ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

Esra MAHSERECİ KARABULUT

INVESTIGATION OF DEEP LEARNING APPROACHES FOR BIOMEDICAL DATA CLASSIFICATION

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADANA, 2016

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

INVESTIGATION OF DEEP LEARNING APPROACHES FOR BIOMEDICAL DATA CLASSIFICATION

Esra MAHSERECİ KARABULUT

PhD THESIS

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of degree of the Doctor of Phylosophy by the board of jury on 10/08/2016.

…….....	……	…….....
Assoc. Prof. Dr. Turgay İBRİKÇİ	Prof. Dr. Hamza EROL	Assoc. Prof. Dr. Sami ARICA
SUPERVISOR	MEMBER	MEMBER
…….....	……	
Asst. Prof. Dr. İrem ERSÖZ KAYA	Asst. Prof. Dr. Lütfü SARIBULUT	
MEMBER	MEMBER	

This PhD Thesis is written at the Department of Institute of Natural And Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. M. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

This thesis was financially supported by Ç.U. academic resource foundation FDK-2015-4395

Not: The usage of the presented specific declerations, tables, figures, and photographs either in this thesis or in any other reference without citiation is subject to "The law of Arts and Intellectual Products" number of 5846 of Turkish Republic

ABSTRACT

PhD THESIS

INVESTIGATION OF DEEP LEARNING APPROACHES FOR BIOMEDICAL DATA CLASSIFICATION

Esra MAHSERECİ KARABULUT

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Supervisor :Assoc. Prof. Dr. Turgay İBRİKCİ
Year: 2016, Pages: 95
Jury :Assoc. Prof. Dr. Turgay İBRİKCİ
:Prof. Dr. Hamza EROL
:Assoc. Prof. Dr. Sami ARICA
:Asst. Prof. İrem ERSÖZ KAYA
:Asst. Prof. Dr. Lütfü SARIBULUT

Deep learning is a new and promising area in machine learning to solve problems of artificial intelligence. Such new approaches are required in automated medical decision making since non-automated process is more expensive, labor intensive and thus exposed to human-supplied errors. In the literature, common fields that make use of the deep learning approaches are computer vision, natural language processing and speech recognition. Even though the performance of deep architectures in these fields is reported to be impressive, the use of deep learning in the field of biomedicine is scarce. This study aims to investigate whether the deep learning approaches are also capable of producing successful results in the biomedical area, as well. For this aim, mainly deep learning approaches of Deep Belief Networks and Convolutional Neural Networks are employed. The investigation is carried out by conducting several experiments that include comparison with some well-known state-of-art methods which have been proven to be successful in the relevant literature.

Additionally, in the scope of the thesis, any possible improvements to the results of basic deep learning methodology are also investigated in order to enhance the classification performance. To this end, some preprocessing steps such as feature selection, removal of the imbalance in data, and texture analysis of medical images are applied in the experiments. To properly assess the performance, various biomedical benchmark datasets including low and high dimensional biomedical data, computed tomography images, microarray based cancer data and skin cancer images have been utilized. The experiments show that deep learning approaches show a promising direction towards supporting medical decision making.

Key Words: Deep learning, deep belief networks, convolutional neural networks, deep neural networks, microarray based cancer data

ÖZ

DOKTORA TEZİ

BİYOMEDİKAL VERİ SINIFLANDIRMASINDA DERİN ÖĞRENME YAKLAŞIMLARININ ARAŞTIRILMASI

Esra MAHSERECİ KARABULUT

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Danışman :Doç. Dr. Turgay İBRİKCİ
Yıl: 2016, Sayfa 95
Jüri :Doç. Dr. Turgay İBRİKCİ
:Prof. Dr. Hamza EROL
:Doç. Dr. Sami ARICA
:Yrd. Doç. Dr. İrem ERSÖZ KAYA
:Yrd. Doç. Dr. Lütfü SARIBULUT

Derin öğrenme, yapay zeka problemlerini çözmek için makine öğrenmesinde yeni ve gelecek vadeden bir alandır. Bu tür yeni yaklaşımlar, otomatik medikal karar verme sistemlerinde gereklidir, çünkü otomatik olmayan süreçler daha pahalıdır, yoğun emek ister ve bu yüzden de insan-kaynaklı hatalara maruzdur. Literatürde derin öğrenme yaklaşımlarını kullanan yaygın alanlar bilgisayar görmesi, doğal dil işleme ve konuşma tanımadır. Derin mimarilerin bu alanlardaki performansının etkileyici olduğu bildirilmiş olsa da, biyomedikal alanda derin öğrenme kullanımı oldukça azdır. Bu çalışma, derin öğrenme yaklaşımlarının biyomedikal alanda da başarılı sonuçlar üretip üretmediğini araştırmayı amaçlamaktadır. Bu amaç doğrultusunda, ağırlıklı olarak derin öğrenme yaklaşımlarından Derin İnanç Ağları ve Konvolusyonel Sinir Ağları kullanılmıştır. Araştırma, tanınmış modern yöntemler ile karşılaştırmayı da içeren birçok uygulama ile gerçekleştirilmiştir. İlgili literatürde, seçilip temel alınan bu yöntemlerin başarılı oldukları ispatlanmıştır.

Ayrıca, bu tez kapsamında, temel derin öğrenme yöntemlerinin sınıflandırma performansında olası herhangi bir iyileştirme yapılıp yapılamayacağı da araştırılmıştır. Bu amaçla, özellik seçimi, verideki dengesizliğin giderilmesi ve medikal görüntülerde doku analizi gibi bazı önişleme adımları uygulandı. Doğru performans değerlendirmesi için, düşük ve yüksek boyutlu veriler, bilgisayarlı tomografi görüntüleri, mikro dizilim tabanlı kanser verisi ve deri kanseri görüntülerini de içeren çeşitli biyomedikal veriler kullanıldı. Deneyler derin öğrenme yaklaşımlarının medikal karar vermeyi destekleyen yönde gelecek vadettiğini göstermektedir.

Anahtar Kelimeler: Derin öğrenme, derin inanç ağları, konvolusyonel sinir ağları, derin sinir ağları, mikro dizilim tabanlı kanser verisi

ACKNOWLEDGEMENTS

I am grateful to my supervisor, Assoc. Prof. Turgay İBRİKCİ, whose encouragement, guidance and support motivated me in research and study of the thesis. Furthermore he was always accessible and willing to help in every level of the study. It was a pleasure working with him.

It is a pleasure to thank to my thesis committee members and advisors Prof. Dr. Hamza EROL, Assoc. Prof. Dr. Sami ARICA, Asst. Prof. Dr. İrem ERSÖZ KAYA and Asst. Prof. Dr. Lütfü SARIBULUT for valuable insight they shared and guiding advices.

I would like to express my appreciation to my children Erva and Cengiz for tolerating my busyness during the completion of the thesis. I also thank my parents for their unceasing encouragement and support.

I would like to show my deepest gratitude to my husband, Mustafa KARABULUT. His support and patience has taught me so much about discipline and his experience broadened my perspective in the study of this thesis.

CONTENTS **PAGE**

ABSTRACT I
ÖZ II
ACKNOWLEDGEMENTS III
CONTENTS IV
LIST OF TABLES VI
LIST OF FIGURES VIII
LIST OF ABBREVIATIONS X
1. INTRODUCTION 1
1.1. Deep Learning Solution in Cancer Researches 4
1.2. Deep Learning for Classification of Medical Images 7
2. RELATED WORKS 11
3. MATERIAL AND METHODS 15
3.1. Datasets Utilized in the Study 15
3.1.1. The First Group of the Datasets 15
3.1.2. The Second Group of the Datasets 18
3.1.3. The Third Group of the Datasets 19
3.1.4. The Forth Group of the Datasets 20
3.2. Methods 21
3.2.1. Caffe Deep Learning Framework 21
3.2.2. Generative and Discriminative Models 22
3.2.3. Restricted Boltzman Machine 23
3.2.4. Deep Belief Networks 26
3.2.5. Convolutional Neural Networks (CNN) 28
3.2.6. Stacked Autoencoders (SA) 31
3.2.7. Deep Neural Networks 33
3.2.8. Baseline Methods 35
3.2.9. Information Gain Feature Selection 44
3.2.10. Synthetic Minority Over-Sampling Technique (SMOTE) 45
3.2.11. Local Binary Patterns 46

3.2.12. Block Difference of Inverse Probabilities (BDIP) 47
4. RESULTS AND DISCUSSION 49
4.1. Evaluation Metrics 49
4.2. Employing Discriminative Deep Belief Networks for Low and High Dimensional Biomedical Data 51
4.3. Comparison of Deep Belief Networks and Feedforward Neural Networks in Deep and Shallow Architectures 59
4.4. Emphysema Discrimination from Raw HRCT Images by Convolutional Neural Networks 62
4.5. Employing Deep Belief Networks for Microarray Based Cancer Classification 65
4.6. Probability Estimation of DDBN for Risk of Recurrence of Laryngeal Cancer 71
4.7. Texture analysis of Melanoma Images for Diagnosis using Convolutional Neural Networks 73
5. CONCLUSIONS 79
REFERENCES 81
CURRICULUM VITAE 96

LIST OF TABLES **PAGE**

Table 3.1. Composition of low dimensional datasets. 15
Table 3.2. Composition of high dimensional datasets. 17
Table 3.3. Categories in the dataset 19
Table 3.4. Class distributions and descriptions for melanoma data..... 19
Table 3.5. Class distributions and descriptions for microarray gene expression data..... 20
Table 4.1. Evaluation results of DDBN with one hidden layer by using LDD 52
Table 4.2. Evaluation results of DDBN with two hidden layers by using LDD.... 52
Table 4.3. Evaluation results of DDBN with three hidden layers by using LDD.. 52
Table 4.4. Evaluation results of DDBN with four hidden layers by using LDD... 53
Table 4.5. Evaluation results of Naive Bayes by using LDD 53
Table 4.6. Evaluation results of k-NN by using LDD 53
Table 4.7. Evaluation results of SVM by using LDD..... 54
Table 4.8. Evaluation results of Random Forest by using LDD..... 54
Table 4.9. Evaluation results of DDBN with one hidden layer by using HDD..... 56
Table 4.10. Evaluation results of DDBN with two hidden layers by using HDD ... 56
Table 4.11. Evaluation results of DDBN with three hidden layers using HDD 56
Table 4.12. Evaluation results of DDBN with four hidden layers using HDD 57
Table 4.13. Evaluation results of Naive Bayes by using HDD..... 57
Table 4.14. Evaluation results of k-NN by using HDD..... 57
Table 4.15. Evaluation results of SVM and Random Forest by using HDD 57
Table 4.16. Evaluation results of SVM and Random Forest by using HDD 58
Table 4.17. Accuracy values of NN with respect to number of hidden layers 60
Table 4.18. Accuracy values of DDBN with respect to number of hidden layers... 60
Table 4.19. CPU and GPU processing time while emphysema training of CNN ... 63
Table 4.20. Accuracies of classification of four methods..... 65
Table 4.21. Results of evaluation of DDBN using laryngeal microarray gene expression data..... 66

Table 4.22. Results of evaluation of SVM using laryngeal microarray gene expression data 66
Table 4.23. Results of evaluation of DDBN using bladder microarray gene expression data 67
Table 4.24. Results of evaluation of SVM using bladder microarray gene expression data 67
Table 4.25. Results of evaluation of DDBN using colorectal microarray gene expression data 67
Table 4.26. Results of evaluation of SVM using colorectal microarray gene expression data 67
Table 4.27. Averages of results in each metric from three microarray gene expression data after preprocessing of IG and SMOTE 70
Table 4.28. Results of CNN vs SVM without preprocessing, and by using LBP with P=8 and R=1 77
Table 4.29. Results of CNN vs SVM by using LBP with P=18 and R=2, and BDIP 77

LIST OF FIGURES **PAGE**

Figure 1.1. Interpretation of deep architecture of the brain (Wang and Raj, 2015) 2
Figure 1.2. HRCT scans from middle part of the lung (a) NT (b) CLE (c) PSE .. 8
Figure 1.3. Samples from melanoma dataset 9
Figure 2.1. Examples of correctly recognized handwritten digits that DBN had never seen before 11
Figure 3.1. 61x61 patches from HRCT slices (a) NT (b) CLE (c) PSE 18
Figure 3.2. Parallel computing styles of CPU and GPU 22
Figure 3.3. Unrestricted Boltzman machine with four visible units and three hidden units. 23
Figure 3.4. A Restricted Boltzman Machine with *i* visible units and *j* hidden units. 24
Figure 3.5. A traditional DBN composed of three layers of RBM 26
Figure 3.6. A DDBN composed of a visible unit, two RBM layers and an associative memory to output the target values. 28
Figure 3.7. Two different representations of CNN (a) the propagation of input in the network (b) layer wise representation 30
Figure 3.8. Representation of an autoencoder 31
Figure 3.9. Representation of taking hidden layer as input layer and reconstructing it back. 32
Figure 3.10. A two layered example of stacked autoencoders 32
Figure 3.11. Representation of deep neural network with *n* input and an output . 33
Figure 3.12. View of data that is separable linearly in two dimensional space 38
Figure 3.13. The probable maximum distance between linearly separable data .. 39
Figure 3.14. Representation of support vectors in two dimensional space 39
Figure 3.15. Determining k = 4 nearest neighbors given a new point. 43
Figure 3.16. Generation of binary patterns for a pixel 47
Figure 4.1. A confusion matrix for a binary classification. Correctly classified instances are represented by TN and TP, whereas incorrectly classified instances are represented by FP and FN. 49

Figure 4.2. Representation of accuracies produced by DDBN and other methods using LDD 55
Figure 4.3. Representation of accuracies produced by DDBN and other methods using HDD 58
Figure 4.4. Comparison of NN and DDBN having (a) one hidden layer (b) two hidden layers (c) three hidden layers (d) four hidden layers. 61
Figure 4.5. Accuracy values by iterations 63
Figure 4.6. CPU vs. GPU computing time 64
Figure 4.7. Comparison of DDBN vs. SVM in (a) laryngeal cancer (b) bladder cancer (c) colorectal cancer datasets. The effect of SMOTE and IG preprocessing are also given for each dataset. 69
Figure 4.8. Distribution of laryngeal cancer class labels holding the ideal probabilities of recurrence of the disease 72
Figure 4.9. Probability estimation of DDBN as output decision without preprocessing steps of IG and SMOTE. 73
Figure 4.10. Probability estimation of DDBN as output decision after using preprocessing steps of IG and SMOTE. 73
Figure 4.11. (a) Example instances from melanoma dataset (b) Contours for segmentation (c) Segmented images (d) Cropped images in size of 350x350. 74
Figure 4.12. Confusion matrix of CNN classification 75
Figure 4.13. Confusion matrix of SVM classification 75
Figure 4.14. Images of skin lesions after texture analysis (a) Some skin lesions from melanoma images (b) LBPs of some skin lesions from melanoma image data for P=8, R=1, (c) LBPs of some skin lesions from melanoma image data for P=16, R=2 (d) BDIPs of some skin lesions from melanoma image data 76
Figure 4.15. Representation of results of CNN and SVM with no preprcessing and LBP (P=8, R=1) 78
Figure 4.16. Representation of results of CNN and SVM with LBP (P=8, R=1) and BDIP preprocessing 78

LIST OF ABBREVIATIONS

Acc.	: Accuracy
ADCA	: Adenocarcinoma
ASR	: Automatic Speech Recognition
BDIP	: Block Difference of Inverse Probabilities
BVLC	: Berkeley Vision and Learning Center
CD	: Contrastive Divergence
CLE	: Centrilobular Emphysema
CNN	: Convolutional Neural Network
COPD	: Chronic Obstructive Pulmonary Disease
CPU	: Central Processing Unit
CT	: Computed Tomography
CTG	: Cardiotocogram
CUDA	: Compute Unified Device Architecture
DBN	: Deep Belief Network
DDBN	: Discriminative Deep Belief Networks
DNA	: Deoxyribonucleic Acid
DNN	: Deep Neural Network
EEG	: Electroencephalogram
FPR	: False Positives Rate
GPU	: Graphics Processing Unit
HDD	: High Dimensional Data
HLIF	: High Level Intuitive Features
HRCT	: High Resolution Computed Tomography
IG	: Information Gain
k-NN	: k-Nearest Neighbors
LBP	: Local Binary Pattern
LDD	: Low Dimensional Data
MAE	: Mean Absolute Error
MD	: Mahalanobis Distance
MLP	: Multilayer Perceptron

MNIST	: Mixed National Institute of Standards and Technology
NN	: Neural Networks
NT	: Normal Tissue
PSE	: Paraseptal Emphysema
RBF	: Radial Basis Network
RBM	: Restricted Boltzman Machine
SA	: Stacked Auoencoders
SMOTE	: Synthetic Minority Over-Sampling Technique
SOM	: Self-organizing Maps
SVM	: Support Vector Machine
TPR	: True Positives Rate
UCI	: University of California

1. INTRODUCTION

Deep learning presents a new way of imitating the human brain functionality in representing information. It is a new area in the artificial intelligence research which for decades aims to decipher complex data representation method adopted by the human brain. According to Hebb, father of cognitive psychobiology, it is impossible to understand the nature of learning unless knowing the working principals of the brain (Sejnowski and Tesauro, 1989). Therefore the perfect neurological analysis of the human is essential in order to build human-like thinking machines. Inspiring the human brain, deep learning uses the findings of neuroscience which reveals that the neo-cortex of human brain propagates the data through a complex hierarchy (Lee and Mumford, 2003). In time this provides that observations are represented by their features. The natural motivation of studies in deep architectures is due to the fact that human brain forms a deep architecture in recognition and sensing of knowledge (Meunier et al., 2010). Therefore it is obvious that brains use a deep architecture (Serre et al., 2007, Arel et al., 2010)).

