21 recall: 36.84% (positive class: ADPD)
5	Gradient Boosted Trees	ConfusionMatrix: True: INJ ADPD INJ: 45 36 ADPD: 6 21 AUC (optimistic): 0.919 (positive class: ADPD) AUC: 0.623 (positive class: ADPD) AUC (pessimistic): 0.329 (positive class: ADPD)
		PerformanceVector: accuracy: 55.56% ConfusionMatrix: True: INJ ADPD INJ: 13 10 ADPD: 38 47 precision: 55.29% (positive class: ADPD)
		ConfusionMatrix: True: INJ ADPD INJ: 13 10 ADPD: 38 47 recall: 82.46% (positive class: ADPD)
6	Random Forest	ConfusionMatrix: True: INJ ADPD INJ: 13 10 ADPD: 38 47 AUC (optimistic): 0.616 (positive class: ADPD) AUC: 0.613 (positive class: ADPD) AUC (pessimistic): 0.610 (positive class: ADPD)
		PerformanceVector: accuracy: 61.11% ConfusionMatrix: True: INJ ADPD

Model	Parameters	Results
	Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01 Minimal leaf size: 2 Guess subset ratio: Yes Voting strategy: confidence vote	INJ: 20 11 ADPD: 31 46 precision: 59.74% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 20 11 ADPD: 31 46 recall: 80.70% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 20 11 ADPD: 31 46 AUC (optimistic): 0.671 (positive class: ADPD) AUC: 0.660 (positive class: ADPD) AUC (pessimistic): 0.648 (positive class: ADPD)
7	Linear Regression Min tolerance: 0.05 Ridge: 1.0E-8	PerformanceVector: accuracy: 55.56% ConfusionMatrix: True: INJ ADPD INJ: 22 19 ADPD: 29 38 precision: 56.72% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 22 19 ADPD: 29 38 recall: 66.67% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 22 19 ADPD: 29 38 AUC (optimistic): 0.656 (positive class: ADPD) AUC: 0.643 (positive class: ADPD) AUC (pessimistic): 0.629 (positive class: ADPD)
8	Generalized Linear Model Family: AUTO Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No	PerformanceVector: accuracy: 61.11% ConfusionMatrix: True: INJ ADPD INJ: 27 18 ADPD: 24 39 precision: 61.90% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 27 18 ADPD: 24 39 recall: 68.42% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 27 18 ADPD: 24 39 AUC (optimistic): 0.639 (positive class: ADPD) AUC: 0.639 (positive class: ADPD) AUC (pessimistic): 0.639 (positive class: ADPD)
9	Naive Bayes	PerformanceVector: accuracy: 48.15% ConfusionMatrix: True: INJ ADPD INJ: 50 55 ADPD: 1 2 precision: 66.67% (positive class: ADPD) ConfusionMatrix:

Model	Parameters	Results
10	Logistic Regression	True: INJ ADPD INJ: 50 55 ADPD: 1 2 recall: 3.51% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 50 55 ADPD: 1 2 AUC (optimistic): 0.645 (positive class: ADPD) AUC: 0.645 (positive class: ADPD) AUC (pessimistic): 0.645 (positive class: ADPD)
		PerformanceVector: accuracy: 61.11% ConfusionMatrix: True: INJ ADPD INJ: 27 18 ADPD: 24 39 precision: 61.90% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 27 18 ADPD: 24 39 recall: 68.42% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 27 18 ADPD: 24 39 AUC (optimistic): 0.639 (positive class: ADPD) AUC: 0.639 (positive class: ADPD) AUC (pessimistic): 0.639 (positive class: ADPD)
11	Support Vector Machine	Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No PerformanceVector: accuracy: 62.04% ConfusionMatrix: True: INJ ADPD INJ: 41 31 ADPD: 10 26 precision: 72.22% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 41 31 ADPD: 10 26 recall: 45.61% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 41 31 ADPD: 10 26 AUC (optimistic): 0.650 (positive class: ADPD) AUC: 0.650 (positive class: ADPD) AUC (pessimistic): 0.650 (positive class: ADPD)
		Kernel type: dot C: 1.0 Convergence epsilon: 0.001 PerformanceVector: accuracy: 62.04% ConfusionMatrix: True: INJ ADPD INJ: 41 31 ADPD: 10 26 precision: 72.22% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 41 31 ADPD: 10 26 recall: 45.61% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 41 31 ADPD: 10 26 AUC (optimistic): 0.650 (positive class: ADPD) AUC: 0.650 (positive class: ADPD) AUC (pessimistic): 0.650 (positive class: ADPD)

4.10 Results for SET10

Models were fed with data set SET10 with splitting data as 90% being training set and 10% being test data. Parameters and results can be found in Table 4.10. From

the table Random Forest resulted in highest AUC with 0.952 (accuracy: 81.48%, precision: 74.03% and recall 100.00%)

Table 4.10 Results for SET10

Model	Parameters	Results
1	Neural Nets	PerformanceVector: accuracy: 83.33% ConfusionMatrix: True: INJ ADPD INJ: 35 2 ADPD: 16 55 precision: 77.46% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 35 2 ADPD: 16 55 recall: 96.49% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 35 2 ADPD: 16 55 AUC (optimistic): 0.909 (positive class: ADPD) AUC: 0.909 (positive class: ADPD) AUC (pessimistic): 0.909 (positive class: ADPD)
		Training cycle: 200 Learning rate: 0.01 Momentum: 0.9 Error epsilon: 1.0E-4
2	Deep Learning	PerformanceVector: accuracy: 85.19% ConfusionMatrix: True: INJ ADPD INJ: 41 6 ADPD: 10 51 precision: 83.61% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 41 6 ADPD: 10 51 recall: 89.47% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 41 6 ADPD: 10 51 AUC (optimistic): 0.943 (positive class: ADPD) AUC: 0.943 (positive class: ADPD) AUC (pessimistic): 0.943 (positive class: ADPD)
		Activation: Rectifier Epochs:10.0 Compute variable importance: yes
3	Fast Large Margin	PerformanceVector: accuracy: 83.33% ConfusionMatrix: True: INJ ADPD INJ: 35 2 ADPD: 16 55 precision: 77.46% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 35 2 ADPD: 16 55 recall: 96.49% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 35 2 ADPD: 16 55
		Solver: L2 SVM Dual C: 1.0 Epsilon: 0.01

Model	Parameters	Results
4	Decision Tree	AUC (optimistic): 0.933 (positive class: ADPD)
		AUC: 0.933 (positive class: ADPD)
		AUC (pessimistic): 0.933 (positive class: ADPD)
		PerformanceVector: accuracy: 78.70% ConfusionMatrix: True: INJ ADPD INJ: 48 20 ADPD: 3 37 precision: 92.50% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 48 20 ADPD: 3 37 recall: 64.91% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 48 20 ADPD: 3 37 AUC (optimistic): 0.965 (positive class: ADPD) AUC: 0.873 (positive class: ADPD) AUC (pessimistic): 0.781 (positive class: ADPD)
5	Gradient Boosted Trees	PerformanceVector: accuracy: 80.56% ConfusionMatrix: True: INJ ADPD INJ: 30 0 ADPD: 21 57 precision: 73.08% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 30 0 ADPD: 21 57 recall: 100.00% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 30 0 ADPD: 21 57 AUC (optimistic): 0.934 (positive class: ADPD) AUC: 0.918 (positive class: ADPD) AUC (pessimistic): 0.902 (positive class: ADPD)
		Number of trees: 50
		Reproducible: no
		Maximal depth: 5
6	Random Forest	Min rows:10.0
		Min split improvements: 1.0E-5
		Number of bins:20
		Learning rate: 0.01
7	Linear Regression	Sample rate: 1.0
		Distribution: AUTO
		Number of trees: 100
		Criterion: gain_ratio
8	Decision Tree	Maximal depth: 10
		Apply pruning: Yes
		Confidence: 0.1
		Apply prepruning: Yes
9	Decision Tree	Minimal gain: 0.01
		Minimal leaf size: 2
		Guess subset ratio: Yes
		Voting strategy: confidence vote
10	Linear Regression	Min tolerance: 0.05
		Ridge: 1.0E-8
		PerformanceVector: accuracy: 82.41%

