

NOVEMBER 2019

M.Sc. in Electronics and Computer Engineering

NIHAT YILMAZ ŞİMŞEK

**T.C.
HASAN KALYONCU UNIVERSITY
GRADUATE SCHOOL OF
NATURAL AND APPLIED SCIENCES**

**A COMPARATIVE STUDY OF DEEP LEARNING
METHODS FOR CLASSIFICATION OF RNA-SEQ
CANCER DATA**

**M.Sc. THESIS
IN
ELECTRONICS AND COMPUTER ENGINEERING**

**BY
NIHAT YILMAZ ŞİMŞEK
NOVEMBER 2019**

**A Comparative Study of Deep Learning Methods for
Classification of RNA-Seq Cancer Data**

M.Sc. Thesis

In

Electronics and Computer Engineering

Hasan Kalyoncu University

Supervisor

Asst. Prof. Dr. Bülent HAZNEDAR

Nihat Yılmaz ŞİMŞEK

November 2019

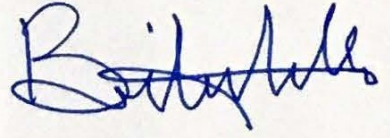
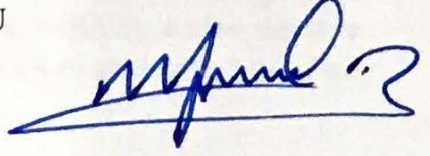



© 2019 [NİHAT YILMAZ ŞİMŞEK]



**GRADUATE SCHOOL OF NATURAL &
APPLIED SCIENCES INSTITUTE
MSc ACCEPTANCE AND APPROVAL FORM**

Electronics-Computer Engineering M.Sc. (Master Of Science) programme student **Nihat Yılmaz ŞİMŞEK** prepared and submitted the thesis titled " **A Comparative Study Of Deep Learning Methods For Classification Of RNA-Seq Cancer Data**" defended successfully on the date of 29/11/2019 and accepted by the jury as an M.Sc. thesis.

<u>Position</u>	<u>Title, Name and Surname</u> <u>Department/University</u>	<u>Signature:</u>
M.Sc. Supervisor	Assist. Prof. Dr. Bülent HAZNEDAR Computer Engineering Department Hasan Kalyoncu University	
Jury Head	Assoc. Prof. Dr. M. Fatih HASOĞLU Computer Engineering Department Hasan Kalyoncu University	
Jury Member	Assist. Prof. Dr. Mücahid GÜNAY Computer Engineering Department Kahramanmaraş Sütçü İmam University	

This thesis is accepted by the jury members selected by the institute management board and approved by the institute management board.


Prof. Dr. Mehmet KARPUZCU

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.

Nihat Yılmaz ŐİMŐEK

ABSTRACT
A COMPARATIVE STUDY OF DEEP LEARNING METHODS FOR
CLASSIFICATION OF RNA-Seq CANCER DATA

ŞİMŞEK, Nihat Yılmaz

M.Sc. in Electronic - Computer Engineering

Supervisor: Asst. Prof. Dr. Bülent HAZNEDAR

November 2019

99 pages

Cancer is one of the most important causes of deaths today. Millions of people die because of cancer every year, while millions of people are diagnosed with cancer. Cancer is a gene disease. As a result of mutations in genes, cells become abnormal and uncontrolled division is the main cause of cancer disease. Therefore, gene expression is very important in the diagnosis and classification of cancer. RNA-Seq data stores information of many genes. Many of these genes found on RNA-Seq data have nothing to do with cancer. Finding which genes cause cancer and then diagnosing the type of cancer is a long time process. Decision support systems can be developed using classification algorithms or deep learning methods to shorten this process and assist doctors in the diagnosis process.

The aim of this thesis is to analyze the cancer type using classical methods, artificial neural networks and deep learning methods by using RNA-Seq datasets created with genes obtained from previously diagnosed cancer patients. First, gene selection is made using wrapper methods to reduce the size of the RNA-Seq data set. The selected genes are then used in the classification process. For classification, decision trees, random forests, support vector machines, artificial neural networks and deep learning methods are used. After this study, which method works better in cancer classifications is examined. The method developed according to the results is expected to help doctors in the process of cancer classification.

Keywords: Cancer, Gene Expression, RNA-Seq, Classification, Deep Learning

ÖZET

RNA-Seq KANSER VERİLERİNİN SINIFLANDIRILMASI İÇİN DERİN ÖĞRENME YÖNTEMLERİNİN KARŞILAŞTIRMALI BİR ÇALIŞMASI

ŞİMŞEK, Nihat Yılmaz

Yüksek Lisans, Elektronik Bilgisayar Mühendisliği

Tez Danışmanı: Dr. Öğr. Üyesi. Bülent HAZNEDAR

Kasım 2019

99 sayfa

Kanser günümüzde ölüm sebeplerinin en başında gelmektedir. Her yıl milyonlarca insan kanserden ölürken, milyonlarca insana ise kanser teşhisi konmaktadır. Kanser bir gen hastalığıdır. Genlerde meydana gelen mutasyonlar sonucu hücrelerin anormal hale gelmesi ve kontrolsüz bir şekilde bölünmesi kanser hastalığının başlıca sebebidir. Bu nedenle kanser hastalığının teşhisinde ve sınıflandırılmasında gen ifadeleri büyük bir öneme sahiptir. RNA-Seq verileri birçok genin bilgilerini saklamaktadır. RNA-Seq verileri üzerinde bulunan bu genlerden birçoğunun kanserle bir ilgisi yoktur. Hangi genlerin kansere sebep olduğunu bulmak ve sonrasında kanser türünün teşhisi çok fazla zaman isteyen bir süreçtir. Bu süreci kısaltmak ve doktorlara teşhis sürecinde yardımcı olmak için sınıflandırma algoritmaları veya derin öğrenme metotları kullanılarak karar destek sistemleri geliştirilebilir.

Bu tezin amacı, daha önce kanser teşhisi konulmuş hastalardan elde edilen genler ile oluşturulmuş RNA-Seq veri kümeleri kullanılarak kanser türünün klasik yöntemler, yapay sinir ağları ve derin öğrenme metotları kullanılarak analiz edilmesidir. Öncelikle RNA-Seq veri kümesinin boyutunu azaltmak için sarmal yöntemler kullanılarak gen seçimi yapılır. Daha sonra seçilen genler sınıflandırma işleminde kullanılır. Sınıflandırma için karar ağaçları, rastgele ormanlar, destek vektör makineleri, yapay sinir ağları ve derin öğrenme kullanılır. Bu çalışmadan sonra kanser sınıflandırmalarında hangi yöntemin daha iyi sonuç verdiği incelenir. Elde edilen sonuçlara göre geliştirilen metodun kanser sınıflandırması sürecinde doktorlara yardımcı olması beklenmektedir.

Anahtar Kelimeler: Kanser, Gen İfadesi, RNA-Seq, Sınıflandırma, Derin Öğrenme



To My Beloved Parents and Lovely Fiance

ACKNOWLEDGEMENTS

I want to thank my supervisor Asst. Prof. Dr. Bülent HAZNEDAR who supported and encouraged me during Master process. It was a big proud to study with him due to his advices, guidance and motivation.

I want to thank my lovely fiance Ana Sigeti for supporting me in each stage of my work.

A special thanks to my mother, my father and my siblings who taught me how to stand up in difficult moments.



TABLE OF CONTENTS

	Pages
ABSTRACT.....	iv
ÖZET.....	v
ACKNOWLEDGEMENTS.....	vii
TABLE OF CONTENTS.....	viii
LIST OF TABLES.....	xi
LIST OF FIGURES.....	xiv
LIST OF SYMBOLS AND ABBREVIATIONS.....	xvi
CHAPTER 1.....	1
1. INTRODUCTION.....	1
CHAPTER 2.....	4
2. BACKGROUND AND LITERATURE REVIEW.....	4
2.1. CANCER.....	4
2.2. GENE EXPRESSION.....	6
2.3. MICROARRAY AND RNA-Seq TECHNOLOGY.....	8
2.3.1. Microarray.....	8
2.3.1.1. DNA Microarrays.....	8
2.3.1.1.1. Oligonucleotide Arrays.....	8
2.3.1.1.2. cDNA arrays.....	9
2.3.1.2. Protein Microarray.....	9
2.3.2. RNA-Seq Technology.....	10
2.3.3. Microarray vs RNA-Seq.....	12
2.4. LITERATURE REVIEW.....	12

CHAPTER 3	16
3. METHODS AND MATERIALS	16
3.1. FEATURE SELECTION METHODS.....	16
3.1.1. Filter Methods	16
3.1.1.1. Correlation Based Feature Selection(CFS).....	17
3.1.1.2. Fast Correlation Based Feature Selection (FCBF).....	17
3.1.1.3. Relief-Based Feature Selection	18
3.1.1.4. Markov Blanket Filter (MBF).....	18
3.1.2. Wrapper Methods.....	18
3.1.2.1. Hill-Climbing Search	19
3.1.2.2. Best-First Search	20
3.1.2.3. Probabilistic Search.....	20
3.1.3. Embedded Methods.....	21
3.1.3.1. SVM-RFE (Recursive Feature Elimination for Support Vector Machines).....	21
3.1.3.2. FS-P (Feature Selection - Perceptron).....	21
3.2. CLASSIFICATION ALGORITHMS	22
3.2.1. Naive-Bayes Classifier	22
3.2.2. K-Nearest Neighbor	23
3.2.3. Decision Trees.....	24
3.2.4. Random Forest	25
3.2.5. Support Vector Machines.....	27
3.2.6. Artificial Neural Networks.....	28
3.2.7. Deep Learning.....	30
3.2.7.1. Development of Deep Learning.....	30

3.2.7.2. Deep Learning Models.....	31
3.2.7.2.1. Autoencoder	31
3.2.7.2.2. Deep Belief Network.....	32
3.2.7.2.3. Convolutional Neural Network.....	33
3.2.7.2.4. Recurrent Neural Network	34
3.2.7.3. Deep Learning Frameworks.....	35
3.3. Evaluation Methods of Classification Results	35
3.3.1. Mean Absolute Error (MAE)	35
3.3.2. Root mean squared error (RMSE).....	36
3.3.3. Confusion Matrix	36
3.3.3.1. Recall.....	37
3.3.3.2. Precision.....	37
3.3.3.3. F-Measure	37
CHAPTER 4	39
4. RESULTS AND DISCUSSIONS	39
4.1. EXPERIMENTAL DATASETS.....	39
4.2. GENE SELECTION	40
4.3. EXPERIMENTAL RESULTS FOR LUNG CANCER DATASET	41
4.4. EXPERIMENTAL RESULTS FOR RENAL CELL CANCER DATASET ..	64
CHAPTER 5	87
5. CONCLUSION.....	89
REFERENCES.....	91

LIST OF TABLES

Table 3.1: Confusion matrix table.....	36
Table 4.1: Description of Datasets.	40
Table 4.2: Selected Genes for Renal Cell Cancer.	40
Table 4.3: Selected Genes for Lung Cancer.....	41
Table 4.4: Confusion matrix of decision tree classifier.	42
Table 4.5: Classification report of decision tree classifier.....	42
Table 4.6: Confusion matrix of random forest classifier.	43
Table 4.7: Classification report of random forest classifier.....	43
Table 4.8: Confusion matrix of SVM with linear kernel type.	44
Table 4.9: Classification report of SVM with linear kernel type.....	44
Table 4.10: Confusion matrix of SVM with polynomial kernel type.	44
Table 4.11: Classification report of SVM with polynomial kernel type.....	45
Table 4.12: Confusion Matrix of SVM with RBF Kernel Type.	45
Table 4.13: Classification report of SVM with RBF kernel type.....	46
Table 4.14: Confusion matrix of artificial neural network.....	46
Table 4.15: Classification report of artificial neural network.	47
Table 4.16: Comparison of Classification Algorithms.	47
Table 4.17: Confusion matrix of SGD Optimizer.....	49
Table 4.18: Classification report of SGD Optimizer.	50
Table 4.19: Confusion matrix of RMSProp Optimizer.....	51
Table 4.20: Classification report of RMSProp Optimizer.....	52
Table 4.21: Confusion matrix of Adagrad Optimizer.	53
Table 4.22: Classification report of Adagrad Optimizer.....	54
Table 4.23: Confusion matrix of Adadelta Optimizer.....	56
Table 4.24: Classification report of Adadelta Optimizer.	56
Table 4.25: Confusion matrix of Adam Optimizer.	58
Table 4.26: Classification report of Adam Optimizer.....	58
Table 4.27: Confusion matrix of Adamax Optimizer.	60
Table 4.28: Classification report of Adamax Optimizer.....	60

Table 4.29: Confusion matrix of Nadam Optimizer.	62
Table 4.30: Classification report of Nadam Optimizer.	62
Table 4.31: Comparison of results with different optimizers.....	63
Table 4.32: Comparison of results for Lung Cancer dataset.....	63
Table 4.33: Confusion Matrix of Decision Tree Classifier.	64
Table 4.34: Classification Report of Decision Tree Classifier.	65
Table 4.35: Confusion Matrix of Random Forest Classifier.....	65
Table 4.36: Classification Report of Random Forest Classifier.....	66
Table 4.37: Confusion Matrix of SVM with Linear Kernel Type.	66
Table 4.38: Classification Report of SVM with Linear Kernel Type.	67
Table 4.39: Confusion Matrix of SVM with Polynomial Kernel Type.	67
Table 4.40: Classification Report of SVM with Polynomial Kernel Type.	68
Table 4.41: Confusion Matrix of SVM with RBF Kernel Type.	68
Table 4.42: Classification Report of SVM with RBF Kernel Type.....	69
Table 4.43: Confusion Matrix of Artificial Neural Network.	69
Table 4.44: Classification Report of Artificial Neural Network.	70
Table 4.45: Comparison of Classification Algorithms for RCC.....	70
Table 4.46: Confusion Matrix of SGD Optimizer.	72
Table 4.47: Classification Report of SGD Optimizer.	73
Table 4.48: Confusion Matrix of RMSProp Optimizer.	74
Table 4.49: Classification Report of RMSProp Optimizer.	75
Table 4.50: Confusion Matrix of Adagrad Optimizer.....	76
Table 4.51: Classification Report of Adagrad Optimizer.	77
Table 4.52: Confusion Matrix of Adadelta Optimizer.	78
Table 4.53: Classification Report of Adadelta Optimizer.....	79
Table 4.54: Confusion Matrix of Adam Optimizer.....	80
Table 4.55: Classification Report of Adam Optimizer.	81
Table 4.56: Confusion Matrix of Adamax Optimizer.....	82
Table 4.57: Classification Report of Adamax Optimizer.....	83
Table 4.58: Confusion Matrix of Nadam Optimizer.	84

Table 4.59: Classification Report of Nadam Optimizer.....	85
Table 4.60: Comparison of results with different optimizers for RCC.....	86
Table 4.61: Comparison of results for RCC.....	86
Table 4.62: Comparison with average values.	87
Table 4.63: All Results.....	88



LIST OF FIGURES

Figure 3.1: Filter method algorithm.....	16
Figure 3.2: Wrapper method algorithm.....	18
Figure 3.3: Embedded method algorithm.	21
Figure 3.4: A decision tree for mammal classification problem.....	25
Figure 3.5: Random Forest Classification Simplified.....	26
Figure 3.6: Optimal hyperplane.	27
Figure 3.7: Structure of artificial neural networks.....	29
Figure 3.8: Neural networks vs Deep learning architecture.....	30
Figure 3.9: Basic structure of autoencoder.	31
Figure 3.10: Architecture of a deep belief network (DBN).....	32
Figure 3.11: The basic structure of CNN on digit classification.	33
Figure 3.12: The basic structure of RNN.....	34
Figure 4.1: Model loss during training and testing for SGD optimizer.....	48
Figure 4.2: Model accuracy during training and testing for SGD optimizer.....	49
Figure 4.3: Model loss during training and testing for RMSProp optimizer.....	50
Figure 4.4: Model accuracy during training and testing for RMSProp optimizer....	51
Figure 4.5: Model loss during training and testing for Adagrad optimizer.	53
Figure 4.6: Model accuracy during training and testing for Adagrad optimizer.	53
Figure 4.7: Model loss during training and testing for Adadelat optimizer.....	55
Figure 4.8: Model accuracy during training and testing for Adadelat optimizer.....	55
Figure 4.9: Model loss during training and testing for Adam optimizer.	57
Figure 4.10: Model accuracy during training and testing for Adam optimizer.	57
Figure 4.11: Model loss during training and testing for Adamax optimizer.....	59
Figure 4.12: Model accuracy during training and testing for Adamax optimizer.....	59
Figure 4.13: Model loss during training and testing for Nadam optimizer.....	61
Figure 4. 14: Model accuracy during training and testing for Nadam optimizer.....	61
Figure 4.15: Model loss during training and testing for SGD optimizer.	71
Figure 4.16: Model accuracy during training and testing for SGD optimizer.	72
Figure 4.17: Model loss during training and testing for RMSProp optimizer.	73

Figure 4.18: Model accuracy during training and testing for RMSProp optimizer. .	74
Figure 4.19: Model loss during training and testing for Adagrad optimizer.	75
Figure 4.20: Model accuracy during training and testing for Adagrad optimizer. ...	76
Figure 4.21: Model loss during training and testing for Adadelta optimizer.....	77
Figure 4.22: Model accuracy during training and testing for Adadelta optimizer....	78
Figure 4.23: Model loss during training and testing for Adam optimizer.	79
Figure 4.24: Model accuracy during training and testing for Adam optimizer.	80
Figure 4.25: Model loss during training and testing for Adamax optimizer.....	81
Figure 4.26: Model accuracy during training and testing for Adamax optimizer.....	82
Figure 4.27: Model loss during training and testing for Nadam optimizer.....	83
Figure 4.28: Model accuracy during training and testing for Nadam optimizer.....	84

LIST OF SYMBOLS AND ABBREVIATIONS

ANN	Artificial neural network
CFS	Correlation based feature selection
CNN	Convolutional neural network
DT	Decision trees
FCBF	Fast correlation based feature selection
KICH	Kidney chromophobe carcinomas
KIRC	Kidney renal clear cell
KIRP	Kidney renal papillary cell
KNN	K-nearest neighbor
LUAD	Lung adenocarcinoma
LUSC	Lung squamous cell with carcinoma
MAE	Mean absolute error
MBF	Markov blanket filter
NGS	Next generation sequencing
RCC	Renal cell cancer
RF	Random forest
RMSE	Root mean squared error
SVM	Support vector machines
TCGA	The cancer genome atlas

CHAPTER 1

1. INTRODUCTION

Cancer is a disease caused by the uncontrolled growth of abnormal cells in a part of the body. Any part of the body can be affected by cancer. There are more than 100 different kinds of cancer. Liver, stomach, prostate, colorectal and lung cancer are the most known kinds of cancer for men. The most diagnosed cancer types among women are lung, cervix, breast, thyroid and colorectal cancers. Cancer is the most important reason of death before age 70 in 91 of 172 countries with respect to estimates of the World Health Organization (WHO) in 2015. According to estimation, there can be 18,1 million new cancer patients and 9,6 million deaths caused by cancer in 2018 (Bray et al., 2018). These informations show the importance of early diagnosis. Because early diagnosis is the most important issue for millions of people to get rid of cancer.

Cancer is primarily a genetic disease. Generally, it starts with a series of mutations on a single cell that becomes an abnormal cell. Then the abnormal cell divides uncontrollably and can spread throughout the tissue, organ or body. Gene mutations related with cancer can be inherited from parents or they can be occurred through somatic mutations. Diagnosis and classification of cancer by gene expression has great importance at this point. It can be understood that the type of tumor which is taken from a tissue. If it is malignant, then the subtype of this tumor and finally the stage of the cancer can be detected. These processes are difficult and time-consuming.

Genes are small sections of DNA that contains all the information about a cell. All the genetic information is stored by this DNA for using to create every protein which is the cell needs. A particular set of instructions codes for a particular protein is present in each gene. Gene expression is the process that contains the necessary information for the formation of a gene. Shortly we can say gene expression shows the activation status of a gene during making a protein. Microarray is a tool which used in laboratory for detecting the expression of many of the genes at same moment. To make a microarray analysis, mRNA molecules are obtained from two different people.

First reference could be a healthy reference and other one is an individual who has disease like cancer. The data which is gathered from microarrays can be used for diagnosis and classification of human cancer (Perez-Diez et al., 2007). Microarray technology gave its place to RNA-Seq technology because of a few main advantages and RNA-Seq technology become the major principle in gene-expression researches (Ritchie et al., 2015).

Support Vector Machines (SVM) and RNA-Seq data used in identifying and analyzing different cancer subtypes by Zhang et al. (2017). The high dimension and number of features affected the success and time of diagnosis.

RNA-Seq data has a big dimensionality with many genes. During the diagnosis of cancer, most of the genes are not relevant. For example, in human genome there are nearly 25,000 coding genes and 291 of them observed that caused to cancer (Futreal et al., 2004). This study showed that the number of genes can be minimized for using the classification or diagnosis of cancer. If the genes which are irrelevant for cancer eliminated, performance and success of SVM would be increased. It can be very important development for cancer diagnosis. Because small sample size with high dimensionality is the main challenge of RNA-Seq data.

One of the basic concept in microarray is feature selection or RNA-Seq data. Genes are the features used to classify or diagnosis cancer in RNA-Seq datasets. Feature selection hugely impacts the performance of your classification or diagnosis algorithm. Feature Selection is the method where features which are the optimal genes for classification, prediction or diagnosis can be selected automatically or manually. Unrelated features in your data can reduce the accuracy of the models and cause learning based on irrelevant features of your model. The most important advantages of using feature selection are decreasing overfitting, decreasing training time and increasing accuracy of model. There are three main kinds of feature selection methods in machine learning. These are filter method, wrapper method and embedded methods. Feature selection placed in data preprocessing step before using data as input for classical or machine learning models.

Several types feature selection methods and classification algorithms have been studied for the diagnosis of cancer using with microarray data. Support Vector Machine, Bayesian, Decision Tree, Artificial Neural Networks, Random Forest and

Bayesian classification techniques have been used in recent studies. For example, diagnosis of cancer made using microarray data by A. Statnikov et al. (2005) and the best classification performance for leukemia data is achieved by using SVM with the accuracy of 97.5%. Deep learning is another method for cancer classification. Deep learning methods includes more stacked layers. It can increase the efficiency and accuracy.

In this study, machine learning classical algorithms and deep learning method are compared to find the best classification model for cancer classification. Renal cell cancer and lung cancer RNA-Seq datasets are selected for application of algorithms. Before the evaluation of algorithms feature selection method applied on them. Wrapper method is chosen for feature selection. The selected gene subsets is evaluated by using four different machine learning techniques, Decision Trees (DT), Support Vector Machines (SVM), Random Forest (RF), Artificial Neural Networks (ANN) and Deep Learning. SVM, RF, DT and ANN's performance on big data is proven by many studies and generally used for gene expression data analyzing. The Multilayer Perceptron classifier is one of the most widely used classifiers in deep learning techniques. The obtained results of algorithm comparisons given at the end of study. The results will provide better and clear understanding about the application of feature selection and classification algorithms when studying on RNA-Seq data.

The content of this thesis is organized as the followings: Chapter 2 gives comprehensive knowledge about the biological background of the cancer, microarray and rna-seq technology and gene expression datasets in addition, related works about classification algorithms and feature selection. In Chapter 3, all feature selection methods and reason of chosen method are explained then classical machine learning algorithms and deep learning method explained in detail. In Chapter 4, experimental results of two different RNA-Seq datasets with feature selection and for classification algorithms are applied and expressed in detail. Lastly, Chapter 5 presents final comments on the topic and future studies.

CHAPTER 2

2. BACKGROUND AND LITERATURE REVIEW

2.1. CANCER

Cancers are caused by mutation of genes related with growth control. The probability that mutations will cause cancer lies in the problem that DNA recovery systems do not work well and the number of mutations, and thus the emergence of cancer is more likely.

Tumor suppressor genes often are genes for DNA- repair proteins because they suppress tumor growth when minimum one replica of the gene isn't affected with a harmful mutation. Tumors develop much faster when both copies of gene are mutated. People who inherit one normal copy of gene and one copy with a defect in a single tumor-suppressor allele are more susceptible to developing the disease because it is enough that one normal gene of copy develop an error to further the cancer growth.

Some tumor-suppressor genes aren't specific to particular kinds of cancer. More than half of tumors are caused because gene for a protein named as p53 is mutated. This protein helps manage the destiny of damage cells.

That plays a primary role about sensing DNA damage in particular double-stranded ruptures. After perceiving harm, the protein either supports a DNA-fixing path or activates the apoptosis path causing to cell death.

Plenty mutation in the p53 gene are sporadic what can be expressed as they happen in somatic cells rather than to be inherited. Li-Fraumeni syndrome have human beings who inherit a detrimental mutation in a replica of the gene p53 and they are more likely to develop different cancer kinds than the population what this mutation does not occur.

Often two characteristics of cancer cells make them particularly undefended to spies that harm DNA molecules. Primary characteristic is that they divide often and second is that their DNA copy paths are more effective than they are in many cells. Cyclophosphamide and cisplatin are some of several agents widely used in cancer

chemotherapy, but they also act by damaging DNA. Normal cells are more able to avoid the effect of the induced damage than are cancer cells, providing a therapeutic window for specifically killing cancer cells (Jeremy et al., 2011).

In most of human cancers fatty acid synthase is overexpressed and its expression is correlated with tumor malignancy. The fatty acids are used as precursors for the synthesis of phospholipids, which are then incorporated into membranes in the rapidly growing cancer cells.

Inhibitors of fat acids synthase have tested by researches on mice to observe if the inhibitors leisurely tumor development. These inhibitors make really slow tumor development, in appearance by containing apoptosis. Other surprising observation has made and to the result; mice acted with inhibitors of the condensing enzyme observed exceptional weight loss because they consumed less food. So, fatty acid synthase inhibitors are nominees both as antitumor and as antiobesity drugs (Jeremy et al., 2011).

Obesity is an important reason for the endometrium cancer but a weak reason for postmenopausal breast cancer in adult life. Obesity also increases the risk of being cancers like kidney and colon.