Deep learning approaches take their roots from neural networks where ‘deep’ refers to several layers stacked on top of each other. A deep learning architecture is constructed by stacking layers each of which is supposed to extract higher level representations, i.e. features. For example, an image is made up of pixels, and a deep model learns the picture edges, object parts and objects as propagating through the layers. Shallow models (e.g. one-hidden layered neural networks, support vector machines, decision trees etc.) don’t learn high level features or multiple levels of representation, and deep architectures are potentially greatly more powerful than shallow ones. (Bengio, 2009). Larochelle et al. (2007) indicated the impressive performance of deep architectures in harder learning problems incorporating factors of variation. All the deep learning approaches try to learn feature hierarchies, i.e. representations of data (see Figure 1.1). By this way, approaches of deep learning are capable of being employed in classification tasks, natural language processing, image processing, speech recognizing etc., and they are potentially much more powerful than shallow architectures (Bengio, 2009).

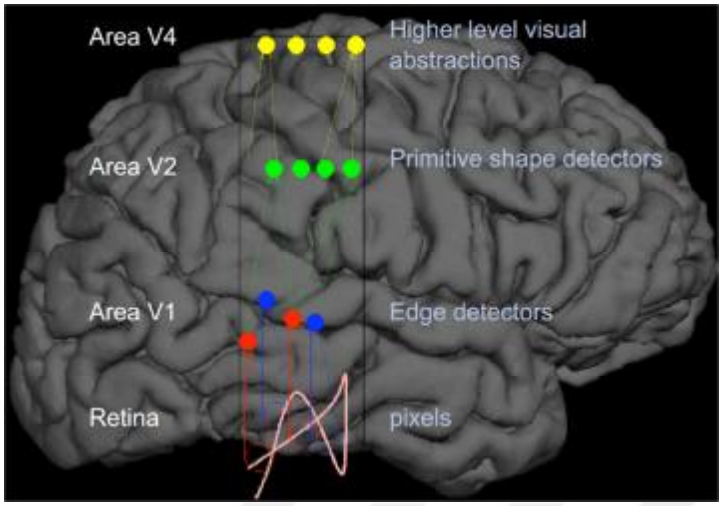


Figure 1.1. Interpretation of deep architecture of the brain (Wang and Raj, 2015)

In this study, we will present four mainstream deep learning approaches; discriminative deep belief networks (DDBN), convolutional neural networks (CNN), stacked autoencoders (SA, autoassociators) also known as denoising autoencoders and deep neural networks (DNN). Each approach has its own advantages and usage areas, for example CNNs has superiority for use of two dimensional data such as images and videos (Arel et al., 2010). Stacked autoencoders are capable of encoding-decoding input data, data compression or dimensionality reduction. DNN are simply a traditional artificial neural network having multiple hidden layers rather than a one hidden-layered neural network known as shallow neural network. A DDBN is a variant of Deep Belief Networks (DBN) which we will focus on in this study with respect to its potential of supervised classification power. Hinton et al. (2006) introduced DBN both as generative and discriminative variant models. The discriminative one is the DDBN which is used for supervised classification purposes. According to their idea, drawbacks of a traditional neural network trained by back propagation can be overcome by unsupervised pre-learning stage of DDBN. For example, back propagation starts the training by assigning random weight values. Additionally back propagation is very slow when the number of hidden layers is more than one. It is very likely to get stuck in poor local optima. It requires labeled data for training of the model, however almost all data are unlabeled in real scenarios. However, DDBN partitions the training process into two phases:

unsupervised greedy layer wise pretraining and fine tuning the whole network. The unsupervised greedy layer-wise training provides an initialization which replaces the random initialization of back propagation. This procedure employs an unsupervised generative learning algorithm for each layer which is constituted by a Restricted Boltzmann Machine (Freund and Haussler, 1994). After greedy layer-wise training phase, the model is fine-tuned in either supervised or unsupervised manner. If the model will be used as a generative model, it is unsupervised. If it will be used for a classification task it is a model of DDBN.

To evaluate DDBN in low and high dimensional biomedical data, the experiments are conducted according to benchmark datasets used in literature and comparative results are gained. A comprehensive understanding of deep learning approaches are required for evaluation of performance of a deep learning approach in medical decision making tasks by use of common benchmarking datasets utilized in the relevant literature and real life medical datasets. In the study, several numerical experiments are conducted including contrasting and comparing DDBN with some leading regular machine learning algorithm.

In order to evaluate the performance of the DDBN approach on low and high dimensional biomedical data exactly, the classification results are compared to four essential classification methods; Naive Bayes, k-Nearest Neighbors (k-NN), Support Vector Machine (SVM) and Random Forest. All the four aforementioned baseline methods have been successfully applied to predictive data mining in biomedicine in literature. Naïve Bayes and k-NN have a common use in comparative medical classification studies (Salem et al., 2009; Beckonert et al., 2003; Kononenko et al., 2001), due to their power in learning such as other sophisticated models. Furthermore, SVM and Random Forest have become a standard data analysis tool in bioinformatics even in evaluation and mining in high dimensional data (Peng et al., 2010; Boulesteix et al., 20102). The comparisons between DDBN and other methods are primarily based on scalar performance measures of accuracy, mean absolute error (MAE), true positives rate (TPR), false positives rate (FPR) pertaining to the classification. The major aim of this study is to investigate whether the deep learning approaches that are proved themselves in prior studies in areas of speech recognition,

natural language processing, computer vision can also produce such improved results in medical and biomedical data.

1.1. Deep Learning Solution in Cancer Researches

Accurate diagnosis of cancer is of great importance due to the global increase in new cancer cases. Cancer is the second-leading cause of death in the United States, coming after the heart disease (USCS Working Group, 2013). The name cancer refers to more than a hundred diseases characterized by out of control growth and multiplication of the cells. These cells may form a tumor which may be benign or malignant. A malignant tumor is cancerous and can spread to other parts of the body. There are four main types of cancer; carcinoma, sarcoma, leukemia, lymphoma. Carcinoma develops in skin or tissues of internal organs like lung, prostate, breast and lung. Whereas sarcoma develops in certain tissues like bone, muscle or fat. Leukemia is cancer of white blood cells which are produced by bone marrows whereas lymphoma is a cancer of the lymphatic system which is a network of vessels and glands constructing a part of immune system.

For the purpose of cancer diagnosis, use of microarray technology along with computer aided methods is increasing rapidly. DNA microarray technology generates features for monitoring expression of genes on genomic level for researching cancer. Diagnosis from such a microarray gene expression data is shown to be more effective compared to traditional methods (Lu and Han, 2003; Su et al., 2007, Tarca et al., 2006). Analysis over gene expression data for the purpose of diagnosis can be done by using additional laboratory experiments, but such experiments are costly and labor intensive. Alternatively, as a replacement to experiments for diagnosis, methods from artificial intelligence, specifically, machine learning methods, can also be utilized to perform computer-based diagnosis of the gene expression data. Because computer-based diagnosis has advantages over laboratory experiments such as low cost and diagnosis speed, research over computer-based methods has been gaining attention and remains essential.

In the literature, several machine learning techniques are evaluated for computer-based diagnosis by using gene expression profiles. The first study that belongs to Golub et al. (1999) is about clustering acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) data by self-organizing maps (SOM). The subsequent studies include specific algorithms with application to specific gene expression profiles (Khan et al., 2001; Samarasinghe et al., 2015) and comparative analysis of different methods (Dudoit et al., 2002; Lee et al., 2005; Yang and Daiman, 2014; Vural et al., 2015). Support Vector Machine (SVM) is the most prominent of them since its ability of handling high dimensional data lead to successful application for classification of microarray gene expression data (Furey et al., 2000; Brown et al., 2000; Ramaswamy et al., 2005; Mukherjee, 2003; Statnikov et al., 2005). Pirooznia et al. (2008) achieved a comparative study among several machine learning methods on microarray gene expression data from different cancer types. They employed methods of SVM, RBF Neural Nets, MLP Neural Nets, Bayesian Network, Decision Tree and Random Forest methods and in almost all cases SVM outperformed the others. This is the reason that motivated us to choose SVM as a baseline method while evaluating the performance of DDBN in cancer classification.

While analyzing microarray gene expression data, a problem that should be handled is the high dimensionality of the data which causes the classifier to be overfitting to the data and increases the computational load. Even with limited number of samples in the data, there are thousands of features, i.e. genes. Therefore prior to the classification, feature selection is essential for identifying the subset of genes which are relevant for predicting the classes of samples (Kumar et al., 2013; Bolón-Canedo et al., 2015; Fernández-Navarro et al., 2012; Chen and Zhao, 2008). Additionally, imbalanced class distribution in the data is also another handicap for efficient training of a classifier, as well. One way to deal with this problem is using oversampling techniques such as the Synthetic Minority Over-Sampling Technique (SMOTE) (Chawla et al., 2002; Chawla, 2003; Estabrooks et al., 2004; Karabulut and Ibrikci, 2014). Blagus and Lusa (2013) performed feature selection before using

SMOTE in high dimensional gene expression data showed that substantial benefit can be obtained by use of SMOTE along with k-nearest neighbors classifier.

Being a recent approach, deep learning has not been investigated in application of classification of microarray gene expression data in the literature as comprehensively as it should have been. Therefore, research over the deep learning approach in this field is quite inadequate and as well as promising with the consideration of the fact that it is a proven method in other fields of machine learning (Hinton et al., 2006; Mohamed et al., 2012; Vinyals et al., 2011; Bengio et al., 2007; Sarikaya et al., 2011; Huang et al., 2012; Sainath et al., 2011). In this study, we also aimed to show that DDBN is capable of being a successful decision support model for cancer data in addition to its proven success in other subfields of artificial intelligence recently (Hinton et al., 2006; Mohamed et al., 2012; Vinyals et al., 2011; Bengio et al., 2007; Sarikaya et al., 2011; Huang et al., 2012; Sainath et al., 2011). For this purpose, three microarray gene expression datasets are analyzed: laryngeal cancer, bladder cancer and colorectal carcinomas. The results of DDBN with SVM in terms of accuracy, sensitivity, specificity, precision and F-measure metrics. In the preprocessing phase, we employed Information Gain (IG) feature selection method for discovering predictive genes, and SMOTE to overcome the problems aroused by the imbalanced nature of the data. Therefore, we attempted to propose a general method based on DDBN to diagnose cancer cases by using gene expression data (Karabulut and Ibrikci, 2017). This model attempts to keep the diagnosis performance stable even with datasets containing imbalanced class distribution and high number of genes.

The relevant literature studies -described in related works section- that utilized deep learning approach in microarray data generally aimed at finding features or reducing dimensionality of the data in an unsupervised manner. This study diverges from them in that it aims at evaluating a deep learning method, namely DDBN, in supervised classification of cancer cases for the purpose of diagnosis.

1.2. Deep Learning for Classification of Medical Images

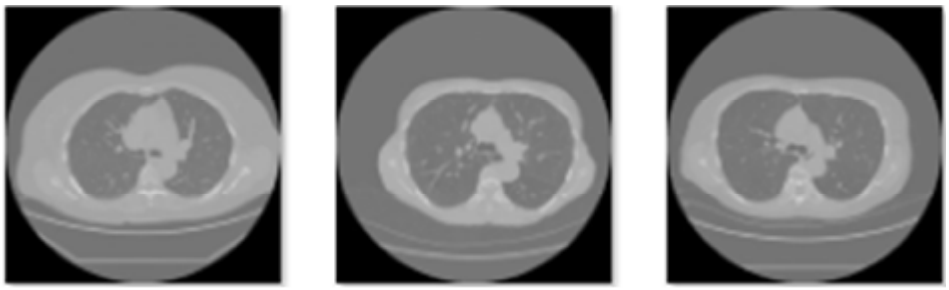
In this thesis, we presented computer based evaluation results on two medical images; the images of lung related to emphysema disease and the images of melanoma skin cancer.

Emphysema is a disease of the lungs described as decrease in the amount of oxygen transferred to the blood and therefore causes shortness of breath. Emphysema is mostly caused by smoking and it is characterized by elasticity loss of the tissue surrounding the alveoli limiting the expanding and shrinking of airspaces. Emphysema with chronic bronchitis is referred to as Chronic Obstructive Pulmonary Disease (COPD) which is the fourth leading cause of death in the United States (HHS, 2003) and affects 5% of the world population (Vos et al., 2012).

A reliable technique for imaging the pathologies of emphysema is the high resolution computed tomography (HRCT). The possibility of other medical conditions related to lung can be detected and eliminated by HRCT scanning. Two important subtypes of emphysema are centrilobular emphysema (CLE) and paraseptal emphysema (PSE) which can be distinguished via HRCT images. CLE is the most common of the subtypes that arise from long-term cigarette smoking and usually involves the upper half of the lungs. PSE can occur both in smokers and non-smokers, and it is mostly found in younger patients compared to CLE (Satoh et al., 2001). Figure 1.2. represents three example HRCT scans of size 512x512 where normal tissue (NT) describes the non-emphysematous healthy lung.

In the literature, there are several approaches for automated discrimination of emphysema subtypes by using machine learning approaches (Gangeh et al., 2010; Azim and Niranjan, 2014; Ibrahim and Mukundan, 2014). In most of the studies, as a pre-work, textural features of HRCT images are discovered by using texture analysis methods such as kernel density estimation of local histograms (Mendoza et al., 2012), Riesz transfom (Depeursinge et al., 2012), different intensity measures (Ibrahim and Mukundan, 2014), density mask (Müller et al., 1988; Kinsella et al. 1990), smoothing and thresholding (Friman et al., 2002). Sørensen et al. (2010) achieved the best in a way that they preprocessed the emphysema data by combining

textural features using local binary patterns (LBPs) and classified by k-nearest network (k-NN) method.



(a) (b) (c)

Figure 1.2. HRCT scans from middle part of the lung (a) NT (b) CLE (c) PSE

As a deep learning approach Convolutional Neural Networks (CNN) has an architecture design for learning 2D input data. When an image is fed to a CNN, it extracts features from raw pixels at the first layer. As the data is processed through layers, higher-level informative input is generated for the softmax classification layer.

In this paper, discrimination between emphysema subtypes and non-emphysematous normal tissue is achieved by CNN that has never been used for classification of CT lung images in the literature. The dataset is provided by Sørensen et al. (2010) as grey-scale 16 bit tiff images with a resolution of 61 × 61 pixels. It comprises 168 patches from 115 lung HRCT slices. Each patch is labelled as NT, CLE or PSE. Experimental results are obtained by using Caffe environment. Caffe, (Jia, 2013) is a deep learning framework developed by the Berkeley Vision and Learning Center (BVLC) with the support of community contributors. Our CNN model is trained in a supervised fashion in the Caffe deep learning framework. We explored the direct performance of CNN on emphysema medical image data without using any of texture analysis preprocessing. The studies on classification of direct raw images of emphysema are rare in the literature. From this point of view, to our knowledge this study presents the best performance with respect to processing time and accuracy without any preprocessing of the input images (Karabulut and Ibrikci, 2015).

The other data of medical image analyzed in the thesis is images of melanoma skin cancer. Melanoma is a type of skin cancer that occurs in melanocytes cells which color the skin and produce melanin pigments. Comparing to other skin cancer types melanoma is less common but it is very dangerous. 75 % of deaths caused by skin cancers are melanoma cancer (Jerant et al., 2002). Melanoma cells make more melanin than normal so melanoma tumors occur which are generally brown or black (see Figure 1.3). Moles on the body are mostly benign melanoma, but sun exposure and artificial ultraviolet light are two main causes of malignant melanoma. It tends to spread to other parts of body, therefore early detection is very important, and it is curable in early stages.

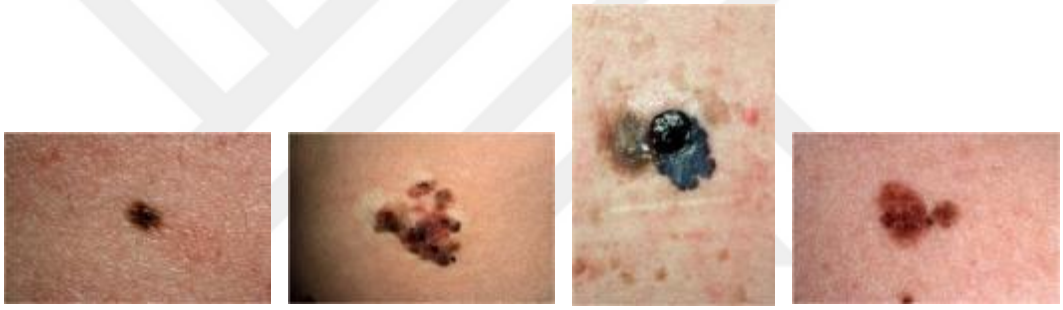


Figure 1.3. Samples from melanoma dataset

A mole on the body can be suspected whether it is malignant melanoma according to ABCD rule (Nachbar et al., 1994), in which A stands for asymmetry, B is border irregularity, C is color changes or many different colors, and D is diameter more than 6 mm. By adding a fifth criterion E, evolution, the rule is improved. E criterion implies the changes in morphology of the lesion in time. ABCD rule has been accepted by a worldwide point of view, however it may not be accurate in suspecting benign melanomas or in small malignant ones. Additionally dermatologists screen the melanoma at high cost. For a first step of computer based diagnostic system skin automatic lesion segmentation studies are carried out (Glaister, 2013; Othman et al., 2014). Amelard et al. (2015) studied on such a computer based diagnosis by defining high level intuitive features (HLIFs) as a feature extraction framework. These features are determined automatically by using

ABCD rules and the obtained feature vector is fed to Support Vector Machine (SVM) for classification.