Model	Parameters	Results
		ConfusionMatrix: True: INJ ADPD INJ: 34 2 ADPD: 17 55 precision: 76.39% (positive class: ADPD)
		ConfusionMatrix: True: INJ ADPD INJ: 34 2 ADPD: 17 55 recall: 96.49% (positive class: ADPD)
		ConfusionMatrix: True: INJ ADPD INJ: 34 2 ADPD: 17 55 AUC (optimistic): 0.939 (positive class: ADPD) AUC: 0.939 (positive class: ADPD) AUC (pessimistic): 0.939 (positive class: ADPD)
8	Generalized Linear Model	PerformanceVector: accuracy: 83.33%
		ConfusionMatrix: True: INJ ADPD INJ: 36 3 ADPD: 15 54 precision: 78.26% (positive class: ADPD)
		ConfusionMatrix: True: INJ ADPD INJ: 36 3 ADPD: 15 54 recall: 94.74% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 36 3 ADPD: 15 54 AUC (optimistic): 0.936 (positive class: ADPD) AUC: 0.936 (positive class: ADPD) AUC (pessimistic): 0.936 (positive class: ADPD)
9	Naive Bayes	PerformanceVector: accuracy: 84.26%
		ConfusionMatrix: True: INJ ADPD INJ: 46 12 ADPD: 5 45 precision: 90.00% (positive class: ADPD)
		ConfusionMatrix: True: INJ ADPD INJ: 46 12 ADPD: 5 45 recall: 78.95% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 46 12 ADPD: 5 45 AUC (optimistic): 0.903 (positive class: ADPD) AUC: 0.903 (positive class: ADPD) AUC (pessimistic): 0.903 (positive class: ADPD)
10	Logistic Regression	PerformanceVector: accuracy: 83.33%
		ConfusionMatrix: True: INJ ADPD INJ: 36 3 ADPD: 15 54 precision: 78.26% (positive class: ADPD)

Model	Parameters	Results
11	Support Vector Machine	ConfusionMatrix: True: INJ ADPD INJ: 36 3 ADPD: 15 54 recall: 94.74% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 36 3 ADPD: 15 54 AUC (optimistic): 0.936 (positive class: ADPD) AUC: 0.936 (positive class: ADPD) AUC (pessimistic): 0.936 (positive class: ADPD)
		PerformanceVector: accuracy: 87.04% ConfusionMatrix: True: INJ ADPD INJ: 40 3 ADPD: 11 54 precision: 83.08% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 40 3 ADPD: 11 54 recall: 94.74% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 40 3 ADPD: 11 54 AUC (optimistic): 0.939 (positive class: ADPD) AUC: 0.939 (positive class: ADPD) AUC (pessimistic): 0.939 (positive class: ADPD)

4.11 Results for SET11

Models were fed with data set SET07 with splitting data as 90% being training set and 10% being test data. Parameters and results can be found in Table 4.11. From the table Support Vector Machine resulted in highest AUC with 0.940 (accuracy: 87.04%, precision: 87.72% and recall 87.72%)

Table 4.11 Results for SET11

Model	Parameters	Results
1	Neural Nets	Training cycle: 200 Learning rate: 0.01 Momentum: 0.9 Error epsilon: 1.0E-4 PerformanceVector: accuracy: 83.33% ConfusionMatrix: True: INJ ADPD INJ: 35 2

Model	Parameters	Results
2	Deep Learning	ADPD: 16 55 precision: 77.46% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 35 2 ADPD: 16 55 recall: 96.49% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 35 2 ADPD: 16 55 AUC (optimistic): 0.926 (positive class: ADPD) AUC: 0.924 (positive class: ADPD) AUC (pessimistic): 0.922 (positive class: ADPD)
		PerformanceVector: accuracy: 85.19% ConfusionMatrix: True: INJ ADPD INJ: 38 3 ADPD: 13 54 precision: 80.60% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 38 3 ADPD: 13 54 recall: 94.74% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 38 3 ADPD: 13 54 AUC (optimistic): 0.910 (positive class: ADPD) AUC: 0.908 (positive class: ADPD) AUC (pessimistic): 0.906 (positive class: ADPD)
3	Fast Large Margin	Solver: L2 SVM Dual C: 1.0 Epsilon: 0.01 PerformanceVector: accuracy: 82.41% ConfusionMatrix: True: INJ ADPD INJ: 35 3 ADPD: 16 54 precision: 77.14% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 35 3 ADPD: 16 54 recall: 94.74% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 35 3 ADPD: 16 54 AUC (optimistic): 0.937 (positive class: ADPD) AUC: 0.936 (positive class: ADPD) AUC (pessimistic): 0.934 (positive class: ADPD)
4	Decision Tree	Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01 PerformanceVector: accuracy: 86.11% ConfusionMatrix: True: INJ ADPD INJ: 45 9 ADPD: 6 48 precision: 88.89% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 45 9

Model	Parameters	Results
		ADPD: 6 48 recall: 84.21% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 45 9 ADPD: 6 48 AUC (optimistic): 0.964 (positive class: ADPD) AUC: 0.914 (positive class: ADPD) AUC (pessimistic): 0.864 (positive class: ADPD)
5	Gradient Boosted Trees Number of trees: 50 Reproducible: no Maximal depth: 5 Min rows:10.0 Min split improvements: 1.0E-5 Learning rate: 0.01 Sample rate: 1.0 Distribution: AUTO	PerformanceVector: accuracy: 82.41% ConfusionMatrix: True: INJ ADPD INJ: 36 4 ADPD: 15 53 precision: 77.94% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 36 4 ADPD: 15 53 recall: 92.98% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 36 4 ADPD: 15 53 AUC (optimistic): 0.929 (positive class: ADPD) AUC: 0.904 (positive class: ADPD) AUC (pessimistic): 0.879 (positive class: ADPD)
6	Random Forest Number of trees: 100 Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01 Minimal leaf size: 2 Guess subset ratio: Yes Voting strategy: confidence vote	PerformanceVector: accuracy: 77.78% ConfusionMatrix: True: INJ ADPD INJ: 27 0 ADPD: 24 57 precision: 70.37% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 27 0 ADPD: 24 57 recall: 100.00% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 27 0 ADPD: 24 57 AUC (optimistic): 0.940 (positive class: ADPD) AUC: 0.939 (positive class: ADPD) AUC (pessimistic): 0.937 (positive class: ADPD)
7	Linear Regression Min tolerance: 0.05 Ridge: 1.0E-8	PerformanceVector: accuracy: 83.33% ConfusionMatrix: True: INJ ADPD INJ: 36 3 ADPD: 15 54 precision: 78.26% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 36 3 ADPD: 15 54 recall: 94.74% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD

Model	Parameters	Results
8	Generalized Linear Model	INJ: 36 3 ADPD: 15 54 AUC (optimistic): 0.937 (positive class: ADPD) AUC: 0.936 (positive class: ADPD) AUC (pessimistic): 0.934 (positive class: ADPD)
		PerformanceVector: accuracy: 83.33% ConfusionMatrix: True: INJ ADPD INJ: 37 4 ADPD: 14 53 precision: 79.10% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 37 4 ADPD: 14 53 recall: 92.98% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 37 4 ADPD: 14 53 AUC (optimistic): 0.938 (positive class: ADPD) AUC: 0.936 (positive class: ADPD) AUC (pessimistic): 0.934 (positive class: ADPD)
		Family: AUTO Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No
9	Naive Bayes	PerformanceVector: accuracy: 79.63% ConfusionMatrix: True: INJ ADPD INJ: 29 0 ADPD: 22 57 precision: 72.15% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 29 0 ADPD: 22 57 recall: 100.00% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 29 0 ADPD: 22 57 AUC (optimistic): 0.914 (positive class: ADPD) AUC: 0.912 (positive class: ADPD) AUC (pessimistic): 0.911 (positive class: ADPD)
		PerformanceVector: accuracy: 83.33% ConfusionMatrix: True: INJ ADPD INJ: 37 4 ADPD: 14 53 precision: 79.10% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 37 4 ADPD: 14 53 recall: 92.98% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 37 4 ADPD: 14 53 AUC (optimistic): 0.938 (positive class: ADPD) AUC: 0.936 (positive class: ADPD) AUC (pessimistic): 0.934 (positive class: ADPD)
		Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No
10	Logistic Regression	PerformanceVector: accuracy: 83.33% ConfusionMatrix: True: INJ ADPD INJ: 37 4 ADPD: 14 53 precision: 79.10% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 37 4 ADPD: 14 53 recall: 92.98% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 37 4 ADPD: 14 53 AUC (optimistic): 0.938 (positive class: ADPD) AUC: 0.936 (positive class: ADPD) AUC (pessimistic): 0.934 (positive class: ADPD)

Model	Parameters	Results
11	Support Vector Machine	PerformanceVector:
		accuracy: 87.04%
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 44 7
		ADPD: 7 50
		precision: 87.72% (positive class: ADPD)
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 44 7
		ADPD: 7 50
		recall: 87.72% (positive class: ADPD)
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 44 7
ADPD: 7 50		
AUC (optimistic): 0.942 (positive class: ADPD)		
AUC: 0.940 (positive class: ADPD)		
AUC (pessimistic): 0.938 (positive class: ADPD)		

4.12 Results for SET12

Models were fed with data set SET07 with splitting data as 90% being training set and 10% being test data. Parameters and results can be found in Table 4.12. From the table Gradient Boosted Trees resulted in highest AUC with 0.704 (accuracy: 61.11%, precision: 58.82% and recall 87.72%)

Table 4.12 Results for SET12

Model	Parameters	Results
1	Neural Nets	PerformanceVector:
		accuracy: 67.59%
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 39 23
		ADPD: 12 34
		precision: 73.91% (positive class: ADPD)
		ConfusionMatrix:
		True: INJ ADPD
		INJ: 39 23
		ADPD: 12 34
		recall: 59.65% (positive class: ADPD)
		ConfusionMatrix:
		True: INJ ADPD
INJ: 39 23		
ADPD: 12 34		
AUC (optimistic): 0.686 (positive class: ADPD)		

Model	Parameters	Results
2	Deep Learning Activation: Rectifier Epochs:10.0 Compute variable importance: yes	AUC: 0.673 (positive class: ADPD) AUC (pessimistic): 0.660 (positive class: ADPD)
		PerformanceVector: accuracy: 58.33% ConfusionMatrix: True: INJ ADPD INJ: 18 12 ADPD: 33 45 precision: 57.69% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 18 12 ADPD: 33 45 recall: 78.95% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 18 12 ADPD: 33 45 AUC (optimistic): 0.659 (positive class: ADPD) AUC: 0.647 (positive class: ADPD) AUC (pessimistic): 0.634 (positive class: ADPD)
3	Fast Large Margin Solver: L2 SVM Dual C: 1.0 Epsilon: 0.01	PerformanceVector: accuracy: 57.41% ConfusionMatrix: True: INJ ADPD INJ: 23 18 ADPD: 28 39 precision: 58.21% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 23 18 ADPD: 28 39 recall: 68.42% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 23 18 ADPD: 28 39 AUC (optimistic): 0.662 (positive class: ADPD) AUC: 0.649 (positive class: ADPD) AUC (pessimistic): 0.637 (positive class: ADPD)
		PerformanceVector: accuracy: 58.33% ConfusionMatrix: True: INJ ADPD INJ: 20 14 ADPD: 31 43 precision: 58.11% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 20 14 ADPD: 31 43 recall: 75.44% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 20 14 ADPD: 31 43 AUC (optimistic): 0.770 (positive class: ADPD) AUC: 0.633 (positive class: ADPD) AUC (pessimistic): 0.496 (positive class: ADPD)
4	Decision Tree Criterion: gain_ratio Maximal depth: 10 Apply pruning: Yes Confidence: 0.1 Apply prepruning: Yes Minimal gain: 0.01	PerformanceVector: accuracy: 58.33% ConfusionMatrix: True: INJ ADPD INJ: 20 14 ADPD: 31 43 precision: 58.11% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 20 14 ADPD: 31 43 recall: 75.44% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 20 14 ADPD: 31 43 AUC (optimistic): 0.770 (positive class: ADPD) AUC: 0.633 (positive class: ADPD) AUC (pessimistic): 0.496 (positive class: ADPD)
		PerformanceVector: accuracy: 58.33% ConfusionMatrix: True: INJ ADPD INJ: 20 14 ADPD: 31 43 precision: 58.11% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 20 14 ADPD: 31 43 recall: 75.44% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 20 14 ADPD: 31 43 AUC (optimistic): 0.770 (positive class: ADPD) AUC: 0.633 (positive class: ADPD) AUC (pessimistic): 0.496 (positive class: ADPD)
5	Gradient Boosted Trees Number of trees: 50 Reproducible: no Maximal depth: 5	PerformanceVector: accuracy: 61.11% ConfusionMatrix:

Model	Parameters	Results
9	Naive Bayes	True: INJ ADPD INJ: 23 18 ADPD: 28 39 recall: 68.42% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 23 18 ADPD: 28 39 AUC (optimistic): 0.662 (positive class: ADPD) AUC: 0.650 (positive class: ADPD) AUC (pessimistic): 0.637 (positive class: ADPD)
		PerformanceVector: accuracy: 48.15% ConfusionMatrix: True: INJ ADPD INJ: 50 55 ADPD: 1 2 precision: 66.67% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 50 55 ADPD: 1 2 recall: 3.51% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 50 55 ADPD: 1 2 AUC (optimistic): 0.654 (positive class: ADPD) AUC: 0.641 (positive class: ADPD) AUC (pessimistic): 0.628 (positive class: ADPD)
		PerformanceVector: accuracy: 57.41% ConfusionMatrix: True: INJ ADPD INJ: 23 18 ADPD: 28 39 precision: 58.21% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 23 18 ADPD: 28 39 recall: 68.42% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 23 18 ADPD: 28 39 AUC (optimistic): 0.662 (positive class: ADPD) AUC: 0.650 (positive class: ADPD) AUC (pessimistic): 0.637 (positive class: ADPD)
10	Logistic Regression	Solver: AUTO Reproducible: no Use regularization: No Standardize: Yes Non-negative coefficients: No True: INJ ADPD INJ: 23 18 ADPD: 28 39 recall: 68.42% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 23 18 ADPD: 28 39 AUC (optimistic): 0.662 (positive class: ADPD) AUC: 0.650 (positive class: ADPD) AUC (pessimistic): 0.637 (positive class: ADPD)
11	Support Vector Machine	Kernel type: dot C: 1.0 Convergence epsilon: 0.001 True: INJ ADPD INJ: 40 32 ADPD: 11 25 precision: 69.44% (positive class: ADPD) ConfusionMatrix: True: INJ ADPD INJ: 40 32 ADPD: 11 25 recall: 43.86% (positive class: ADPD) ConfusionMatrix:

Model	Parameters	Results
		True: INJ ADPD
		INJ: 40 32
		ADPD: 11 25
		AUC (optimistic): 0.668 (positive class: ADPD)
		AUC: 0.655 (positive class: ADPD)
		AUC (pessimistic): 0.643 (positive class: ADPD)

All results for different data sets are listed in Table 4.13. The table shows performance of the eleven algorithms for twelve different data sets.