Epstein-Barr virus is generating mononucleosis, but sometimes it happens carcinogenic. It is estimated to cause globally to almost half the cancers of the upper pharynx and some gastric cancers.

The human immunodeficiency virus (HIV) can be reason for lymphoma- kind of cancer described by an unusual proliferation of lymphoid tissue and also tissue cancer known as Kaposi's sarcoma (Trichopoulos et al., 1996).

Smoking cigarettes is the most major reason of lung cancer. For non-smoker, smoker has a 20 times increased risk of lung cancer. Some other risk factors contain exposure to asbestos, radon lines, and air pollution in professional and indoor environments, as well as increased age, genetic sensitivity, and perhaps low fruit, vegetable and micronutrient options (Kamangar et al., 2006).

Cancer incidence and types of cancer age, local environmental factors, diet, genetics and so on. It depends on many factors.

Elder patients can have more aggressive or lazier cancers than younger patients because of various tumor types and tumor-host mutual effect. Elder patients have a pretty higher ratio of tumors of secret histology, grade, hormone receptor status, and lymph node interest (Bouchardy et al., 2007).

Cancer is not simply a disease of mutant growth factors and cell cycle control proteins, it is also a metabolic disease. The realization that there is a metabolic components of cancer gives to cancer researchers new ideas about control of cancer (Jeremy et al., 2011).

There are not enough treatments for cancer. The aim of chemotherapy is to destroy rapidly growing abnormal cells with drugs. Hormone therapy includes using drugs that modify how specific hormones duty or eliminate the body's ability to produce them. Immunotherapy uses drugs and other treatment methods to strengthen the immune system and destroy cancer cells. To give an example of these treatments, the following two treatment methods can be said; checkpoint inhibitors and adoptive cell transfer. Radiation therapy is using high-dose radiation to destroy cancer cells. For humans with blood-interested cancers, such as leukemia or lymphoma stem cell transplant may be helpful. Another part of the treatment in patients with cancer tumors is surgical intervention. It can also destroy lymph nodes to decrease or protect the disease's outspread in another tissues of the body (Rachael, 2018).

2.2. GENE EXPRESSION

Proteins are encoded by genes and proteins manage the cell function. Hence, each particular cell's duties are determined by the thousands of genes expressed. Besides, each progression of data transferring from DNA to RNA gives a possible checkpoint to the cell for regulation of its capacities by altering the sum and kind of proteins it generates. The measure of a specific protein in a cell shows the stability among that protein's synthetic and degradative biochemical paths at any given time. Creation of a protein starts with transcription which is DNA to RNA and continue with translation which is RNA to protein on the synthetic part of this stability. In this manner, control of these procedures has a primary job in figuring out which proteins are available in a cell and how much. Moreover, the manner by what a cell forms its RNA transcripts and recently generated proteins additionally incredibly impacts protein levels.

The amount and type of mRNA molecules present in a cell gives the capacity of that cell. Truth be told, in every second many transcripts are created in each cell. According to measurement, it is not surprising that the essential checkpoint for gene expression is generally at the absolute starting point of the protein generation procedure (the beginning of transcription). An effective checkpoint generated by RNA transcription because numerous proteins may be produced using one mRNA particle.

Prepare a transcript gives an additional rule for eukaryotes and it can be thinking the existence of a nucleus. A transcript translation begins while the transcript process is in progress because ribosomes' closeness to the new mRNA particle for prokaryotes. For eukaryotes, transcripts are altered in the nucleus after, they send to cytoplasm for translation, in any case.

Furthermore, eukaryotic transcripts are more complex than prokaryotic transcripts. For example, the necessary transcripts combined by RNA polymerase include series and they are not a piece of the old RNA. These series called as introns and they are removing the old mRNA that break up the nucleus, before others. The rest parts of the transcript called as exons and they have protein-coding parts. Exons create the old mRNA and they grafted as a group. In addition, eukaryotic transcripts are identified at the ends that affect their security and translation.

Obviously, there are numerous situations where cells have to react rapidly to exchange ecological situations. With that conditions, the regulatory checkpoint can proceed skillfully when transcription is done. For instance, early growth in many animals related to translational control because tiny transcription happens after fertilization throughout the first few cell divisions. Therefore, eggs include numerous maternally started mRNA transcripts as a ready hold for post-fertilization interpretation.

On the most important side of the equalization, cells may quickly modify their protein degrees thanks to the enzymatic breakdown of RNA transcripts and present protein atoms. These two activities bring about diminished measures of specific proteins. Regularly, this breakdown is connected to explicit occasions in the cell. A genuine case of how protein breakdown is connected to cell occasions given by the eukaryotic cell cycle. The cycle is isolated into a few stages, every one of which is defined by unmistakable cyclin proteins that go about as essential controllers for that phase. Before a cell may advance from one period of the cell cycle to the following, the

cycline describing that particular cycle period must be corrupted. A cycline's inability to corrupt prevents the cycle from going on.

2.3. MICROARRAY AND RNA-Seq TECHNOLOGY

2.3.1. Microarray

Microarray technology is that lets quantitative, synchronous displaying and expression of many genes (Afshari, 2002). The basic complementarity is the essential rule behind microarray. 'A' matches some 'T', while 'C' connects with 'G'. In a microarray, a huge number of spots, which has countless DNA parts from a specific gene, are placed in a rectangular framework. When the example of relevance includes numerous duplicates of mRNA, numerous bindings are happened, demonstrating that the transcribed mRNA gene is extremely described. The amount of hybridization may be resolved because in the experiment every mRNA copy is named with a fluorescent or radioactive label and a more brilliant signal is recognized when more duplicates bind (Kuo et al., 2003). Generally, every microarray analysis includes five separate experimental phases – biological query, biochemical reaction, pattern preparation, finding, data visualization and modeling (Schena et al., 1998). According to Leung and Cavalieri (2003), a characteristic microarray experiment require fluorescent naming, pattern extraction, cohybridization, scanning and in the end statistical analysis. Techniques of microarrays primarily can be divided into microarrays of DNA and microarrays of proteins.

2.3.1.1. DNA Microarrays

DNA microarrays supply a fundamental and indigenous tool for systematically and comprehensively discovering the genome. As experimental vehicles, the strength and intelligibility of DNA microarrays are reproduced from the elegant originality and proximity of the relevant base-pairing (Brown and Botstein, 1999). DNA microarrays are sequences of oligonucleotides and various cDNA strains.

2.3.1.1.1. Oligonucleotide Arrays

Oligonucleotide arrays also called as DNA chips. They are very small parallel analytical gadgets including oligonucleotides libraries. They automatically detected or fixed on solid supports as characterized by their own area of identification of each oligonucleotide (Tillib and Mirzabekov, 2001). Oligonucleotide arrays include small DNA parts. Gene Chips produced by firms such as Affymetrix are one of the

industrially accessible oligonucleotide microarrays. Every array includes ten to hundred distinct oligonucleotide tests. Usually, Affymetrix GeneChip Arrays are produced in a fixed glass cartridge as a substratum as a single array caged.

2.3.1.1.2. cDNA arrays

cDNA arrays are made by automatically selecting single examples of filtered cDNA clones onto a solid support. It includes some long DNA parts. There are some fundamental standards behind cDNA arrays preparation contains:

- 1) Selecting the objectives for printed on the array directly from some databases like GenBank, UniGene, dbEST or from any related library randomly.
- 2) Arraying the chosen cDNA objectives on the given area of closed with glass microscope slide from inside a PC-controlled quick robot.
- 3) Naming the absolute RNA fluorescently from test and reference examples using reverse transcription dyes with one round.
- 4) Pooling the fluorescent target for hybridization under strict situations.
- 5) The measurement of the laser excited associated targets using a scanning confocal laser microscope.
- 6) Finally, images which are from scanner checking by bringing into a computer program in which they are pseudo-colored and combined.

There are some techniques for arraying cDNA such as piezoelectric printing, micro spotting and photolithography. There are some companies produce arrays for commercial purpose. For example, Affymetrix, Silicon genetics, Genome systems, Xenon, Biodiscovery, Invitrogen, Genetix, etc.

2.3.1.2. Protein Microarray

Despite the fact that it has a few limitations, DNA microarray is exceptionally useful. The numerous genes expression levels are dependent to important post-transcriptional regulation and post-translational change like acetylation, glycosylation, proteolysis, phosphorylation etc. unpleasantly affect the numerous proteins. Clearly, a nucleic acid-based array is not sufficient to see these effects and for particular applications, DNA microarrays dreary example preparation prerequisites make them unpractical (Kodadek, 2001). The solution is to break up protein rather than to make inferences directly possible by means of protein microarrays dependent on RNA levels. The

microarrays of protein otherwise known as protein chips are nothing more than grids with low levels of purified high-density proteins.

Protein microarrays may be mainly classified in three categories (Hall et al., 2007): analytical, reverse phase and functional microarrays.

The first microarray form is analytical microarrays, where microprints or micro structuring processes are used to immobilize biomolecular recognition molecules in a heterogeneous matrix. The most popular ones are hapten and antibody.

Functional protein microarrays are second form of microarray consisting a set of autonomous proteins or protein domains that are full-length. These spotted arrays are used to evaluate local protein feature or binding characteristics. Functional protein microarrays used efficiently to discover or depict the enzyme behaviours of enzymes in recognition of proteins and protein relationships, protein and small molecule relationships, cell and DNA relationships.

Reverse phase protein microarray is the other microarray form. These RPAs include the immobilization rather than a single test of all proteins found in individual cell or tissue populations different than functional and antibody arrays. Some of the present RPA apps involve a pathological information of some complicated disorders like cancer, stroke and traumatic brain injury (Gyorgy et al., 2010), new anti-invasion processes and anti-metastatic operations such as 3-(50-hydroxymethyl-20-furyl)-1-benzylindazole (Hong et al., 2010), quantitative study of down-regulated DNA, disease development and biomarker searching, are distinguished.

2.3.2. RNA-Seq Technology

The RNA-seq is named as RNA-sequencing. Next-Generation Sequencing (NGS) is used in RNA-Seq for looking at the RNA amount and sections (Wang et al., 2009). It analyzes the gene expression transcriptome stored in our RNA.

RNA-seq enables the transcriptome, the entire RNAs cells, including mRNA, rRNA and tRNA, to be researched and found. To link understanding of genomes with its functional protein expression is crucial to understanding the transcriptome. RNA-seq can demonstrate us which genes are enabled, their expression level, and when they are activated or shut off in the cell (Ozsolak and Milos, 2011). This allows researchers to gain a deeper understanding of cell genetics and evaluate changes that may indicate disease. The most common methods used by RNA-seq are differential gene expression

analysis, RNA processing, transcriptional profiling and SNP detection (Han et al., 2015).

Transcriptome can be used for showing all the texture for a gene. This gene can be expression of an unknown feature. That can show the importance of it and this can provide scientists with essential information on gene feature. It also collects some data about alternative splicing events. These events generate different transcripts from a single gene sequence. DNA sequences will not collect these events. It may also recognize post-transcriptional modifications that happen during mRNA processing, like polyadenylation and 5' capping (Ozsolak and Milos, 2011).

Previous RNA-Seq techniques used Sanger technology for sequencing. That technology was novel in that years, but it was also expensive, incorrect and low-output method. Only in the last times, we have been able to take benefit's all RNA-seq's opportunities with usage of adverted NGS technology (Schuster, 2008).

Turning the RNA population to cDNA fragments is the first stage in the method. cDNA fragments are a cDNA library. Therefore, the RNA can be inserted into a workflow of NGS. At each end of the fragment's adjusters are then added. These adapters comprise protective components, such as the amplification component and the main sequence location, which allow processing. NGS analyzes then the cDNA library, produces short sequences that either one or both sides of the fragment conform to each other. The size of the sequence of the library will be differs according to the methods to which the output data are used. Sequencing usually takes place using single-read or paired-end sequencing techniques. Single-read sequencing is a low cost and quicker method (for a comparison, around 1% of the Sanger sequencing cost). Sequences the cDNA from just one end, while paired-end techniques sequence from both ends, and are therefore more costly and takes more time (Genome Sequencing, 2019; Advantages of paired-end and single-read sequencing, 2019).

A selection between protocols for each strand-specific and non-strand-specific must be done. The past technique is the data about which DNA strand has been adapt is kept. The importance of additional information acquired from strand-specific protocols make them an advantageous choice.

These reads, from which millions will be produced at the end of the workflow, can then be matched to a reference genome and edited to create an RNA sequence map that spans the transcriptome (Zhao et al., 2015).

2.3.3. Microarray vs RNA-Seq

RNA-seq is commonly considered superior to other technologies like hybridization of the microarray. There are several explanations why the RNA-seq well-respected status:

- Capability to identify new transcripts: Unlike arrays, the RNA-Seq technology requires no specific samples for organisms or transcript. It detects singular nucleotide variants, gene fusions, new transcripts, indels (small insertions and deletions) and other unknown alterations that arrays can not detect (Wilhelm and Landry, 2009).
- Greater dynamic range: The estimation of gene expression is restricted by low-end backdrop and high-end sample saturation by array-hybridization technology. RNA-Seq technology creates distinct, digital sequencing read counts, and can quantify statement across a wider dynamic range ($>10^5$ for RNA-Seq vs. 10^3 for arrays) (Wilhelm and Landry, 2009).
- Advanced specificity and sensitivity: RNA-Seq technology can identify a greater percentage of differentially expressed genes, especially genes with low expression, compared to microarrays (Wang et al., 2014; Liu et al., 2015).
- Rare and low-abundance transcripts can be detected simply by improving the spread range of the sequence to identify rarity, single transcripts per cell or poorly-expressed genes.

2.4. LITERATURE REVIEW

With technological advances, studies in cancer field have increased. Microarrays created in laboratory environments have an important role in cancer diagnosis. Several methods have been applied to diagnose cancer using microarray datasets, which are composed of multiple gene expressions. These studies started with statistical methods and included artificial intelligence methods and deep learning in time. Today, the use of RNA-Seq data along with microarrays has started to increase with technological advances. Since the number of studies performed with RNA-Seq is low, the studies carried out with microarrays are emphasized in the literature section of this study.

Tran et al. (2011) have focused their work on microRNA (miRNA) data. Using the microarray data set used by Gloub et al. (2005), they classified samples as tumors and normal cells. It has 151 miRNA properties for each sample in the dataset containing 223 samples. Using the SVM as the classification algorithm, Tran and his colleagues applied 3 different kernel types on the dataset. As a result of the tests performed with RBF, an accuracy rate of 92.00%, 95.00% with Linear test and 93.00% with Polynomial tests were obtained.

Another study was conducted by Statnikov et al. (2005). In this study, 11 different cancer microarray datasets containing binary or multiple classification were determined. And support vector machines, k-nearest neighbors, backpropagation neural networks and probabilistic neural networks classification methods were applied on all datasets. Because of the high dimension of the dataset, gene number was reduced by 3 different feature selection methods before classification methods were applied. In addition, it was observed whether the number of genes had an effect on the classification by using different gene numbers each time. At the end of the tests, SVM applied to leukemia data with an accuracy rate of 97.5% was the most successful.

In another study, Huang et al. (2010) applied three different classification algorithms to four different cancer microarray datasets. Decision trees, SVM and KNN were used as the classification algorithm, while hepatatox, colon cancer, lymph cancer and leukaemia were used as the dataset. At the end of the tests, decision trees applied to leukemia data with an accuracy rate of 96.6% was the most successful.

Furey et al. (2000) applied the SVM classification algorithm on ovarian cancer dataset. The algorithm has also been applied to previous datasets. The highest accuracy rate was obtained from leukemia data with 91%. Peng (2006) studied prostate cancer data and achieved an accuracy of 95.1% with the SVM algorithm.

In another study, Han et al. (2011) analyzed leukemia, brain tumor, prostate and colon cancer data with four different classification algorithms. Gene selections were made in datasets before classification algorithms were applied. Accordingly, subsets of five, ten, twenty, fifty and one hundred genes were created for the datasets. Later, algorithms were applied and most Naive Bayes algorithm achieved 91.1% accuracy rate on colon cancer.

Chen et al. (2011) and colleagues used the SVM classification algorithm on 999 samples from the Wisconsin Diagnostic Breast Cancer (WDBC) cell database. The dataset was preprocessed before the algorithm was applied. Training and test data are divided into three different data sets: 50-50%, 70-30% and 80-20%, respectively. SVM was then applied and an accuracy of 96.00% was obtained in the dataset divided into 80-20%.

In another study, Lee et al. (2005) applied several different classification algorithms to seven different cancer microarray datasets. Data sets containing binary and multiple classifications were pre-processed before classification. Filtering was performed to reduce the dataset properties and the genes that had no effect on the classification were screened. Then classification algorithms were applied and SVM gave the best results with 94% accuracy.

Many studies have been conducted for microarray datasets and many algorithms have been tested. However, there are not many studies of RNA-seq datasets with these algorithms. A poisson linear discriminant analysis (PLDA) classifier was suggested by Witten (2011). This classifier is a developed from Fisher's linear discriminant analysis which is improved for high-dimensional count data sets. PLDA reduce the differences of class to recognize a gene subset and a Poisson log linear model is implemented for data classification. This algorithm was improved by Dong et al. (2015) for generating another classification technique dependent to the negative binomial (NB) distribution.

Researchers presented different methods for making high dimensional RNA-Seq data continuous and less complex, such as microarray data. Law et al. (2014) recently proposed variance modeling at the observational level (voom) method for change of RNA-Seq. The voom method uses the log counts for finding relation between mean and variance then gives precision weights for downstream analysis. Voom method was then combined with the limma (Ritchie et al., 2015) method. After this combination, the new method gave the better performance when compared with counting methods.

Zararsiz et al. (2017) applied seventeen different classifier algorithms to four different RNA-Seq datasets. Among the classification algorithms applied to the cervical, Alzheimer's, renal cell cancer (RCC) and Lung cancer RNA-Seq data sets, SVM and RF gave the best results. Since RCC and Lung cancer were used in this thesis, these

two datasets were chosen as reference. At the end of SVM tests, an accuracy rate of 93.5% was obtained for RCC, and 94.8% for Lung cancer.



CHAPTER 3

3. METHODS AND MATERIALS

3.1. FEATURE SELECTION METHODS

A feature is a process being observed that have individual measurable property (Girish and Ferat, 2014). Large amount of data which has many features have generated such as video, photo, text and voice because of the greatly development of technology. The numbers of features and variables are used by machine learning applications have enlarged from tens to hundreds in the past decades and machine learning algorithms use a features set for making classification. The most important problem in classification difficulty is huge count of irrelevant features or variables. Feature selection can be expressed as a process of selecting the most useful features to use in machine learning algorithms. Useless and irrelevant features increase the time-consuming of training algorithm and affect the result of algorithm. Feature Selection helps having meaningful features, making easy to apply machine learning algorithms on relevant features, decreasing overfitting, removing data repetition, decreasing training time of machine learning models and increasing classification performance of model (Guyon and Elisseeff, 2003). There are some feature selection methods to find most suitable features for a machine learning algorithm. These methods can be filtered in three groups: wrapper method, embedded methods, filter methods. In this chapter, these three methods are explained and compared for finding most suitable subset which affects the success of classification and estimation achievement.

3.1.1. Filter Methods

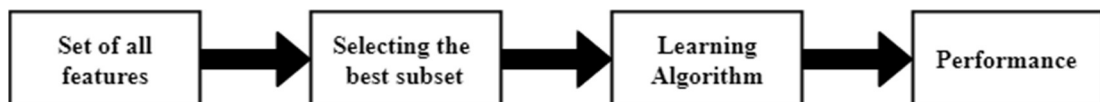


Figure 3.1: Filter method algorithm.

Filter methods use techniques for rating features as an essential criteria for ordering feature selection. Through checking the relational features of the data, filter methods assess the features' importance. The features are scored using an appropriate ranking

principle and a threshold is used to discard features less than threshold. Then, this feature subset is given to classification algorithm as input data. Filter methods use the measures like distance, information, correlation and consistency (Sasikala et al., 2014) to determine the relation of features. One of the greatest importance of filter methods is to choose features regardless of the machine learning model. Features selected using filter methods can be given to any model of machine learning as input data. Another strength of filter methods is being very fast (Yvan et al., 2007). The filter methods' disadvantage is that lack communication with the system of machine learning, generating general outcomes and poor classification achievements (Hall and Smith, 1998). It is possible to categorize filter methods as univariate and multivariate. Feature dependencies are ignored by univariate filter methods. These dependencies may be induced when compared with other feature selection methods by selecting irrelevant features and worst classification results. Moreover, the system of multivariate filter methods includes dependency on the classification model independently. In addition, we also quantify the dependency between each feature pair by evaluating category significance as univariate. Some of the commonly used filter methods are represented in the following sub-sections:

3.1.1.1. Correlation Based Feature Selection (CFS)

CFS method gives the optimal feature subsets by choosing features that are extremely correlated and uncorrelated for classification. Irrelevant features are not correlated with the class because of that they are ignored. Unnecessary features should be excluded as they are extremely correlated with rest of one or more features. In order to select a feature, the correlation amount and the estimation capacity are taken into consideration.

The formula of the CFS:

$$CFS = \frac{k\bar{r}_{cf}}{\sqrt{k + k(k - 1)\bar{r}_{ff}}}$$

CFS is the ranking score, \bar{r}_{cf} shows feature-class correlation, \bar{r}_{ff} is the average correlation among features and k is the feature numbers of a feature subset.

3.1.1.2. Fast Correlation Based Feature Selection (FCBF)

FCBF is a feature selection process which begins to selection with all features and then symmetrical uncertainty is used for measuring the feature dependencies. The best

subset is found by using sequential search strategy backward as selection method. Inside the algorithm, there are ending conditions when there are no more features to add. Normally, FCBF is faster than other forms of correlation-based selection (Senliol et al., 2008).

3.1.1.3. Relief-Based Feature Selection

Relief is a feature selection method for filtering features which calculates a value for each feature to use a requirement for the estimation function or a reference to the target approach. These values are considered as feature weights ($W[X]$ = weight of feature 'X'), or score of feature between -1(worst) to +1(best) range. This shows us that Relief algorithm can be used in limited classification problems like binary and there is no chance for handling with missing data (Urbanowicz et al., 2017). Relief algorithm divides samples into the classes and calculate near-hit and near-miss values. After that, feature's weight is updated as follow:

$$W_i = W_i - \text{diff}(x_i, \text{near} - \text{hit}_i)^2 + \text{diff}(x_i, \text{near} - \text{miss}_i)^2$$

Relief algorithm uses nearest neighbor method to find the near-hit instance which means the most similar sample in the indifferent class and near-miss instance which is the most similar sample in the different class.

3.1.1.4. Markov Blanket Filter (MBF)

MBF algorithm selects independent features from the class name thus, these removed features do not affect accuracy. There is no need a specific feature ranking in MBF, nor there is no need to limit the number of parents allowed for each node. Thanks to this property, MBF is both more common and more applicable for domain applications where the use of no prior experience to shorten the learning process.

3.1.2. Wrapper Methods

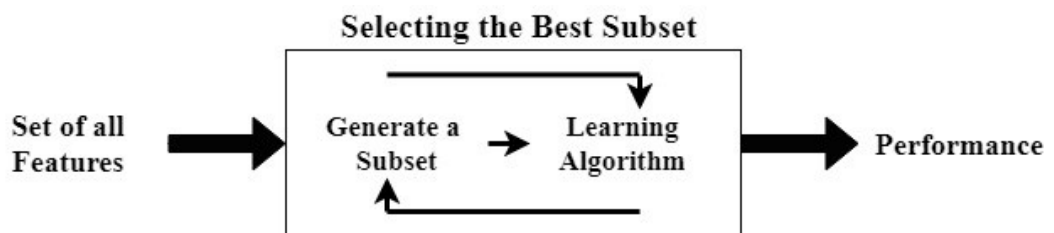


Figure 3.2: Wrapper method algorithm.

Another feature selection method is the wrapper method. It is used for creating last version of algorithm which will be used make a final classifier for feature subset selection. So, A is a classifier and S is a feature set, then wrapper method looks for in the subset domain of S and trained classifier A is tested on each subset. Then results are compared using by cross-validation method. Wrapper method is more computationally expensive than other feature selection techniques but it is better to have a good biases which is suitable for learning algorithm and it provides a better performance.

The wrapper method has an important problem about searching the domain of feature subsets. In microarray analysis, when searching the best subsets, enumeration on all possible property sets is generally very difficult in the high-dimensional problems. Other than, there is no an algorithm to perform this optimization traceable. Feature selection is often an NP-hard problem, but many studies have recently developed numerous heuristic scans to perform this research effectively. A comprehensive review article (Russell and Norvig, 1995) can be read for more information about search intuition.

It is possible to see the search when creating a search tree attached to the status field (in our situation, a specific feature subset is responded by each node for trees, and adjoining nodes communicate to two different subset of properties, with one property). This is the first set of features that are selected as root, empty or random. In every search stage, a leaf node is selected by the search algorithm in the tree to expand by applying an operator to the feature subset respond to the node for generating a child. For a better understanding, two different search methods are explained below.

3.1.2.1. Hill-Climbing Search

Hill-climbing search also named as greedy search or called as steepest ascent is one the simplest search methods. Actually, there is no even need to ensure a search tree for performing this search because all the algorithm makes the best modifications on the feature subset locally. In fact, the algorithm develops the last node and transmits it to the child with the best accuracy based on cross-validation. The process continues until no better results are obtained at the last node. Hill-climbing search has an important issue because of the existence of local maximum. This local maximum points are

plateaux and ridges of the value surface of the evaluation function. Simulated annealing is used to break that wrong sub-optimality.

3.1.2.2. Best-First Search

Best-First search method is more autonomous when compared to Hill-Climbing search method. Essentially, in best-first search, the the most valuable leaf that produced until now is selected. To do so, a record which is obtained from search tree must given with the tree boundary. There is no stop at the moment when values of node stop rising for discovering the state space more fully but continue extend the tree untill no improvement was obtained in the last k expansions.