In this study for analysis of melanoma images we evaluated the raw pixel intensity values for Convolutional Neural Network (CNN) and SVM classification. In the experiments we employed texture analysis methods of Local Binary Pattern (LBP) and Block Difference of Inverse Probabilities (BDIP). The results are compared both from aspect of these texture analysis methods and classification methods, i.e. CNN and SVM (Karabulut and Ibrikci, 2016).

2. RELATED WORKS

Hinton et al. (2006) put forward the idea of training strategies of deep architectures by studying training of deep belief networks (DBN). Their approach based on greedy layer-wise unsupervised pre-training followed by supervised or unsupervised fine-tuning. If the system is aimed to be used as discriminative, the fine tuning is performed by back propagation. Therefore the pretraining stage of the algorithm is unsupervised but can also be applied to the labeled data. The MNIST database of handwritten digits containing 60000 training images and 10000 test images are used in their study and a generalization performance of 1.25% errors is obtained. Figure 2.1 represents the digits the DBN model was able to recognize in their study, some of which are difficult to classify even by a human.

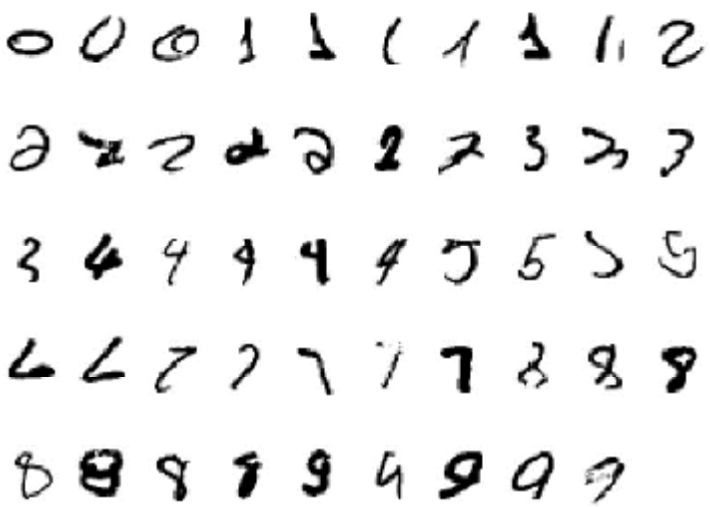


Figure 2.1. Examples of correctly recognized handwritten digits that DBN had never seen before

Tamilselvan and Wang (2013) proposed multi sensor system health diagnosis methodology using DBN, with the aim of handling the complexity of sensory signals for failure diagnosis applications. Experimental results are presented for two case studies; first case utilizes 2008 IEEE PHM challenge data for aircraft engine health diagnosis, and the second case is power transformer mechanical fault diagnosis. Their methodology is compared with four existing diagnosis techniques; support

vector machine SVM, neural network (NN), self-organizing maps (SOM) and Mahalanobis distance (MD) classifiers. The results of both two cases show that DBN based approach produces higher correct classification rates compared to the four techniques.

In literature there are studies also for achieving natural language processing, automatic speech recognition (ASR) and image classification by deep learning approach. Collobert and Weston (2008) employed deep learning for natural language processing. Mohamed et al. (2012) reported the first application of deep learning of acoustic modeling and showed in their study that a better phone recognition can be achieved by deep belief networks compared to a typical ASR system which uses hidden markov models. Siniscalchi et al. (2013) asserted that deep neural networks (DNN), i.e. artificial neural networks having multiple hidden layers, are successful especially in recognition of speech tasks. They showed that DNNs are able to increase the accuracy of classification of phonetic attributes and phonemes. DNN outperforms multilayer perceptron (MLP) having single hidden layer by 90% and 86.6% of frame-level attribute estimation and phoneme prediction respectively. Another study with the aim of ASR by deep learning approach is carried out by Vinyals and Ravuri (2011). They drew the conclusion of the superiority of DBN over MLP by results of 10.1% and 15.4% phone error rates respectively.

A study for image classification is carried out by Krishevsky et al. (2012) in which convolutional DNN is employed. Supervised learning is used in the method and 1.2 million images from ImageNet dataset is utilized (Deng et al., 2009). According to the results of their study the authors assert that such a method is capable of achieving record breaking results. In another study, Hörster and Lienhart (2008) derived a low dimensional image representation appropriate for image retrieval by using deep networks. For classification of MNIST data, Bengio et al. (2007) compared five learning networks; DBN with unsupervised pretraining, deep network of stacked auto associators with pretraining, DNN with supervised pretraining, DNN without pretraining and one layered shallow neural network. The result of their study showed that deep architectures produce better results than shallow one. Ranzato et. al. (2007) employed deep convolutional network for

handwritten digit classification. They used 600,000 digits augmenting MNIST data by distorted data and investigated the error rate of back propagation alone and unsupervised layer-by-layer pretraining. These two methods produced error rates of 0.49% and 0.39% respectively.

The studies of application of deep learning approaches on biomedical data are sparse. Plis et al. (2014) and Brosch et al. (2013) applied DBN to neuroimaging data and showed the capability of DBN in learning physiologically important representations and detecting modes of variations. Li et al. (2014) employed a variety of DBN models for identifying informative risk factors and predicting bone disease progression and their results showed that DBN can capture characteristics for different patient groups and select risk factors validated by the medical literature. In another study (Li et al., 2013), DBN is used for effectively extracting critical information from thousands of features of electroencephalogram (EEG) data in an unsupervised manner. Hua et al. (2015) introduced models of a deep belief network and a convolutional neural network in the context of nodule classification in computed tomography images. Fractal analysis and scale invariant feature transform methods with feature computing steps were implemented for comparison. Their experimental results suggest that deep learning methods have the capability of achieving better results and in various domains of applications they are promising as computer aided diagnosis methods.

In one of the few studies that utilized deep learning approach with microarray data, Gupta et al. (2015) demonstrated the empirical effectiveness of using deep autoencoders as a pre-processing step for clustering of gene expression data. Another contribution was made by Fakoor at al. (2013) that reported unsupervised feature learning can be used for cancer detection and cancer type analysis from gene expression data by deep stacked autoencoders. Ibrahim et al. (2014) proposed a deep and active learning based method called MLFS (Multi-level gene/MiRNA feature selection) for selecting genes from expression profiles. Their experiments show that the approach outperforms classical feature selection methods in hepatocellular carcinoma, lung cancer and breast cancer. In another study (Denas, 2014) deep models are applied to functional genomic data to build low dimensional

representations of multiple tracks of experimental functional genomics data. Briefly, relevant literature studies that utilized deep learning approach in microarray data generally aimed at finding features or reducing dimensionality of the data in an unsupervised manner. This study diverges from them in that it aims at evaluating a deep learning method, namely DDBN, in supervised classification of cancer cases for the purpose of diagnosis.

3. MATERIAL AND METHODS

3.1. Datasets Utilized in the Study

3.1.1. The First Group of the Datasets

For this group of datasets, the study investigates the DDBN behavior in classification of biomedical data in two categories; low dimensional data (LDD) and high dimensional data (HDD). All the eight LDD are downloaded from University of California, Irvine (UCI) machine learning repository (Newman et al., 1998). Missing values in both LDD and HDD are replaced by the average of values of the corresponding attribute. Table 3.1 summarizes the numerical properties of LDD.

Table 3.1. Composition of low dimensional datasets.

No	Data	# of instances	# of attributes	# of classes	missing values
1	Arrhythmia	452	279	2	Yes
2	Heart disease (Cleveland)	303	13	5	Yes
3	Vertebral column (2C)	310	6	2	No
4	Diabetes (Pima Indians)	768	8	2	No
5	Wisconsin breast cancer	699	9	2	Yes
6	Mammographic mass	961	5	2	Yes
7	CTG	2126	21	3	No
8	Parkinson	194	22	2	No

Arrhythmia data is used with the aim of distinguishing whether the cardiac arrhythmia is present or not. The original dataset has 16 classes including different types of arrhythmia. Data is altered to include two class labels for records by assigning ‘present’ and ‘absent’ for cardiac arrhythmia to have a balanced data. 245 of 452 instances are labeled as present and 207 are absent (Lichman, 2013).

Heart disease data is obtained from the clinical and noninvasive test results of 303 patients collected in Cleveland Clinic Foundation (Detrano et al., 1989). Class

values are integers from 0 to 4 where 0 refers to absence of heart disease, and others refers to presence of levels of the disease. In the dataset there are 164, 55, 36, 35, 13 instances labeled as 0, 1, 2, 3 and 4 respectively.

Vertebral column data is available both as including two class labels (2C) and three class labels (3C). We used the first one (2C) in which the task is distinguishing the patients as 'normal' or 'abnormal'. 310 patients are represented by attributes derived from the shape and orientation of the pelvis and lumbar spine (Berthonnaud et al., 2005). There are 100 normal and 210 abnormal labelled record in the data.

Diabetes data includes 768 records derived from patients who are females at least 21 years old of Pima Indian heritage. Each record is labelled as 'tested_positive' and 'tested_negative' indicating the presence and absence of the disease. There are 268 'tested_positive' instances and 500 'tested_negative' instances in the data (Lichman, 2013).

Wisconsin breast cancer tumor data is collected at University of Wisconsin Hospitals Madison (Wolberg and Mangasarian, 1990). The samples are provided from needle aspiration from human breast cancer tissue. Nine attributes of the data are clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nuclei, and mitoses. There are 699 samples 458 of which are benign and 241 are malignant.

Mammographic mass data is derived from mammography, a method for screening breast cancer (Elter et al., 2007). Each record has a label of 'malignant' or 'benign' indicating the severity of the disease. There are 445 malignant and 516 benign instances in the data.

CTG (cardiotocogram) data includes simultaneous recordings of both fetal heart rate (FHR) and uterine contractions of a pregnant (Lichman, 2013). By using 21 given attributes data can be classified according to FHR pattern class or fetal state class code. In this study, fetal state class code is used as target attribute instead of FHR pattern class code. Each record has a label of 'normal', 'suspect' or 'pathologic', indicating if the fetus is suffering from lack of oxygen. There are 1655 normal, 295 suspicious and 176 pathologic instances in the data.

Parkinson's data includes biomedical voice measurements taken from 31 people (Little et al., 2007). The data is used to identify healthy people from diseased ones according to the status attribute which is determined as 1 for patients and 0 for healthy people. There are 147 records labelled as 1, and 48 records labelled as 0 in the data.

Table 3.2 summarizes the numerical properties of HDD. The p-53 mutants data (Danziger et al., 2009) is publicly available from University of California, Irvine (UCI) machine learning repository. Breast cancer data (Van't Veer et al., 2002), lung cancer data (Gordon et al., 2002) and ovarian cancer data (Petricoin et al., 2002) are obtained from Kent Ridge Biological Data Set Repository (Li and Liu, 2002). Laryngeal cancer data (Fountzilas et al., 2013) is also publicly available from BioGPS gene annotation portal (Wu et al., 2009).

Table 3.2. Composition of high dimensional datasets.

No	**Data**	**# of instances**	**# of attributes**	**# of classes**	**missing values**
1	p-53 mutants (site 3)	114	5409	2	Yes
2	Breast Cancer	97	24481	2	No
3	Lung Cancer	181	12533	2	No
4	Ovarian Cancer	253	15154	2	No
5	Laryngeal Cancer	109	22287	2	No

p-53 mutants data includes the features of mutant p53 proteins which have the ability of tumor suppressor activity. Class labels are 'active' and 'inactive' representing transcriptionally competent p53 and cancerous p53 respectively. There are 63 active and 51 inactive labelled records. Attributes 1- 4826 keep 2D electrostatic and surface based values and attributes 4827-5408 keep 3D distance based value. Class labels are in attribute 5409.

Breast cancer data includes records of 24481 genes of patients labeled as 'relapse' and 'non- relapse'. Relapse indicates that distance metastases (spread to other parts of body) developed within five years. Non-relapse indicates that the patient remains healthy after the diagnosis of breast cancer for five years. There are 46 relapse and 51 non-relapse records in the dataset.

Lung cancer data is constituted by 181 tissue records labeled as ‘malignant pleural mesothelioma’ (MPM) and ‘adenocarcinoma’ (ADCA). MPM and ADCA are two types of lung cancer. MPM is diagnosed in people exposed to asbestos and develops on a thin membrane of a lung called the pleura. ADCA lung cancer arises in glandular cells found in tissue that lines the lung and make substances such as mucus. There are 31 MPM and 150 ADCA records in the dataset.

Ovarian cancer data is collected from women owing a high risk of ovarian cancer because of family or the history of cancer. The aim is to detect proteomic patterns in serum that distinguish normal and ovarian cancer. There are a total of 253 records 91 of which are labelled as ‘normal’ and 162 are labelled as ‘cancer’.

Laryngeal cancer data is used for detecting patients that are likely to recur after treatment. Class labels are as 1 and 0, which indicate reccurrence and non-reccurence of laryngeal carcinoma respectively. 75 of 109 patients recorded to have suffered the recurrence of the cancer, and 34 have not.

3.1.2. The Second Group of the Datasets

The dataset is provided by Sørensen et al. (2010) as grey-scale 16 bit tiff images with a resolution of 61 × 61 pixels. It comprises 168 patches from 115 lung HRCT slices. Each patch is labelled as NT, CLE or PSE. Figure 3.1. presents samples patches from the dataset. CT scanning was performed using General Electric (GE) equipment (LightSpeed QX/i; GE Medical Systems, Milwaukee, WI, USA) with four detector rows and using the following parameters: in-plane resolution 0.78 x 0.78 mm, slice thickness 1.25 mm, tube voltage 140 kV, and tube current 200 mAs. The slices were reconstructed using a high-spatial-resolution (bone) algorithm (Sørensen et al., 2010; Shaker et al., 2008).

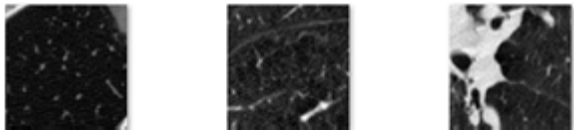


Figure 3.1. 61x61 patches from HRCT slices (a) NT (b) CLE (c) PSE

HRCT scans are from 39 people each of which belongs to one of the three categories: a) 9 healthy and non-smokers, b) 10 healthy and smokers (without COPD) and c) 20 smokers and COPD patients. Each patch is manually annotated with one of the three classes; NT, CLE or PSE. Table 3.3 gives class descriptions and the number of patches in each class.

Table 3.3. Categories in the dataset

Category	**Description**	**# of samples**
1 (NT)	annotated in never smokers	59
2 (CLE)	annotated healthy smokers and smokers with COPD	50
3 (PSE)	annotated healthy smokers and smokers with COPD	59

3.1.3. The Third Group of the Datasets

Amerald et al. (2015) constituted the melanoma dataset summarized in Table 3.4. They extracted the skin lesion images from Dermatology Information System (DermIS, 2016) and DermQuest (Galderma, 2016) by additionally including the segmentation contour partner of each image. They segmented the images manually with the aim of eliminating the effect of automatic segmentation on accuracy.

Table 3.4. Class distributions and descriptions for melanoma data

	Malignant	**Benign**
Dermatology Information System	43	26
DermQuest	76	61
Total	119	87

There are a total of 206 images in dataset. 119 of them labelled as malignant and 87 of them labelled as benign. The images are acquired via a digital camera in varying and nonstandard environmental conditions.

3.1.4. The Forth Group of the Datasets

Microarray gene expression datasets of laryngeal cancer, bladder cancer and colorectal cancer are taken from BioGPS (Wu et al., 2009) which is a customizable and extensible gene annotation portal and also a source for information about genes. It is supported by the U.S. National Institute of General Medical Sciences. It presents its gene annotation data to scientists by a flexible search interface. Additionally it has a role of content aggregator of many other gene annotation portals, therefore most of its content is obtained from other online resources.

Table 3.5. Class distributions and descriptions for microarray gene expression data

Type of microarray gene expression data	**# of genes**	**Negatives**		**Positives**	
		# of samples	**description**	**# of samples**	**description**
Laryngeal cancer (Fountzilas et al., 2013)	22284	75	no recurrence of disease	34	recurrence of disease
Bladder cancer (Urquidi et al., 2012)	54676	40	non-cancer urothelial cells	52	urothelial cancer cells
Colorectal cancer (Watanabe et al., 2009)	54676	57	without lymph node metastasis	32	with lymph node metastasis

Laryngeal microarray gene expression data was obtained using Affymetrix U133A Genechips. Tumor tissues of 66 laryngeal cancer patients were profiled and gained gene expressions are obntained. By adding age, duration of DFS (disease free survival) in months and the grade of the cancer a total of 22287 featured dataset is obtained. Using Affymetrix U133 Plus 2.0 arrays platform bladder cancer gene expression data were obtained from exfoliated urothelia sampling for the evaluation of patients with suspected bladder cancer. The data was collected from 92 subjects and a total of 54676 genes are used. The aim of researchers was to identify urothelial cell transcriptomic signatures associated with bladder cancer.

The expression profiles of colorectal cancer were obtained from 89 patients using Affymetrix Human Genome U133 Plus 2.0 arrays. The aim of the researchers in collecting this data is to identify whether there is lymph node metastases in patient or not, because the existence of lymph node metastases give the physicians an idea about the prognosis of the colorectal cancer. Table 3.5 summarizes the properties of three microarray gene expression data.