Table 4.13 Overview of the Results

		1					2					3				
		Neural Nets					Deep Learning					Fast Large Margin				
		Accuracy	Kappa	Precision	Recall	AUC	Accuracy	Kappa	Precision	Recall	AUC	Accuracy	Kappa	Precision	Recall	AUC
with	from ICs and BCs	72.03%	0.500	#	#	#	83.05%	0.691	#	#	#	*	*	#	#	#
	Date & Time	77.97%	0.589	#	#	#	80.51%	0.637	#	#	#	*	*	#	#	#
	from ICs	56.78%	0.196	#	#	#	55.93%	0.204	#	#	#	*	*	#	#	#
	from BCs	78.81%	0.614	#	#	#	81.36%	0.663	#	#	#	*	*	#	#	#
	without	77.12%	0.573	#	#	#	77.97%	0.592	#	#	#	*	*	#	#	#
	Date & Time	60.1%	0.273	#	#	#	55.08%	0.147	#	#	#	*	*	#	#	#
with	from ICs and BCs	78.70%	#	75.00%	89.47%	0.906	84.26%	#	82.26%	89.47%	0.913	52.78%	#	52.78%	100.00%	0.503
	Date & Time	86.11%	#	86.21%	87.72%	0.915	84.26%	#	79.41%	94.74%	0.927	52.78%	#	52.78%	100.00%	0.503
	from ICs	64.81%	#	63.77%	77.19%	0.697	54.63%	#	54.88%	78.95%	0.685	52.78%	#	52.78%	100.00%	0.503
	from BCs	83.33%	#	77.46%	96.49%	0.909	85.19%	#	83.61%	89.47%	0.943	83.33%	#	77.46%	96.49%	0.933
	without	83.33%	#	77.46%	96.49%	0.924	85.19%	#	80.60%	94.74%	0.908	82.41%	#	77.14%	94.74%	0.936
	Date & Time	67.59%	#	73.91%	59.65%	0.673	58.33%	#	57.69%	78.95%	0.647	57.41%	#	58.21%	68.42%	0.649
		4					5					6				
		Decision Tree					Gradient Boosted Trees					Random Forest				
		Accuracy	Kappa	Precision	Recall	AUC	Accuracy	Kappa	Precision	Recall	AUC	Accuracy	Kappa	Precision	Recall	AUC
with	from ICs and BCs	67.80%	0.390	#	#	#	76.27%	0.562	#	#	#	71.19%	0.455	#	#	#
	Date & Time	68.64%	0.406	#	#	#	77.12%	0.575	#	#	#	71.19%	0.455	#	#	#
	from ICs	55.93%	0.206	#	#	#	51.69%	0.104	#	#	#	56.78%	0.193	#	#	#
	from BCs	71.19%	0.456	#	#	#	75.42%	0.541	#	#	#	79.66%	0.619	#	#	#
	without	68.64%	0.409	#	#	#	75.42%	0.541	#	#	#	71.19%	0.455	#	#	#
	Date & Time	57.63%	0.229	#	#	#	50.85%	0.084	#	#	#	56.78%	0.194	#	#	#
with	from ICs and BCs	74.07%	#	67.90%	96.49%	0.871	77.78%	#	74.63%	87.72%	0.902	82.41%	#	75.00%	100.00%	0.949
	Date & Time	75.93%	#	68.67%	100.00%	0.884	80.56	#	77.27%	89.47%	0.893	77.78%	#	70.37%	100.00%	0.943
	from ICs	61.11%	#	77.78%	36.84%	0.623	55.56%	#	55.29%	82.46%	0.613	61.11%	#	59.74%	80.70%	0.660
	from BCs	78.70%	#	92.50%	64.91%	0.873	80.56%	#	73.08%	100.00%	0.918	81.48%	#	74.03%	100.00%	0.952
	without	86.11%	#	88.89%	84.21%	0.914	82.41%	#	77.94%	92.98%	0.904	77.78%	#	70.37%	100.00%	0.939
	Date & Time	58.33%	#	58.11%	75.44%	0.633	61.11%	#	58.82%	87.72%	0.704	57.41%	#	57.33%	75.44%	0.684
		7					8					9				
		Linear Regression					Generalized Linear Model					Naive Bayes				
		Accuracy	Kappa	Precision	Recall	AUC	Accuracy	Kappa	Precision	Recall	AUC	Accuracy	Kappa	Precision	Recall	AUC
with	from ICs and BCs	*	*	#	#	#	78.81%	0.612	#	#	#	21.19%	0.083	#	#	#
	Date & Time	*	*	#	#	#	79.66%	0.629	#	#	#	22.03%	0.084	#	#	#
	from ICs	*	*	#	#	#	55.08%	0.172	#	#	#	15.25%	0	#	#	#
	from BCs	*	*	#	#	#	78.81%	0.612	#	#	#	20.34%	0.077	#	#	#
	without	*	*	#	#	#	76.27%	0.560	#	#	#	21.19%	0.077	#	#	#
	Date & Time	*	*	#	#	#	52.54%	0.116	#	#	#	12.71%	0.008	#	#	#
with	from ICs and BCs	82.41%	#	76.39%	96.49%	0.939	84.26%	#	81.25%	91.23%	0.939	84.26%	#	88.46%	80.70%	0.903
	Date & Time	83.33%	#	78.26%	94.74%	0.936	86.11%	#	80.88%	96.49%	0.941	79.63%	#	72.15%	100.00%	0.916
	from ICs	55.56%	#	56.72%	66.67%	0.643	61.11%	#	61.90%	68.42%	0.639	48.15%	#	66.67%	3.51%	0.645
	from BCs	82.41%	#	76.39%	96.49%	0.939	83.33%	#	78.26%	94.74%	0.936	84.26%	#	90.00%	78.95%	0.903
	without	83.33%	#	78.26%	94.74%	0.936	83.33%	#	79.10%	92.98%	0.936	79.63%	#	72.15%	100.00%	0.912
	Date & Time	55.56%	#	56.72%	66.67%	0.643	57.41%	#	58.21%	68.42%	0.650	48.15%	#	66.67%	3.51%	0.641
		10					11									
		Logistic Regression					Support Vector Machine									
		Accuracy	Kappa	Precision	Recall	AUC	Accuracy	Kappa	Precision	Recall	AUC					
with	from ICs and BCs	*	*	#	#	#	*	*	#	#	#					
	Date & Time	*	*	#	#	#	*	*	#	#	#					
	from ICs	*	*	#	#	#	*	*	#	#	#					
	from BCs	*	*	#	#	#	*	*	#	#	#					
	without	*	*	#	#	#	*	*	#	#	#					
	Date & Time	*	*	#	#	#	*	*	#	#	#					
with	from ICs and BCs	84.26%	#	81.25%	91.23%	0.939	87.96%	#	84.38%	94.74%	0.942					
	Date & Time	86.11%	#	80.88%	96.49%	0.941	86.11%	#	85.00%	89.47%	0.941					
	from ICs	61.11%	#	61.90%	68.42%	0.639	62.04%	#	72.22%	45.61%	0.650					
	from BCs	83.33%	#	78.26%	94.74%	0.936	87.04%	#	83.08%	94.74%	0.939					
	without	83.33%	#	79.10%	92.98%	0.936	87.04%	#	87.72%	87.72%	0.940					
	Date & Time	57.41%	#	58.21%	68.42%	0.650	60.19%	#	69.44%	43.86%	0.655					

* Cannot handle polynomial label

4.13 Parameter Optimization

Parameter optimization was performed in Deep Learning for learning rate and learning momentum. Steps were set to 100 for both parameters and 10,000 combinations were tested in the model. It can be concluded from the results that there is an increase in performance which is listed in Table 4.14 in bold figures.