3.1.2.3. Probabilistic Search

It is appropriate to focus on searching in search domain areas for major search issues that have yielded good results in the past but still allow improvement in discovery (as opposed to greedy techniques). For doing that, samples can be taken only from important distribution portion of feature combinations which was seen before. Define a random variable $z \in \{0, 1\}^n$: a string of n bits that shows whether every n features are relevant. For example, an incremental method, a parametric probabilistic model can be considered to learn the distribution of a dependency tree for a random variable z or even a more complex Bayesian network.

A dependency tree model can be shown as below:

$$p(z) = p(z_r) \prod_{i \neq r} p(z_i | z_{\pi_i})$$

Where z_r is the root node and π_i indexes the parent of node i . This tree is different than the search tree which talked about before where a node shows a feature subset and the size of the tree expands during search up to 2^n . Every node corresponds to and pointer random variable related to the inclusion or exclusion of the particular feature in a dependency tree, and the tree has a fixed size. Any specific composition of feature subset that can be selected is an example from the distribution stated by this tree. The Chow-Liu algorithm (Chow and Liu, 1968) can be used for finding the best tree model which is suitable for the data in a collection of previously tested feature subsets (meaning increasing the likelihood of previously tested samples). After that, a depth first tree-traversal can be applied in given the tree model that lets sampling the candidate property subgroups from a concentrated subdomain that is more likely to

contain better solutions than random search. More detailed explanation of this algorithm can be reached in (Baluja and Davies, 1997).

3.1.3. Embedded Methods

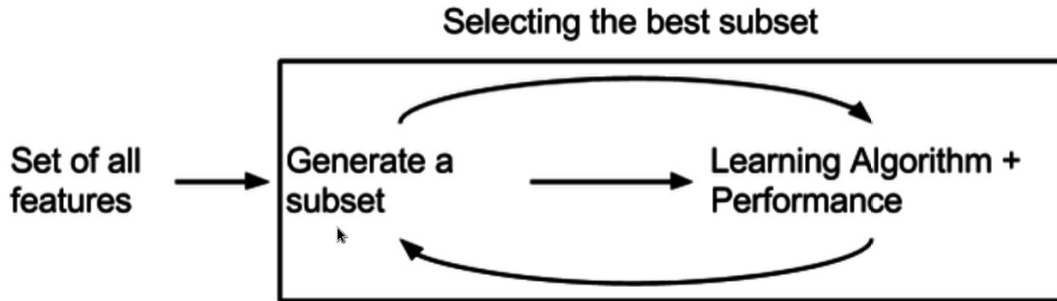


Figure 3.3: Embedded method algorithm.

When compared to other feature selection methods, embedded methods are very different because of different feature selection way and learning interaction. There is no learning in Filter methods. A machine learning is used by Wrapper methods to see the features subsets' quality with no need knowledge about the particular classification or regression function architecture because of that it can be used with any machine learning algorithm. It is the most important advantage of wrapper methods. Compared to filter and wrapper methods, the feature selection part not splitted from the learning in embedded techniques because the class of functions architecture has a critical role for feature selection. For example, Weston et al. (2000) have calculated the importance of a feature by limiting it to SVM only, so the developed method cannot be used for other classification methods that can only be used in SVM.

3.1.3.1. SVM-RFE (Recursive Feature Elimination for Support Vector Machines)

The SVM-RFE method, repetitively applies the feature selection with using SVM classifier that removes the irrelevant features by itself. There are two different test types of these methods: for the solution of more complicated problems a linear and nonlinear kernels are used (Rakotomamonjy, 2003).

3.1.3.2. FS-P (Feature Selection - Perceptron)

FS-P (Feature Selection-Perceptron) (Mejia-Lavalle et al., 2006) is an embedded selection technique based on a perceptron. A perceptron is a kind of an ANN considered a simple type of feedforward neural network: a linear classifier. The essential concept of this technique is using supervised learning for training of a

perceptron. Dependency weights can be used like pointers that have the most appropriate properties and give a ranking.

3.2. CLASSIFICATION ALGORITHMS

Classification is a method for splitting a data into various number of classes which are desired with assigned label to every class. Classification, which is used in machine learning and statistics, is a supervised learning method. Input datas are used by computer program for learning and then it uses that knowledge for classification of new data. The input data which is given to computer program can be a simple binary class (like recognizing whether the person is female or male) or it can be multi-class (like recognizing the color of car) too. Speech recognition, biometric identification and object detection are some applications of classification problems. Microarray analyzing is also an application area for classification. It is used to classify genes as healthy or diseased. In this thesis, some of the classification algorithms will be used to classify RNA-seq data for lung and kidney cancer detection. After that results of that algorithms will be compared with another developed method for this thesis. Some of the most known classification algorithms in machine learning explained in the following for better understood.

3.2.1. Naive-Bayes Classifier

The main idea at the behind of Bayesian classifiers is selection of the most similar class for a given sample specified by its feature vector. The process of learning in classifier can be shown with that formula, $P(X|C) = \prod_{i=1}^n P(X_i|C)$ where $X = (X_1, \dots, X_n)$ is a feature vector and class is shown with C. In spite of that unrealistic guess, naive Bayes which is called the resulting classifier is quite good with practical, generally in contest with many sophisticated techniques. The efficiency of Naive Bayes has been used for a lot of problems such as text classification, systems performance management and medical diagnosis (Domingos and Pazzani, 1997; Langley et al., 1992). The naive Bayes' success in the presence of property dependence is expressed as follows; there is no connection between classification error optimality and probability distribution quality. Rather, an optimal classifier is provided by combination of real and predicted distributions reach an agreement on the class which is the most possible.

3.2.2. K-Nearest Neighbor

First studies about the K-Nearest-Neighbor (kNN) was started at the beginning of 1950s. kNN can be applied to many different problems like pattern recognition, text mining, economy, medical and agriculture etc (Imandoust and Bolandraftar, 2013). kNN is a non-parametric algorithm. There is no need any previous information about dataset for kNN and it supposes that samples in the datasets are independently and equally shared, for this reason, the samples which are similar to each other have the similar classification (Syed, 2014). kNN is also called as lazy learner. It simply keeps all the given training data as input without making no changes or makes just small changes then waits until a test data for given to it for classification. All processing or calculations are applied at the same time of test data classification. New or unknown data is classified by comparing with similar training data. When there is an unknown data, a kNN classifier looks for the pattern domain in the “k” nearest-neighbors or nearest data to the unknown data. “Nearest” means a distance metric (Han and Kamber, 2006). There are different kind of distance metrics like Euclidean, Minkowski and Manhattan distance. For finding the “k” nearest training data to the unknown data one of these distance metrics are used. The general Euclidean distance $d(x, y)$ is usually used as the distance function (Hechenbichler and Schliep, 2004). Distance function for Euclidean distance is calculated as below between 2 data x and y .

$$d(x, y) = \sqrt{\sum_{i=1}^n (a_i(x) - a_i(y))^2}$$

When total number of features is shown with "n" and "a" shows the feature value in samples x (test data) and y .

The selection of “k” is very important for creating a kNN model. It is one of the most important component of kNN model which may strongly affect the accuracy of predictions. “k” is very important because a little value of it leads to a big change in estimations for any given problem. Another option is assigning “k” value as large can lead to a large model bias. Therefore, assigned “k” value could be enough big to decrease the probability of classification error and enough small so that the “k” nearest point is close enough to the query point. When “k”=1, the unknown data is set the class of the training data that is nearest to it in pattern domain. It is good to select “k” to be an odd number as this prevent connection between votes (Jiang et al., 2006). The value

of “k” shouldn’t be selected as a multiplier of class numbers. It prevents connections when class numbers are greater than two.

3.2.3. Decision Trees

Decision tree is a classification method which explained as an iterative part of the sample domain. The Decision Tree has nodes which creates a Rooted Tree. This tree is a Directed Tree with a root node which does not have incoming edges. For every other nodes, there is a completely one incoming edge. The name of these all other nodes are leaves (Rokach and Maimon, 2002). Some of following deficiencies are seen while studying the Decision Tree Classification.

- DT is not very useful for prediction problems where the purpose is to estimate the value of a continuous properties.
- DT has tendency to errors in classification problem with too much classes and comparatively with lots of classes and comparatively little number of training samples.
- The process of developing a DT is computational costly. For every node, every splitting area must be put in order before finding its best split. Field combinations are used and an investigation is needed for optimum combining weights in some algorithms.
- Many DT algorithms control only one area in a timeBecause of that situation there can be rectangular classification boxes. These boxes block the actual distribution of records in the decision field. By controlling these deficiencies, the decision tree needed to be transformed to a new architecture.

Each leaf node has a category tag in a decision tree. Leaf nodes containing root and other internal nodes provide test cases of properties of various records with different properties. The root nodes shown in Figure 3.4, as an example. To differentiate it from warm-blooded vertebrate animals, it uses the body temperature feature (Komal and Lalita, 2015).

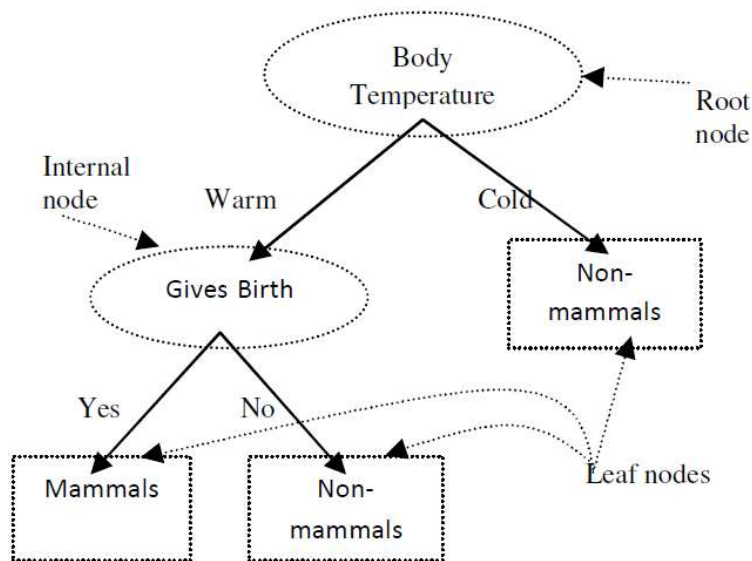


Figure 3.4: A decision tree for mammal classification problem.

Since not all cold-blooded vertebrates are mammals, the root node has produced a non-labeled leaf knot as the right child. A following function that can be giving birth, will be used when vertebrates are warm-blooded to differentiate mammals from other warm-blooded animals, often birds. Once a decision tree is formed, a test record can be easily identified. Beginning with the root node, the test situation will be applied to the record and then according to test result, suitable branch will be followed.

There are, in theory, many exponential decision trees which could be generated from a specific feature sets. Although few of the trees are more sensitive than other trees, it is not computationally possible to find the most suitable tree due to the search field's exponential size. Nonetheless, in order to stimulate a reasonably accurate decision tree within a reasonable period of time, successful algorithms have been developed. Generally speaking, these algorithms use a greedy strategy that grows a decision tree by making a number of locally optimal decisions about which function to use for data separation. Such an algorithm is the algorithm of Hunt, which is the basis of various available algorithms for decision tree induction, including ID3, C4.5 and CART.

3.2.4. Random Forest

The first studies about random forest has been presented in the University of California in 2001 by Breiman (2001). It is created from many other different and completely independent classifiers (decision tree). Given a test data as input to the new classifier

can be classified according to ranking of results from every different classification. Figure 3.5 shows that the whole process of Random Forest Classification.

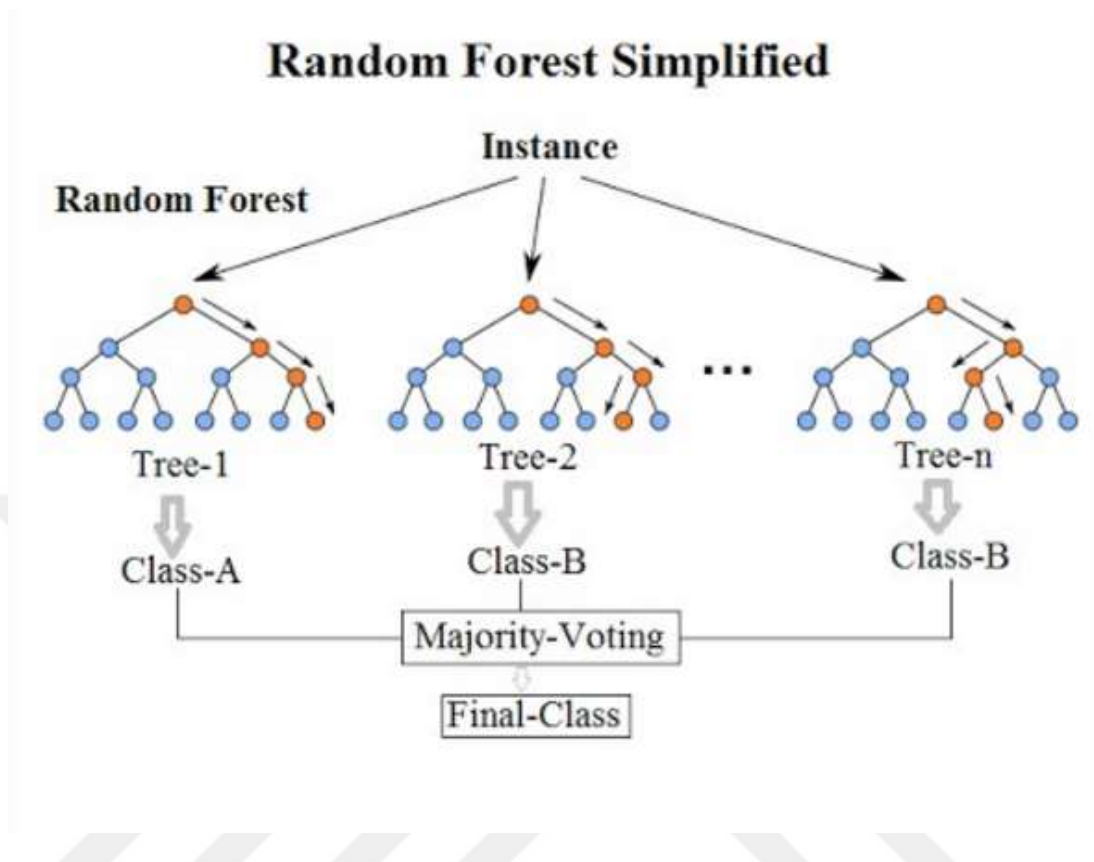


Figure 3.5: Random Forest Classification Simplified.

The next steps can be used for creating a random forest classifiers (Breiman, 2001):

- i. Give the variable named 'N' a proper value, which is the number of elements of each subset of features.
- ii. Depending on the value of N, select a new feature subset m from the whole feature set at random.
- iii. Train the data set with the feature sub-set to build a decision tree for each training set group.
- iv. Select a new m that is different from others and do the whole thing again until all the subsets of the features are used. A random classifier of the forest is completed.
- v. Give the test data and determine the class name of the study according to the ranking results of each classification.

Random forest is ocured with a huge amount of decision trees. The randomness is the most important operation with choosing of examples subset and feature subset for

creating a random forest. It is very important to making independent decision trees, decreasing classification success and have better generalization skills (Breiman, 2001). The random operation is used to have training subset from original examples with bagging method. This process is important for providing the independence of every preparation subset. The selected feature subsets are used as training dataset. Rating of all features with respect to importance and results of every training results can affect the final decision. N variable is very critical for random forest because of strengthness and correlation. Strengthness and correlation can be changed for a better result with the value of N. The advantage of random operation in random forest is increasing the accuracy of classifier. It is very fast to create a single decision tree and random forest uses the parallel use of these decision trees which decrease the classification time.

3.2.5. Support Vector Machines

The first studies about Support Vector Machine made by Vapnik and his friends for finding solution to pattern classification and regression problems. SVMs are group of related supervised learning techniques used for classification and regression (Vapnik, 1995). It is implemented to solve problems like nonlinear, local minimum and high dimension. The structural risk minimization (SRM) principal is created base of SVM. The most important advantage of SVM is decreasing the classification error and increasing the geometric margin simultaneously. The basic idea behind of the SVM algorithm is to create a decision plane (see Figure 3.6) with an N-dimensional space (N is the property number) that clearly classifies the data points.

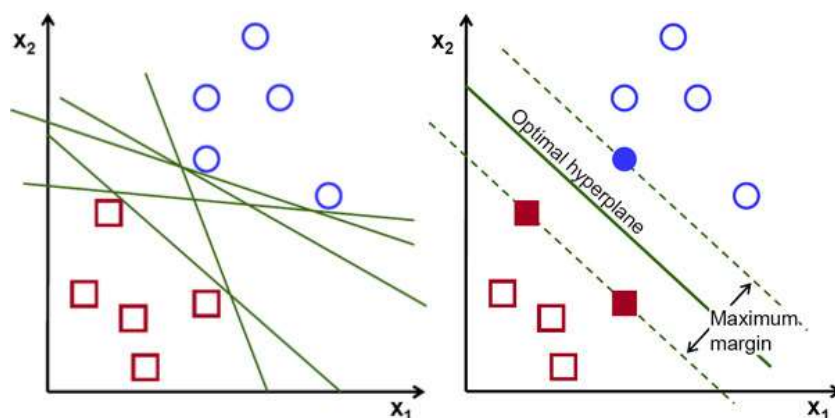


Figure 3.6: Optimal hyperplane.

SVM generates a hyperlane by using a linear model to apply nonlinear class boundaries on some nonlinear mapping input vectors into a high-dimensional feature space. There are some unknown and nonlinear dependence for instance in mapping

function $y = f(x)$ between some high-dimensional input vector x and scalar output y . No information is given about essential joint probability functions and a contribution should be given to distribution-free learning. The data set for training $D = \{(x_i, y_i) \in X \times Y\}, I = 1, I$ where I means training data pairs and the amount of it is the same with training data set D . Usually y_i is defined as d_i where d means expected target value. As a result, SVM is one member of supervised learning methods.

SVM has three main advantages;

- 2 parameters are enough to select these are upper limits and the kernel.
- There are unique optimal and global solutions for a linearly constrained quadratic problems.
- And using SRM principle make a good generalization of performance.

SVM is applied to many problems by researchers due to these main advantages (Siklasi et al., 2009; Breiman, 1996). There are a few important limitations of SVM (Bhavsar et al., 2012):

- Selecting the kernel parameter is the biggest problem of SVM.
- The second problem in SVM is speed and size. It is computationally expensive because of large training sets.
- Another important problem and research area in SVM is optimal design for multiclass issues.

3.2.6. Artificial Neural Networks

Artificial Neural Networks (ANNs) are adaptive nonlinear data processing systems which merge many processing units with a series of features like self-organizing, self-adapting and real-time learning (McCulloch and Pitts, 1943). Studies on the ANNs has been significantly increased from 1980s and ANNs are applied to many problems in different areas. Many problems have occurred while studies on ANNs were increasing. For example, structure and parameter choice of the networks, data set selection for training, stating the initial values are the some of these problems.

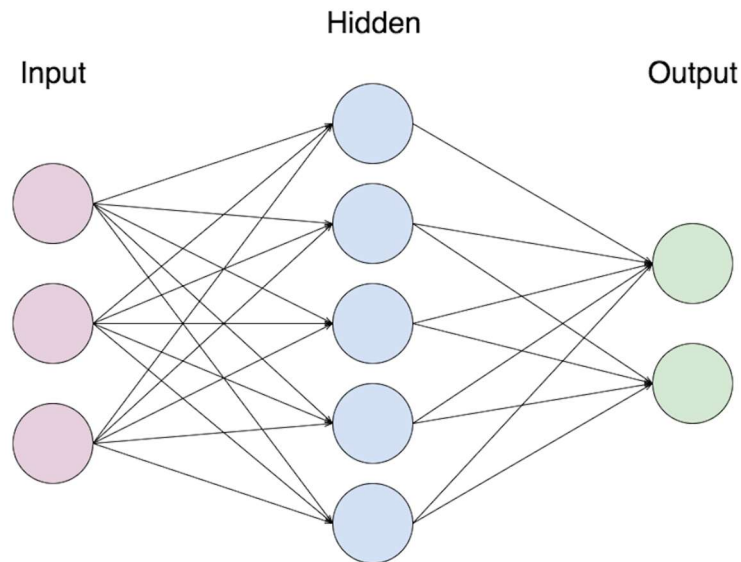


Figure 3.7: Structure of artificial neural networks.

The ANN has become to be a very fresh and useful model which could be used for solving of problem and machine learning. ANN has designed similar to human brain nervous systems for managing the information. It works with the same functionalities of that biological model. The solution way of solving the problems of an ANN can be thought as machine type of brain performance (Haykin, 2009). The unique information-processing ability is the major element of brain. It uses the “neurons” which are connected to each other and works together to solve the specific problems. The synaptic relationship of neurons makes the learning process in human brain and ANN (Stergiou and Siganos, 1996).

The ANNS are also using training data set for learning process. It iteratively changes the values of weights to reach the desired output values. There are three main learning types in ANNs, these are supervised, unsupervised and reinforcement learning. The basic idea at behind of supervised learning is comparing the actual and desired result. Back propagation and other optimization algorithms are used for decreasing error in result with iteratively adjust the weights. Reinforcement learning is seperated from other supervised learning because it just checks actual output is correct or not. Finding the best correlation of the input data is the basic principal of unsupervised learning. There is just finding a rule for updating weights.

3.2.7. Deep Learning

3.2.7.1. Development of Deep Learning

Deep learning is developed from ANNs and it is one of today's leading areas in machine learning. Research started in the 1940s on ANNs. By analyzing the characteristics of neurons (McCulloch and Pitts, 1943), McCulloch and colleagues proposed the McCulloch-Pitts (MP) model. In the learning process, Hebb and colleagues developed a theory that describes the characteristics of neurons (Hebb, 1949). This theory has played a major role in neural network development. Later, Rosenblatt discovered the perceptron algorithm, a supervised learning type and binary classification (Rosenblatt, 1958). Minsky and Papert found that the perceptron algorithm had too many limits in theory, and the development of artificial neural networks was negatively affected. However, Hopfield proposed the Hopfield network in the early 1980s (Hopfield, 1982). So neural networks became popular again. Hinton developed the Boltzmann machine using the simulated annealing algorithm (Ackley et al., 1985). In the 1990s, various machine learning techniques like SVM (Cortes and Vapnik, 1995) emerged. The fact that these methods give good results in theory and practice contributed to the improvement of ANNs and the advancement of studies in this direction. Hinton introduced the concept of deep learning in Science magazine in 2006. Thus, machine learning studies revived. The basic principle of Deep Learning and comparison with neural network is given in Figure 3.8 (KDNuggets, 2019).

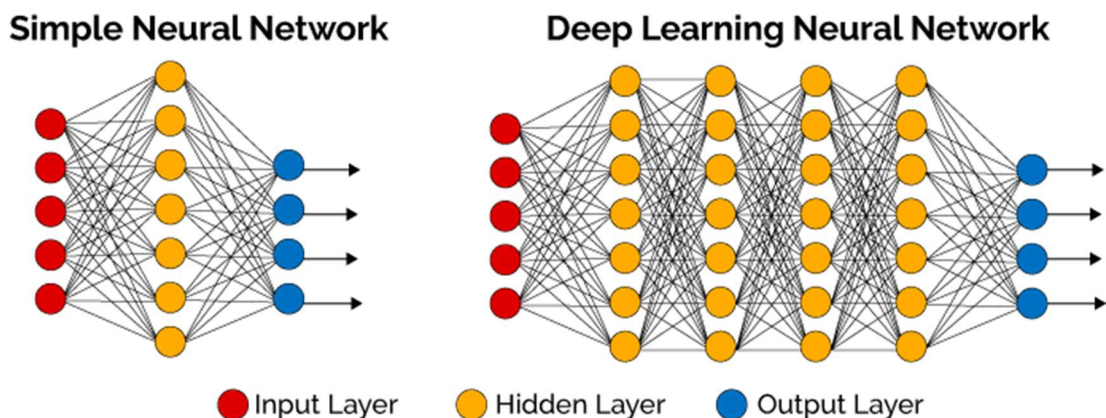


Figure 3.8: Neural networks vs Deep learning architecture.

Deep learning models use hierarchical structures to connect layers together. These models transform the low-level properties of data into high-level abstract features, which can make deep learning models superior to other machine learning methods. One of the powerful features that distinguish deep learning from other methods is that

it is based on data, not user experience (Du et al., 2016). Thus, deep learning works independently from the users. Technological advances increase the performance of computers and the data obtained to reach a very large size, contributes to increase the deep learning usage. In this thesis, deep learning models and applications will be briefly mentioned.

3.2.7.2. Deep Learning Models

Today, deep learning has many different models. The most well known and used are Autoencoder (AE), Deep Belief Network (DBN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN).

3.2.7.2.1. Autoencoder

Autoencoder is a deep learning model that is often used to process very high dimensional data (Du et al., 2016). Autoencoder learns to represent the data set by reducing the size of the data. By passing the given input A through a series of weighting and mapping operations, the output of the lower dimensional B can be obtained. The inverse of these operations is then applied to convert B-dimensional data to A-dimensional data. The other process is to update the network weights to reach the smallest value of the error function $L(A, A')$. The operating principle of Autoencoder is shown in Figure 3.9. (Autodencoder, 2019).

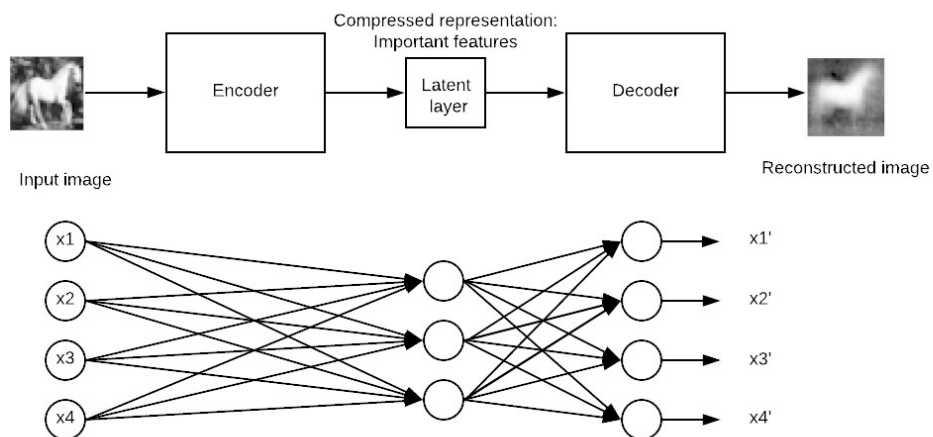


Figure 3.9: Basic structure of autoencoder.