3.2. Methods

3.2.1. Caffe Deep Learning Framework

Caffe is a fast open-source framework to implement CNNs and other deep learning networks (DNNs) easily for researching and exploring purposes. It is maintained and developed by the Berkeley Vision group (Jia, 2013). It supports a wide variety of architectures and efficient implementations vital machine learning tasks such as prediction and learning. It provides state-of-the-art deep learning models for research projects and industrial applications in areas of computer vision, speech and multimedia processing. Caffe is licensed under the BSD license. It is designed to be a C++ library but it has bindings for other programming languages such as Python and Matlab.

Computational complexity is high in deep learning models, so Caffe provides means to use both GPU and CPU processing in a parallel fashion. Caffe is claimed to be the fastest CNN implementation (Jia, 2013). High speed is achieved by Compute Unified Device Architecture (CUDA) GPU computation which is capable of processing over 40 million images in a day with a single K40 or Titan GPU (approximately 2 ms per image) (Jia, 2013). Such a performance level is of great importance for deep models research.

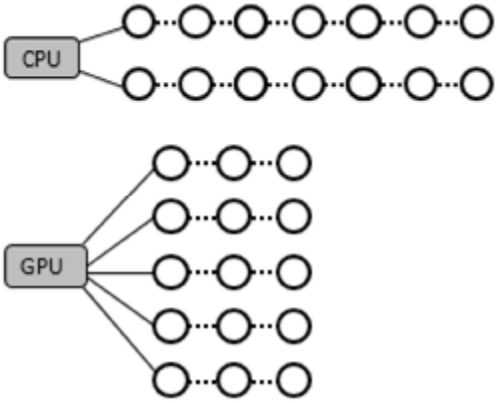


Figure 3.2. Parallel computing styles of CPU and GPU

GPU computing puts emphasis on parallel computing rather than higher processing unit performance, whereas CPU has low number of parallel units that are powerful in processing as represented in Figure 3.2. However, parallelism outperforms powerful unit performance.

3.2.2. Generative and Discriminative Models

Generative methods are model distributions of the input and output variables, therefore a mapping between inputs and target outputs are optimized. They are full probabilistic models using all the variables, i.e. both input and output variables.

$$p(x, y) = p(x_1, \dots, x_n, y_1, \dots, y_n) \tag{1}$$

where x is input vector containing n elements. Principal Component Analysis, Gaussian Mixture Model, Hidden Markov model, Naïve Bayes and Restricted Boltzmann machine are some examples for generative models. Discriminative models are conditional probabilistic models which samples the output variables conditional on the input variables.

$$p(y|x) = p(y_1, \dots, y_n|x_1, \dots, x_n) \tag{2}$$

Discriminative approaches focuses on learning instead of modeling, the relation between input and output variables are not obvious. They are referred to as

classification approaches. Support Vector Machines, Decision Trees, Neural Networks, Logistic Regression are examples of discriminative models.

3.2.3. Restricted Boltzman Machine

RBM is a type of neural network which is an unsupervised probabilistic generative model (Freund and Haussler, 1994). RBMs are focused in deeper after being proposed as building blocks of DBN, as employed in successful applications. The idea depends on that the hidden neurons extract relevant features from the training data. It is a feature learner having stochastic binary units, i.e. units that have values based on probability. An RBM is derived from Boltzman Machine, which is unrestricted form of the RBM as represented in Figure 3.4. An RBM has two layers; a visible layer (input layer) and a hidden layer as represented in Figure 3.5.

In a stochastic unit a value between 0 and 1 is calculated by the weighted input from other units plus a bias. The word ‘restricted’ refers to the two-part structure of the model, which disallows direct interaction between hidden units, or between visible units. Distribution of binary vectors are modelled by connection of feature extractors (hidden units) to input variables (visible units) for learning correlations in the data.

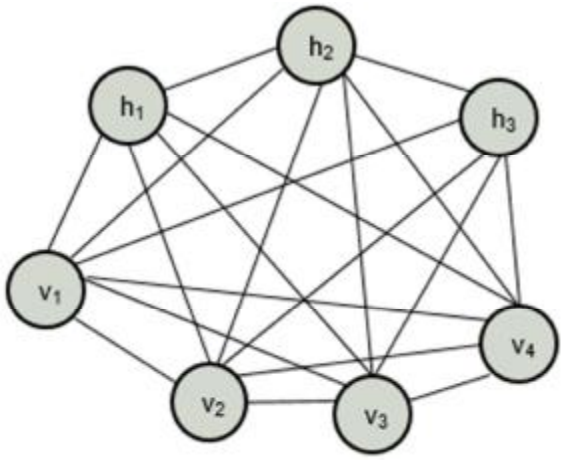


Figure 3.3. Unrestricted Boltzman machine with four visible units and three hidden units.

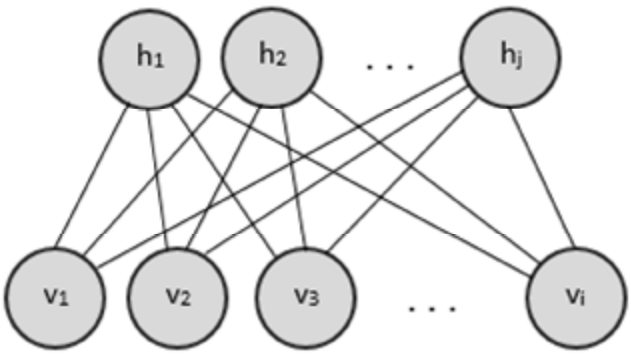


Figure 3.4. A Restricted Boltzman Machine with *i* visible units and *j* hidden units.

An RBM consists of a layer of *i* binary visible units and *j* binary hidden units having bidirectional weighted connections. The energy of a configuration of this network can be defined by

$$E(\mathrm{v}, \mathrm{h}) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{ij} h_j \tag{3}$$

where a_i is bias value of visible unit *i*, and b_j is bias value of hidden unit *j* and w_{ij} is the weight between them. The probability of a visible vector **v** is

$$p(\mathrm{v}) = \frac{1}{Z}\sum_{\mathrm{h}} e^{-E(\mathrm{v}, \mathrm{h})} \tag{4}$$

where *Z* is the normalizing factor calculated by summing all possible configurations of visible and hidden units. After observed variables are fed to visible units as input, a stochastic unit in a RBM has the probability of having value of 1.

$$p(h_j = 1|\mathrm{v}) = \sigma(b_j + \sum v_i w_{ij}) \tag{5}$$

where $\sigma(x) = 1/(1 + e^{-x})$ By using binary states of hidden units, the reconstructed binary states of visible units are

$$p(v_i = 1|\mathrm{h}) = \sigma(a_i + \sum_j h_j w_{ij}) \tag{6}$$

Each training step of an RBM works in a two-phase which can also be called as an encoder-decoder process. In the first phase, the hidden values are calculated according to Equation 5 as an encoding process. Then the visible units are calculated using the values of hidden values, this is the reconstruction of observed input values, i.e. decoding phase. The difference of observed variables and reconstructed values determine the update calculation of weights. This training process is called single step contrastive divergence (CD-1) algorithm. By CD-1, RBM learns a set of feature extractors from input data (Hinton, 2002). The CD-1 is repeated giving the next input until reaching an error of predetermined threshold value, therefore the weight vector is optimized. The steps of CD-1 algorithm are given as follows:

1. Take an input sample v_i, compute the stochastic hidden values and sample a hidden activation vector h_j.
2. Compute the $<v_i h_j>^0$, outer product of v_i and h_j (this is the positive gradient).
3. Calculate a reconstruction of visible units, using h. Then resample the hidden activations from the reconstruction.
4. Compute the $<v_i h_j>^1$ outer product of reconstructed visible values and hidden values from reconstruction (this the negative gradient).
5. Calculate the weight update Δw_{ij} by:

$$\Delta w = \varepsilon(< v_i h_j >^0 - < v_i h_j >^1) \tag{7}$$

where ε is the learning rate. Choosing learning rate perfectly is important, a small value of learning rate results in slow convergence, and a large value can skip the convergence. $< v_i h_j >^0$ is the outer product of input stochastic values and corresponding generated hidden values, and $< v_i h_j >^1$) is the outer product of reconstructed visible units and corresponding generated hidden values.

3.2.4. Deep Belief Networks

A DBN is a probabilistic generative model which is trained by using a Restricted Boltzmann Machine (RBM) to learn one layer of hidden features at a time (Hinton et al., 2006). After training process of the first layer is completed, a second layer of hidden units of RBM is added. The learned features of the first layer are taken as input for the second layer and the second layer is trained in the same fashion. The layers are added one after another until the desired number of layers is reached. Therefore a DBN is obtained by stacks of RBMs. Such a learning strategy of the DBN is called unsupervised greedy layer-wise learning method. The unsupervised greedy layer-wise learning provides an initialization in contrast to the traditional random initialization of neural networks.

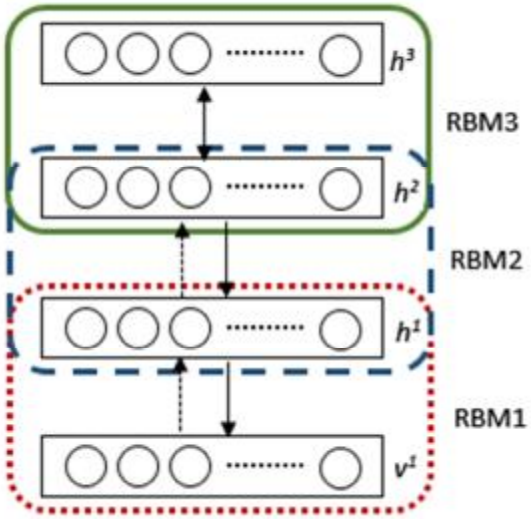


Figure 3.5. A traditional DBN composed of three layers of RBM

The architecture of a traditional DBN consists of a visible layer taking the input data and multiple hidden layers. Each layer of the DBN is trained according to the training procedure of RBM. After training of RBM1 is completed hidden units of RBM2 is added to the model. The hidden activations of RBM1 are fed to RBM2 as visible layer of RBM2 and the procedure is repeated for RBM3.

After the unsupervised greedy layer-wise pre-training is completed, fine tuning the DBN model is required for optimizing the weights. Fine tuning procedure differs according to the aim for which the model will be used. The aim can be a

generative model or a discriminative model. Figure 3.6 represents a traditional DBN that is used for generative purposes, and Figure 3.7 represents a discriminative DBN that is used for supervised classification purposes. In generative DBNs, wake-sleep algorithm is used for fine tuning. In wake phase of this algorithm the model is passed from down to up based on input **v** and the weights are adjusted. In the sleep phase, the model is passed from up to down based on hidden values **h** and the weights are again adjusted. In both phases, the weights are adjusted according to the Equation 8. These two phases are iterated until a convergence is reached.

$$\Delta W_{ji} \propto h_j(h_i - \hat{h}_i) \tag{8}$$

where h_j and h_i are hidden values of layer j and i respectively, h_i is reconstructed hidden values in layer i. At the beginning of the wake phase h_i is actual input values of visible layer, in a similar way in the ending of the sleep phase h_j is actual input values of visible layer.

Discriminative Deep Belief Networks (DDBN)

A Discriminative Deep Belief Network (DDBN) is a variant of DBN, which is used for supervised learning. In DDBN, the fine tuning procedure is performed by using back propagation algorithm. To obtain DDBN from DBN firstly, a new layer which is label units of the data is added like the output layer of a neural network for producing the desired outputs, i.e. o^1. Therefore, the weights of DDBN model are adjusted according to the difference of system output and actual expected values. In this model the last two layers are called associative memory for associating the lower layers to the label value.

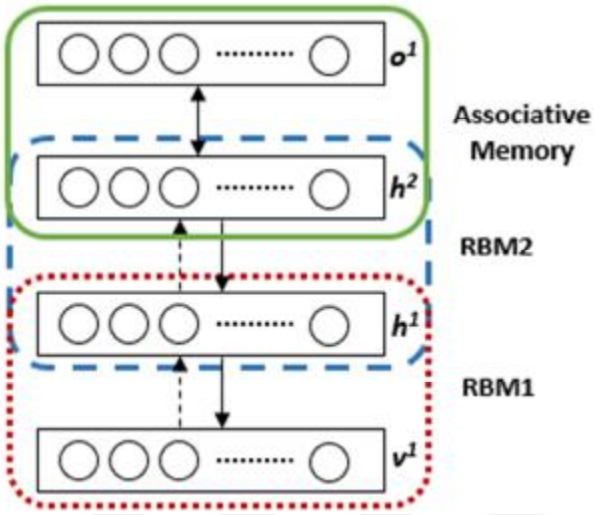


Figure 3.6. A DDBN composed of a visible unit, two RBM layers and an associative memory to output the target values.

DDBNs solve some problems of back propagation (BP) algorithm which is used to train traditional neural networks. BP is very slow when the network has multiple hidden layers. Secondly it is very probable to get stuck in local optima. Additionally, BP requires labelled data for learning, and then for mapping input to an output.

Hinton and Salakhutdinov (2006) show that DDBN often perform better than a neural network trained by only back propagation. The only thing one should pay attention is to choose the correct learning rate. Because a large value of learning rate change the pre-determined weights a lot, and a small value causes a slow convergence.

3.2.5. Convolutional Neural Networks (CNN)

Lecun et al. (1998) introduced CNN which is applied in many areas but showed a superiority when used with two dimensional data such as images and videos (Arel et al., 2010). A CNN is a special type of artificial neural network having the advantage of data dimensionality reduction. The “depth” of the model refers to constituting functions into a series of transformations. These transformations are done to convert inputs to intermediate representations and finally to output. Image

classification by a traditional neural network requires assigning each input pixel of the image to all of the hidden neuron units. This means a vast number of connections and calculations of weights as data is propagated back and forth through the network. In CNN, each neuron at a hidden layer receives regions of the image with size of the filter. The neurons' grid-like layout represents the input image of the next hidden layer and this network strategy goes on according to network design requirements. Thereafter the last grid of neurons is input to the hidden layer of an artificial neural network (ANN) in a full-connection form.

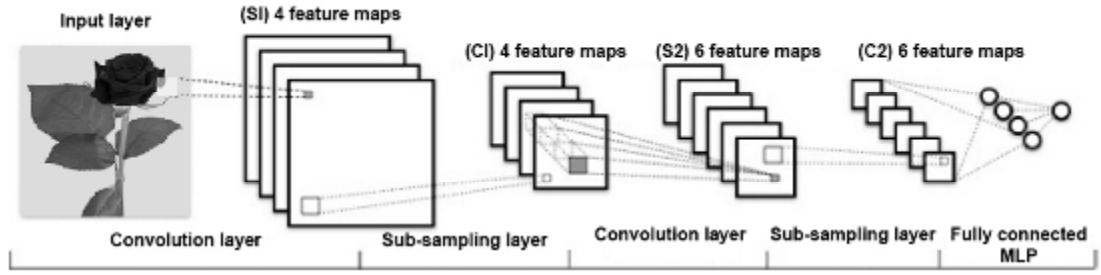
The image is fed to the system and convoluted by different filters (or known as convolution kernels) having pre-determined or learnt coefficients. Suppose *nxn* filter *w* is used, convolution result *x* is calculated by multiplying each image pixel *y* in layer *l-1* by corresponding filter coefficient and the products are all summed up as in Equation 9. The image size decreased *n-1* from number both of rows and columns.

$$x_{i,j}^{l} = \sum_{a}^{n} \sum_{b}^{n} w_{ab} y_{(i+a)(j+b)}^{l-1} \tag{9}$$

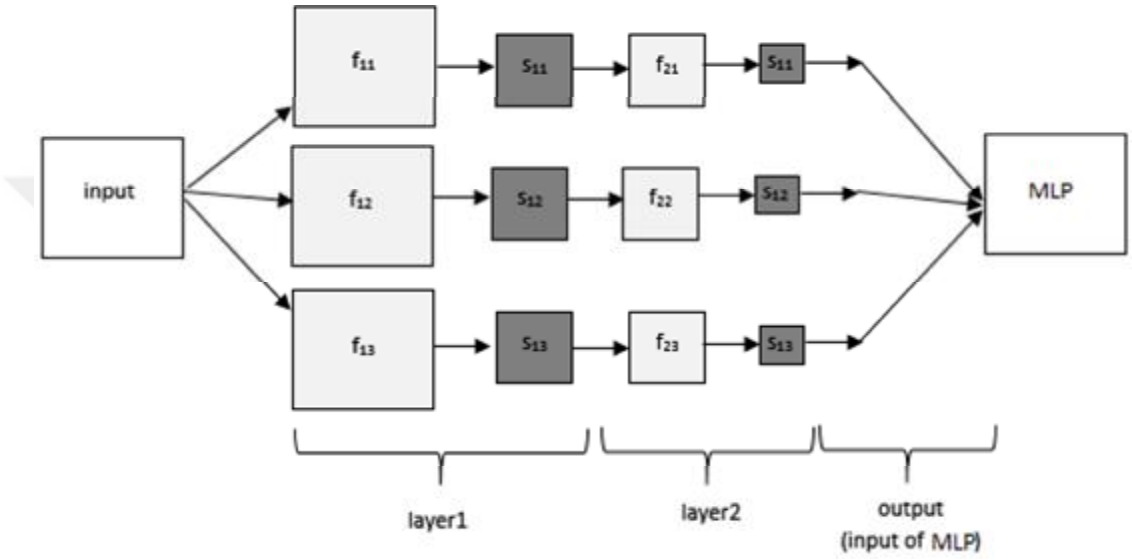
Nonlinear transformation is applied for each value generated by convolution by Equation 10.

$$y_{i,j}^{l} = \frac{1}{1+e^{-x_{i,j}^{l}}} \tag{10}$$

By each filter different features or aspects of input image, i.e. feature maps, come out. The output images of convolution is taken by pooling layer and images are subsampled. This operation step is for reducing the dimensionality and preserving critical information. Although several subsampling operations are in use, the widely used ones are 2x2 averaging or max-pooling. The 2x2 averaging means computing the mean of 4 adjacent pixel values and the max-pooling means taking the pixel having maximum value of these 4 adjacent ones.