Table 4.14 Results for Deep Learning after parameter optimization for all data sets

				2 Deep Learning				
				Accuracy	Kappa	Precision	Recall	AUC
with Nearmiss	with Date & Time	from ICs and RCs	SET01	83.05% 88.14%	0.691 0.790	#	#	#
		From ICs	SET02	80.51% 86.44%	0.637 0.757	#	#	#
		From RCs	SET03	55.93% 69.49%	0.204 0.437	#	#	#
	without Date & Time	from ICs and RCs	SET04	81.36% 85.59%	0.663 0.740	#	#	#
		From ICs	SET05	77.97% 85.59%	0.592 0.746	#	#	#
		From RCs	SET06	55.08% 72.03%	0.147 0.487	#	#	#
without Nearmiss	with Date & Time	from ICs and RCs	SET07	84.26% 94.44%	#	82.26% 96.36%	89.47% 92.98%	0.913 0.979
		From ICs	SET08	84.26% 93.52%	#	79.41% 93.10%	94.74% 94.74%	0.927 0.984
		From RCs	SET09	54.63% 74.07%	#	54.88% 69.33%	78.95% 91.23%	0.685 0.762
	without Date & Time	from ICs and RCs	SET10	85.19% 93.52%	#	83.61% 98.08%	89.47% 89.47%	0.943 0.981
		From ICs	SET11	85.19% 92.59%	#	80.60% 94.55%	94.74% 91.23%	0.908 0.970
		From RCs	SET12	58.33% 71.30%	#	57.69% 65.85%	78.95% 94.74%	0.647 0.808

As seen from the Table 4.13, Deep learning produced better results for data subsets with three outcomes, namely SET01, SET02, SET03, SET04 and SET05 apart from SET06. Random Forest was better in predicting outcomes for SET07, SET08 and SET010. For SET06 and SET09, Neural Nets was more successful than other models. Support Vector Machines and Gradient Boosted Trees showed better performance for SET11 and SET12 respectively.

The performance of deep learning was lower than the performance of the rest of the models for data subsets without polynomial label in which near miss was removed from consequences, however, it was not much lower than performance of other models.

Accuracy and kappa were highest in predicting consequences with Deep Learning for SET01 with 83.05% and 0.691 respectively. Performance of the model decreased for SET02 and SET03 in which consequences were only predicted from immediate causes and root causes respectively. Performance of deep learning dropped to a low of 55.93% accuracy and 0.592 kappa for SET03, therefore, it can be concluded that predicting consequences from root cause is decreasing performance of the model.

Comparing all models in terms of performance of predicting from immediate and root cause, it is clear from the results that immediate causes led to higher kappa and accuracy for all data sets. Performance of the models decreases in predicting outcomes from root causes.

Performance of the naïve bayes was lower than any other model for data subsets SET01 to SET06.

Deep learning was selected for parameter optimization due to performing better in predicting complex data sets that means predicting all consequences including near miss from date and time, all immediate and root causes.

After optimization of learning rate and learning momentum all performance were increased. The highest AUC, accuracy, precision, and recall were for SET07 with 0.979, 94.44%, 96.36% and 92.98% respectively. Figures for deep learning with optimized parameters became higher than the results of the other models for all data subsets.





CHAPTER 5

CONCLUSION

The main objective of this research was to introduce a new accident prevention model called prescriptive model. This model uses machine learning algorithms for predicting outcomes of work-related accidents. Based on the results introduced in this study, predictive models are successful in predicting outcomes of accidents from attributes. Only deep learning was selected for performance optimization due to having higher performance for 5 of 12 data subsets compared with other algorithms.

It was seen that performance of the models decreases with decreasing number of attributes. Predicting outcomes from only root causes showed lower performance than predicting from immediate causes. Applying deep learning on the original data set SET01 showed good performance in comparing deep learning for other data subset.

From the results introduced in Table 4.14, it is clear that performance of the deep learning is improved after parameter optimization for all data subsets.

To sum up, a zero-harm model is possible if consequences is predicted correctly. These predictions can be helpful in taking proactive or generative measures to avoid accidents. The proposed zero-harm or prescriptive accident prevention model represented in Figure 2.9 can be introduced a new accident prevention model. Learning from failures still need to be studied with more sophisticated machine learning algorithms to reveal the potential.

The main limitation is the missing details in the accident investigation reports and reports being prepared by different people.

Reports having different incident investigation format, missing details such as age, experience, education of the injured person or statements might be misleading in predicting outcomes.

Quality of the reports was also open to discussion taking background of the people filling forms and especially their experience and training on this. This was avoided with cross checking all other data available with the incident investigation reports up to an extent.

Immediate causes were selected according to the nature of the work done and may not be effective in representing immediate causes for other industries.

Since this is just an initial idea for application it can be further improved using much more deep machine learning methods and with comprehensive parameter optimization. Due to the limited computing power, only two parameters were selected for optimization.

As a further study, all accidents of a similar project can be selected, and data set can be prepared from the project with the same methodology applied in this study. Processed data set from new project can be used as test set to see how successful the proposed model is.

REFERENCES

- Anderson, J. R., & Kline, P. J. (1979). A learning system and its psychological implications. *Proceedings of the Sixth International Joint Conference on Artificial Intelligence* (pp. 16-21). Tokyo, Japan: Morgan Kaufmann.
- Antaki, G.A. (2003). *Piping and Pipeline Engineering Design, Construction, Maintenance, Integrity, and Repair*. Aiken, South Carolina, U.S.A.
- Arockia, A.B. and Appavu C.S. (2013). *Data Mining Approaches for Aircraft Accidents Prediction, Emerging Trends in Computing, Communication and Nanotechnology*, Pages 25-26.
- Ayhan, B. U., and Tokdemir, O. B. (2019). Predicting the outcome of construction incidents. *Safety Science*, 113, 91–104. <https://doi.org/10.1016/j.ssci.2018.11.001>
- Ayhan, B.U. and Tokdemir, O.B. (2019). Predicting the outcome of construction incidents. *Saf.Sci.* 113 (September 2018), 91–104.
- Bahadori, A. (2017). *Transportation Pipelines*. In *Oil and Gas Pipelines and Piping Systems* (pp. 1–27). <https://doi.org/10.1016/b978-0-12-803777-5.00001-0>.
- Balram, Y., Sravan Kumar, Sridhar Babu, B., Vishu Vardhan, T., & Venkat Ramana, G. (2019). Effect of filler wires on weld strength of dissimilar pulse GTA Monel 400 and AISI 304 weldments. *Materials Today: Proceedings*. <https://doi.org/10.1016/j.matpr.2019.06.759>
- Barkin, E. (May 2011), *CRM + Predictive Analytics: Why It All Adds Up, Destination CRM*.
- Bauer, W. E. (2011). *Right-of-Way. Pipeline Planning and Construction Field Manual*.
- Bevilacqua, M., Ciarapica, F. E., & Giacchetta, G. (2008). Industrial and occupational ergonomics in the petrochemical process industry: A regression trees approach. *Accident Analysis & Prevention*, 40(4), 1468–1479. <https://doi.org/10.1016/j.aap.2008.03.012>
- Breiman, L., Friedman, J.H., Olshen, R.A., and Stone C.J. (1998). *Classification and regression trees*.
- Brownlee, J. (2016). *Master Machine Learning Algorithms*.
- Brownlee, J. (2018). *Basics of Linear Algebra for Machine Learning*.

- Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1956). *A study of thinking*. New York: John Wiley and Sons.
- Bubar, B. G. (2011). *Welding and NDT. Pipeline Planning and Construction Field Manual*.
- Bureau of Labor Statistics, U.S. Department of Labor, *The Economics Daily*, Fatal and nonfatal falls, slips, and trips in the construction industry at <https://www.bls.gov/opub/ted/2021/fatal-and-nonfatal-falls-slips-and-trips-in-the-construction-industry.htm> (visited December 03, 2021).
- Central Intelligence Agency. (2017). *The world factbook*. Retrieved from <https://www.cia.gov/library/publications/the-world-factbook/fields/2117.html> on December 2019.
- Cervantes, J., García Lamont, F., López-Chau, A., Rodríguez Mazahua, L., & Sergio Ruíz, J. (2015). Data selection based on decision tree for SVM classification on large data sets. *Applied Soft Computing*, 37, 787–798. <https://doi.org/10.1016/j.asoc.2015.08.048>
- Cheng, L., Chen, X., De Vos, J., Lai, X. and Witlox, F. (2019). Applying a random forest method approach to model travel mode choice behavior. *Travel Behav. Soc.* 14, 1–10.
- Cianciarullo, M.I. (2019). Green construction– reduction in environmental impact through alternative pipeline water crossing installation, *Journal of Cleaner Production*, Volume 223, Pages 1042-1049.
- Clarke, B.S. and Clarke, J.L. (2018). *Predictive Statistics, Analysis and Inference beyond Models*. Cambridge University Press.
- Clevert, D., Unterthiner, T., & Hochreiter, S. (2015). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *CoRR*, abs/1511.07289.
- Collobert, R., & Bengio, S. (2004). Links between perceptrons, MLPs and SVMs. *Twenty-First International Conference on Machine Learning - ICML '04*. Presented at the Twenty-first international conference. <https://doi.org/10.1145/1015330.1015415>
- Conz, N. (2008). *Insurers Shift to Customer-focused Predictive Analytics Technologies*, Insurance & Technology.
- Das, K. and Vidyashankar, G.S. (2006). *Competitive Advantage in Retail Through Analytics: Developing Insights, Creating Value, Information Management*.

- Davoudi Kakhki, F., Freeman, S. A., & Mosher, G. A. (2019). Evaluating machine learning performance in predicting injury severity in agribusiness industries. *Safety Science*, 117, 257–262. <https://doi.org/10.1016/j.ssci.2019.04.026>
- Davoudi Kakhki, F., Freeman, S. A., & Mosher, G. A. (2019). Evaluating machine learning performance in predicting injury severity in agribusiness industries. *Safety Science*, 117, 257–262. <https://doi.org/10.1016/j.ssci.2019.04.026>
- de Leeuw, J. (2001). Scaling: Multidimensional. In *International Encyclopedia of the Social & Behavioral Sciences* (pp. 13512–13519). <https://doi.org/10.1016/b0-08-043076-7/00501-5>
- Dean, J., Corrado, G. S., Monga, R., Chen, K., Devin, M., Le, Q. V, ... Ng, A. Y. (2012). Large Scale Distributed Deep Networks. *NIPS 2012: Neural Information Processing Systems*, 1–11.
- Dewa, C. K., & Afiahayati. (2018). Suitable CNN Weight Initialization and Activation Function for Javanese Vowels Classification. *Procedia Computer Science*, 144, 124–132. <https://doi.org/10.1016/j.procs.2018.10.512>
- Dey, P.K and Tabucanon, M.T. (1996). Petroleum pipeline construction planning: a conceptual framework. *International Journal of Project Management* Vol. 14, No. 4, pp. 231-240.
- Dietterich, T. G., and Michalski, R. S. (1983). A comparative review of selected methods of learning from examples.
- Dong, X.S., Choi, S.D., Borchardt, J.G., Wang, X., and Largay, J.A., (2013). Fatal falls from roofs among U.S. construction workers. *J. Saf. Res.* 44, 17–24.
- Dorling, D and Gianetto, J. (2015). Pipeline Welding from the Perspective of Safety and Integrity. *Oil and Gas Pipelines: Integrity and Safety Handbook*, First Edition. Edited by R. Winston Revie.
- Dozat, T. (2016). Incorporating Nesterov Momentum into Adam. *ICLR Workshop*, (1), 2013–2016.
- Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12, 2121–2159.
- Eckerson, W. (2007). Extending the Value of Your Data Warehousing Investment, The Data Warehouse Institute.
- Engel, Y. and Etzion O. (2011). Towards proactive event-driven computing. In: *Proceedings of the 5th ACM international conference on distributed event-based system*. New York

- Engel, Y. Etzion, O., and Feldman, Z. (2012). A basic model for proactive event-driven computing. In: Proceedings of the 6th ACM international conference on distributed event-based systems, Berlin
- Esmaeili, B. and Hallowell M. 2012. Attribute-Based Risk Model for Measuring Safety Risk of Struck-by Accidents, Construction Research Congress May 2012, pp. 289–298.
- Everitt, B. (1981). Cluster analysis. London: Heinemann.
- Farnaaz, N. and Jabbar, M.A., 2016. Random forest modeling for network intrusion detection system. *Procedia Comput. Sci.* 89, 213–217.
- Finlay, S. (2014). Predictive Analytics, Data Mining and Big Data. <https://doi.org/10.1057/9781137379283>
- Fisher, D. and Pazzani, M. (1991). Computational Models of Concept Learning.
- Fletcher, H. (2011). The 7 Best Uses for Predictive Analytics in Multichannel Marketing, Target Marketing.
- Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1), 119–139. <https://doi.org/10.1006/jcss.1997.1504>
- Fülöp, L. J., Beszédes, Á., Tóth, G., Demeter, H., Vidács, L., & Farkas, L. (2012). Predictive complex event processing. Proceedings of the Fifth Balkan Conference in Informatics on - BCI '12. Presented at the the Fifth Balkan Conference in Informatics. <https://doi.org/10.1145/2371316.2371323>
- Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn & TensorFlow. Concepts, Tools, And Techniques to Build Intelligent Systems.
- Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of International Conference on Artificial Intelligence and Statistics, pp. 249–256.
- Goh, Y.M., and Ubeynarayana, C.U. (2017). Construction accident narrative classification: an evaluation of text mining techniques. *Accid. Anal. Prev.* 108, 122–130.
- Hackathorn, R. (2003) Minimizing Action Distance. The Data Administration Newsletter. (2003)
- Hannachi, A., & Turner, A. G. (2013). Isomap nonlinear dimensionality reduction and bimodality of Asian monsoon convection. *Geophysical Research Letters*, 40(8), 1653–1658. <https://doi.org/10.1002/grl.50351>