3.2.7.2.2. Deep Belief Network

The Deep Belief Network model is a deep learning model created by a limited number of restricted Boltzmann machines (RBMs). RBM is a productive artificial neural network model consisting of Boltzmann machines. Although RBM is similar to the Boltzmann machine's two-layer neuron structure, a neuron doesn't have any connection with other one in the same layer with only the whole connection among the visual layer and the hidden layer. There is main structure of RBM is depicted in Figure 3.10 (DeepBeliefNetwork, 2019).

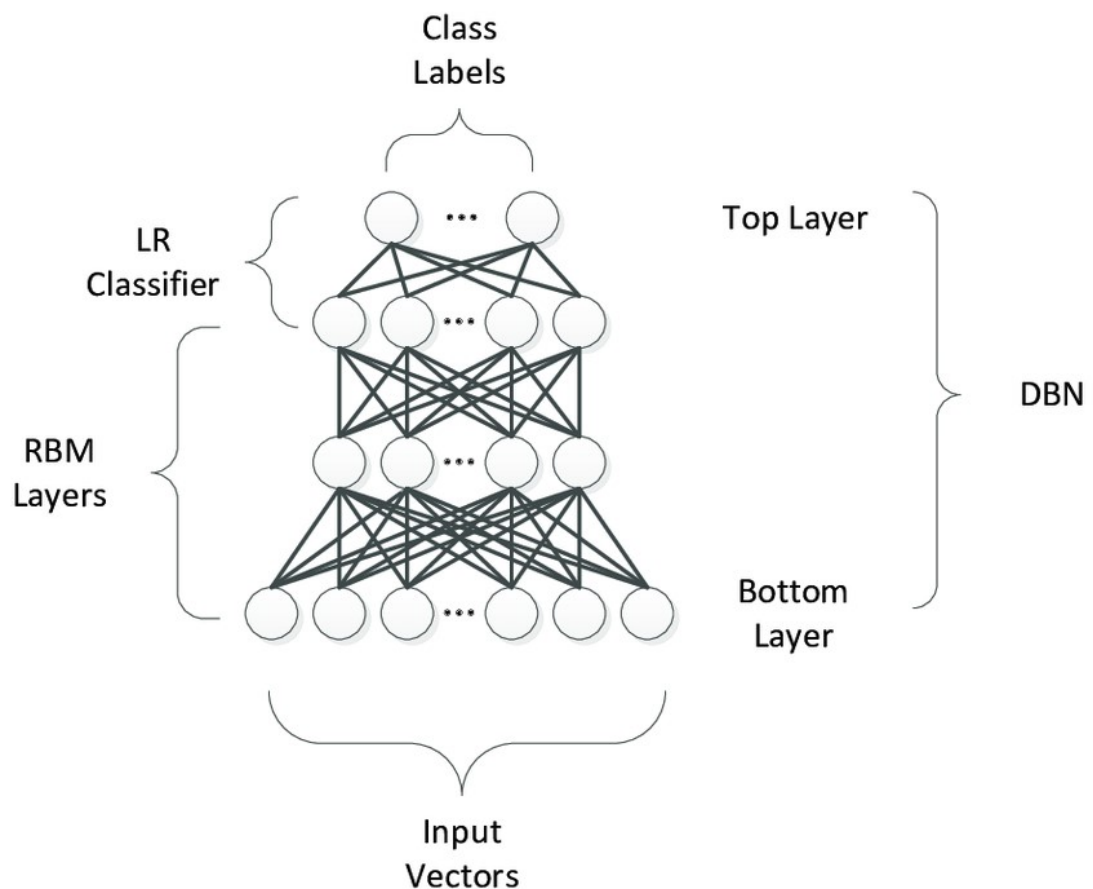


Figure 3.10: Architecture of a deep belief network (DBN).

After increasing the number of the hidden layers of RBM, we can get deep Boltzmann machine. Then, we adopt a top-down directed connection near the visual layer so that we can get DBN model. When training the network, the greedy unsupervised layer-wise pre-training method can be used to get the network weights. It only trains one layer at a time with the output of the lower layer being used as the input of the higher layer. Then, back-propagation algorithm is used to fine-tune the whole network.

3.2.7.2.3. Convolutional Neural Network

The receptive field principle proposed in the 1960s and the neocognitron based on the receptive field (Fukushima et al., 1983) proposed in the 1980s is considered the foundation of the Convolutionary Neural Network (CNN). CNN's most important feature is that local receptive field and weight sharing are used by the network. CNN reduces the number of training variables dramatically with these two features and makes the network less complex. A simple CNN structure involves convolutionary layers, pooling layers, and layers that are completely linked. The convolutionary layer is used for extraction of information. Each neuron input in this layer is connected to the previous one's local receptive field. The layer of pooling is used to map functions. It can reduce the information dimension and keep the network structure invariant. Figure 3.11 demonstrates the basic structure of CNN (CNN, 2019).

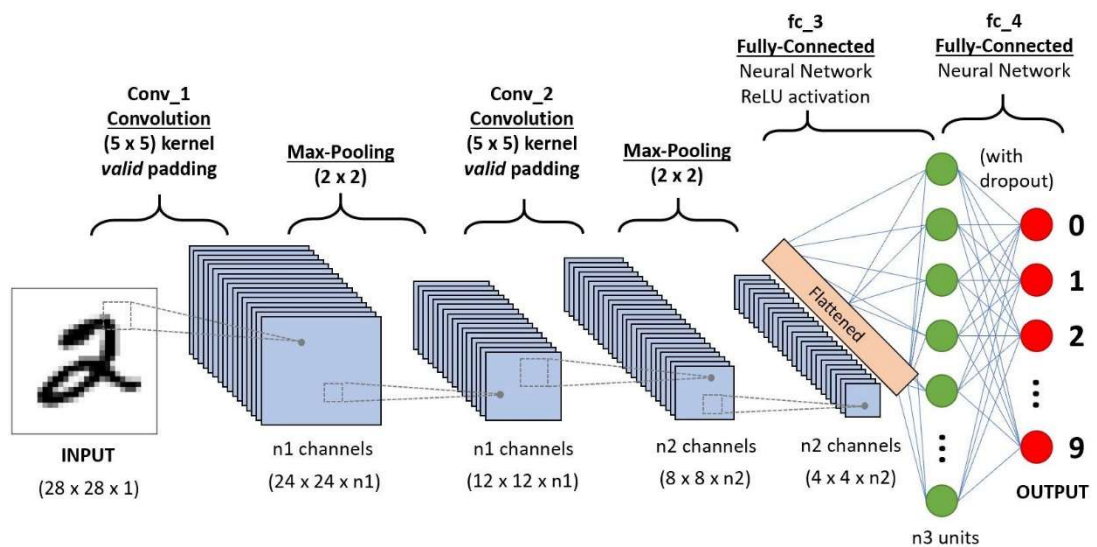


Figure 3.11: The basic structure of CNN on digit classification.

CNN is the most remarkable deep learning model of recent years. It gives productive results in many different fields and different studies. There are many types of CNN structures, such as LeNet (Lecun et al., 1998), AlexNet (Krizhevsky et al., 2012), ZFNet (Zeiler and Fergus, 2014), VGGNet (Simonyan and Zisserman, 2014) and GoogleNet (Szegedy et al., 2015). LeCun proposed a convolutional neural network namely LeNet, and applied to handwriting recognition. AlexNet is mainly used to object detections. After that, ZFNet, VGGNet and GoogleNet were put forward based on AlexNet. At present, CNN is still an active topic with many directions to explore.

Some researchers want to increase the complexity of CNN structures. Others want to combine CNN with other traditional machine learnings.

3.2.7.2.4. Recurrent Neural Network

A kind of deep learning system is the Recurrent Neural Network (RNN). RNN has a neural network architecture for feedforward, but has controlled cycles. This structure allows the data to circulate through the network so that each output is not connected to the current input, but to the previous outputs. Figure 3.12 (RNN, 2019) shows the basic structure of RNN.

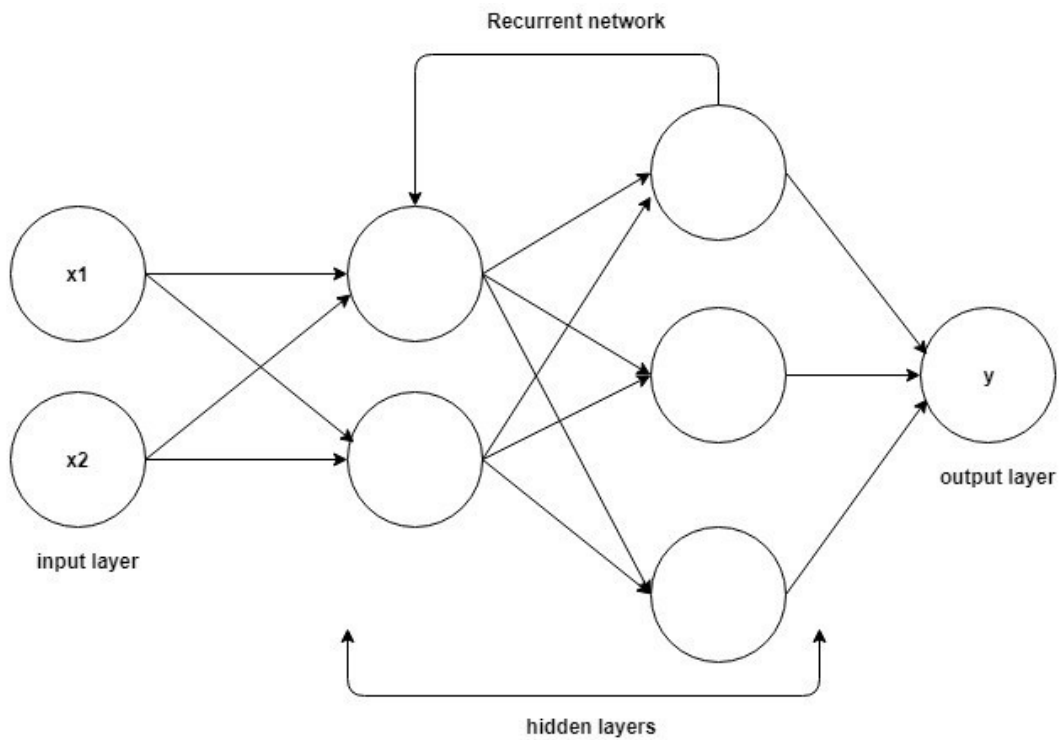


Figure 3.12: The basic structure of RNN.

Traditional RNN was successful in time series problems and faced serious problems in the back propagation process. For this reason, the RNN model was often used for problems requiring short-term memory. To solve this problem, researchers began to develop different structures such as Long Short Term Memory. Unlike RNN, the LSTM structure has a memory and an input-output port so that it stores the information in memory and controls the input-output port in this memory. Thanks to these features, LSTM performs better than RNN in long-term memory tasks.

3.2.7.3. Deep Learning Frameworks

There are many deep learning systems as well as deep learning models. Caffe, TensorFlow, Torch and Theano are the most popular ones.

Caffe (Jia et al., 2014) is a deep learning framework that is more suitable for CNN models. It contains many libraries such as MKL, OpenBLAS and cuBLAS. Caffe has a set of tools to be used for training, predicting, fine-tuning and so force. The configuration files of Caffe are simple to set up. And the Matlab and Python interfaces it provided are convenient to use.

TensorFlow (Abadi et al., 2016) is a large-scale machine learning framework which provides an interface for machine learning algorithms to execute. It has been used in many fields, including speech recognition, computer vision, robotics, information retrieval, and natural language processing. Tensorflow is developed from DistBelief.

Torch (Collobert et al., 2002) can support most of the machine learning algorithms. It includes most popular algorithms and models such as multi-layer perceptrons, support vector machines, Gaussian mixture models, hidden Markov models, spatial and temporal convolutional neural networks, AdaBoost, Bayes classifiers and so on.

Theano (Al-Rfou et al., 2016) is a framework based on Python. It can support some unsupervised and semi-supervised learning approaches as well as supervised learning approaches, such as logistic regression, multi-layer perceptron, deep CNN, AE, RBM, and DBN. Thanks to these functions, Theano is usually be used for teaching at aboard. However, Theano has a weakness that its speed is too slow.

3.3. Evaluation Methods of Classification Results

In this section, evaluation criteria for comparison of classical methods are mentioned.

3.3.1. Mean Absolute Error (MAE)

The MAE finds the average magnitude of errors in a series of estimates, regardless of their direction. It calculates the accuracy for continuous variables. The equation can be found in library references. The MAE is the average of the absolute values of the differences between the estimate and the coincident observation relative to the validation example. MAE is a linear score; this means that all individual differences are on average equal weight.

The mean absolute error is given by:

$$MAE = \frac{\sum_{i=1}^n |x_i - y_i|}{n}$$

The mean absolute error is an average absolute value of errors $|x_i - y_i|$, where x_i is the prediction and y_i is the target value.

3.3.2. Root mean squared error (RMSE)

RMSE is a quadratic scoring principle which also calculates the error's average magnitude. It is the square root of the mean differences in squares between prediction and real observation.

The mean absolute error is given by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (x_i - y_i)^2}{n}}$$

The root mean square error is shown above, where x_i represents the prediction class and y_i represents the result of truly classified values. RMSD is a measure of truthiness that used for compare prediction errors of different models. It compares these errors not between datasets because it depends on the scale (Hyndman et al., 2006).

3.3.3. Confusion Matrix

Forecasted results of a classification issue's summary is called confusion matrix. The key to the confusion matrix is to summarize and classify the number of true and false estimates by the degree of counting. It also shows when your classification model interrupt with predictions. This tells us not only the errors that made by classifiers, more significant it tells types of errors made.

Table 3.1: Confusion matrix table.

	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

The terms which are given in Table 3.1 are described as below,

- Class 1: means correct prediction
- Class 2: means wrong prediction
- True Positive (TP): Target class is positive, and predicted class is positive.
- False Negative (FN): Target class is positive, but predicted class is negative.
- True Negative (TN): Target class is negative, and predicted class is negative.
- False Positive (FP): Target class is negative, but predicted class is positive.

After creation of confusion matrix, classification accuracy is calculated as below:

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

3.3.3.1. Recall

Recall can be calculated as the division of the total number of truly classified positive samples by the total number of positive samples. High Recall shows the class is truly recognized (small number of FN).

Recall formula is given below:

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

3.3.3.2. Precision

Precision can be calculated as division of total number of truly classified positive samples by the total number of predicted correct samples. High Precision shows an example classified as positive is truly positive (small number of FP).

Precision formula is shown below:

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

3.3.3.3. F-Measure

The F-measure can be used for a measure to represents both Precision and Recall measures. F-measure calculation uses harmonic mean instead of arithmetic mean as to decrease the excessive values more. The result of F-measure always will be close to the smaller value of Precision or Recall.

Formulation of F-measure is shown below:

$$F - Measure = \frac{2 * Recall * Precision}{Recall + Precision}$$



CHAPTER 4

4. RESULTS AND DISCUSSIONS

In this chapter, experimental results for gene selection and classification of two different cancer datasets is completed. Wrapper method is applied for the selections of genes and five different classification algorithms used to classify selected gene subsets. RNA-Seq datasets are created with current gene expression technologies. It gives many information about a gene but size of RNA-Seq data is too much for analyzing.

4.1. EXPERIMENTAL DATASETS

Renal cell cancer and lung cancer are two different dataset studied on in this thesis. Detailed information about these datasets can be found below.

Renal cell cancer dataset: First dataset is renal cell cancer (RCC) which is an RNA-Seq dataset provided by The Cancer Genome Atlas (TCGA) (Saleem et al., 2013). There are many dataset for researchers to study, download and analyse in TCGA which is a comprehensive community resource platform. There are 1,020 RCC samples with 20,531 RNA transcript for each sample in dataset which is taken from TCGA. This RNA-Seq data has 606, 323 and 91 specimens from the kidney renal papillary cell (KIRP), kidney renal clear cell (KIRC) and kidney chromophobe carcinomas (KICH), respectively. These three types of cancers are most known subtypes of RCC (account for nearly 90%-95% of the total malignant kidney tumors in adults) and separated as three different classes in this study (Goyal et al., 2013).

Lung cancer dataset: In addition to RCC, another RNA-Seq dataset used from TCGA is lung cancer dataset. It was obtained with RCC dataset for this study. There are 1,128 samples in dataset and each sample includes 20,531 transcripts. There are two different classes that are lung adenocarcinoma (LUAD) and lung squamous cell with carcinoma (LUSC) with 576 and 552 class sizes, respectively. In this thesis, these two different lung cancer types are used as classes.

Table 4.1: Description of Datasets.

Dataset	Number of Samples	Number of Genes	Number of Samples (each class)			Provided Platform
			KIRP	KIRC	KICH	
Renal Cell Cancer	1,020	20,531				The Cancer Genome Atlas
			606	323	91	
Lung Cancer	1,128	20,531	LUAD	LUSC		The Cancer Genome Atlas
			576	552		

4.2. GENE SELECTION

In this thesis, Wrapper Method was used for gene selection. There are 20,531 genes in one sample and it is computationally very expensive to train it. Some of these genes are same in the each sample and there is no any effect on classification. Python programming language used on tensorflow environment for applying correlation method for genes. Random Forest Regressor used as estimator in training model. Dataset was splited into training set and test set. %80 of dataset used for traning and %20 for testing. Kfold cross validation accepted as 5 for increasing the dataset variance. After running the program, best suited 50 genes selected for classification.

Table 4.2: Selected Genes for Renal Cell Cancer.

ACBD7	CLDND2	LDLR	OBSCN	RGS22
ADAMTS18	CORO7	LOC283663	OR52W1	SCD
ADCY8	DENND1A	LOC285735	OTOP2	SNORD111B
C20orf96	EN2	LOC650293	PER4	SNX10
C2orf61	FABP7	MAP1B	PGBD1	SPIN2B
C2orf83	GPR133	MCART6	PIP5KL1	TCEB3C
C6orf223	GPR144	MOS	PISD	TREML1
CCL7	HIST1H2BA	NACA2	PLAC1	WASF1
CCND2	KCNH4	NAMPT	PRSS42	WDR64
CCNO	KLB	NLRP10	PSMG1	ZFAT

Table 4.3: Selected Genes for Lung Cancer.

ANKRD23	DEFB104A	LCE6A	PSMD1	SPP2
BNIP3	DRD5	LHPP	PTAR1	SRP14
C21orf121	EMR4P	LOC647288	RAD51L1	STXBP4
CCDC122	EP300	LOC728410	RORC	TNIP3
CCS	HINT2	MPL	SHC1	TREML2P1
CLCA2	HTA	NKX2-1	SHD	TRIO
CLK3	KCNN1	NR2F6	SLK	WFDC2
CTU2	KIAA1671	OR2C1	SMARCD1	ZAR1L
CUEDC2	KRT74	OR4F21	SNORD115- 37	ZDHHC19
DCUN1D4	KRTDAP	PEX13	SPARCL1	ZNF841

After selection of these genes, 5 different classification methods applied on preprocessed samples with new genes subsets. Classification results will be explained in next section.

4.3. EXPERIMENTAL RESULTS FOR LUNG CANCER DATASET

After gene selection, classification algorithms were applied on lung cancer dataset. Firstly, classical algorithms were applied and then the results were obtained with deep learning methods. Classical methods are Decision Tree, Random Forest, three different types of Support Vector Machines and Artificial Neural Networks. Afterwards, deep learning methods were applied with 7 different optimizers and all these results were compared in the tables.

First, the Decision Tree classification algorithm was applied to the lung cancer dataset. 70% of the dataset is reserved for training and 30% for the test. After training, the algorithm was tested for 30% test data and classification was completed with an accuracy of 91.74%. The LUAD cancer type is designated 0, while the LUSC cancer type is designated 1. At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.4 and Table 4.5.

Confusion matrix of classification result:

Table 4.4: Confusion matrix of decision tree classifier.

		Predicted Values	
		LUAD	LUSC
Actual Values	LUAD	168	8
	LUSC	20	143

Classification Report:

Table 4.5: Classification report of decision tree classifier.

	Precision	Recall	F-measure	Support
0	0.89	0.95	0.92	176
1	0.95	0.88	0.91	163
Micro avg	0.92	0.92	0.92	339
Macro avg	0.92	0.92	0.92	339
Weighted avg	0.92	0.92	0.92	339

After the Decision Tree, the Random Forest algorithm was applied. Again 70% of the data set is reserved for training and 30% for the test. Initially, the number of trees was randomly assigned to 100. Then the model was applied on the train set and then tested. As a result of the test, the model reached an accuracy rate of 93.51%. At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.6 and Table 4.7.

Confusion matrix of classification result:

Table 4.6: Confusion matrix of random forest classifier.

		Predicted Values	
		LUAD	LUSC
Actual Values	LUAD	171	5
	LUSC	17	146

Classification Report:

Table 4.7: Classification report of random forest classifier.

	Precision	Recall	F-measure	Support
0	0.91	0.97	0.94	176
1	0.97	0.90	0.93	163
Micro avg	0.94	0.94	0.94	339
Macro avg	0.94	0.93	0.93	339
Weighted avg	0.94	0.94	0.93	339

After Random Forest, the support vector machines algorithm, which is one of the most popular classical methods, has been applied. The data set was again divided into 70% training and 30% test. There are different type kernels of SVM. In this thesis linear, polynomial and RBF kernels used and three different result obtained. As a result of training and testing with linear kernel, an accuracy rate of 89.38% was obtained. At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.8 and Table 4.9.

Confusion matrix of classification result:

Table 4.8: Confusion matrix of SVM with linear kernel type.

		Predicted Values	
		LUAD	LUSC
Actual Values	LUAD	160	16
	LUSC	20	143

Classification Report:

Table 4.9: Classification report of SVM with linear kernel type.

	Precision	Recall	F-measure	Support
0	0.89	0.91	0.90	176
1	0.90	0.88	0.89	163
Micro avg	0.89	0.89	0.89	339
Macro avg	0.89	0.89	0.89	339
Weighted avg	0.89	0.89	0.89	339

After linear kernel, SVM applied with polynomial kernel. As a result of training and testing with polynomial kernel, an accuracy rate of 87.61% was obtained. At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.10 and Table 4.11.

Confusion matrix of classification result:

Table 4.10: Confusion matrix of SVM with polynomial kernel type.

		Predicted Values	
		LUAD	LUSC
Actual Values	LUAD	155	21
	LUSC	21	142

Classification Report:

Table 4.11: Classification report of SVM with polynomial kernel type.

	Precision	Recall	F-measure	Support
0	0.88	0.88	0.88	176
1	0.87	0.87	0.87	163
Micro avg	0.88	0.88	0.88	339
Macro avg	0.88	0.88	0.88	339
Weighted avg	0.88	0.88	0.88	339

Finally, SVM applied with RBF kernel. As a result of training and testing with polynomial kernel, an accuracy rate of 92.04% was obtained. At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.12 and Table 4.13.

Confusion matrix of classification result:

Table 4.12: Confusion Matrix of SVM with RBF Kernel Type.

		Predicted Values	
		LUAD	LUSC
Actual Values	LUAD	169	7
	LUSC	20	143

Classification Report:

Table 4.13: Classification report of SVM with RBF kernel type

	Precision	Recall	F-measure	Support
0	0.89	0.96	0.93	176
1	0.95	0.88	0.91	163
Micro avg	0.92	0.92	0.92	339
Macro avg	0.92	0.92	0.92	339
Weighted avg	0.92	0.92	0.92	339

After SVM, finally the ANN has been applied to Lung cancer dataset. Firstly, dataset was again splitted into 70%-30% for training and test. As a result of training and testing with Neural Networks, an accuracy rate of 89.97% was obtained. After classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.14 and Table 4.15.

Confusion matrix of classification result:

Table 4.14: Confusion matrix of artificial neural network.

		Predicted Values	
		LUAD	LUSC
Actual Values	LUAD	160	16
	LUSC	18	145

Classification Report:

Table 4.15: Classification report of artificial neural network.

	Precision	Recall	F-measure	Support
0	0.90	0.91	0.90	176
1	0.90	0.89	0.90	163
Micro avg	0.90	0.90	0.90	339
Macro avg	0.90	0.90	0.90	339
Weighted avg	0.90	0.90	0.90	339

The comparison of the 4 classical classification algorithms applied before deep learning methods is given in the Table 4.16.

Table 4.16: Comparison of Classification Algorithms.

Classifier	MAE	RMSE	Classification Accuracy
Decision Tree	0.09	0.29	91.74%
Random Forest	0.08	0.25	93.51%
Linear SVM	0.10	0.32	89.38%
Polynomial SVM	0.35	0.35	87.61%
RBF SVM	0.08	0.28	92.04%
Artificial Neural Networks	0.10	0.32	89.97%

When the results were compared, linear kernel gave the best results among SVM kernels, while Random Forest gave the best results in classical classification algorithms. After this stage, deep learning methods were applied and the results were compared.

A Keras sequential model was used for the classification. Keras, which is written with Python, is a high-level neural networks API. It is capable of running with TensorFlow, CNTK or Theano. It was developed with a focus on enabling fast experimentation. RELU has been used as activation function. 7 different optimizers with one activation function applied on the dataset. Then results are compared.

SGD Optimizer: SGD stands for stochastic gradient descent optimizer. It contains support for momentum, learning rate decay, and Nesterov momentum. SGD optimizer tested on dataset and the following results obtained; training accuracy: 87.04% and test accuracy: 90.27%. Figures shown below during training and testing.