(a)



(b)

Figure 3.7. Two different representations of CNN (a) the propagation of input in the network (b) layer wise representation

The convolution and sampling stages constitute one layer of the deep architecture of a CNN as represented in Figure 3.8. There isn't any greedy layer-wise pre-training like in Deep Belief Networks (DBN), but many layers can be used one after another. After the last sampling process, a traditional ANN continues. The resulting feature maps are categorized by the ANN. The final output of convolutional and subsampling layers is fed to a fully connected layer. More than one fully connected layers can be used and finally a softmax output layer of ANN is utilized for the final decision. The training is achieved in a supervised manner, and the traditional backpropagation algorithm can be used to train the whole model.

3.2.6. Stacked Autoencoders (SA)

An autoencoder is a type of artificial neural network which aims to reconstruct its own inputs. Therefore it is capable efficient encodings of the input, or it can be used for dimensionality reduction when the number of hidden units is lower than number of input units.

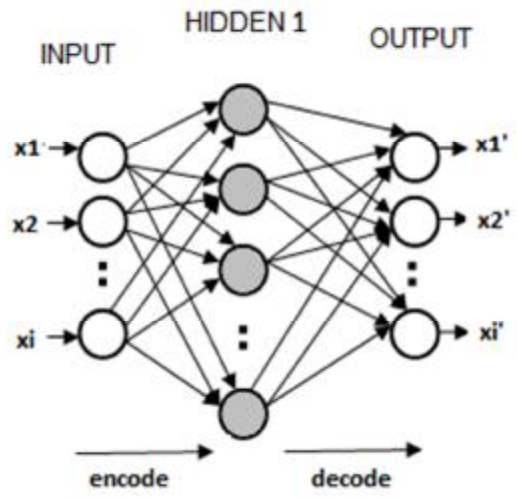


Figure 3.8. Representation of an autoencoder

The model is trained in a way that output target values are input values as represented in Figure 3.9. The weights are updated according to the difference between input values and produced values. For obtaining stacked autoencoders, after the model is trained and the weight values are determined the output layer is removed, and additional hidden layer is added, and hidden-output weights are input of the new hidden layer. At this time the model is trained for reconstructing these new input values. Therefore the greedy layer-wise approach is used also in stacked autoencoders.

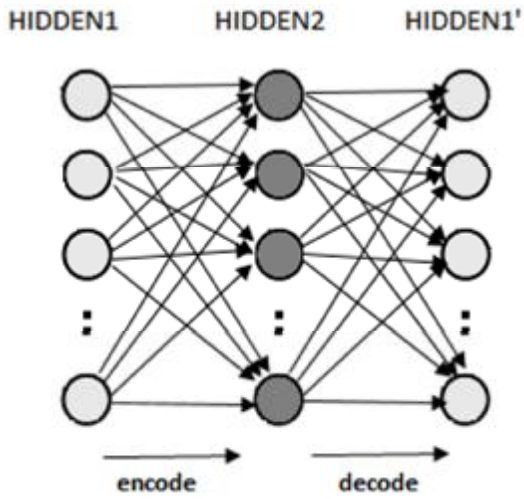


Figure 3.9. Representation of taking hidden layer as input layer and reconstructing it back.

After training the model in this step hidden1 to hidden2 layer weight values are determined. In this way required number of hidden layers can be added one after another in a greedy manner. Figure 3.10 represents the next step after generating the model in Figure 3.9.

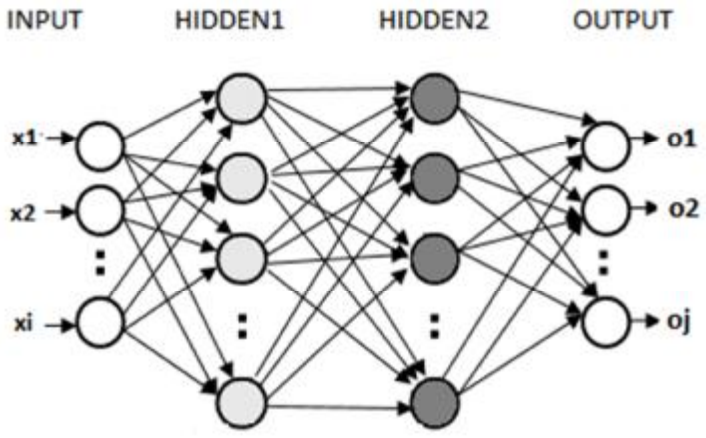


Figure 3.10. A two layered example of stacked autoencoders

After desired number of hidden layers are added in greedy layer-wise training fashion a last layer of output is added as represented in Figure 3.11. After this a supervised training is required for fine tuning the model. For that reason target class values are used by some back propagation way and therefore hidden2-output weight values are also determined.

3.2.7. Deep Neural Networks

Deep neural networks (DNNs) are simply traditional neural networks (NNs) having more than one hidden layer. NNs are computer systems that are inspired by biological systems. NNs are made up of many neurons and this system learns with examples as humans. Information is gained from environment in the process of learning by network. An NN is capable of performing some tasks that is impossible to be done by linear programming.

Information is not in a database or file, but in the neuron connections directly. This information, gained by learning, is used to decide on a new example that the network has never seen. A wide range of problems can be solved by using this property. Multilayer neural networks consist of an input layer, hidden layers, and an output layer. They are deep when they have multiple hidden layers as represented in Figure 3.12.

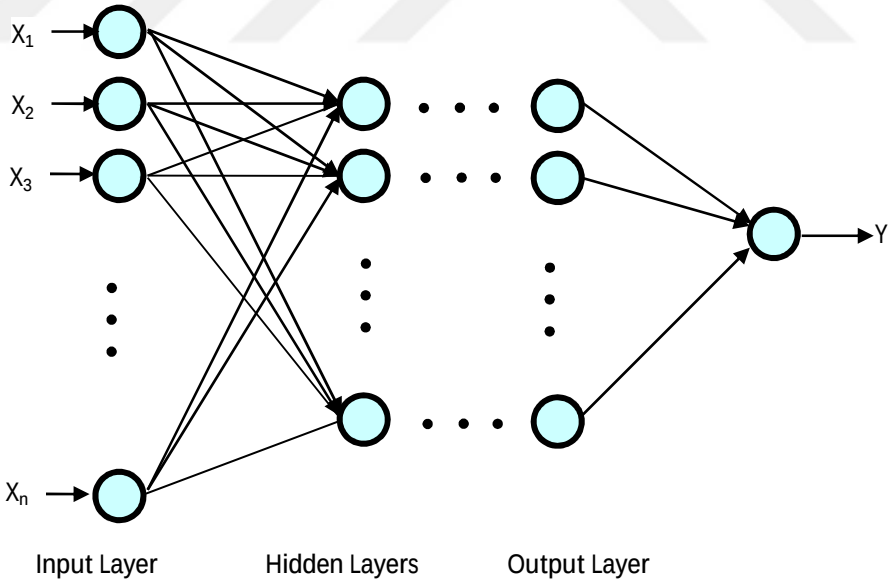


Figure 3.11. Representation of deep neural network with *n* input and an output

The power of ANNs occurs when more than one neuron is interconnected. When training data set includes two linearly separable classes, after a number of training iterations, the perceptron learns (Jain et al., 1996). A single layer network is made up of only input and output layer. There is no other layer of neurons, namely

no hidden layers. Learning is achieved by changing weights in each epoch, and epoch is an iteration of presenting the training data set to the network once (Haykin, 1999).

A neural network may have more than one input and an output layer. If there is an extra layer that supplies input to output layer of neurons, this layer is called hidden layer. Each layer has a W weight vector, b bias vector, and an output vector. There are many substantial points to be carefully decided while designing neural networks (Haykin, 1999). Some of them are number of hidden layers in the network, number of neurons in each hidden layer, finding a global optimum solution to prevent local minimum, finding an optimal solution in an acceptable time, testing the validity of network.

Back propagation (BP) algorithms are widely used in training multilayer networks. Weights and bias values are tried to be optimized according to the error of the network by using a BP algorithm. Initially the NN has random weights, it is required to update the weights to get the required output. The difference between expected output and produced output is the error of the model and this error acts as a feedback to update the weights. The gradient of error tells whether the error is increasing or decreasing, therefore the weight value is changed according to that gradient. This weight update procedure is called gradient descent, and repeated for all the weights. Due to the propagation of error is from output nodes to input nodes, the method named as backpropagation.

The standard back propagation algorithm for a feed forward network having layers of $q = 1,2,3,...Q$ is given as:

I. Network weight vector w is initialized randomly.
II. The output values are calculated in a forward manner according to the mapping function of the network:

$$y_i^q = f\left(\sum_i y_i^{q-1} w_{ij}^q\right) \tag{11}$$

where y_i^q is output of node *i* at layer *q*, w_{ij}^q is the weight value from node *i* to node *j* in layer *q*.

III. The errors at output nodes are calculated:

$$\delta_i^Q = (y_i^Q - y_i^p) f'(H_i^Q) \quad (12)$$

where H_i^Q is input of node i in last layer *Q*.

IV. In layers of $q = Q, Q-1, Q-2, \ldots 1$ for all *i* units in in hidden layers error values are calculated by back propagating

$$\delta_i^{q-1} = f'(H_i^{q-1}) \sum_i \delta_i^q w_{ij}^q \quad (13)$$

V. All the weights are updated by:

$$\Delta w_{ij}^q = \eta \delta_i^q y_i^{q-1} \quad (14)$$

$$w_{ij}' = w_{ij} + \Delta w_{ij}^q \quad (15)$$

where w_{ij}' is the new value of weight of connection from node *i* to node *j*.

VI. The steps through II to IV is repeated until the error is lower than a predetermined threshold value.

3.2.8. Baseline Methods

Naïve Bayes

Naïve Bayes classifier is a statistical learning method based on Bayes theorem. The model outputs the probabilities of class membership of a sample to each class.

Bayes' theorem is beneficial for calculating conditional probabilities. Bayes theorem can be defined as follows:

1. Assume A and B are two events, and Equation 16 gives the formula of the conditional probability of A, in such a condition that B has already occurred.

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \tag{16}$$

2. Therefore,

$$P(A \cap B) = P(A \mid B)P(B) \tag{17}$$

$$P(A \cap B) = P(B \mid A)P(A) \tag{18}$$

3. When right sides of Equation 17 an Equation 18 are equated Bayes' theorem is obtained:

$$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)} \tag{19}$$

Despite its simple nature, Naïve Bayes is one of the most efficient and well-known algorithms (Minsky, 1961). It estimates the class probabilities of a given sample and selects the class with maximum probability value as the decision. Naïve Bayes algorithm assumes that each attribute value of a sample is independent of other attributes, but the class value of the sample affects all the other attributes. This assumption, also known as "the conditional independence", simplifies the probability calculation. This independence assumption works very efficiently for problems in medical fields, probably connected to the fact that the chosen symptoms are independent to some degree (Sierra et al., 2001)

Naïve Bayes classifier is also called simple Bayesian classifiers and works as follows:

1. Assume that X is a data sample and has $X=\{x_1, x_2, \dots, x_n\}$ attribute values. Assume that there are m classes in the data set as $C_1, C_2, \dots, C_m$. Using Bayes' theorem the Equation 20 can be used to determine the class of X sample for each of the classes (the maximum class value is going to be regarded):

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)} \tag{20}$$

2. In Naïve Bayes classification it is assumed that x_i values are conditionally independent and therefore the computation is simplified. And now to calculate $P(X \mid C_i)$ the following equation can be used (Han and Kamber, 2000):

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) \tag{21}$$

3. $P(X)$ value will be the same in all class probability calculations and because the maximum class probability is going to be selected as class we can discard $P(X)$ from calculations.

$$\arg\max\{P(X \mid C_i)P(C_i)\} \tag{22}$$

Support Vector Machine (SVM)

SVM is a supervised machine learning method that depends on the statistical learning model proposed by Cortes and Vapnik (1995) to use with classification and regression tasks. The statistical learning theory, which is also known as the Vapnik-Chernovenkis (VC) theory, analyses the problem of estimating the function from the

training data. The models of statistical learning may be either parametric or nonparametric. SVM is a nonparametric model meaning that it has no assumptions on probability distributions of the input data and thus the model structure is not specified in advance.

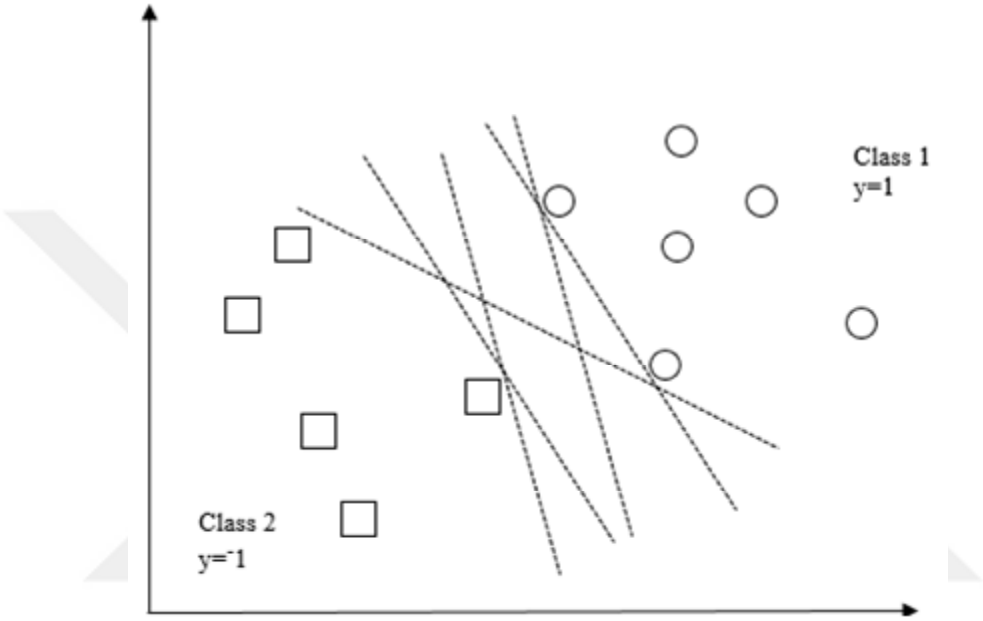


Figure 3.12. View of data that is separable linearly in two dimensional space

In SVM classification, the inputs are defined as *n*-dimensional feature vectors which are mapped to *m*-dimensional feature space by kernel functions, where $m>n$. Inputs are classified linearly by a hyperplane in such a higher dimensional space. By using the training data, SVM learns a decision surface, i.e. the hyperplane that separates the input vectors into two different classes for binary classification as represented in Figure 3.13. Therefore the aim of SVM classification is to design an optimum H_0 hyperplane that can be expressed by

$$g(\vec{x}) = \vec{w}^T\vec{x} + \omega_0 \tag{23}$$

where $\vec{w}$ is vector of weights and $\vec{x}$ is the input vector. The $\vec{w}$ vector determines the generalizing ability of learning.

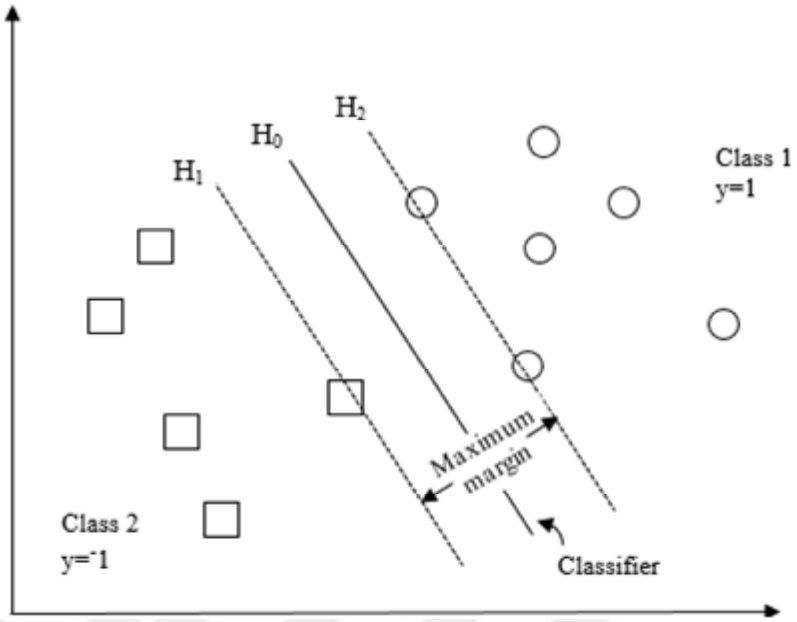


Figure 3.13. The probable maximum distance between linearly separable data

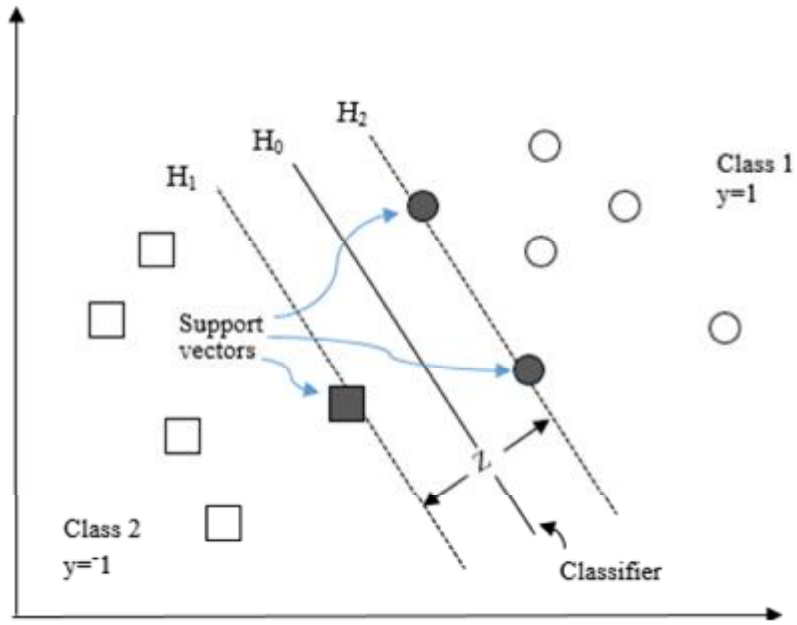


Figure 3.14. Representation of support vectors in two dimensional space

Figure 3.14 shows that there can be more than one solution in determining the hyperplane but the best one leaves maximum margin from both classes. This margin is determined according to the closest data points called support vectors as represented in Figure 3.15. H_1 and H_2 are the hyperplanes parallel to H_0 plane intersecting the samples closest to H_0 hyperplane. For input vectors from class 1 the

hyperplane function produces values larger than 1, and for input vectors from class 2 it produces values smaller than -1.

$$g(\vec{x}) \geq 1, \forall \vec{x} \in class\ 1$$
$$g(\vec{x}) \leq -1, \forall \vec{x} \in class\ 2 \quad (24)$$

Let *Z* be the margin distance from support vectors of one of the classes to the hyperplane, then

$$Z = \frac{|g(\vec{x})|}{\|\vec{w}\|} = \frac{1}{\|\vec{w}\|} \quad (25)$$

where we can see the total margin from both classes is $2/\|\vec{w}\|$ which must be maximized by minimizing the term $\|\vec{w}\|$ for maximum separability. In case the two classes are non-separable, the amount proportional to number of misclassified samples must also be minimized as in Equation 26,

$$\min_{\vec{w},\omega_0,\xi} \frac{1}{2}\|\vec{w}\|^2 + C\sum_1^N \xi_i \quad (26)$$

where ξ_i is the error parameter determined by input vectors that are not separated linearly and located on the wrong side of hyperplane. C is a predetermined constant parameter to control the penalty for misclassification. If this value is too small, many input vectors of misallocated are accepted, otherwise very few input vectors on wrong side are required.