- Hayes-Roth, F., & McDermott, J. (1978). An interference matching technique for inducing abstractions. *Communications of the ACM*, 21, 401-410.
- Haykin, S. (2005). *Neural Networks; A Comprehensive Foundation*. Ontario, Canada.
- He, K., Zhang, X., Ren, S., & Sun, J., (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034.
- He, X., Chen, W., Nie, B., & Zhang, M. (2010). Classification technique for danger classes of coal and gas outburst in deep coal mines. *Safety Science*, 48(2), 173–178. <https://doi.org/10.1016/j.ssci.2009.07.007>
- Hollnagel, E., Wears, R. L., & Braithwaite, J. (2015). *From Safety-I to Safety-II: A White Paper*. University of Southern Denmark. <https://doi.org/10.13140/RG.2.1.4051.5282>
- HSE. Health and Safety Executive. (2014). *Annual Report and Account 2013/2014*. Her Majesty's Stationery Office, London.
- Hunt, E., Marin, J., & Stone, P. (1966). *Experiments in induction*. New York: Academic Press.
- International Labor Organization. (2016). *Safety and Health at Work*. Retrieved from <http://www.ilo.org/ilostat/> on December 2019.
- Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.
- Jain, A. K., & Dubes, R. C. (1988). *Algorithms for cluster analysis*. Englewood Cliffs, NJ: Prentice Hall.
- Jiang, P., Maghrebi, M., Crosky, A., & Saydam, S. (2017). Unsupervised Deep Learning for Data-Driven Reliability and Risk Analysis of Engineered Systems. In *Handbook of Neural Computation* (pp. 417–431). <https://doi.org/10.1016/b978-0-12-811318-9.00023-5>
- Kaiser, M.J. (2017). Offshore pipeline construction cost in the U.S. Gulf of Mexico, *Marine Policy*, Volume 82, Pages 147-166.
- Kang, K and Ryu H. (2019). Predicting types of occupational accidents at construction sites in Korea using random forest model, *Safety Science*, Volume 120, Pages 226-236

- Kang, K., & Ryu, H. (2019). Predicting types of occupational accidents at construction sites in Korea using random forest model. *Safety Science*, 120, 226–236. <https://doi.org/10.1016/j.ssci.2019.06.034>
- Kecman, V. (2005). Support Vector Machines, An Introduction, *StudFuzz* 177, 1–47.
- Kingma, D. P., & Ba, J. L. (2015). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 1–13.
- Korn, S. (2011). The Opportunity for Predictive Analytics in Finance, *HPC Wire*.
- Krumeich, J., Werth, D., & Loos, P. (2015). Prescriptive Control of Business Processes. *Business & Information Systems Engineering*, 58(4), 261–280. <https://doi.org/10.1007/s12599-015-0412-2>
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. <https://doi.org/10.1007/978-1-4614-6849-3>
- Langley, P. (1987). A general theory of discrimination learning. In D. Klahr, P. Langley, & D. Neches (Eds.), *Production system models of learning and development*. Cambridge, MA: MIT Press.
- Larson, D., & Chang, V. (2016). A review and future direction of agile, business intelligence, analytics and data science. *International Journal of Information Management*, 36(5), 700–710.
- Lepenioti, K., Bousdekis, A., Apostolou, D., & Mentzas, G. (2020). Prescriptive analytics: Literature review and research challenges. *International Journal of Information Management*, 50, 57–70. <https://doi.org/10.1016/j.ijinfomgt.2019.04.003>
- Li, L.P. and Wang, S. (2004). Trends in injury-related incidence and mortality among inpatients in Guangdong Province in 1997–2001. *Biomed. Environ. Sci.* 17, 333–340.
- Liao, C.W. and Perng, Y.H. (2008). Data mining for occupational injuries in the Taiwan construction industry. *Saf. Sci.* 46 (7), 1091–1102
- Lowd, D. and Domingos, P. (2005). Naïve Bayes Models for Probability Estimation. In: *Proceedings of 22nd International Conference on Machine Learning*, Bonn, Germany, 529-536.
- Lucas, J., Sun S., Zemel, R. and Grosse, R. (2019). *Aggregated Momentum: Stability Through Passive Damping*. University of Toronto.

- Masetic, Z., and Subasi, A. (2016). Congestive heart failure detection using random forest classifier. *Comput. Methods Programs Biomed.* 130, 54–64.
- Matías, J. M., Rivas, T., Martín, J. E., & Taboada, J. (2008). A machine learning methodology for the analysis of workplace accidents. *International Journal of Computer Mathematics*, 85(3–4), 559–578. <https://doi.org/10.1080/00207160701297346>
- McDonald, M. (2010). New Technology Taps 'Predictive Analytics' to Target Travel Recommendations, *Travel Market Report*.
- McKay, L. (2009). The New Prescription for Pharma, *Destination CRM*
- Medin, D. (1983). Structural principles of categorization. In T. Tighe and B. Shepp (Eds.), *Perception, cognition, and development*. Hillsdale, NJ: Lawrence Erlbaum.
- Miner, G., Delen, D., Elder, J., Fast, A., Hill, T., and Nisbet, R. (2012) *Practical Text Mining and Statistical Analysis for Non-Structured Text Data Applications*. Academic Press, Waltham MA.
- Minsky, M., & Papert, S. (1969). *Perceptrons*. M.I.T. Press.
- Misiurek, K. and Misiurek, B. (2017). Methodology of improving occupational safety in the construction industry on the basis of the TWI program. *Saf. Sci.* 92, 225–231.
- Mistikoglu, G., Gerek, I.H., Erdis, E., Usmen, P.M., Cakan, H., and Kazan, E.E., (2015). Decision tree analysis of construction fall accidents involving roofers. *Expert Syst. Appl.* 42, 2256–2263.
- Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence*, 18, 203–226.
- Mohammadi, A., Tavakolan, M., and Khosravi, Y. (2018). Factors influencing safety performance on construction projects: a review. *Saf. Sci.* 109, 382–397.
- Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., & Damas, L. (2016). Time-evolving O-D matrix estimation using high-speed GPS data streams. *Expert Systems with Applications*, 44, 275–288. <https://doi.org/10.1016/j.eswa.2015.08.048>
- Nazeri, Z., Donohue, G. and Sherry, L. (2008). Analyzing Relationships between Aircraft Accidents and Incidents, *International Conference on Research in Air Transportation (ICRAT)*, Pages 185–190.

- Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. Doklady ANSSSR (translated as Soviet.Math.Docl.), vol. 269, pp. 543– 547.
- Nguyen, L.D., Tran, D.Q., and Chandrawinata, M.P., (2016). Predicting safety risk of working at heights using Bayesian networks. J. Construct. Eng. Manage. 142 (9), 04016041.
- Nyce, C. (2007). Predictive Analytics White Paper (PDF), American Institute for Chartered Property Casualty Underwriters/Insurance Institute of America, p. 1.
- Olson, D. L., Delen, D., & Meng, Y. (2012). Comparative analysis of data mining methods for bankruptcy prediction. Decision Support Systems, 52(2), 464–473. <https://doi.org/10.1016/j.dss.2011.10.007>
- Oz, N. E., Mete, S., Serin, F., & Gul, M. (2018). Risk assessment for clearing and grading process of a natural gas pipeline project: An extended TOPSIS model with Pythagorean fuzzy sets for prioritizing hazards. Human and Ecological Risk Assessment: An International Journal, 1–18.
- Ozanne, H.S. (2011). Route Selection. Pipeline Planning and Construction Field Manual.
- Oztekin A., Kong ZJ, and Delen D. (2010). Development of a structural equation modeling-based decision tree methodology for the analysis of lung transplantations. Decis Support Syst 2011; 51:155–66. doi: 10.1016/j.dss.2010.12.004.
- Pang, S., & Gong, J. (2009). C5.0 Classification Algorithm and Application on Individual Credit Evaluation of Banks. Systems Engineering - Theory & Practice, 29(12), 94–104. [https://doi.org/10.1016/s1874-8651\(10\)60092-0](https://doi.org/10.1016/s1874-8651(10)60092-0)
- Pazzani, M. J., & Sarrett, W. (1990). Average case analysis of conjunctive learning algorithms. Proceedings of the Seventh International Conference on Machine Learning (pp. 339-347). Austin, TX: Morgan Kaufmann.
- PCRA. (2018). Pipeline Construction Risk Assessment. Global Maritime.
- Phark, C., Kim, W., Yoon, Y.-S., Shin, and G., Jung, S. (2018). Prediction of issuance of emergency evacuation orders for chemical accidents using machine learning algorithm, Journal of Loss Prevention in the Process Industries, doi: 10.1016/j.jlp.2018.08.021.
- Pipeline and Gas Journal. (2018). 2017. Worldwide Construction Report <https://pgjonline.com/> on December 2019.