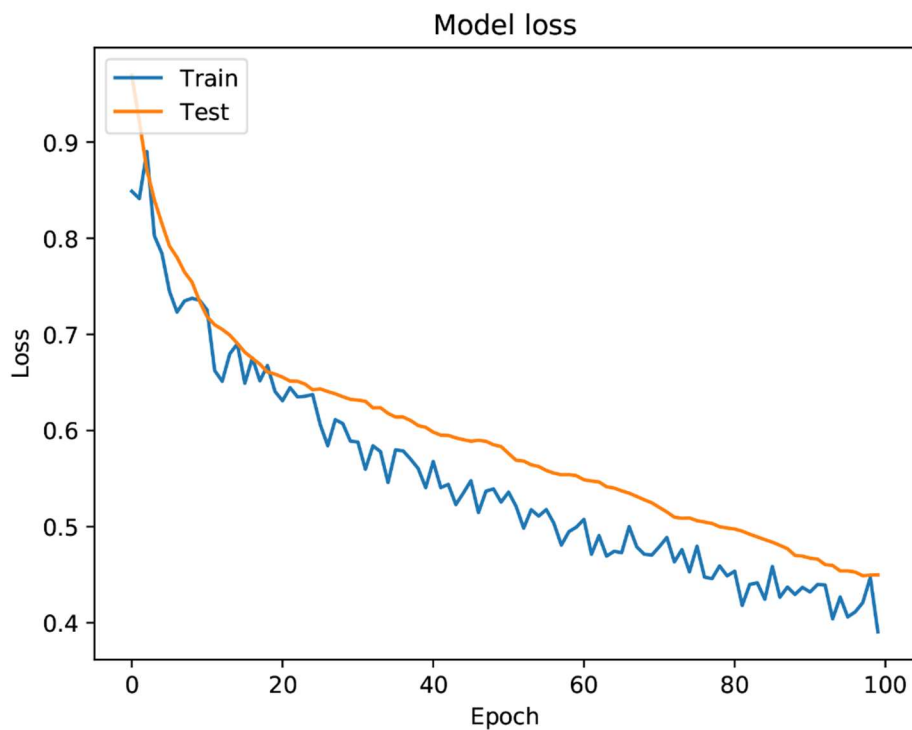


Figure 4.1: Model loss during training and testing for SGD optimizer.

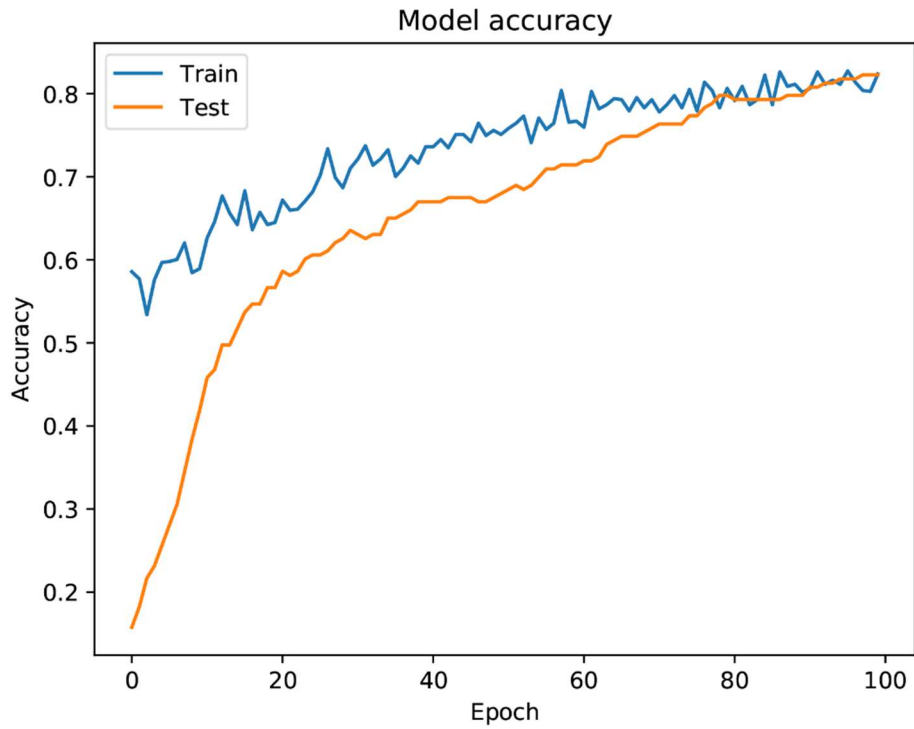


Figure 4.2: Model accuracy during training and testing for SGD optimizer.

At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.17 and Table 4.18.

Confusion matrix of classification result:

Table 4.17: Confusion matrix of SGD Optimizer.

		Predicted Values	
		LUAD	LUSC
Actual Values	LUAD	162	11
	LUSC	15	151

Classification Report:

Table 4.18: Classification report of SGD Optimizer.

	Precision	Recall	F-measure	Support
0	0.92	0.94	0.93	173
1	0.93	0.91	0.92	166
Micro avg	0.92	0.92	0.92	339
Macro avg	0.92	0.92	0.92	339
Weighted avg	0.92	0.92	0.92	339

RMSProp Optimizer: The RMSprop optimizer limit oscillations in the orthogonal direction. So, the learning rate can be improved and the algorithm takes longer stages as it progresses in the horizontal direction more quickly. RMSProp optimizer tested on dataset and the following results obtained; training accuracy: 96.67% and test accuracy: 93.86%. Figures shown below during training and testing.

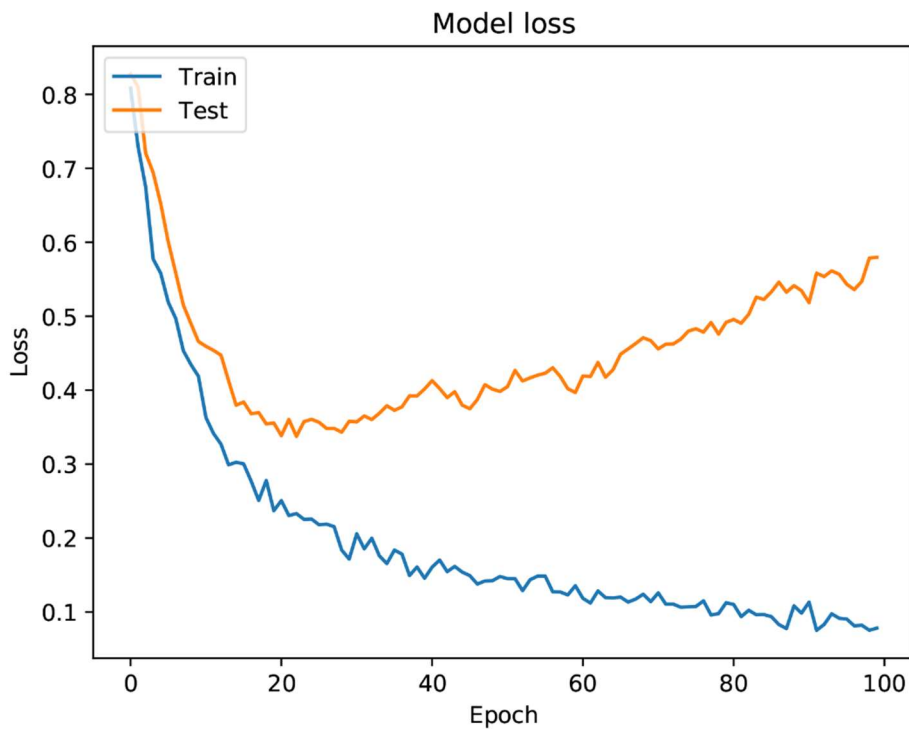


Figure 4.3: Model loss during training and testing for RMSProp optimizer.

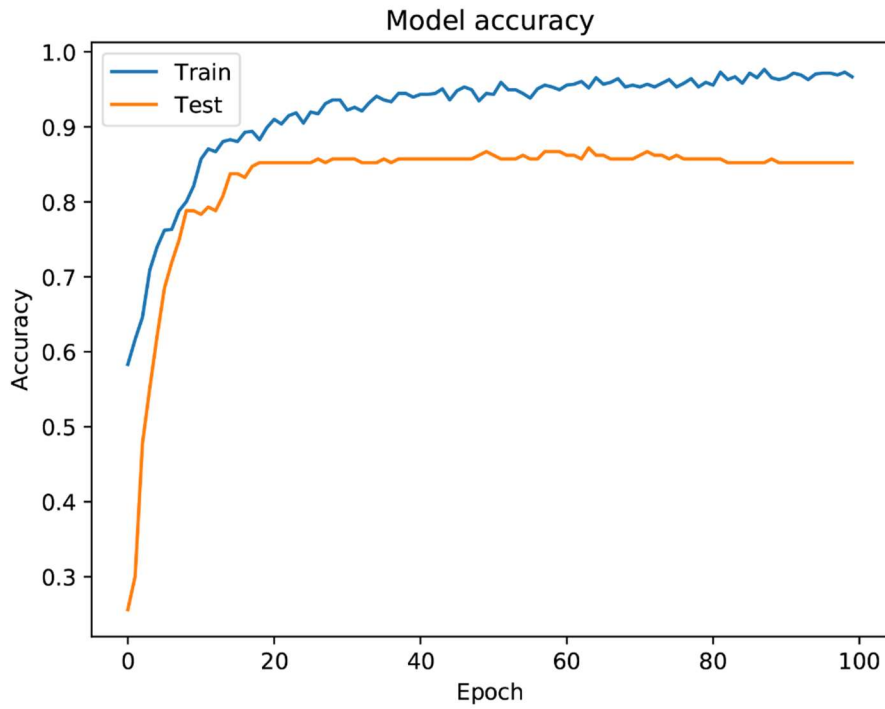


Figure 4.4: Model accuracy during training and testing for RMSProp optimizer.

At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.19 and Table 4.20.

Confusion matrix of classification result:

Table 4.19: Confusion matrix of RMSProp Optimizer.

		Predicted Values	
		LUAD	LUSC
Actual Values	LUAD	162	11
	LUSC	11	155

Classification Report:

Table 4.20: Classification report of RMSProp Optimizer.

	Precision	Recall	F-measure	Support
0	0.94	0.94	0.94	173
1	0.93	0.93	0.93	166
Micro avg	0.94	0.94	0.94	339
Macro avg	0.94	0.94	0.94	339
Weighted avg	0.94	0.94	0.94	339

Adagrad Optimizer: It implemented with checking how often a parameter is updated in training and it has parameter-specific learning rates. The learning rate will be lower if parameter is more updated. Adagrad optimizer tested on dataset and the following results obtained; training accuracy: 95.21 % and test accuracy: 93.81%. Figures shown below during training and testing.

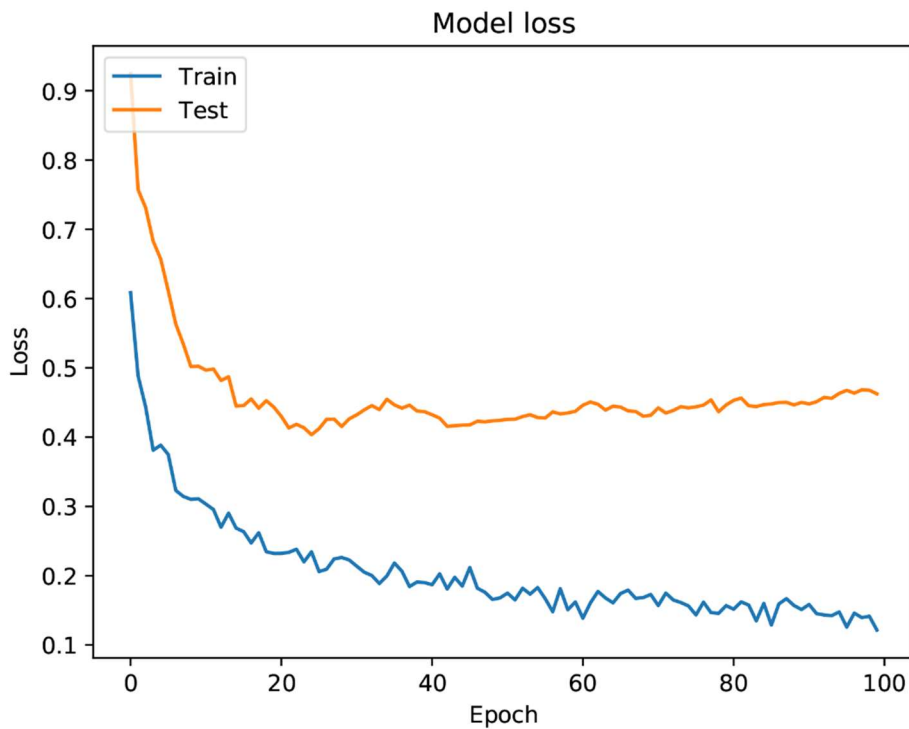


Figure 4.5: Model loss during training and testing for Adagrad optimizer.

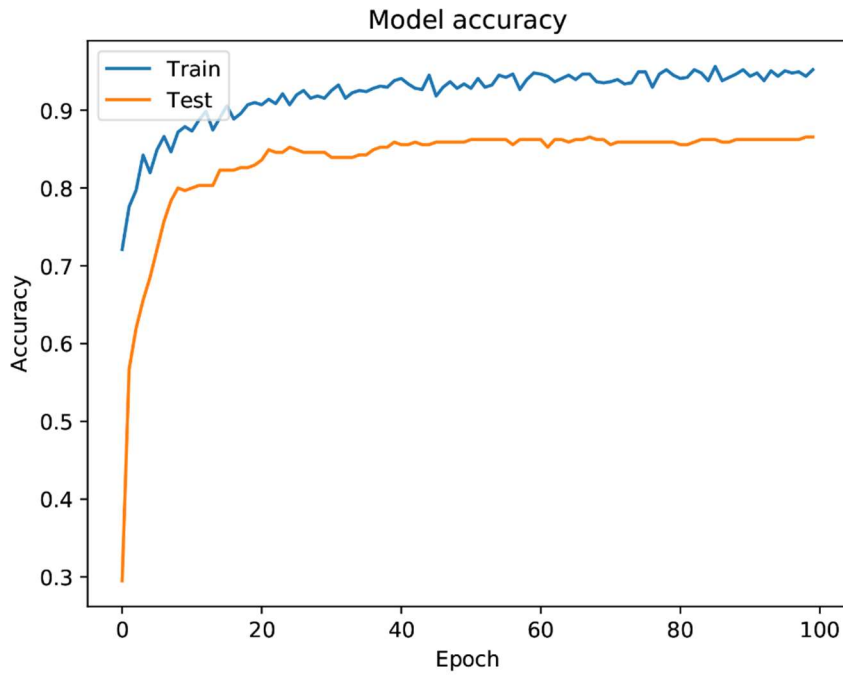


Figure 4.6: Model accuracy during training and testing for Adagrad optimizer.

At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.21 and Table 4.22.

Confusion matrix of classification result:

Table 4.21: Confusion matrix of Adagrad Optimizer.

		Predicted Values	
		LUAD	LUSC
Actual Values	LUAD	165	8
	LUSC	12	154

Classification Report:

Table 4.22: Classification report of Adagrad Optimizer.

	Precision	Recall	F-measure	Support
0	0.93	0.95	0.94	173
1	0.95	0.93	0.94	166
Micro avg	0.94	0.94	0.94	339
Macro avg	0.94	0.94	0.94	339
Weighted avg	0.94	0.94	0.94	339

Adadelta Optimizer: Adadelta Optimizer: It is more powerful form of Adagrad. It adjusts learning rate based on change of gradient, instead of gathering all past gradients. Thus, Adadelta keeps going to learn even when there are many updates. It is not necessary that set an initial learning rate in Adadelta, when Adagrad and Adadelta compared. Nonetheless, initial learning level and decay factor can be assigned in this edition of Adadelta as in other optimizers. Adadelta optimizer tested on dataset and the following results obtained; training accuracy: 94.83 % and test accuracy: 95.54%. Figures shown below during training and testing.

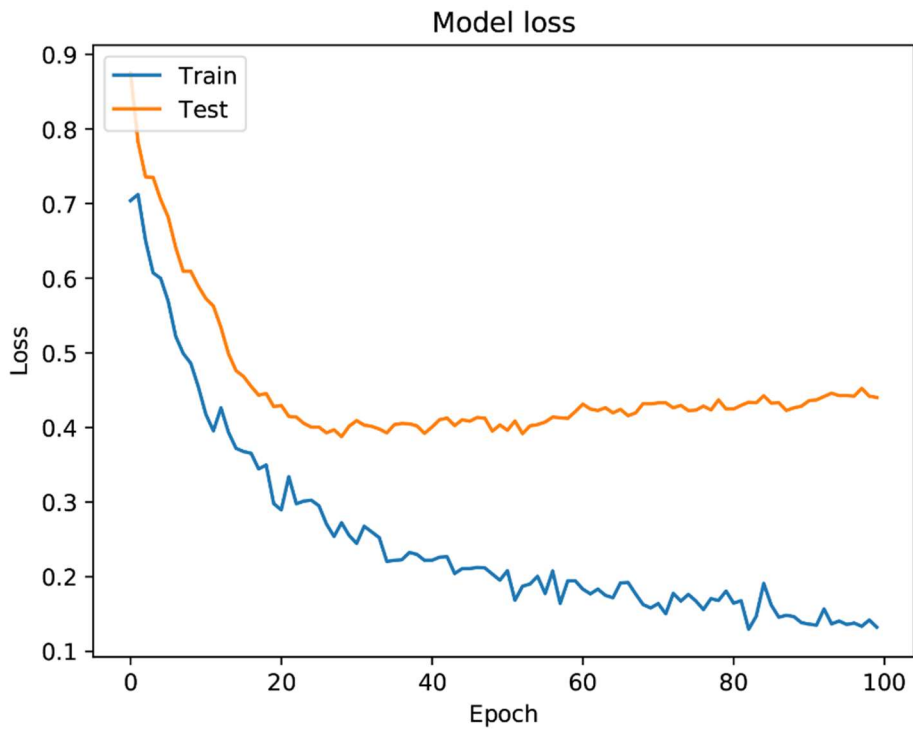


Figure 4.7: Model loss during training and testing for Adadelta optimizer.

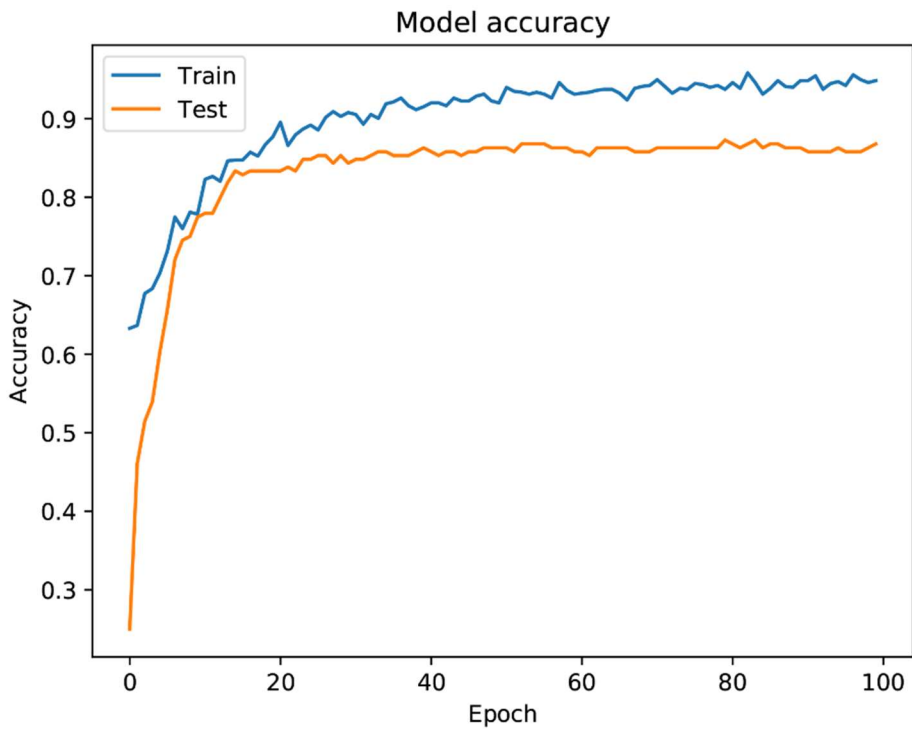


Figure 4.8: Model accuracy during training and testing for Adadelta optimizer.

At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.23 and Table 4.24.

Confusion matrix of classification result:

Table 4.23: Confusion matrix of Adadelata Optimizer.

		Predicted Values	
		LUAD	LUSC
Actual Values	LUAD	165	8
	LUSC	10	156

Classification Report:

Table 4.24: Classification report of Adadelata Optimizer.

	Precision	Recall	F-measure	Support
0	0.94	0.95	0.95	173
1	0.95	0.94	0.95	166
Micro avg	0.95	0.95	0.95	339
Macro avg	0.95	0.95	0.95	339
Weighted avg	0.95	0.95	0.95	339

Adam Optimizer: AdaM stands for Adaptive Momentum. It combines the Momentum and RMS prop in a single approach making AdaM a very powerful and fast optimizer. Adam optimizer tested on dataset and the following results obtained; training accuracy: 96.76 % and test accuracy: 93.81%. Figures shown below during training and testing.

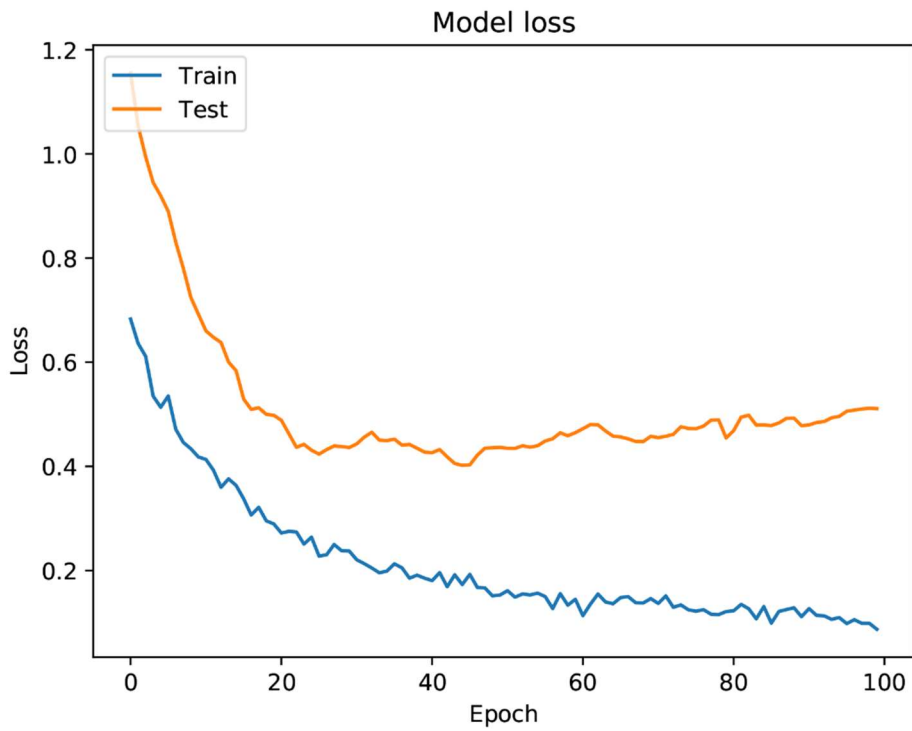


Figure 4.9: Model loss during training and testing for Adam optimizer.

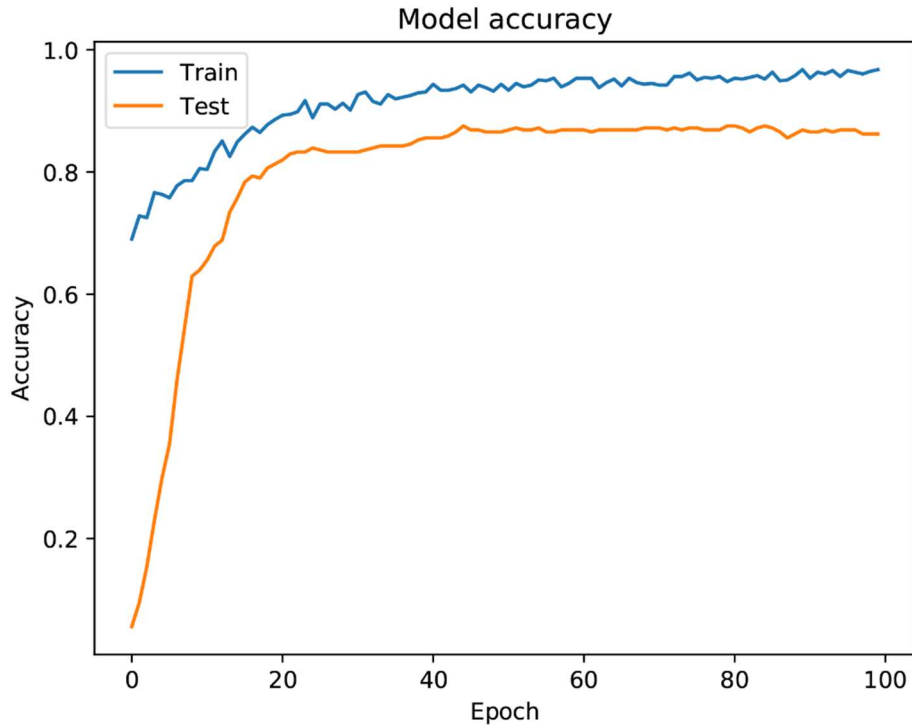


Figure 4.10: Model accuracy during training and testing for Adam optimizer.

At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and

weighted average also calculated. The classification report is shown in Table 4.25 and Table 4.26.

Confusion matrix of classification result:

Table 4.25: Confusion matrix of Adam Optimizer.

		Predicted Values	
		LUAD	LUSC
Actual Values	LUAD	163	10
	LUSC	11	155

Classification Report:

Table 4.26: Classification report of Adam Optimizer.

	Precision	Recall	F-measure	Support
0	0.94	0.94	0.94	173
1	0.94	0.93	0.94	166
Micro avg	0.94	0.94	0.94	339
Macro avg	0.94	0.94	0.94	339
Weighted avg	0.94	0.94	0.94	339

Adamax Optimizer: It is a different form of Adam but based on the infinity norm. Adamax optimizer tested on dataset and the following results obtained; training accuracy: 92.97 % and test accuracy: 92.86%. Figures shown below during training and testing.