The search for the optimum hyperplane in SVM is a nonlinear optimization problem solved by using Karush Kuhn Tucker conditions which use Lagrange multipliers

$$L(\vec{w}, \omega_0, \lambda) = \frac{1}{2}\|\vec{w}\|^2 - \sum_{i=1}^N \lambda_i (y_i((\vec{w}^T\vec{x}_i + \omega_0) - 1) \quad (27)$$

where y_i is the corresponding class label of $\vec{x}_i$.

What makes the SVM approach being effective is the kernel functions that map the input vectors to feature vectors. By an appropriate mapping, data can be transformed to another dimensionality in which it can be separated by a hyperplane. Quadratic, polynomial, sigmoid, linear and radial basis functions are some common kernel functions.

Random Forest (RF)

Breiman (2001) introduced the Random Forests algorithm which are an ensemble of decision trees for classification and regression. Training process is construction of a number of decision trees and output of the model is an agreed decision of the member trees. The method inspires the Bagging idea in the opinion that if a group of weak learners come together they may form a strong learner. A random subset of the training data is used for construction of a member decision tree. After the model is trained each test row is passed to obtaining an output prediction value.

The algorithm of training each tree is given as:

I. Assume N is the number of cases in the training data, a random number of samples from N cases are sampled, and used for training of a tree in the model.
II. Assume M is the number of features in the data and m a constant number that holds a random subset of M. To split the node in a tree the best split on these m is used. (In standard trees each node is split using the best split among all variables. If all of the features were used for each tree the member trees would be nearly similar to each other, this situation lead to a higher error rate (L. Breiman 2001).
III. Choose a training set for the tree by sampling n times with replacement from N available cases. The rest of the cases are used for error estimation for the model.
IV. The tree is grown fully without any pruning.

Algorithm steps are applied for each tree to be used in the model. After completion of this training process of the model new data samples are predicted by aggregating the predictions of all trees in the model. If the model is used for classification the decision is according to majority votes, or if the model is used for regression the decision is the average of outputs of member trees.

In case of microarray gene expression data classification Random Forest is stated to be a robust machine learning approach by Díaz-Uriarte and De Andres (2006). They concluded their study by the statement that "Because of its performance and features, random forest and gene selection using random forest should probably become part of the standard tool-box of methods for class prediction and gene selection with microarray data".

K-Nearest Neighbors

K-nearest neighbor (Dasarathy, 1990) is instance-based method which has been used in a wide variety of applications in literature. The principal idea behind the method is that a new sample is classified according to the samples of observations that have known classification labels. The distance of new sample to the samples in the training data is calculated, and the samples having minimum distance to the new samples are selected. The number of samples to be selected is k. The method is simple and very accurate, but the performance is directly affected by the number of samples, because the distances are calculated one by one for each sample in data.

Distance functions are used for the calculation of distance for i and j points. Euclidean, Manhattan and Minhowski distance functions are given in Equation 28, Equation 29 and Equation 30, respectively.

$$E_{d(i,j)} = \sqrt{\sum_{i=1}^{k}(x_i - x_j)^2} \tag{28}$$

$$Mt_{d(i,j)} = \sum_{i=1}^{k}|x_i - y_j| \tag{29}$$

$$Mw_{d(i,j)} = \left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q} \quad (30)$$

k-Nearest Neighbors algorithms is the application of the following steps in a dataset including observation values:

I. The number of the nearest neighbors (the k parameter) is determined.
II. All the distances from each data point to the new sample is calculated.
III. The data points in dataset are sorted according to the distances calculated in the previous step, and k ones having the minimum distance are selected.
IV. Labels of the selected data points are determined, and the label having the majority in them is selected.
V. The selected label is accepted as the label of new sample.

According to the examples represented in Figure 3.16, 11 data samples of circles and squares are given nearest neighbors are determined for $k = 4$. The decision label for the sample is circle, due to its majority in nearest neighbors.

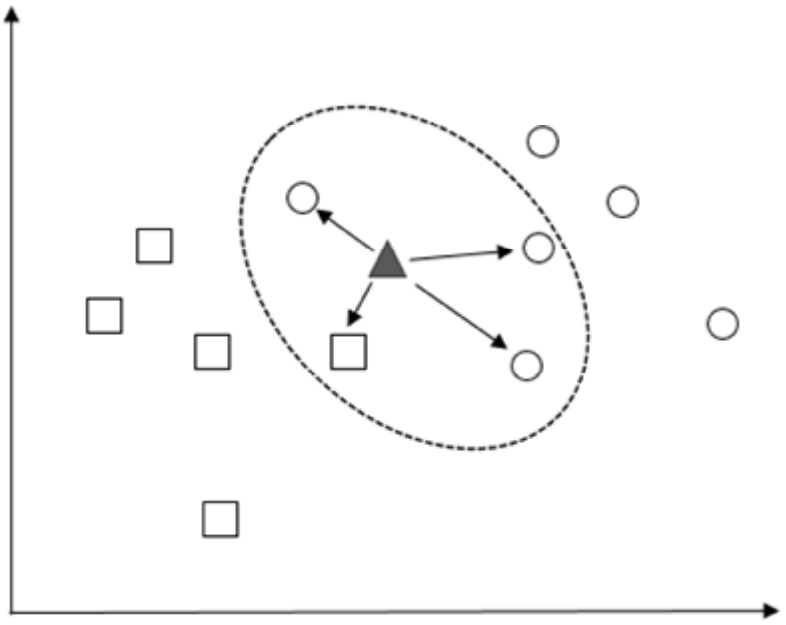


Figure 3.15. Determining k = 4 nearest neighbors given a new point.

The determination of label of new sample according to the majority class label may not be convenient in some cases. Instead of this, a method called inverse distance-weighted voting can be applied. This weighted voting method depends on the calculation of Equation 31:

$$d(i,j) = def \begin{cases} \infty & if\ d(i,j) = 0 \\ \frac{1}{d(i,j)^2} & otherwise \end{cases} \quad (31)$$

where ***d(i,j)*** a distance value obtained by a distance function. For each class labels ***d(i,j)*** values are summed and the class label having the highest value is returned.

3.2.9. Information Gain Feature Selection

Information Gain (IG) measures the reduction in uncertainty about a class variable, therefore it is entropy based filter method. Formally, assume ***Y*** is the class attribute of a data set and ***X*** is a given feature such that both are discrete. The IG of ***X*** is the reduction of uncertainty of ***Y*** values when ***X*** values are known. This uncertainty is measured as H(Y), the entropy of Y. IG of X is the difference between entropy of ***Y*** and entropy of ***Y*** after ***X*** values are observed, and is calculated as in Equation 32.

$$IG(Y;X) = H(Y) - H(Y|X) \quad (32)$$

Entropy of ***Y*** is calculated with Equation 33.

$$H(Y) = -\sum_{y \in Y} p(y) log_2(p(y)) \quad (33)$$

where ***y*** is a value of ***Y*** class feature, and ***p(y)*** is the probability of ***Y=y***. And entropy of ***Y*** after ***X*** values are observed is calculated as:

$$H(Y|X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) log_2(p(y|x)) \quad (34)$$

3.2.10. Synthetic Minority Over-Sampling Technique (SMOTE)

SMOTE, proposed by Chawla et al. (2002), is an over-sampling technique for avoiding decrease in classifier performance caused by class imbalance in the dataset. Classifiers including the decision tree and the neural networks are observed to bias towards a major class when learning from an imbalanced dataset (Chawla, 2003). Instead of replicating instances, like traditional over-sampling methods, SMOTE creates new instances from minor class 'synthetically'. It assumes that a minor class instance and its closest vectors have the same class value.

A sample having the class value of minor class is called a minority sample. In SMOTE, each instance is considered as a vector, and synthetic samples are generated along the line between the minority sample and its nearest neighbor. Assuming T is the number of minority samples in dataset, the SMOTE algorithm uses two parameters; N is the percent of T to be generated as synthetic samples; i.e. oversampling ratio, and k is the number of nearest neighbors of a minority sample. Therefore the input of algorithm is T number of minority samples, and output is synthetic samples that count to $(N/100)*T$. If N is selected as a number smaller than 100, then minority class samples are selected randomly. In our study, as a result of experiments to optimize the parameters of SMOTE, it is observed that best results are obtained with parameters of $N = 100$ and $k = 5$. By using N as 100 all minority class samples are used and for each of them k-nearest neighbors are found. One neighbor sample is selected from k-nearest neighbors randomly for each minority class sample. A vector x_{new} is calculated by taking the difference between the minority class sample x_n and the neighbor sample. Finally, a new synthetic sample vector x_{new} is generated by:

$$x_{new} = x + (x - x_n)\gamma \quad (35)$$

where γ is a random value between *0* and *1*.

In the case of microarray gene expression data classification, especially laryngeal and colorectal data are imbalanced. However SMOTE is applied to all three data of laryngeal, bladder and colorectal. The aim is to minimize the potential misclassification caused by an unbalanced structure of data.

3.2.11. Local Binary Patterns

Local Binary Patterns (LBP) is a local feature summarizer not only for better texture classification, also face detection, face recognition and image segmentation. Ojala et al. (1996) introduced the original version of LBP, in which center pixel of 3x3 image block is labelled according to its intensity of neighbors. The pixels of the image is thresholded by the center of the block and a binary code is produced. The decimal correspondence of this binary code is called the LBP code. Component-wise multiplication is done with this code by the weight vector of powers of 2. LBP is extended for further texture and image analysis by using different number of neighbors and distance of the neighbors as defined in Equation 36.

$$LBP_{P,R}(x_i) = \mu(x_p - x_i)2^p \tag{36}$$

Where

$$\mu(k) = \begin{cases} 1, k \geq 0 \\ 0, k < 0 \end{cases} \tag{37}$$

In Equation 36, P represents the number of neighbor pixels, and R is distance of neighbors to the central pixel. Different LBP analysis of textures can be achieved by employing different values for P and R. For $P=8$ and $R=1$, a circular eight pixels of neighborhood is used in 3x3 sub-image.

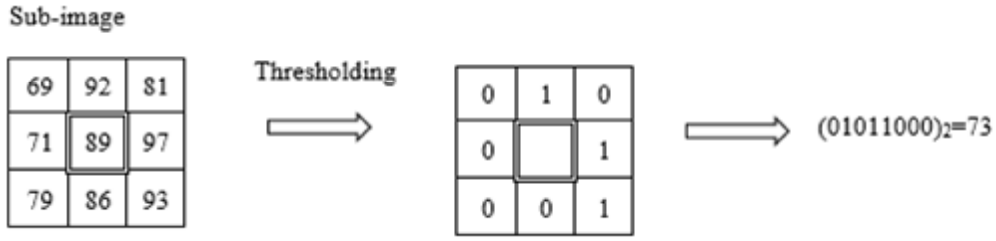


Figure 3.16. Generation of binary patterns for a pixel

When $LBP_{8,1}$ is applied to a *3x3* sub-image, *256 (2^8)* different binary patterns can be produced after thresholding as Figure 3.17 represents. When the image is rotated even if slightly, a completely different LBP code would be produced for the same sub-images. This situation results in dependence on point of view when taking the image. To eliminate this imperfection each binary code is shifted in anti-clock wise direction until achieving the minimum corresponding decimal.

$$LBP_{P,R}^{ri} = \min_{k} ROR(LBP_{P,R}, k) \tag{38}$$

where ***ROR(x,i)*** is the function of a bitwise rotation of bit sequence *x* by *k* steps. For binary code above 01011000 is shifted once 10110000 is obtained. For second, third, fourth and fifth shift we gain 01100001, 11000010, 10000101, 00001011 binary codes respectively. The fifth one is the minimum we would obtain therefore the LBP code for this block is eleven. The approach can be applied to gain the maximum code in the same way. Considering all the blocks in the image all the LBP codes will be in a standard mode and rotation invariant.

3.2.12. Block Difference of Inverse Probabilities (BDIP)

BDIP is proposed by Chun et al. (2003) for extracting sketch features considering local intensities. They studied on a variant of the difference of inverse probabilities (DIP) (Ryoo and Kim, 1988) for discovering edges and valleys by their block based approach. BDIP is defined as the difference between number of pixels in a block and the ratio of sum of pixel intensities to the maximum intensity value in that block.

$$BDIP = M^2 - \frac{\sum_{(i,j)\in B} I(i,j)}{\max_{(i,j)\in B} I(i,j)} \quad (39)$$

where M^2 is the block size, e.g. *2x2*, B is the block of pixels. *I(i,j)* is the intensity of pixels at coordinates of *i* and *j* in block B. To apply BDIP, the image is split into blocks and for each block the Equation 39 is applied to all blocks. For a block size of *2x2* the image is reduced to its half in both width and height.

4. RESULTS AND DISCUSSION

4.1. Evaluation Metrics

The calculation of most of the metrics requires the outcomes of the classifier system to be labeled with four possible states, as true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). True Positives (TP) refers to the number of samples predicted as positive and actually they are positive. False Positives (FP) refers to the number of samples predicted as positive but actually they are negative. False Negatives (FN) refers to the number of samples predicted as negative but actually they are positive. True Negatives (TN) refers to the number of samples predicted as negative and actually they are negative. These possible states take place in a confusion matrix as represented in Figure 4.1.

		Classified As	
		Negatives	Positives
Actual Class Value	Negatives	TN	FP
	Positives	FN	TP

Figure 4.1. A confusion matrix for a binary classification. Correctly classified instances are represented by TN and TP, whereas incorrectly classified instances are represented by FP and FN.

Once the output for each sample is labeled, quantitative metrics can be calculated for the evaluated algorithm. TPR (True Positives Rate) is the ratio of number of correctly predicted positives to the total of actual positives.

$$TPR = \frac{TP}{FN + TP} \tag{40}$$

FPR (False Positives Rate) is the ratio of incorrectly predicted positives to the total of actual negatives, and defined as:

$$FPR = \frac{FP}{FP + TN} \tag{41}$$

Accuracy is the measure to show the classifier's ability of correctly predicting the given samples in overall the data and defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{42}$$

The F-measurement (F-m) is described as a harmonic mean of two other classification metrics; precision and recall.

$$F - measure = \frac{precision.recall}{precision + recall} \tag{43}$$

Precision is the ratio of truly classified positives to all predicted positives defined as:

$$Precision = \frac{TP}{TP + FP} \tag{44}$$

Sensitivity (recall), also known as TPR, is the ratio of number of correctly predicted positives to the total of actual positives, and defined as:

$$Sensitivity\ (Recall) = \frac{TP}{FN + TP} \tag{45}$$

Specificity is the ratio of number of correctly predicted negatives to the total

of actual negatives, and defined as:

$$Specificity = \frac{TN}{TN + FP} \tag{46}$$

Mean Absolute Error (MAE) is a metric for measuring how far the predicted values from the actual values by giving the average of absolute errors.

$$MAE = \frac{1}{n}\sum_{i}^{n}|f_i - y_i| \tag{47}$$

where f_i denotes for predicted values, y_i denotes for actual values, and n is the number of instances in data.