- Poh, C.Q., Ubeynarayana, C.U., and Goh, Y.M. (2018). Safety leading indicators for construction sites: a machine learning approach. *Automation in Construction*.
- Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks: The Official Journal of the International Neural Network Society*, 12(1), 145–151. [http://doi.org/10.1016/S0893-6080\(98\)00116-6](http://doi.org/10.1016/S0893-6080(98)00116-6)
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1, 81-106.
- Quinlan, J.R. (2007). “C5”. <http://www.rulequest.com>
- Robinson, S.D., Irwin, W.J., Kelly, T.K., and Wu, X.O., (2015). Application of machine learning to mapping primary causal factors in self reported safety narratives. *Saf. Sci.* 75, 118–129.
- Rosenblatt, F. (1957). The perceptron — a perceiving and recognizing automaton (Technical Report 85-460-1). Cornell Aeronautical Laboratory.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536. <https://doi.org/10.1038/323533a0>
- Sánchez, A.S., Fernández, P.R., Lasheras, F.S., Juez, F.J.D.C., Nieto, P.J.G. (2011). Prediction of work-related accidents according to working conditions using support vector machines. *Appl. Math. Comput.* 218, 3539–3552. <https://doi.org/10.1016/j.amc.2011.08.100>.
- Sanmiquel, L., Bascompta, M., Rossell, J., Anticoi, H., & Guash, E. (2018). Analysis of Occupational Accidents in Underground and Surface Mining in Spain Using Data-Mining Techniques. *International Journal of Environmental Research and Public Health*, 15(3), 462. <https://doi.org/10.3390/ijerph15030462>
- Sarkar, S., Vinay, S., Raj, R., Maiti, J., & Mitra, P. (2019). Application of optimized machine learning techniques for prediction of occupational accidents. *Computers & Operations Research*, 106, 210–224. <https://doi.org/10.1016/j.cor.2018.02.021>
- SGK (Social Security Institution), “2020 Statistical Yearbooks”, SGK Yayini, Ankara, 2020
- Shmueli, G., Patel, N. R., & Bruce, P. C. (2007). *Data mining for business intelligence*. New Jersey: John Wiley & Sons Inc.
- Siegel, E. (2016). *Predictive Analytics: the Power To Predict Who Will Click, Buy, Lie, Or Die*. Hoboken, N.J: Wiley.

- Šikšnys, L., & Pedersen, T. B. (2016). Prescriptive analytics. In L. Liu, & M. Özsu (Eds.). *Encyclopedia of database systems*. New York, NY: Springer.
- Simon, H. A., & Lea, G. (1974). Problem solving and rule induction: A unified view. In L. W. Gregg (Ed.), *Knowledge and cognition*. Hills dale, NJ: Lawrence Erlbaum.
- Smarra, F., Jain, A., de Rubeis, T., Ambrosini, D., D’Innocenzo, A., and Mangharam, R. (2018). Data-driven model predictive control using random forests for building energy optimization and climate control. *Appl. Energy*.
- Srivastava, N., Hinton, G., & Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 15. 1929-1958.
- Stevenson, E. (2011). "Tech Beat: Can you pronounce health care predictive analytics? Times-Standard.
- Suárez Sánchez, A., Riesgo Fernández, P., Sánchez Lasheras, F., de Cos Juez, F. J., & García Nieto, P. J. (2011). Prediction of work-related accidents according to working conditions using support vector machines. *Applied Mathematics and Computation*, 218(7), 3539–3552. <https://doi.org/10.1016/j.amc.2011.08.100>
- Tam, C.M., Tong, T.K.L. and Chan, K.K. (2006). Rough set theory for distilling construction safety measures. *Construct. Manage. Econ.* 24 (11), 1199–1206.
- Tam, C.M., Zeng, S.X., and Deng, Z.M. (2004). Identifying elements of poor construction safety management in China. *Saf. Sci.* 42 (7), 569–586.
- Tanaka, K., Kinkyo, T., and Hamori, S. (2016). Random forests-based early warning system for bank failures. *Economics Letters* 148, 118–121.
- Tang, L., Cai, F., and Ouyang, Y. (2018). Applying a nonparametric random forest algorithm to assess the credit risk of the energy industry in China. *Technol. Forecast. Soc. Chang*.
- Tarkhaneh, O., & Shen, H. (2019). Training of feedforward neural networks for data classification using hybrid particle swarm optimization, Mantegna Lévy flight and neighborhood search. *Heliyon*, 5(4), e01275. <https://doi.org/10.1016/j.heliyon.2019.e01275>
- Tian,J. Rong,H &Zhao,T. (2015). Hybrid Safety analysis method based on SVM and RST: An application to carrier landing of aircraft, *School of Reliability and Systems Engineering*, Vol. 80, Pages 56-65.

- Tixier, A.J.P., Hallowell, M.R., Rajagopalan, B., and Bowman, D. (2016). Application of machine learning to construction injury prediction. *Autom. Constr.* 69, 102–114.
- Vere, S. A. (1980). Multilevel counterfactuals for generalization of relational concepts and productions. *Artificial Intelligence*, 14, 139–164.
- Vomfell, L., Härdle, W.K., Lessmann, S. (2018). Improving crime count forecasts using twitter and taxi data. *Decis. Support Syst.* 113, 73–85.
- Wang, Z., Lai, C., Chen, X., Yang, B., Zhao, S., and Bai, X. (2015). Flood hazard risk assessment model based on random forest. *J. Hydrol.* 527, 1130–1141.
- Wininger, G. A. (2011). Pipeline Construction. Pipeline Planning and Construction Field Manual.
- Witten, I.H. and Frank, E. (2005). Data Mining. Practical Machine Learning Tools and Techniques.
- Woergoetter, F., & Porr, B. (2008). Reinforcement learning. *Scholarpedia*, 3(3), 1448. <https://doi.org/10.4249/scholarpedia.1448>
- WSBC. (2017). Workers' Compensation Board of British Columbia. Vancouver, Canada.
- Yi, W., Chan, A. P. C., Wang, X., & Wang, J. (2016). Development of an early-warning system for site work in hot and humid environments: A case study. *Automation in Construction*, 62, 101–113. <https://doi.org/10.1016/j.autcon.2015.11.003>
- Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method.
- Zhang, S., Teizer, J., Lee, J.K., Eastman, C.M., and Venugopal, M. (2013). Building Information Modeling (BIM) and safety: automatic safety checking of construction models and schedules. *Autom. Constr.* 29, 183–195.
- Zhu, W., Zeng, N., and Wang, N. (2010). Sensitivity, Specificity, Accuracy, Associated Confidence Interval and ROC, Analysis with Practical SAS Implementations. *Health Care and Life Sciences*.
- Zinter, G. (2019). In-The-Ditch Pipeline Inspection. *Oil and Gas Pipelines: Integrity and Safety Handbook*, First Edition. Edited by R. Winston Revie.
- zur Muehlen, M., Shapiro, R. (2010). Business Process Analytics. In: vom Brocke, J., Rosemann, M. (eds.): *Handbook on Business Process Management*, Vol. 2, pp. 137–157. Springer, Berlin.