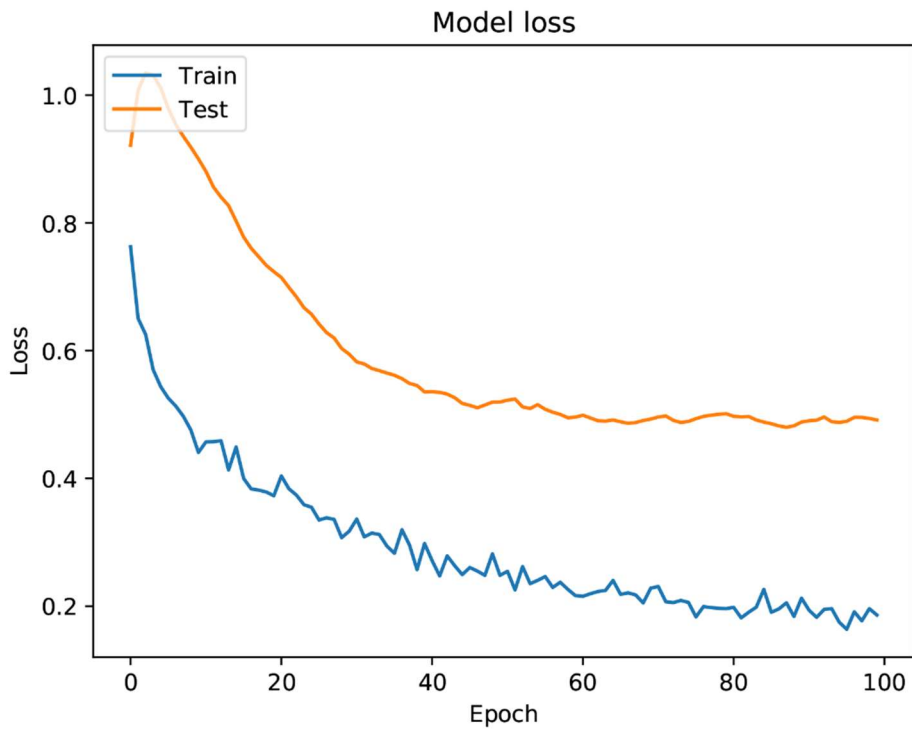


Figure 4.11: Model loss during training and testing for Adamax optimizer.

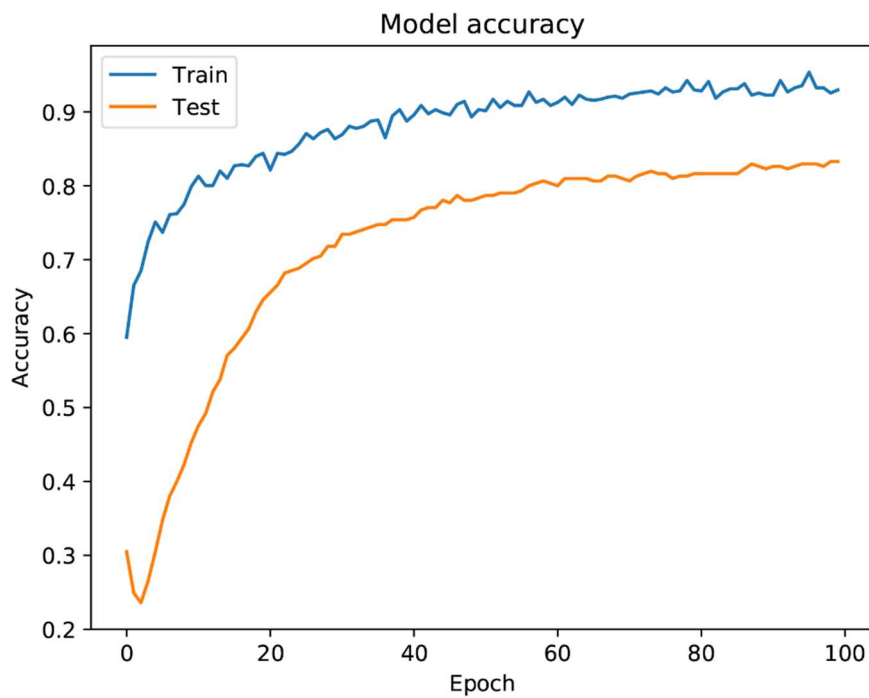


Figure 4.12: Model accuracy during training and testing for Adamax optimizer.

At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and

weighted average also calculated. The classification report is shown in Table 4.27 and Table 4.28.

Confusion matrix of classification result:

Table 4.27: Confusion matrix of Adamax Optimizer.

		Predicted Values	
		LUAD	LUSC
Actual Values	LUAD	166	7
	LUSC	13	153

Classification Report:

Table 4.28: Classification report of Adamax Optimizer.

	Precision	Recall	F-measure	Support
0	0.93	0.96	0.94	173
1	0.96	0.92	0.94	166
Micro avg	0.94	0.94	0.94	339
Macro avg	0.94	0.94	0.94	339
Weighted avg	0.94	0.94	0.94	339

Nadam Optimizer: Nadam stands for Nesterov Adam optimizer. It is similar to Adam optimizer but its structure is basically like RMSprop with Nesterov momentum. Nadam optimizer tested on dataset and the following results obtained; training accuracy: 98.45 % and test accuracy: 94.69%. Figures shown below during training and testing.

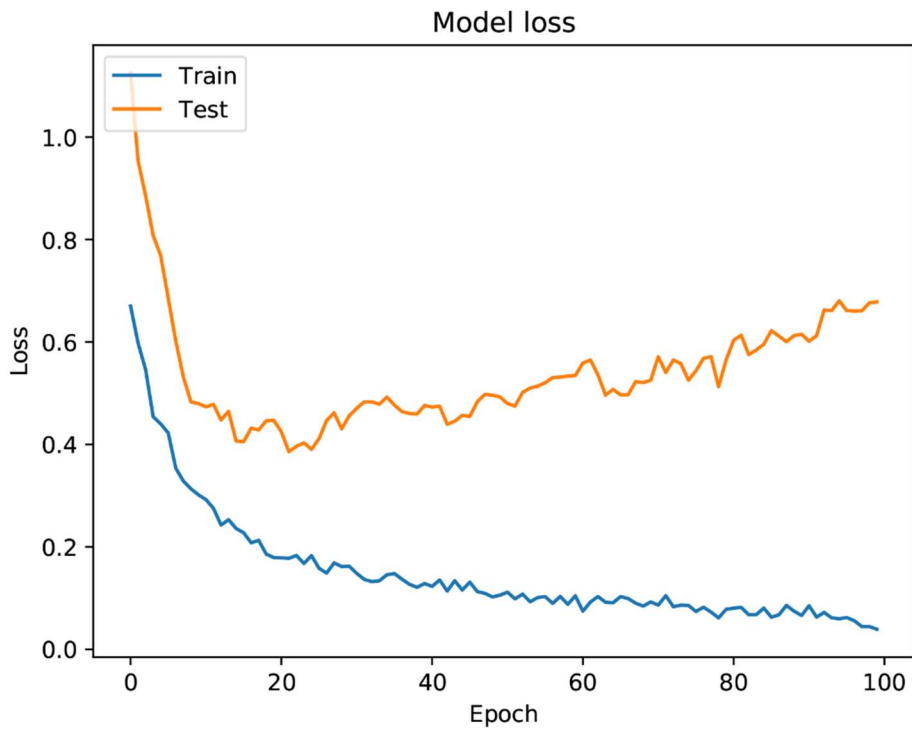


Figure 4.13: Model loss during training and testing for Nadam optimizer.

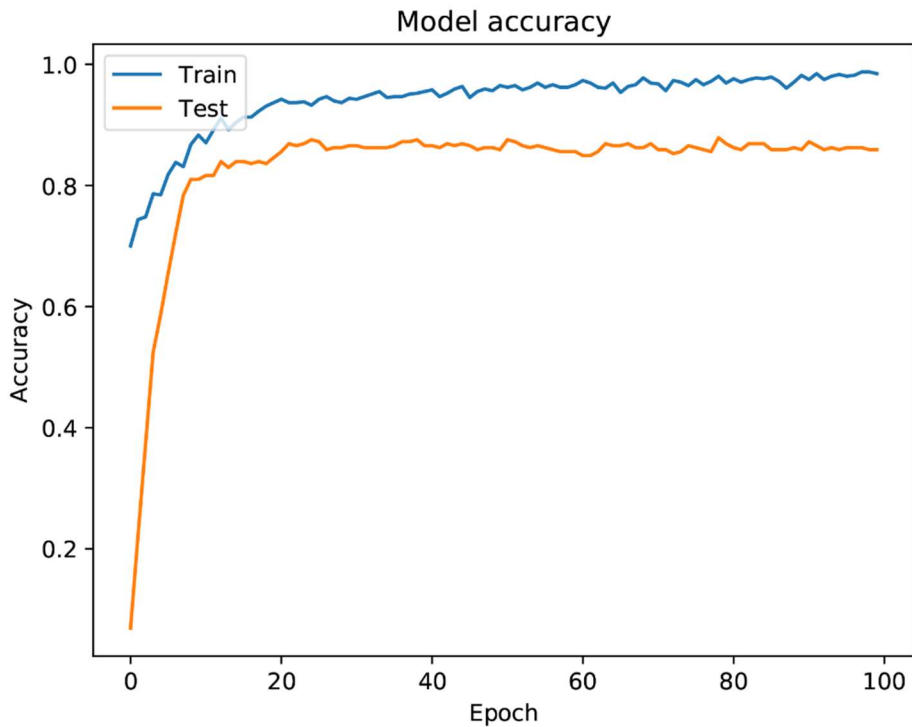


Figure 4. 14: Model accuracy during training and testing for Nadam optimizer.

At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and

weighted average also calculated. The classification report is shown in Table 4.29 and Table 4.30.

Confusion matrix of classification result:

Table 4.29: Confusion matrix of Nadam Optimizer.

		Predicted Values	
		LUAD	LUSC
Actual Values	LUAD	162	11
	LUSC	9	157

Classification Report:

Table 4.30: Classification report of Nadam Optimizer.

	Precision	Recall	F-measure	Support
0	0.95	0.94	0.94	173
1	0.93	0.95	0.94	166
Micro avg	0.94	0.94	0.94	339
Macro avg	0.94	0.94	0.94	339
Weighted avg	0.94	0.94	0.94	339

The training and test results with seven different optimizers and their graphs are given above. Each optimizer was briefly described and different results were obtained in each. Here it is observed that the optimizers affect the success of the model. When the results are examined, it is seen that the best result in model training is provided by Nadam optimizer. The highest success was achieved by Adadelata optimizer in testing. The results are combined in the Table 4.31.

Table 4.31: Comparison of results with different optimizers.

Optimizer	MAE	RMSE	Training Accuracy	Test Accuracy
SGD	0.18	0.27	82.37%	89.47%
RMSProp	0.8	0.23	96.67%	93.86%
Adagrad	0.09	0.21	95.21%	93.81%
Adadelta	0.07	0.21	94.83%	95.54%
Adam	0.08	0.21	96.76%	93.81%
Adamax	0.11	0.22	92.97%	92.86%
Nadam	0.08	0.22	98.45%	94.69%

The results obtained by classical methods were compared with the deep learning model. In the comparison, linear SVM with the best results from SVM and Adadelta with the best results were used in the deep learning model. The comparison based on the test results is shown in the Table 4.32.

Table 4.32: Comparison of results for Lung Cancer dataset.

Classifier	MAE	RMSE	Result
Decision Tree	0.09	0.29	91.74%
Random Forest	0.08	0.25	93.51%
SVM (RBF)	0.08	0.28	92.04%
Artificial Neural Networks	0.10	0.32	89.97%
Deep Learning Model (Adadelta)	0.07	0.21	95.54%

When the results were compared it was observed that the results obtained with the deep learning model were better. RNA-Seq lung cancer data were analyzed and 95.54% accuracy of the disease class was determined.

4.4. EXPERIMENTAL RESULTS FOR RENAL CELL CANCER DATASET

After gene selection, classification algorithms were applied on renal cell cancer dataset. Firstly, classical algorithms were applied and then the results were obtained with deep learning methods. Classical methods are Decision Tree, Random Forest, three different types of Support Vector Machines and Artificial Neural Networks. Afterwards, deep learning methods were applied with 7 different optimizers and all these results were compared in the table. Same process applied to Renal Cell Cancer dataset too.

Firstly, the DT classification method was applied to the RCC dataset. 70% of the dataset is reserved for training and 30% for the test. After training, the algorithm was tested for 30% test data and classification was completed with an accuracy of 90.52%. The KICH cancer type is designated 0, KIRC is designated as 1 and the KIRP cancer type is designated 2. After classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.33 and Table 4.34.

Confusion matrix of classification result:

Table 4.33: Confusion Matrix of Decision Tree Classifier.

		Predicted Values		
		KICH	KIRC	KIRP
Actual Values	KICH	22	4	0
	KIRC	6	78	7
	KIRP	6	6	177

Classification Report:

Table 4.34: Classification Report of Decision Tree Classifier.

	Precision	Recall	F-measure	Support
0	0.65	0.85	0.73	26
1	0.89	0.86	0.87	91
2	0.96	0.94	0.95	189
Micro avg	0.91	0.91	0.91	306
Macro avg	0.83	0.88	0.85	306
Weighted avg	0.91	0.91	0.91	306

After the Decision Tree, the Random Forest algorithm was applied. Again 70% of the data set is reserved for training and 30% for the test. Initially, the number of trees was randomly assigned to 100. Then the model was applied on the train set and then tested. As a result of the test, the model reached an accuracy rate of 91.83%. At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.35 and Table 4.36.

Confusion matrix of classification result:

Table 4.35: Confusion Matrix of Random Forest Classifier.

		Predicted Values		
		KICH	KIRC	KIRP
Actual Values	KICH	23	2	1
	KIRC	3	79	9
	KIRP	2	8	179

Classification Report:

Table 4.36: Classification Report of Random Forest Classifier.

	Precision	Recall	F-measure	Support
0	0.82	0.88	0.85	26
1	0.89	0.87	0.88	91
2	0.95	0.95	0.95	189
Micro avg	0.92	0.92	0.92	306
Macro avg	0.89	0.90	0.89	306
Weighted avg	0.92	0.92	0.92	306

After Random Forest, the support vector machines algorithm has been applied. The dataset was again splitted into 70%-30% for training and test. There are different type kernels of SVM. In this section again linear, polynomial and RBF kernels used and three different result obtained. As a result of training and testing with linear kernel, an accuracy rate of 87.91% was obtained. At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.37 and Table 4.38.

Confusion matrix of classification result:

Table 4.37: Confusion Matrix of SVM with Linear Kernel Type.

		Predicted Values		
		KICH	KIRC	KIRP
Actual Values	KICH	21	4	1
	KIRC	12	72	7
	KIRP	4	9	176

Classification Report:

Table 4.38: Classification Report of SVM with Linear Kernel Type.

	Precision	Recall	F-measure	Support
0	0.57	0.81	0.67	26
1	0.89	0.79	0.82	91
2	0.96	0.93	0.94	189
Micro avg	0.88	0.88	0.88	306
Macro avg	0.79	0.84	0.81	306
Weighted avg	0.89	0.88	0.88	306

After linear kernel, SVM applied with polynomial kernel. As a result of training and testing with polynomial kernel, an accuracy rate of 84.64% was obtained. At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.39 and Table 4.40.

Confusion matrix of classification result:

Table 4.39: Confusion Matrix of SVM with Polynomial Kernel Type.

		Predicted Values		
		KICH	KIRC	KIRP
Actual Values	KICH	17	7	2
	KIRC	8	69	14
	KIRP	7	9	173

Classification Report:

Table 4.40: Classification Report of SVM with Polynomial Kernel Type.

	Precision	Recall	F-measure	Support
0	0.53	0.65	0.59	26
1	0.81	0.76	0.78	91
2	0.92	0.92	0.92	189
Micro avg	0.85	0.85	0.85	306
Macro avg	0.75	0.78	0.76	306
Weighted avg	0.85	0.85	0.85	306

Finally, SVM applied with RBF kernel. As a result of training and testing with polynomial kernel, an accuracy rate of 83.01% was obtained. At the end of the classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.41 and Table 4.42.

Confusion matrix of classification result:

Table 4.41: Confusion Matrix of SVM with RBF Kernel Type.

		Predicted Values		
		KICH	KIRC	KIRP
Actual Values	KICH	9	10	7
	KIRC	1	72	18
	KIRP	0	16	173

Classification Report:

Table 4.42: Classification Report of SVM with RBF Kernel Type.

	Precision	Recall	F-measure	Support
0	0.90	0.35	0.50	26
1	0.73	0.79	0.76	91
2	0.87	0.92	0.89	189
Micro avg	0.83	0.83	0.83	306
Macro avg	0.84	0.68	0.72	306
Weighted avg	0.83	0.83	0.82	306

After SVM, finally the ANN has been applied to RCC dataset. Firstly, dataset was again splitted into 70%-30% for training and test. As a result of training and testing with Neural Networks, an accuracy rate of 89.22% was obtained. After classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.43 and Table 4.44.

Confusion matrix of classification result:

Table 4.43: Confusion Matrix of Artificial Neural Network.

		Predicted Values		
		KICH	KIRC	KIRP
Actual Values	KICH	16	7	3
	KIRC	6	76	9
	KIRP	2	6	181

Classification Report:

Table 4.44: Classification Report of Artificial Neural Network.

	Precision	Recall	F-measure	Support
0	0.67	0.62	0.64	26
1	0.85	0.84	0.84	91
2	0.94	0.96	0.95	189
Micro avg	0.89	0.89	0.89	306
Macro avg	0.82	0.80	0.81	306
Weighted avg	0.89	0.89	0.89	306

The comparison of the 4 classical classification algorithms applied before deep learning methods is given in the Table 4.45.

Table 4.45: Comparison of Classification Algorithms for RCC.

Classifier	MAE	RMSE	Classification Accuracy
Decision Tree	0.11	0.39	90.52%
Random Forest	0.09	0.33	91.83%
Linear SVM	0.13	0.41	87.91%
Polynomial SVM	0.18	0.49	84.64%
RBF SVM	0.19	0.48	83.01%
Artificial Neural Networks	0.12	0.39	89.22%

When the results were compared, linear kernel gave the best results among SVM kernels, while Random Forest gave the best results in classical classification algorithms. After this stage, deep learning methods were applied and the results were compared.

A sequential model was used for the classification. RELU has been used as activation function. 7 different optimizers with this activation function applied on the dataset. Then results are compared.

SGD Optimizer: SGD optimizer tested on dataset and the following results obtained; training accuracy: 90.57% and test accuracy: 94.23%. Figures shown below during training and testing.

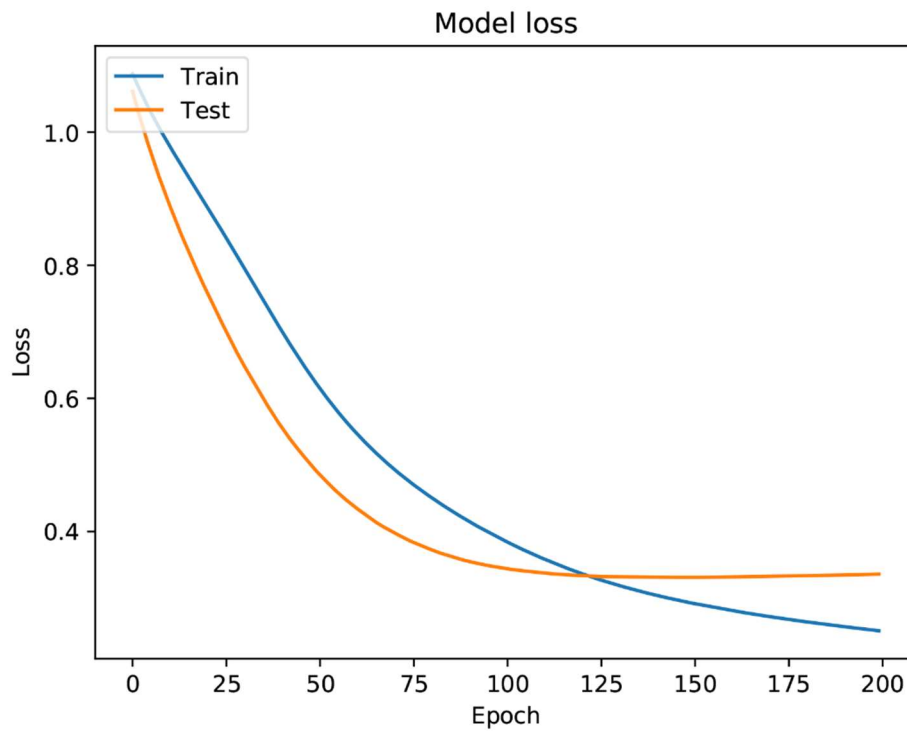


Figure 4.15: Model loss during training and testing for SGD optimizer.

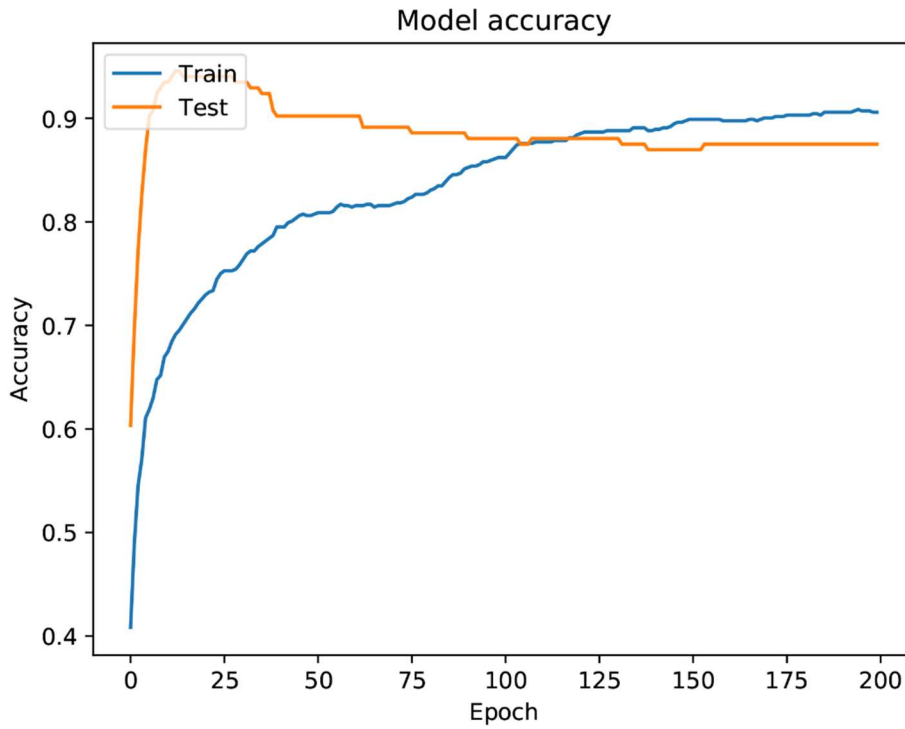


Figure 4.16: Model accuracy during training and testing for SGD optimizer.

After classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.46 and Table 4.47.

Confusion matrix of classification result:

Table 4.46: Confusion Matrix of SGD Optimizer.

		Predicted Values		
		KICH	KIRC	KIRP
Actual Values	KICH	20	3	4
	KIRC	8	76	13
	KIRP	8	0	174

Classification Report:

Table 4.47: Classification Report of SGD Optimizer.

	Precision	Recall	F-measure	Support
0	1.00	0.11	0.20	27
1	0.96	0.78	0.86	97
2	0.91	0.96	0.93	182
Micro avg	0.93	0.83	0.87	306
Macro avg	0.96	0.62	0.67	306
Weighted avg	0.94	0.83	0.85	306

RMSProp Optimizer: RMSProp optimizer tested on dataset and the following results obtained; training accuracy: 100% and test accuracy: 95.19%. Figures shown below during training and testing.

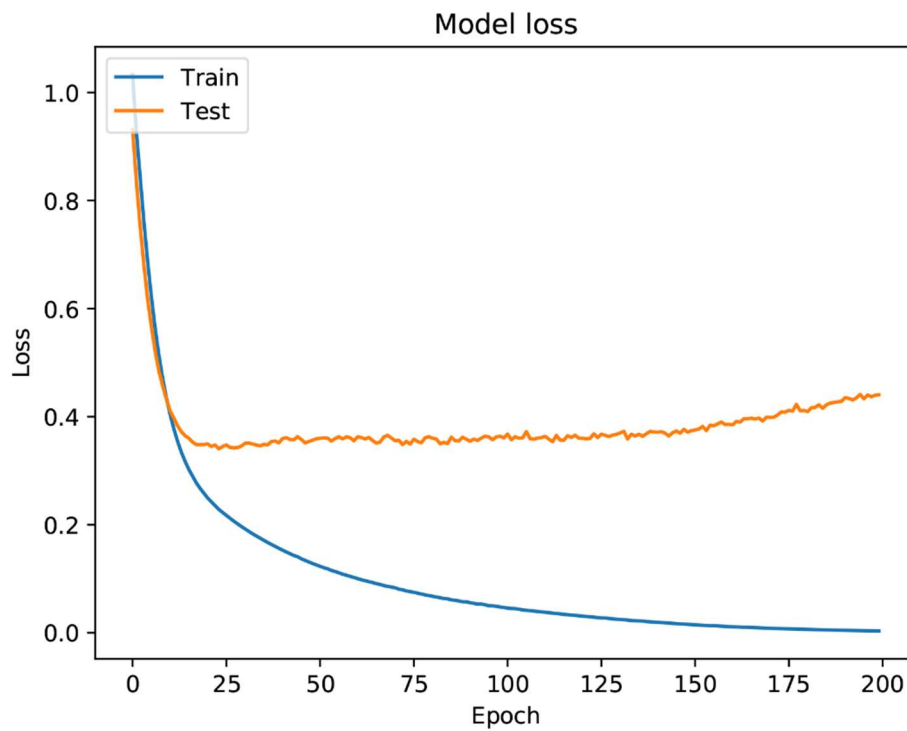


Figure 4.17: Model loss during training and testing for RMSProp optimizer.