4.2. Employing Discriminative Deep Belief Networks for Low and High Dimensional Biomedical Data

Table 4.1, Table 4.2, Table 4.3 and Table 4.4 show the results of classification of DDBN and presents the values of evaluation metrics by using one, two, three and four hidden layers, respectively. In each hidden layer 10 units are used. As the average values verify, generally the optimum results are achieved by using single hidden layer in a DDBN architecture for LDD. Comparing to one-hidden layer a slight improvement is gained in heart disease data by utilizing two additional hidden layers. A four hidden layered DDBN architecture is not able to converge in six of eight data. The evaluation results which are presented italic in Table 4.4 mean that all the instances are classified to one class of data.

Table 4.1. Evaluation results of DDBN with one hidden layer by using LDD

	DDBN (1 hid. lay.)			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
Arrhythmia	0.279	0.705	0.265	72.13
Heart disease (Cleveland)	0.472	0.557	0.200	55.75
Vertebral column (2C)	0.214	0.857	0.240	82.58
Diabetes (Pima Indians)	0.308	0.654	0.205	75.00
Wisconsin breast cancer	0.046	0.946	0.028	96.28
Mammographic mass	0.179	0.766	0.132	82.10
CTG	0.160	0.861	0.188	86.07
Parkinson's	0.193	0.925	0.500	81.92
Average	**0.231**	**0.784**	**0.220**	**78.98**

Table 4.2. Evaluation results of DDBN with two hidden layers by using LDD

	DDBN (2 hid. lay.)			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
Arrhythmia	0.335	0.604	0.275	65.92
Heart disease (Cleveland)	0.473	0.561	0.172	56.10
Vertebral column (2C)	0.217	0.844	0.378	76.45
Diabetes (Pima Indians)	0.310	0.616	0.210	73.04
Wisconsin breast cancer	0.049	0.930	0.035	95.28
Mammographic mass	0.239	0.790	0.154	81.90
CTG	0.162	0.850	0.174	84.99
Parkinson's	0.207	0.891	0.458	80.34
Average	**0.249**	**0.761**	**0.232**	**76.75**

Table 4.3. Evaluation results of DDBN with three hidden layers by using LDD

	DDBN (3 hid. lay.)			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
Arrhythmia	0.309	0.763	0.396	69.00
Heart disease (Cleveland)	0.543	0.570	0.423	57.03
Vertebral column (2C)	0.262	0.906	0.305	82.28
Diabetes (Pima Indians)	0.310	0.652	0.192	75.26
Wisconsin breast cancer	0.045	0.934	0.031	95.71
Mammographic mass	0.247	0.806	0.170	81.89
CTG	0.253	0.815	0.492	81.47
Parkinson's	0.306	0.812	0.537	73.58
Average	**0.284**	**0.782**	**0.318**	**77.03**

Table 4.4. Evaluation results of DDBN with four hidden layers by using LDD

	DDBN (4 hid. lay.)			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
Arrhythmia	0.415	0.808	0.666	59.07
Heart disease (Cleveland)	*0.646*	*0.000*	*0.000*	*54.10*
Vertebral column (2C)	*0.486*	*0.000*	*0.000*	*67.74*
Diabetes (Pima Indians)	*0.452*	*0.000*	*0.000*	*65.10*
Wisconsin breast cancer	0.287	0.941	0.244	78.26
Mammographic mass	*0.495*	*0.000*	*0.000*	*53.69*
CTG	*0.372*	*0.000*	*0.000*	*77.86*
Parkinson's	*0.391*	*0.000*	*0.000*	*75.37*
Average	**0.443**	**0.219**	**0.114**	**66.40**

Table 4.5. Evaluation results of Naive Bayes by using LDD

	Naive Bayes			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
Arrhythmia	0.231	0.898	0.382	76.99
Heart disease (Cleveland)	0.190	0.561	0.190	56.11
Vertebral column (2C)	0.220	0.733	0.130	77.74
Diabetes (Pima Indians)	0.284	0.612	0.156	76.30
Wisconsin breast cancer	0.041	0.975	0.048	95.99
Mammographic mass	0.237	0.836	0.262	78.36
CTG	0.091	0.872	0.121	87.16
Parkinson's	0.307	0.619	0.083	69.23
Average	**0.200**	**0.763**	**0.172**	**77.24**

Table 4.6. Evaluation results of k-NN by using LDD

	k-NN			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
Arrhythmia	0.364	0.800	0.556	63.72
Heart disease (Cleveland)	0.188	0.551	0.247	55.12
Vertebral column (2C)	0.186	0.833	0.220	81.61
Diabetes (Pima Indians)	0.300	0.530	0.206	70.18
Wisconsin breast cancer	0.047	0.917	0.028	95.28
Mammographic mass	0.258	0.757	0.256	75.03
CTG	0.063	0.911	0.165	91.06
Parkinson's	0.041	0.966	0.042	96.41
Average	**0.181**	**0.783**	**0.215**	**78.55**

Table 4.7. Evaluation results of SVM by using LDD

	SVM			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
Arrhythmia	0.241	0.853	0.353	75.88
Heart disease (Cleveland)	0.272	0.571	0.209	57.10
Vertebral column (2C)	0.213	0.938	0.530	78.71
Diabetes (Pima Indians)	0.227	0.541	0.102	77.34
Wisconsin breast cancer	0.030	0.963	0.026	97.00
Mammographic mass	0.208	0.834	0.244	79.19
CTG	0.250	0.963	0.285	88.76
Parkinson's	0.128	0.993	0.500	87.18
Average	**0.196**	**0.832**	**0.281**	**80.14**

Table 4.8. Evaluation results of Random Forest by using LDD

	Random Forest			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
Arrhythmia	0.352	0.886	0.280	80.97
Heart disease (Cleveland)	0.205	0.568	0.240	56.77
Vertebral column (2C)	0.205	0.881	0.260	83.55
Diabetes (Pima Indians)	0.314	0.590	0.166	74.87
Wisconsin breast cancer	0.059	0.959	0.031	96.57
Mammographic mass	0.248	0.769	0.169	80.23
CTG	0.063	0.953	0.110	95.34
Parkinson's	0.167	0.946	0.186	90.26
Average	**0.202**	**0.819**	**0.180**	**82.32**

One hidden layered DDBN produced the best results for LDD. According to the evaluation results, one hidden layered DDBN produces approximately an average of results among of others. Random Forest outperforms the other methods including DDBN considering average accuracy. With respect to MAE, k-NN outperforms others, and with respect to TPR, SVM outperforms the others. Achieving the value of 0.172 Naive Bayes wins at the FPR metric. According to these results, it is obvious that DDBN's performance is comparable with the other algorithms. Therefore, DDBN approach does not show any advantage over the traditional classification algorithms for low dimensional biomedical data.

Figure 4.2. shows the status of DDBN comparing to other methods. If vertebral column and Wisconsin breast cancer data is excluded, all other data are best

classified by other methods, however DDBN is never the most left behind. Note that the performance of a classifier depends on data, for example for arrhythmia k-NN is worst of five methods, and however it is the best for Parkinson's disease by accuracy of 96.4%.

As a deep model DDBN tends to extract a relevant set of features from the input layer, it improves the recognition task by the next layer according to these features. However when the amount of attributes of training data is limited the model is the best with a single layer without fulfilling the essential function. Namely a single hidden layered DDBN is not able to compensate a real deep model, but when the additional layers are used the sparse of features to be extracted obstruct DDBN in case of LDD.

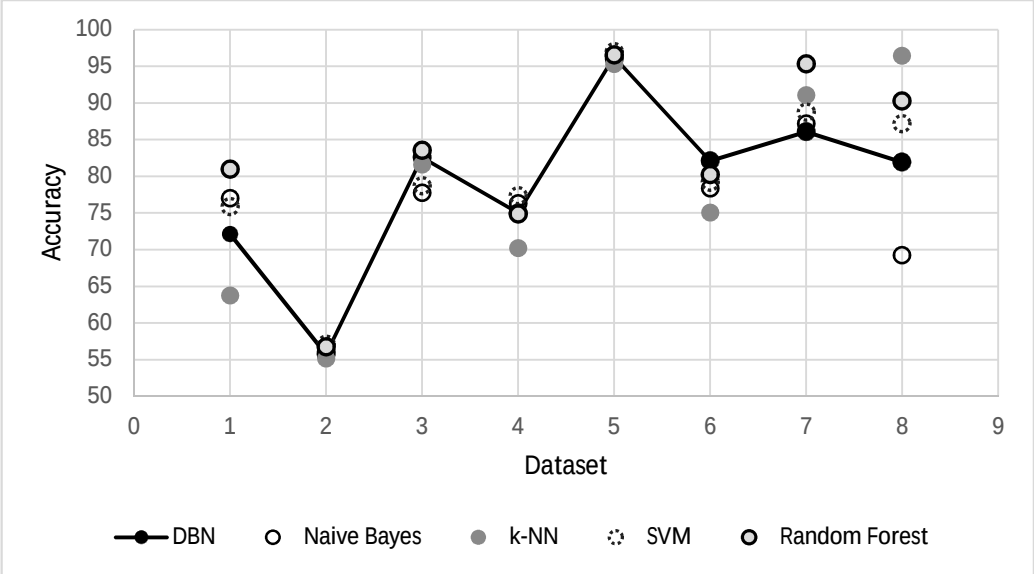


Figure 4.2. Representation of accuracies produced by DDBN and other methods using LDD

The results of four architectures of discriminative DDBN in terms of number of hidden layers are represented in Table 4.9, Table 4.10, Table 4.11 and Table 4.12. We used 50 nodes in the first hidden layer, and 10 in each of the other hidden layers. Table 4.9, Table 4.10, Table 4.11 and Table 4.12 show that two-hidden layer DDBN produces the best results on HDD by the average MAE, TPR, FPR and Acc. values of 0.153, 0.741, 0.153 and 81.19% respectively. As it can be seen in Table 4.11 and

Table 4.12, only results on lung cancer data are failed in three and four layered DDBN. Considering Table 4.4 in the previous section, four hidden layered DDBN failed in six of eight LDD, and considering Table 4.11 and Table 4.12 three or four hidden layered DDBN is failed in only one of five HDD. Therefore it is obvious that high dimensionality in input data is essential for DDBN evaluation. The previous hidden layer acts as a feature detector for the next hidden layer, besides for aforementioned HDD it is obvious that the hidden layers come upon next to the second hidden layer are redundant in data of such dimensionality.

Table 4.9. Evaluation results of DDBN with one hidden layer by using HDD

	DDBN (1 hid. lay.)			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
p-53 mutants (site 3)	0.446	0.587	0.451	57.27
Breast Cancer	0.510	0.457	0.471	49.11
Lung Cancer	0.040	0.871	0.000	97.84
Ovarian Cancer	0.014	0.994	0.022	98.83
Laryngeal Cancer	0.358	0.412	0.253	64.22
Average	**0.274**	**0.664**	**0.239**	**73.45**

Table 4.10. Evaluation results of DDBN with two hidden layers by using HDD

	DDBN (2 hid. lay.)			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
p-53 mutants (site 3)	0.046	0.651	0.275	68.48
Breast Cancer	0.351	0.652	0.294	67.89
Lung Cancer	0.036	0.903	0.000	98.39
Ovarian Cancer	0.009	1.000	0.011	99.62
Laryngeal Cancer	0.322	0.500	0.187	71.56
Average	**0.153**	**0.741**	**0.153**	**81.19**

Table 4.11. Evaluation results of DDBN with three hidden layers using HDD

	DDBN (3 hid. lay.)			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
p-53 mutants (site 3)	0.383	0.698	0.471	62.27
Breast Cancer	0.439	0.587	0.431	57.67
Lung Cancer	*0.234*	*0.000*	*0.000*	*83.22*
Ovarian Cancer	0.016	0.993	0.033	98.45
Laryngeal Cancer	0.411	0.294	0.293	57.79
Average	**0.297**	**0.514**	**0.246**	**71.88**

Table 4.12. Evaluation results of DDBN with four hidden layers using HDD

	DDBN (4 hid. lay.)			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
p-53 mutants (site 3)	0.508	0.810	0.980	45.98
Breast Cancer	0.438	0.391	0.372	51.54
Lung Cancer	*0.314*	*0.000*	*0.000*	*83.22*
Ovarian Cancer	0.148	0.975	0.374	85.32
Laryngeal Cancer	0.446	0.118	0.227	56.88
Average	**0.371**	**0.459**	**0.390**	**64.58**

Table 4.13. Evaluation results of Naive Bayes by using HDD

	Naive Bayes			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
p-53 mutants (site 3)	0.368	0.508	0.216	63.16
Breast Cancer	0.454	0.043	0.000	54.64
Lung Cancer	0.017	0.968	0.013	98.34
Ovarian Cancer	0.075	0.895	0.022	92.49
Laryngeal Cancer	0.349	0.500	0.280	65.14
Average	**0.253**	**0.583**	**0.106**	**74.75**

Table 4.14. Evaluation results of k-NN by using HDD

	k-NN			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
p-53 mutants (site 3)	0.345	0.730	0.431	65.79
Breast Cancer	0.394	0.457	0.255	60.82
Lung Cancer	0.050	0.742	0.000	95.58
Ovarian Cancer	0.044	0.975	0.066	96.05
Laryngeal Cancer	0.423	0.324	0.307	57.59
Average	**0.251**	**0.646**	**0.212**	**75.17**

Table 4.15. Evaluation results of SVM and Random Forest by using HDD

	SVM			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
p-53 mutants (site 3)	0.375	0.730	0.510	62.50
Breast Cancer	0.320	0.696	0.333	68.04
Lung Cancer	0.006	0.968	0.000	99.44
Ovarian Cancer	0.000	1.000	0.000	100.0
Laryngeal	0.431	0.294	0.307	56.88
Average	**0.226**	**0.738**	**0.230**	**77.37**

Table 4.16. Evaluation results of SVM and Random Forest by using HDD

	Random Forest			
Data	**MAE**	**TPR**	**FPR**	**Acc.**
p-53 mutants (site 3)	0.419	0.762	0.510	64.29
Breast Cancer	0.461	0.522	0.255	63.92
Lung Cancer	0.126	0.903	0.000	98.34
Ovarian Cancer	0.181	0.994	0.088	96.44
Laryngeal	0.410	0.147	0.067	68.81
Average	**0.319**	**0.666**	**0.184**	**78.36**

Table 4.13, Table 4.14, Table 4.15 and Table 4.16 show that the two-hidden layered DDBN outperforms other DDBN architectures considering all evaluation metrics. In p53 mutants and laryngeal cancer data, it also surpasses other four algorithms. In classification of breast, lung and ovarian cancer data, SVM is the best predictor, especially in the classification of ovarian cancer with a result of 100% accuracy rate. However the difference between the results of DDBN and SVM are very slight according to accuracy metric. The performances of these two algorithms in breast cancer, lung cancer and ovarian cancer datasets are 67.89% to 68.04%, 98.39% to 99.44% and 99.62% to 100%, respectively. Figure 4.3 represents the comparison of the classification performance of the methods.

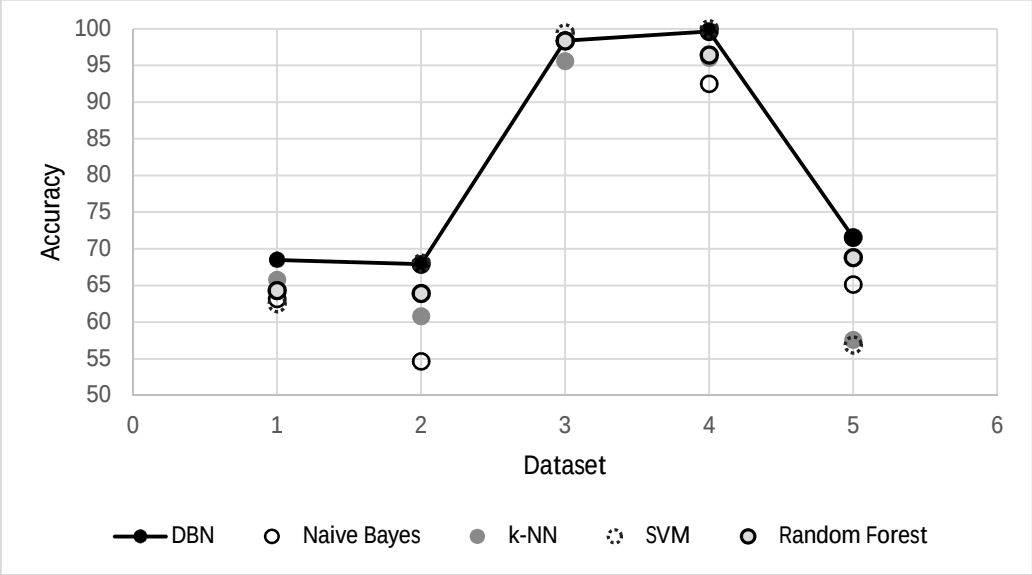


Figure 4.3. Representation of accuracies produced by DDBN and other methods using HDD

As a result, the DDBN approach appears to be superior in HDD classification tasks whereas its performance is close to mediocre in LDD classification. This consequence may be linked to the fact that each RBM layer in the DDBN model is a feature learner and thus works as a feature reducing tool. This way, in HDD classification tasks, DDBN has the potential to outperform successful biomedical data analysis tools. Moreover, since each RBM layer acts as a feature detector for the upper layer, in the case of two layered architecture, the features of features are best fit by the DDBN model.

4.3. Comparison of Deep Belief Networks and Feedforward Neural Networks in Deep and Shallow Architectures

For both DDBN and NN, the experiments are carried out according to the network parameters of learning rate and momentum are 0.1 and 0.2 respectively. Additionally number of hidden units used in each hidden layer is calculated as average of number of features and classes of data. The evaluation results which are presented italic in Table 4.17 mean all the instances are classified to one class of data. The visual representation of the comparison is given in Figure 4.4.