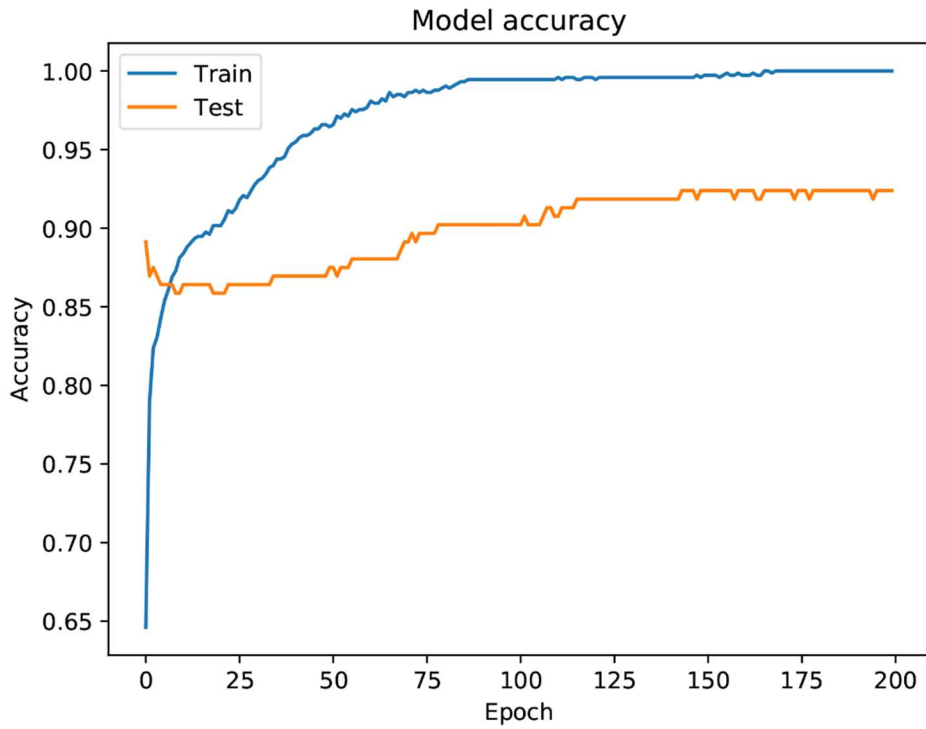


Figure 4.18: Model accuracy during training and testing for RMSProp optimizer.

After classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.48 and Table 4.49.

Confusion matrix of classification result:

Table 4.48: Confusion Matrix of RMSProp Optimizer.

		Predicted Values		
		KICH	KIRC	KIRP
Actual Values	KICH	22	1	4
	KIRC	4	86	7
	KIRP	1	6	175

Classification Report:

Table 4.49: Classification Report of RMSProp Optimizer.

	Precision	Recall	F-measure	Support
0	0.91	0.74	0.82	27
1	0.92	0.89	0.91	97
2	0.94	0.96	0.95	182
Micro avg	0.93	0.92	0.93	306
Macro avg	0.92	0.86	0.89	306
Weighted avg	0.92	0.92	0.92	306

Adagrad Optimizer: Adagrad optimizer tested on dataset and the following results obtained; training accuracy: 99.45 % and test accuracy: 95.19%. Figures shown below during training and testing.

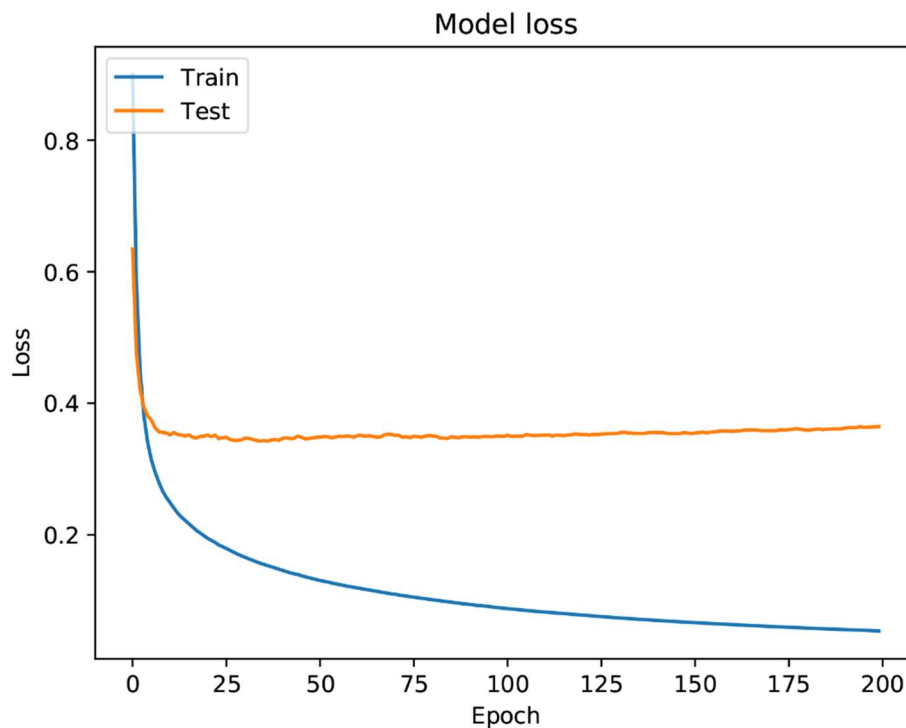


Figure 4.19: Model loss during training and testing for Adagrad optimizer.

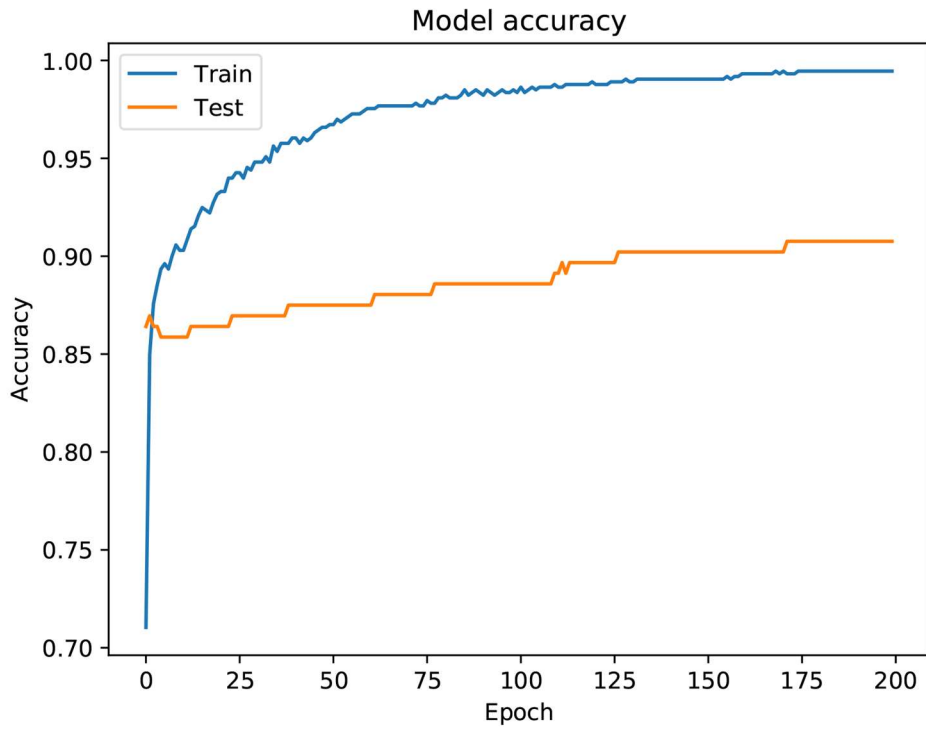


Figure 4.20: Model accuracy during training and testing for Adagrad optimizer.

After classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.50 and Table 4.51.

Confusion matrix of classification result:

Table 4.50: Confusion Matrix of Adagrad Optimizer.

		Predicted Values		
		KICH	KIRC	KIRP
Actual Values	KICH	23	1	3
	KIRC	2	84	11
	KIRP	1	3	178

Classification Report:

Table 4.51: Classification Report of Adagrad Optimizer.

	Precision	Recall	F-measure	Support
0	0.90	0.67	0.77	27
1	0.95	0.87	0.91	97
2	0.93	0.98	0.95	182
Micro avg	0.93	0.92	0.92	306
Macro avg	0.93	0.84	0.88	306
Weighted avg	0.93	0.92	0.92	306

Adadelta Optimizer: Adadelta optimizer tested on dataset and the following results obtained; training accuracy: 100% and test accuracy: 96.15%. Figures shown below during training and testing.

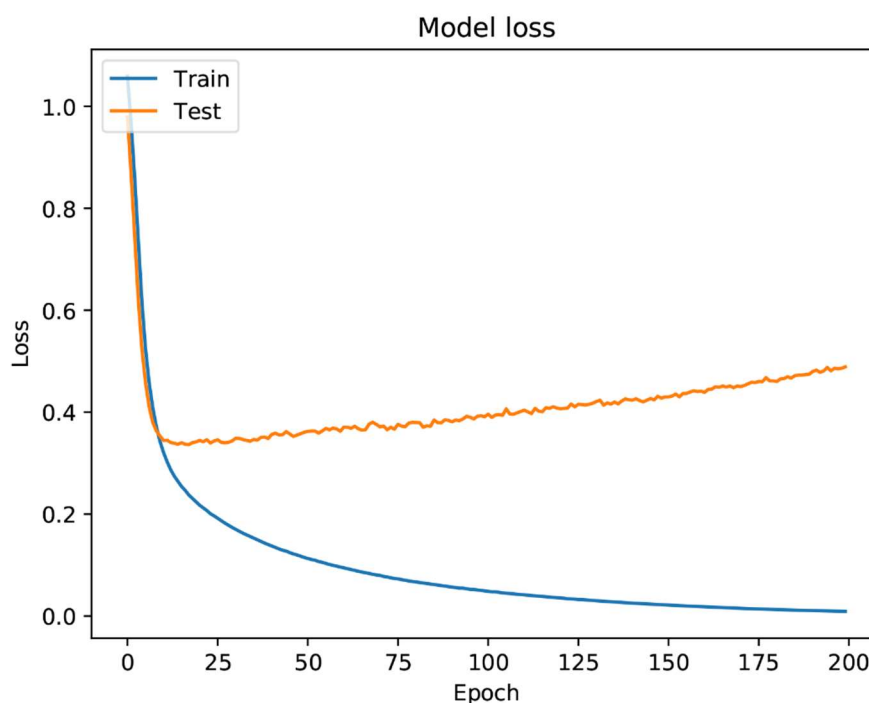


Figure 4.21: Model loss during training and testing for Adadelta optimizer.

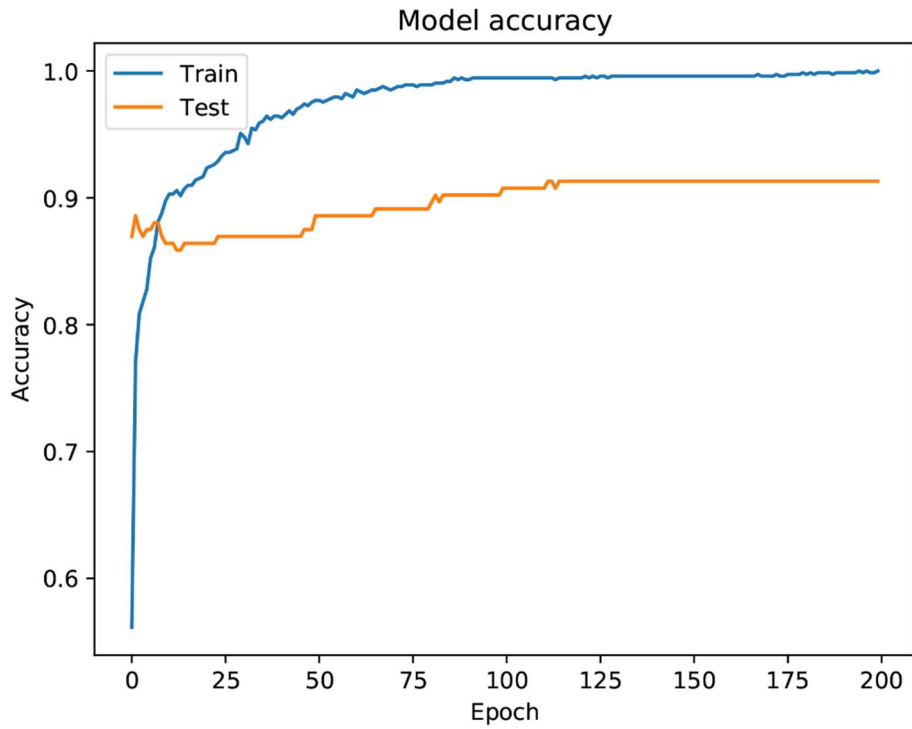


Figure 4.22: Model accuracy during training and testing for Adadelata optimizer.

After classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.52 and Table 4.53.

Confusion matrix of classification result:

Table 4.52: Confusion Matrix of Adadelata Optimizer.

		Predicted Values		
		KICH	KIRC	KIRP
Actual Values	KICH	24	1	2
	KIRC	4	85	8
	KIRP	1	3	178

Classification Report:

Table 4.53: Classification Report of Adadelata Optimizer.

	Precision	Recall	F-measure	Support
0	0.90	0.70	0.79	27
1	0.96	0.88	0.91	97
2	0.95	0.98	0.96	182
Micro avg	0.95	0.92	0.93	306
Macro avg	0.94	0.85	0.89	306
Weighted avg	0.95	0.92	0.93	306

Adam Optimizer: Adam optimizer tested on dataset and the following results obtained; training accuracy: 99.59% and test accuracy: 96.15%. Figures shown below during training and testing.

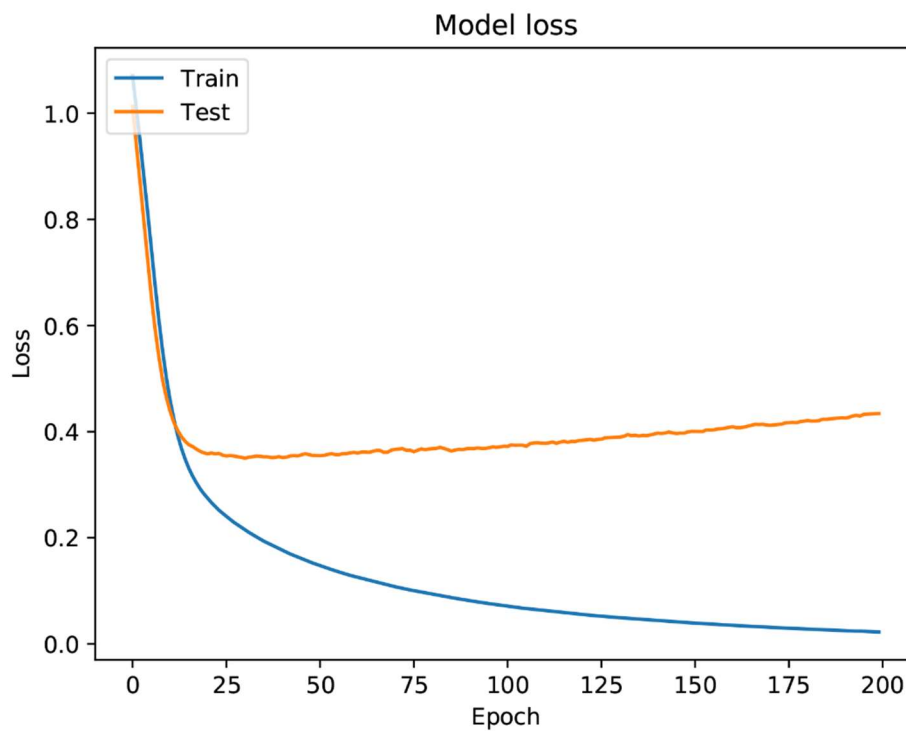


Figure 4.23: Model loss during training and testing for Adam optimizer.

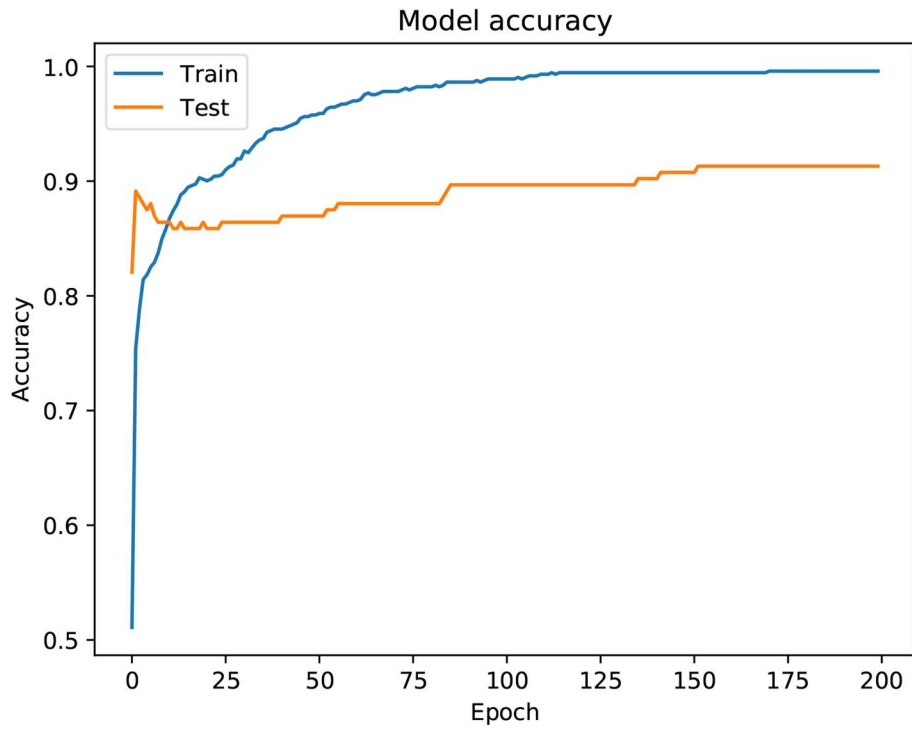


Figure 4.24: Model accuracy during training and testing for Adam optimizer.

After classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.54 and Table 4.55.

Confusion matrix of classification result:

Table 4.54: Confusion Matrix of Adam Optimizer.

		Predicted Values		
		KICH	KIRC	KIRP
Actual Values	KICH	19	3	5
	KIRC	1	91	5
	KIRP	5	6	171

Classification Report:

Table 4.55: Classification Report of Adam Optimizer.

	Precision	Recall	F-measure	Support
0	0.85	0.63	0.72	27
1	0.91	0.94	0.92	97
2	0.94	0.94	0.94	182
Micro avg	0.93	0.91	0.92	306
Macro avg	0.90	0.84	0.86	306
Weighted avg	0.93	0.91	0.92	306

Adamax Optimizer: Adamax optimizer tested on dataset and the following results obtained; training accuracy: 99.45 % and test accuracy: 96.15%. Figures shown below during training and testing.

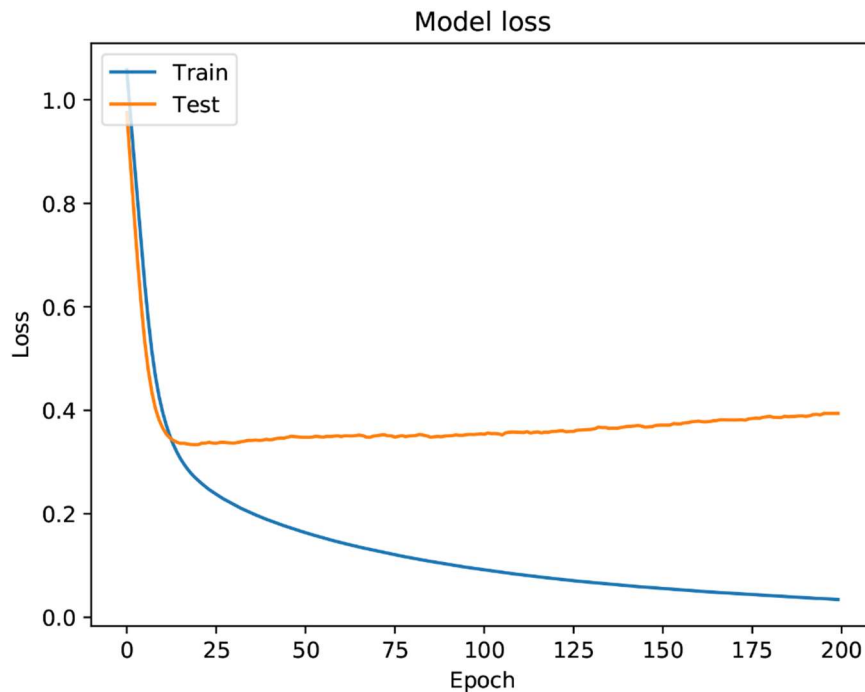


Figure 4.25: Model loss during training and testing for Adamax optimizer.

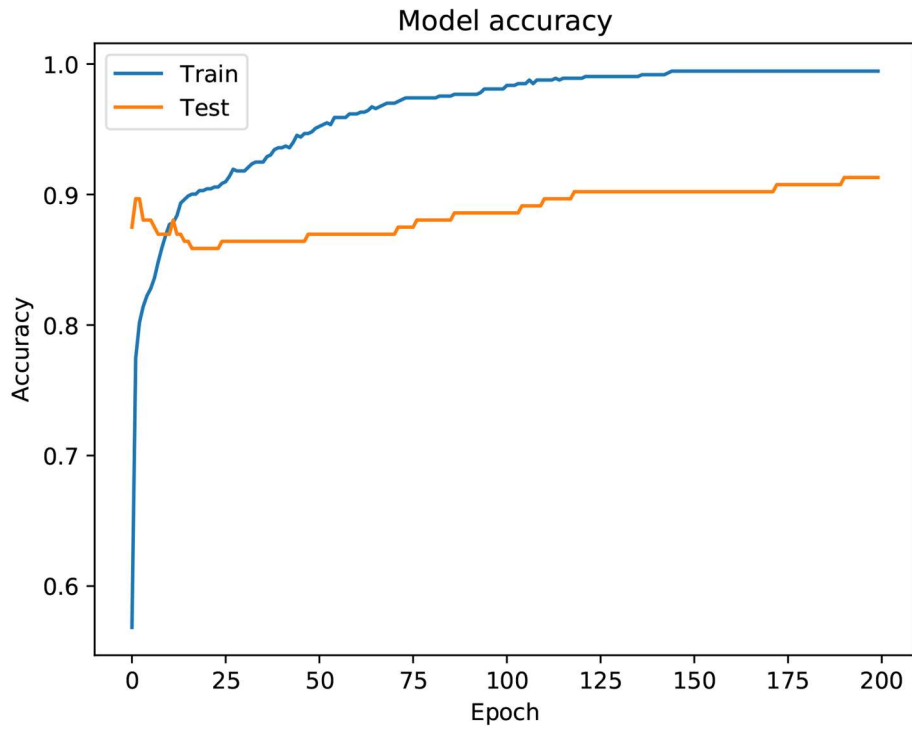


Figure 4.26: Model accuracy during training and testing for Adamax optimizer.

After classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.56 and Table 4.57.

Confusion matrix of classification result:

Table 4.56: Confusion Matrix of Adamax Optimizer.

		Predicted Values		
		KICH	KIRC	KIRP
Actual Values	KICH	23	2	2
	KIRC	8	80	9
	KIRP	10	1	171

Classification Report:

Table 4.57: Classification Report of Adamax Optimizer.

	Precision	Recall	F-measure	Support
0	0.88	0.52	0.65	27
1	0.96	0.82	0.89	97
2	0.94	0.94	0.94	182
Micro avg	0.94	0.87	0.90	306
Macro avg	0.93	0.76	0.83	306
Weighted avg	0.94	0.87	0.90	306

Nadam Optimizer: Nadam optimizer tested on dataset and the following results obtained; training accuracy: 99.86 % and test accuracy: 95.19%. Figures shown below during training and testing.

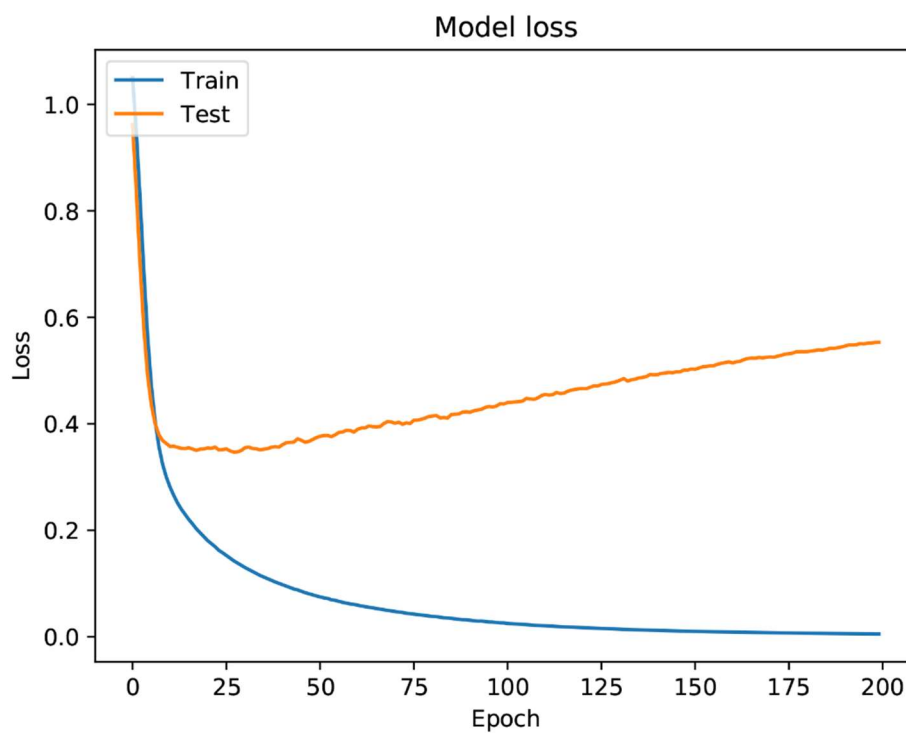


Figure 4.27: Model loss during training and testing for Nadam optimizer.

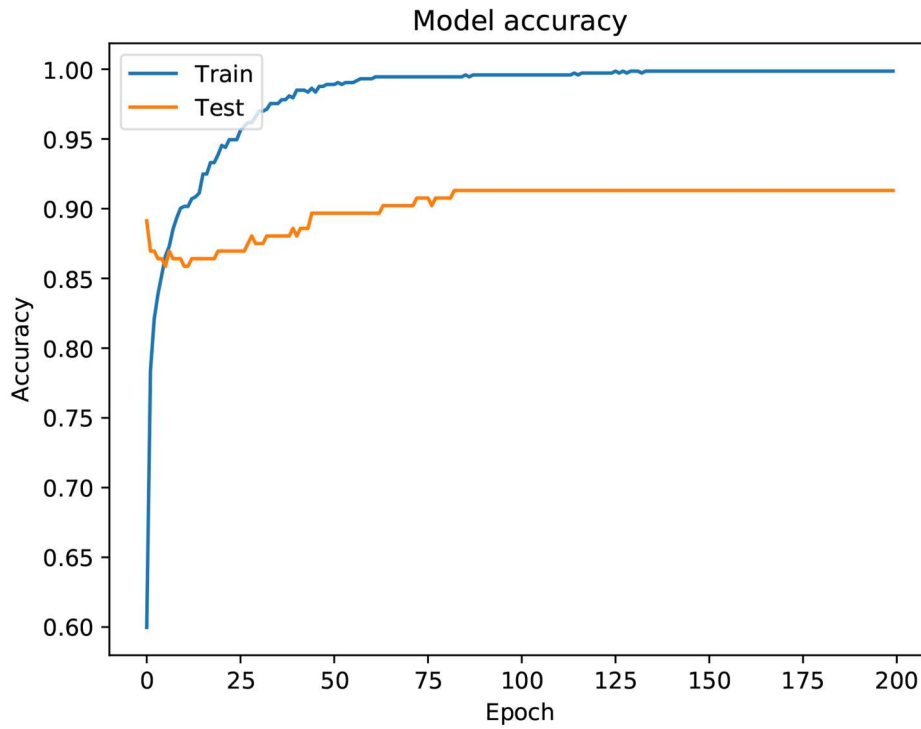


Figure 4.28: Model accuracy during training and testing for Nadam optimizer.

After classification, a confusion matrix was created and precision, recall, f-measure and support values were calculated. Additionally, micro, macro and weighted average also calculated. The classification report is shown in Table 4.58 and Table 4.59.

Confusion matrix of classification result:

Table 4.58: Confusion Matrix of Nadam Optimizer.

		Predicted Values		
		KICH	KIRC	KIRP
Actual Values	KICH	24	1	2
	KIRC	1	87	8
	KIRP	1	7	174

Classification Report:

Table 4.59: Classification Report of Nadam Optimizer.

	Precision	Recall	F-measure	Support
0	0.92	0.81	0.86	27
1	0.92	0.90	0.91	97
2	0.95	0.96	0.95	182
Micro avg	0.93	0.92	0.93	306
Macro avg	0.93	0.89	0.91	306
Weighted avg	0.93	0.92	0.92	306

The training and test results with seven different optimizers and their graphs are given above. Each optimizer applied to Renal Cell Cancer dataset and different results were obtained in each. Here it is observed that the optimizers affect the success of the model. When the results are examined, it is seen that the best result in model training is provided by Adadelta and RMSProp optimizers. The highest success was achieved by Adadelta, Adamax and Adam optimizers in testing. The results are combined in the Table 4.60.

Table 4.60: Comparison of results with different optimizers for RCC.

Optimizer	MAE	RMSE	Training Accuracy	Test Accuracy
SGD	0.17	0.25	90.57%	94.23%
RMSProp	0.07	0.19	100%	95.19%
Adagrad	0.08	0.19	99.45%	95.19%
Adadelat	0.07	0.19	100%	96.15%
Adam	0.08	0.21	99.59%	96.15%
Adamax	0.10	0.21	99.45%	96.15%
Nadam	0.07	0.20	99.86%	95.15%

The results obtained by classical methods were compared with the deep learning model. In the comparison, linear SVM with the best results from SVM and Adadelat with the best results were used in the deep learning model. The comparison based on the test results is shown in Table 4.61.

Table 4.61: Comparison of results for RCC.

Classifier	MAE	RMSE	Result
Decision Tree	0.11	0.39	90.52%
Random Forest	0.09	0.33	91.83%
SVM(Linear)	0.13	0.41	87.91%
Artificial Neural Networks	0.12	0.39	89.22%
Deep Learning Model(Adadelat)	0.07	0.19	96.15%

When the results were compared it was observed that the results obtained with the deep learning model were better. RNA-Seq Renal Cell Cancer dataset were analyzed and 96.15% accuracy of the disease class was determined.

When the results given above were evaluated, deep learning methods gave the best results among the classification algorithm applied after gene selection on RNA-Seq data. Zararsiz et al. (2017) in the study conducted by SVM RCC data with 93.5% accuracy rate in the classification of lung cancer data was 94.8% success was achieved. As a result of the studies in this thesis, developed deep learning model provided an accuracy of 96.15% on RCC data and 95.54% on lung cancer. Thus, it has been observed that deep learning model gives better results compared to classical methods.

Table 4.62: Comparison with average values.

Dataset	Classifier	MAE	RMSE	Accuracy
Lung Cancer	Decision Tree	0.09	0.29	91.74%
	Random Forest	0.08	0.25	93.51%
	SVM Average	0.17	0.31	89.67%
	Artificial Neural Networks	0.10	0.32	89.97%
	Deep Learning Model Average	0.20	0.22	93.43%
Renal Cell Cancer	Decision Tree	0.11	0.39	90.52%
	Random Forest	0.09	0.33	91.83%
	SVM Average	0.16	0.46	85.18%
	Artificial Neural Networks	0.12	0.39	89.22%
	Deep Learning Model Average	0.09	0.20	95.45%

There are three different kernel types of SVM which is used in this study. And seven different optimizer algorithm used for deep learning model. Table 4.62 tabulates the results of classification methods with average values of SVM and deep learning methods.

Table 4.63: All Results.

Dataset	Classifier	MAE	RMSE	Accuracy
Lung Cancer	Decision Tree	0.09	0.29	91.74%
	Random Forest	0.08	0.25	93.51%
	SVM (RBF)	0.07	0.28	92.04%
	Artificial Neural Networks	0.10	0.32	89.97%
	Deep Learning Model (AdaDelta)	0.07	0.21	95.54%
Renal Cell Cancer	Decision Tree	0.11	0.39	90.52%
	Random Forest	0.09	0.33	91.83%
	SVM (Linear)	0.13	0.41	87.91%
	Artificial Neural Networks	0.12	0.39	89.22%
	Deep Learning Model (AdaDelta)	0.07	0.19	96.15%

All results, obtained in this study, are presented in Table 4.63. It can be considered as a summary for comparison of all results.

CHAPTER 5

5. CONCLUSION

One of the most advanced techniques used to store gene expressions is RNA-Seq technology. Thanks to this technology, the gene sequences of the patients can be kept together and used digitally. In particular, these data have a significant role in the diagnosis of cancer-related genes. Using the previous data, a diagnosis can be made for the next patient or cancer type can be detected for a patient. This classification and diagnosis can be very difficult because of high-dimensionality of RNA-Seq dataset.

In this thesis, RNA-Seq datasets belonging to two different types of cancer were classified with using 5 different methods. There are 1,128 samples for the lung cancer and 1,020 samples of renal cell cancer. First, the 20,531 genes for each sample was reduced to 50 genes using wrapper methods. A wrapper method model is applied on genes and most important 50 genes selected for cancer classification. Then, five different classification methods were applied to the two datasets separately and results were compared.

Lung cancer samples divided into two classes; 576 belong to LUAD and 552 belong to LUSC. 70% of the lung cancer dataset is reserved for training, while 30% is reserved for testing. The created deep learning model with AdaDelta optimization algorithm gave the best test result with 95.54% accuracy rate. It has been observed that the method of deep learning gives better results when compared with other methods. It correctly estimated 324 of the 339 samples used for the test.

Renal Cell cancer includes 3 different classes: KICH, KIRC and KIRP. 70% of the RCC dataset is reserved for training and 30% for testing. As a result of the tests, it was observed that the model created by the deep learning method gave better results than the other methods. While the deep learning model provided a success rate of 96.15%, the second closest result was obtained by random forests method with 91.83%. 306 samples allocated for the test and 295 of them were correctly estimated and 11 incorrectly classified.

The models developed for cancer classification were applied on Lung Cancer and Renal Cell Cancer data and a success rate of at least 85% was achieved in all methods. Decision Tree, Random Forests, SVM, ANN and Deep Learning methods were used for classification. There are different kernel types of SVM and in this study Polynomial, Linear and RBF were used for classification. In general, deep learning models gave the better result. In this study, seven different deep learning model used with different optimization algorithms. SGD, RMSProp, Adagrad, AdaDelta, Adam, Adamax and Nadam used for the optimization. AdaDelta optimizer gave the best results when compared to other optimizers. When optimization algorithms compared, adaptive algorithms are better than sophisticated methods. Because adaptive algorithms find the learning rate by themselves and they are very dynamic. AdaDelta is also one part of adaptive algorithms. In this study, AdaDelta optimizer is found to be the best one in deep learning methods.

The fact that the model created by using deep learning method gives better results than the others shows the success of this method in the studies to be done on RNA-Seq data. Training and test success rates can also be increased by using more datasets. Thus, the resulting reliability of the obtained system can be increased. The models created due to the problem of accessing the dataset could be applied to these two cancer data. Models developed for dual classification with lung cancer alone and for multiple classification for renal cell cancer can be applied on other datasets.

In this thesis, RNA-Seq datasets have been successfully used to make the decision support system for lung cancer and renal cell cancer classification. In conclusion, the cancer classification methods, which proposed in this study, gave better results than previous studies. It is shown that these methods can be used for further analysis of RNA-Seq data for specific cancer types.

REFERENCES

A. Krizhevsky, Sutskever, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," Proc. Neural Information and Processing Systems, pp. 1097-1105, 2012.

Advantages of paired-end and single-read sequencing. Retrieved 10 05, 2019 from <https://emea.illumina.com/science/technology/next-generation-sequencing/paired-end-vs-single-read-sequencing.html>

Afshari CA. Perspective: microarray technology, seeing more than spots. *Endocrinology*. 2002; 143(6): 1983-1989.

Autoencoder, 15 05, 2019 Retrived from <https://www.guru99.com/autoencoder-deep-learning.html>

Baluja S. and Davies S. (1997). Using Optimal Dependency-Trees for Combinatorial Optimization: Learning the Structure of the Search Space, Proceedings of the Fourteenth International Conference on Machine Learning.

Baris Senliol, Gokhan Gulgezen, Lei Yu, Zehra Cataltepe, (2008)." Fast Correlation Based Filter (FCBF) with a Different Search Strategy". *Computer and Information Science*, 23rd international symposium.

Bhavsar, Himani P. and Mahesh H. Panchal. "A Review on Support Vector Machine for Data Classification." (2012).

Bouchardy, C., Rapiti, E., Blagojevic, S., Vlastos, A.-T., & Vlastos, G. (2007). Older Female Cancer Patients: Importance, Causes, and Consequences of Undertreatment. *Journal of Clinical Oncology*, **25(14)**, 1858–1869.

Br I, Siklsi D, Szab J, Benczr AA (2009) Linked latent dirichlet allocation in web spam filtering. DOI 10.1145/1531914.1531922

Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A., & Jemal, A. (2018). Global Cancer Statistics 2018: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries.

- Breiman L (1996) Bagging predictors. *Machine learning* 24(2):123–140
- Breiman L.: Random forests. *Mach. Learn.* 45(1), 5–32 (2001)
- Brown PO, Botstein D. Exploring the new world of the genome with DNA microarrays. *Nat Genet.* 1999; 21(1 Suppl): 33-37.
- C. Cortes and V. Vapnik, “Support-Vector Networks,” *Mach. Learn.*, vol. 20, pp. 273-297, 1995.
- C. Stergiou, D. Siganos, *Neural Networks* 1996, 1996.
- C. Szegedy, W. Liu, Y. Jia, and P. Sermanet, “Going deeper with convolutions,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, pp. 1-9, 2015.
- Chen, H., Yang, B., Liu, J., & Liu, D. (2011). A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis. *Expert Systems with Applications*, 38, 9014-9022.
- Chow M.L and Liu C. (1968). Approximating discrete probability distribution with dependency tree, *IEEE Transactions on Information Theory* 14:462-367.
- CNN, 15 05, 2019 Retrived from <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- D. H. Ackley, G. E. Hinton, and T.J. Sejnowski, “A learning algorithm for boltzmann machines,” *Cognitive Sci.*, vol. 9, pp. 147-169, 1985.
- D. O. Hebb, “The organization of behavior,” *J. Appl. Behav. Anal.*, vol. 25, pp. 575–577, 1949.
- Deep Belief Network for Spectral–Spatial Classification of Hyperspectral Remote Sensor Data - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Architecture-of-a-deep-belief-network-DBN_fig1_330246603 [accessed 2 Nov, 2019]
- Dong K, Zhao H, Wan X, Tong T. 2015. NBLDA: negative binomial linear discriminant analysis for RNA-Seq data. *BMC Bioinformatics* 17:369 DOI 10.1186/s12859-016-1208-1.
- Du, Xuedan & Cai, Yinghao & Wang, Shuo & Zhang, Leijie. (2016). Overview of deep learning. 159-164. 10.1109/YAC.2016.7804882.

F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386-408, 1958.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10), 906-914.

Futreal, P. A., Coin, L., Marshall, M., Down, T., Hubbard, T., Wooster, R., Stratton, M. R. (2004). A census of human cancer genes. *Nature Reviews Cancer*, 4(3), 177–183. doi:10.1038/nrc1299

Genome Sequencing: Defining Your Experiment. Retrieved 10 05,2019 from <https://systemsbiology.columbia.edu/genome-sequencing-defining-your-experiment>

Girish Chandrashekar, Ferat Sahin, (2014). "A survey on feature selection methods". *Computers and Electrical Engineering*

Goyal R, Gersbach E, Yang XJ, Rohan SM. 2013. Differential diagnosis of renal tumors with clear cytoplasm. Clinical relevance of renal tumor subclassification in the era of targeted therapies and personalized medicine. *Archives of Pathology & Laboratory Medicine* 137:467-480 DOI 10.5858/arpa.2012-0085-RA.

Gyorgy AB, Walker J, Wingo D, Eidelman O, Pollard HB, Molnar A, Agoston DV . Reverse phase protein microarray technology in traumatic brain injury. *J Neurosci Methods*. 2010; 192(1): 96-101.

Hall DA, Ptacek J, Snyder M. Protein microarray technology. *Mech Ageing Dev*. 2007; 128(1): 161-167.

Hall, M. A. & Smith, L. A. (1998). Practical feature subset selection for machine learning. In C. McDonald (Ed.), *Computer Science '98 Proceedings of the 21st Australasian Computer Science Conference ACSC'98*, Perth.

Hall, M.A.: *Correlation-based Feature Selection for Machine Learning*. PhD thesis, University of Waikato, Hamilton, New Zealand (1999).

Han, B., Li, L., Chen, Y., Zhu, L., & Dai, Q. (2011). A two step method to identify clinical outcome relevant genes with microarray data. *Journal of Biomedical Informatics*, 44, 229-238.

Han, Y., Gao, S., Muegge, K., Zhang, W., & Zhou, B. (2015). Advanced Applications of RNA Sequencing and Challenges. *Bioinformatics and Biology Insights*, 9(Suppl 1), 29–46.

Hong B, Lui VW, Hui EP, Lu Y, Leung HS, Wong EYL, Cheng SH, Ng MHL, Mills GB, Chan ATC. Reverse phase protein array identifies novel anti-invasion mechanisms of YC-1. *Biochem Pharmacol.* 2010; 79(6): 842-852.

Huang, J., Fang, H., & Fan, X. (2010). Decision forest for classification of gene expression data”. *Computers in Biology and Medicine*, 40, 98-704.

Hyndman, Rob J.; Koehler, Anne B. (2006). "Another look at measures of forecast accuracy". *International Journal of Forecasting*. 22 (4): 679–688. CiteSeerX 10.1.1.154.9771. doi:10.1016/j.ijforecast.2006.03.001.

I. Guyon, A. Elisseeff, An introduction to variable and feature selection, *J Mach Learn Res*, 3 (2003), pp. 1157-1182

J. Han and M. Kamber, *Data mining: concepts and techniques*, 2nd ed. Amsterdam; Boston: San Francisco, CA: Elsevier, 2006.

J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” *P. Natl. Acad. Sci. USA*, vol. 79, pp. 2554-8, 1982.

J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 526–532, Cambridge, MA, USA, 2000. MIT Press.

Jeremy M. Berg, John L. Tymoczko, Lubert Stryer. 2011. *Biochemistry*. 7th Edition. Palgrave MacMillan.

K. Fukushima, S. Miyake, and T. Ito, “Neocognitron: A neural network model for a mechanism of visual pattern recognition,” *IEEE Trans. Syst., Man, Cybern.* vol. 1, pp. 826-834, 1983

K. Hechenbichler and K. Schliep, “Weighted k-nearest-neighbor techniques and ordinal classification,” *Institut für Statistik Sonderforschungsbereich*, 386, 2004.

K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv: 1409.1556*, 2014

Kamangar, F., Dores, G. M., & Anderson, W. F. (2006). Patterns of Cancer Incidence, Mortality, and Prevalence Across Five Continents: Defining Priorities to Reduce Cancer Disparities in Different Geographic Regions of the World. *Journal of Clinical Oncology*, 24(14), 2137–2150.

KDNuggets, 15 05, 2019 Retrived from <https://www.kdnuggets.com/2017/12/deep-learning-made-easy-deep-cognition.html>

Kodadek T. Protein microarrays: prospects and problems. *Chem Biol.* 2001; 8(2): 105-115.

Komal A K, Lalita B, “A review on classification using decision tree,” *International Journal of Computing and Technology*, 2, 42-46 (2015).

Kuo WP, Whipple ME, Jenssen TK, Todd R, Epstein JB, Ohno-Machado L, Sonis ST, Park PJ . Microarrays and clinical dentistry. *J Am Dent Assoc.* 2003; 134(4): 456-462.

Law CW, Chen Y, Shi W, Smyth GK. 2014. Voom: precision weights unlock linear model analysis tools for RNA-Seq read counts. *Genome Biology* 15:R29 DOI 10.1186/gb-2014-15-2-r29.

Lee, J. W., Lee, J. B., Park, M., & Song, S. H. (2005). An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics & Data Analysis*, 48(4), 869-885.

Leung YF, Cavalieri D. Fundamentals of cDNA microarray data analysis. *Trends Genet.* 2003; 19(11): 649-659.

Liangxiao Jiang, Harry Zhang, and Zhihua Cai, “Dynamic k-nearest-neighbor naive bayes with attribute weighted,” presented at the International Conference on Fuzzy Systems and Knowledge Discovery, 2006, pp. 365–368.

Lior Rokach and Oded Maimon, “Top-Down Induction of Decision Trees Classifiers –A Survey” *IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS: PART C, VOL. 1, NO. 11, November 2002*, pp 1-12.

Liu Y, Morley M, Brandimarto J, et al. RNA-Seq identifies novel myocardial gene expression signatures of heart failure. *Genomics.* 2015; 105:83–89.

Lu, J., Getz, G., Miska, A. E., Alvarez-Saavedra, J., Lamb, J., Peck, D., Golub, R. T. (2005). MicroRNA expression profiles classify human cancers. *Nature*, 435, 834-838.

- M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. F. Chen, C. Citro, et al., “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” arXiv preprint arXiv: 1603.04467, 2016.
- M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” European Conference on Computer Vision, Zurich, pp. 818- 833, 2014.
- M. E. Syed, “Attribute weighting in k-nearest neighbor classification,” University of Tampere, 2014.
- Mccullochw S, PittsW (1943) A logical calculus of the ideas immanent in nervous activity. *BullMath Biophys* 10(5):115–133.
- Mejia-Lavalle M, Sucar E, Arroyo G (2006) Feature selection with a perceptron neural net. In: *Proceedings of the international workshop on feature selection for data mining*, pp 131–135.
- Ozsolak, F., & Milos, P. M. (2011). RNA sequencing: advances, challenges and opportunities. *Nature Reviews Genetics*, 12(2), 87–98.
- P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.
- P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 399–406, San Jose, CA, 1992. AAAI Press.
- Peng, Y. (2006). A novel ensemble machine learning for robust microarray data classification. *Computers in Biology and Medicine*, 36(6), 553-573.
- Perez-Diez A., Morgun A., Shulzhenko N. (2007) Microarrays for Cancer Diagnosis and Classification. In: *Mocellin S. (eds) Microarray Technology and Cancer Gene Profiling. Advances in Experimental Medicine and Biology*, vol 593. Springer, New York, NY
- R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, et al., “Theano: A Python framework for fast computation of mathematical expressions,” arXiv preprint arXiv: 1605.02688, 2016.
- R. Collobert, S. Bengio, and J. Mariéthoz, “Torch: a modular machine learning software library,” *Idiap*, No. EPFL-REPORT-82802, 2002.

Rachel Nall MSN CRNA. "What to know about cancer." Medical News Today. MediLexicon, Intl., 12 Nov. 2018. Web. 2 Sep. 2019.

Rakotomamonjy A (2003) Variable selection using SVM-based criteria. *J Mach Learn Res* 3:1357–1370.

Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, Smyth G. 2015. Limma Powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research* 43(7):e47 DOI 10.1093/nar/gkv007.

Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, Smyth G. 2015. Limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research* 43(7):e47 DOI 10.1093/nar/gkv007.

RNN, 15 05, 2019 Retrived from <https://towardsdatascience.com/machine-learning-recurrent-neural-networks-and-long-short-term-memory-lstm-python-keras-example-86001ceaaebc>

Russell S. and Norvig P. (1995). *Artificial Intelligence, A Modern Approach*, Prentice Hall, New Jersey.

S. B. Imandoust and M. Bolandraftar, "Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background," *Int. J. Eng. Res. Appl.*, vol. 3, no. 5, pp. 605–610, 2013.

S. Haykin, *Neural Networks and Learning Machines*, third ed., McMaster University Hamilton, Ontario, Canada, 2009.

S. Sasikala, S. Appavu alias Balamurugan, S. Geetha, (2014). "Multi Filtration Feature Selection (MFFS) to improve discriminatory ability in clinical data set". *Applied Computing and Informatics*.

Saleem M, Shanmukha A, Ngonga Ngomo AC, Almeida JS, Decker HF, Deus HF. 2013. Linked cancer genome atlas database. In: *I-SEMANTICS '13-Proceedings of the 9th international conference on semantic systems: 04-06 September 2013-Graz*. 129-134.

Schena M, Heller RA, Theriault TP, Konrad K, Lachenmeier E, Davis RW. *Microarrays: biotechnology's discovery platform for functional genomics. Trends Biotechnol.* 1998; 16(7): 301-306.

Schuster, S. C. (2008). Next-generation sequencing transforms today's biology. *Nature Methods*, 5(1), 16–18.

Statnikov, A., Aliferis, C. F., Tsamardinos, I., Hardin, D., & Levy, S. (2005). A comprehensive evaluation of multiclassification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(5), 631-643.

Tillib SV, Mirzabekov AD. Advances in the analysis of DNA sequence variations using oligonucleotide microchip technology. *Curr Opin Biotechnol*. 2001; 12(1): 53-58.

Tran, D. H., Ho, T. B., Pham, T. H., & Satou, K. (2011). MicroRNA Expression Profiles for Classification and Analysis of Tumor Samples. *IEICE Trans. Inf&Syst.*, 94(3).

Trichopoulos, Dimitrios, et al. "What Causes Cancer?" *Scientific American*, vol. 275, no. 3, 1996, pp. 80–87. JSTOR, www.jstor.org/stable/24993351.

Urbanowicz, Ryan & Meeker, Melissa & LaCava, William & Olson, Randal & Moore, Jason. (2017). Relief-Based Feature Selection: Introduction and Review. *Journal of Biomedical Informatics*. 85. 10.1016/j.jbi.2018.07.014.

V. Vapnik. *The Nature of Statistical Learning Theory*. NY: Springer-Verlag. 1995.

W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115-133, 1943.

Wang C, Gong B, Bushel PR, et al. The concordance between RNA-seq and microarray data depends on chemical treatment and transcript abundance. *Nat Biotechnol*. 2014; 32:926–932.

Wang, Z., Gerstein, M., & Snyder, M. (2009). RNA-seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1), 57–63.

Wilhelm BT, Landry JR. RNA-Seq—quantitative measurement of expression through massively parallel RNA sequencing. *Methods*. 2009; 48:249–57.

Witten DM. 2011. Classification and clustering of sequencing data using a Poisson model. *The Annals of Applied Statistics* 5(4):24932518 DOI 10.1214/11 AOAS493.

Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, et al., “Caffe: Convolutional architecture for fast feature embedding,” Proceedings of the 22nd ACM international conference on Multimedia, Orlando, pp. 675-678, 2014.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, pp. 2278-2324, 1998

Yvan Saeys, Inak Inza, Pedro Larranaga, (2007). “A review of Feature Selection techniques in bioinformatics”. Bioinformatics, Oxford University press

Zararsiz G, Goksuluk D, Klaus B, Korkmaz S, Eldem V, Karabulut E, Ozturk A. 2017. voomDDA: discovery of diagnostic biomarkers and classification of RNA-seq data. PeerJ 5:e3890 <https://doi.org/10.7717/peerj.3890>

Zhang YH, Huang T, Chen L, et al. Identifying and analyzing different cancer subtypes using RNA-seq data of blood platelets. Oncotarget. 2017; 8(50):87494–87511. Published 2017 Sep 15. doi:10.18632/oncotarget.20903

Zhao, S., Zhang, Y., Gordon, W., Quan, J., Xi, H., Du, S., Zhang, B. (2015). Comparison of stranded and non-stranded RNA-seq transcriptome profiling and investigation of gene overlap. BMC Genomics, 16(1).