In the pretraining phase of DDBN on each pair of levels are treated as an RBM, and the hidden variables in one level are used as the visible variables for the next level in a greedy fashion. Therefore the weights are initialized in DDBN. For using this DDBN for discriminative purposes, an additional layer is added corresponding to the class labels activations. During this phase, all the weights are fine-tuned jointly using back-propagation. Fine-tuning in DDBNs is quite similar to training fashion in NNs.

Due to random initialization of weights in NN and the probable local optima issue it can be expected that DDBN performs better than NN. However, as defined in section 4.2 LDD is not convenient for DDBN for classification. As for HDD, due to the huge number of connections and weights NN is not able to converge. Therefore we are not able to do a comparison using HDD.

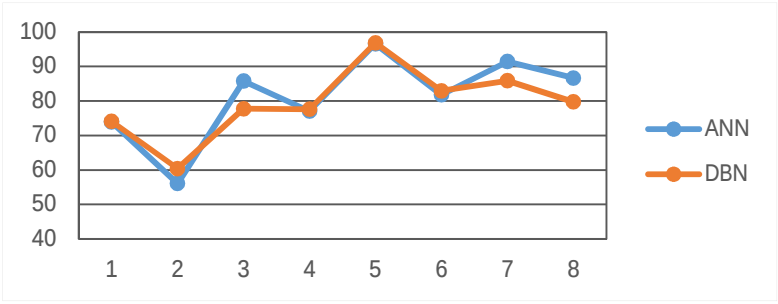
Table 4.17. Accuracy values of NN with respect to number of hidden layers

	NN			
Data	**1 hid.**	**2 hid.**	**3 hid.**	**4 hid.**
Arrhythmia	73.89	74.33	*54.20*	*54.20*
Heart disease (Cleveland)	56.10	62.05	*54.12*	*54.12*
Vertebral column (2C)	85.81	80.97	*67.74*	*67.74*
Diabetes (Pima Indians)	77.08	77.21	*65.10*	*65.10*
Wisconsin breast cancer	96.56	96.56	96.70	*65.52*
Mammographic mass	81.79	81.16	81.16	*53.05*
CTG	91.43	90.12	89.42	*77.84*
Parkinson	86.67	86.67	*75.38*	*75.38*

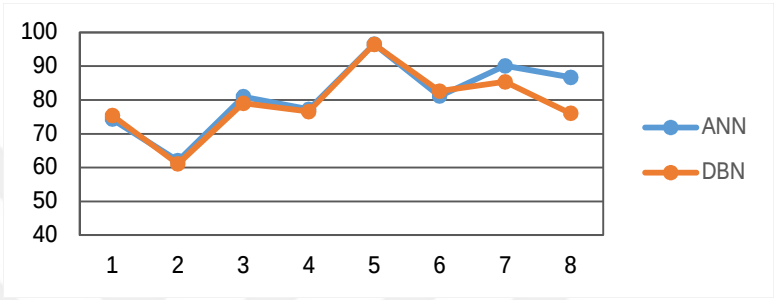
Table 4.18. Accuracy values of DDBN with respect to number of hidden layers

	DDBN			
Data	**1 hid.**	**2 hid.**	**3 hid.**	**4 hid.**
Arrhythmia	74.11	75.44	73.44	*54.20*
Heart disease (Cleveland)	60.39	61.08	55.43	*54.12*
Vertebral column (2C)	77.74	79.03	*67.74*	*67.74*
Diabetes (Pima Indians)	77.61	76.56	71.86	*65.10*
Wisconsin breast cancer	96.86	96.43	96.71	68.12
Mammographic mass	82.94	82.62	82.00	*53.05*
CTG	85.89	85.37	84.71	81.19
Parkinson	79.76	76.05	*75.38*	*75.38*

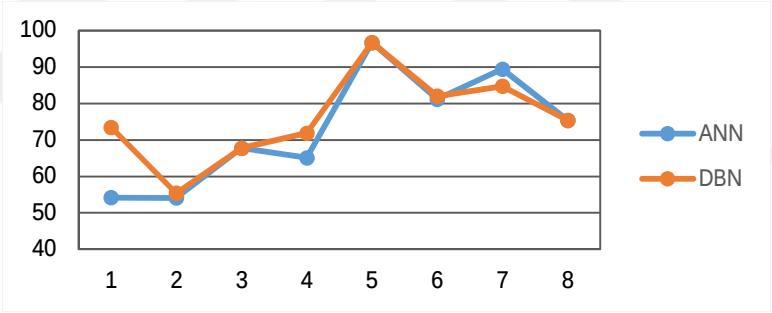
As Figure 4.4 shows NN and DDBN has produced approximate accuracy values in general. But in Figure 4.4 (a) it is clear that NN is better in one hidden-layered architecture for vertebral column, CTG and Parkinson data. In Figure 4.4 (b) for the first six data similar results are obtained, but for CTG and Parkinson, NN outperformed DDBN. In three and four layered architectures the models tend to overfit and classify all the samples to one of the class, therefore they are not convenient for classification for these datasets.



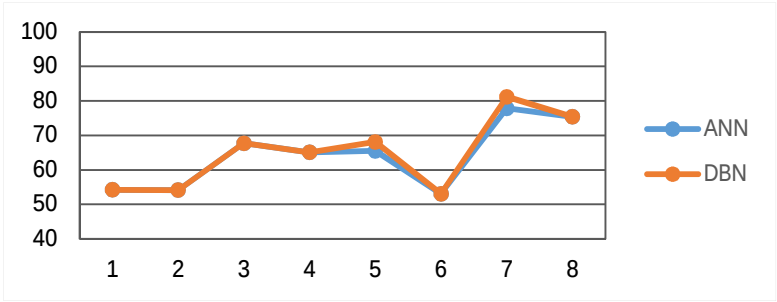
(a)



(b)



(c)



(d)

Figure 4.4. Comparison of NN and DDBN having (a) one hidden layer (b) two hidden layers (c) three hidden layers (d) four hidden layers.

4.4. Emphysema Discrimination from Raw HRCT Images by Convolutional Neural Networks

In order to analyze the classification performance of the CNN model for the emphysema disease, several experiments are conducted. In these experiments, a computer with a CPU of 8 cores each at 2.3 GHz is utilized. Also CUDA GPU acceleration is enabled by use of the graphic processor (GeForce 840M with 384 CUDA cores at 1029 MHz clock speed). Ubuntu Linux is the host operating system on which the experiments are conducted.

Since 61x61 pixel images are used, the network has 3721 (=61x61) input neurons. The dataset is split into two parts for training and testing. Training is achieved via 138 out of 168 patches and the remaining 30 patches are for testing purposes. The network parameters used in model are such that; learning rate is 0.0001, momentum 0.9, weight decay 0.0005, and solver mode is a flag that can be switched to CPU or GPU.

Firstly, the effect of the amount of training over classification performance is investigated. To this end, training is carried out over 100000 iterations and test procedure is repeated at each 200 iterations. When Figure 4.5 is examined, it is clear that up to a certain point (approximately around 20000 iterations) there is a positive relationship between the amount of training and the accuracy. After that point, the accuracy stays almost intact. A maximum accuracy of 84.25% is achieved through these tests. With sufficient number of training iterations, the network proves itself capable of classifying given instances with an accuracy of 80±4.25% which is a mostly adequate performance for automated classification of emphysema disease for decision support systems.

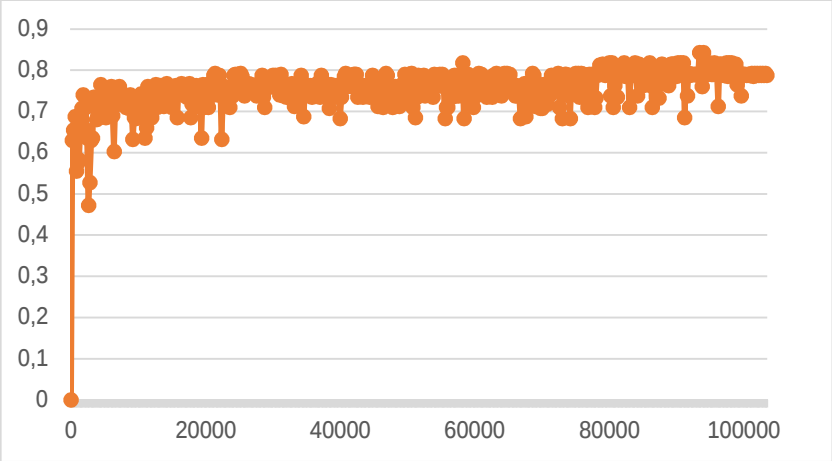


Figure 4.5. Accuracy values by iterations

Secondly, the computation load of the classifier system is examined. Owing to the complex nature of neural networks, a deep CNN is expected to operate relatively slower than shallow counterparts. Caffe framework presents opportunities to deal with this disadvantage such as the utilization of GPU as a parallel processor. As can be seen in Table 4.19, 1000 iterations of the training phase is completed within approximately 142 seconds whereas the GPU powered version of the network completes the same task in almost 29 seconds.

Table 4.19. CPU and GPU processing time while emphysema training of CNN

	Computing time [s]	
#Iterations	**CPU**	**GPU**
0	4.091787	0.380072
200	31.61855	6.136533
400	59.09462	11.99544
600	86.54946	17.80014
800	114.0059	23.58323
1000	142.2619	29.4446

Figure 4.6. reveals that the required computing time related to each iterations is reduced if the GPU processor is utilized.

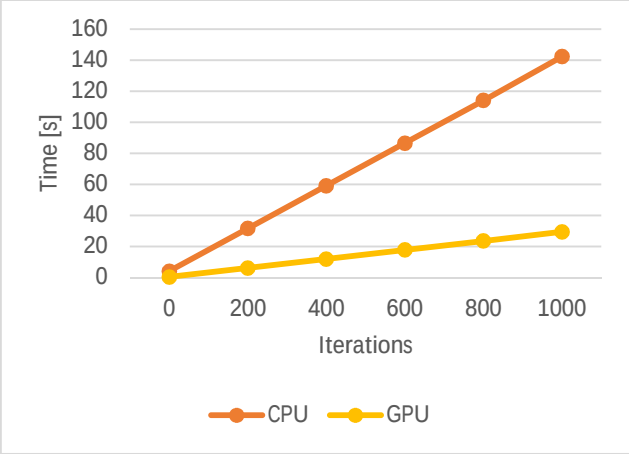


Figure 4.6. CPU vs. GPU computing time

The accuracy of 84.25% achieved in this study is very reasonable for an automated decision support system. However, some literature studies such as that of Sørensen et al. (2010) reached a better accuracy of 95%. The main reason behind the fact is that this study uses the raw image pixels and relies on the intensity values of each pixel for classification purposes whereas Sørensen et al. (2010) utilized more advanced texture techniques such as combining two different texture groups (i.e., local binary patterns and a filter bank based on Gaussian derivatives). In another relevant study, Azim and Niranjan (2014) presented classification results for the same dataset using k-NN, condensed nearest neighbor and Restricted Boltzman Machine (RBM) classifiers that were 46.04%, 46.06% and 47.85%, respectively. It is obvious that CNN is capable of outperforming some other classifiers when sole classifier is used with raw images without any preprocessing. However, the same study (Azim and Niranjan, 2014) also reported that 3-way factored RBM with the input images cropped into the size of 31x31 pixels performed better and produced an accuracy of 86.97%.

As a result, CNN as a classifier proves itself successful in classifying raw emphysema images. Without any preprocessing, the achieved accuracy of 84.25% proves this fact. However, since pathology can be localized in small areas of the lung and the region of interest (ROI) is generally smaller than the whole image, advanced texture techniques and preprocessing are required to improve the performance of CNN in this task.

4.5. Employing Discriminative Deep Belief Networks for Microarray Based Cancer Classification

In order to measure the classification performance of deep learning approach, experiments are conducted over the three datasets mentioned previously. All these experiments were carried out using the ten-folds cross validation technique in which the model is validated by taking the average of ten tests each of which is executed by using one out of ten subsets of the dataset for testing and the remaining parts for training. DDBN experiments are carried out by using the python programming library called 'nolearn' and Weka data mining tool (Hall et al., 2009) was used for SVM experiments. SVM was employed with default parameters set by the Weka environment, i.e. polynomial kernels are used as the kernel function. As for DDBN, we used a two layered structure with 50 and 10 hidden nodes in the first and second hidden layers, respectively. Training of the network is achieved with a learning rate of 0.04 and a momentum of 0.9.

Before the deep analysis of DDBN and SVM comparison, Table 4.20 is presented that enables to comprehend the reason of selection of methods. In Table 3 the methods of Random Forest and k-Nearest Neighbors are included besides DDBN and SVM. The values are accuracies of classification of each method after application of IG and SMOTE preprocessing steps. According to Table 4.20, Random Forest generates considerable results, but in a total assessment performance values of DDBN and SVM are better than Random Forest and k-NN.

Table 4.20. Accuracies of classification of four methods

Data	**DDBN**	**Random Forest**	**SVM**	**k-NN**
Laryngeal	0.944	90.21	0.937	73.43
Bladder	0.947	92.42	0.962	87.12
Colorectal	0.909	87.60	0.826	78.51

Table 4.21, Table 4.22, Table 4.23, Table 4.24, Table 4.24 and Table 4.26 summarize the classification results of DDBN and SVM for laryngeal, bladder and colorectal microarray gene expression data respectively. In each of the tables, results

are grouped into four categories. First category is the “no preprocessing” category in which each data is directly classified by DDBN and SVM without any preprocessing applied to the data. The second category represents the results of classification of DDBN and SVM after selecting discriminative features by the IG filter. As IG produces a score for each feature, we took the top ranking features exceeding 0.1 threshold score, therefore for laryngeal and bladder microarray gene expression data, 251 and 322 features are used, respectively. For colorectal data 300 features are selected by predetermination specified by previous observations. In the third category, the results are based on SMOTE based preprocessing while the last category presents the results obtained by applying both IG and SMOTE subsequently. The SMOTE algorithm produces new synthetic instances whose amount is determined by an algorithm parameter. This parameter is the percent of number of minority samples to be considered while generating new samples. In the study, this parameter is set to 100% and therefore all minority samples are used for new samples. As a result, the number of minor samples is doubled. Another parameter of SMOTE is the number of nearest neighbors which is determined as 5.

Table 4.21. Results of evaluation of DDBN using laryngeal microarray gene expression data

Evaluation metrics	**no preprocessing**	**IG**	**SMOTE**	**IG + SMOTE**
Accuracy	0.716	0.881	0.860	0.944
Sensitivity	0.500	0.853	0.941	0.985
Specificity	0.813	0.893	0.787	0.907
Precision	0.548	0.784	0.800	0.905
F-measure	0.523	0.817	0.865	0.943

Table 4.22. Results of evaluation of SVM using laryngeal microarray gene expression data

Evaluation metrics	**no preprocessing**	**IG**	**SMOTE**	**IG + SMOTE**
Accuracy	0.569	0.872	0.832	0.937
Sensitivity	0.294	0.794	0.971	0.956
Specificity	0.693	0.907	0.707	0.920
Precision	0.303	0.794	0.750	0.915
F-measure	0.299	0.794	0.846	0.935

Table 4.23. Results of evaluation of DDBN using bladder microarray gene expression data

Evaluation metrics	**no preprocessing**	**IG**	**SMOTE**	**IG + SMOTE**
Accuracy	0.533	0.913	0.818	0.947
Sensitivity	0.712	0.942	0.558	0.865
Specificity	0.300	0.875	0.988	1.000
Precision	0.569	0.907	0.967	1.000
F-measure	0.633	0.950	0.708	0.928

Table 4.24. Results of evaluation of SVM using bladder microarray gene expression data

Evaluation metrics	**no preprocessing**	**IG**	**SMOTE**	**IG + SMOTE**
Accuracy	0.707	0.946	0.909	0.962
Sensitivity	0.808	0.923	0.808	0.904
Specificity	0.575	0.975	0.975	1.000
Precision	0.712	0.980	0.855	1.000
F-measure	0.757	0.950	0.875	0.949

Table 4.25. Results of evaluation of DDBN using colorectal microarray gene expression data

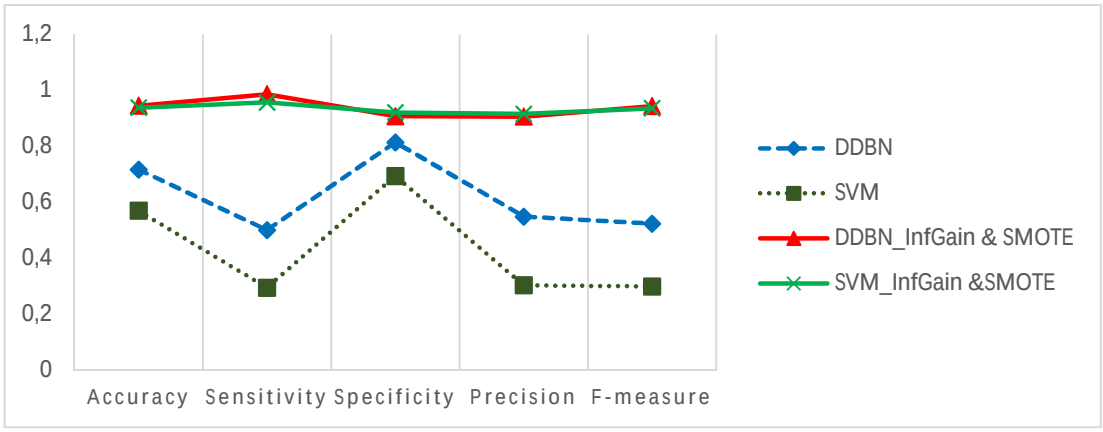
Evaluation metrics	**no preprocessing**	**IG**	**SMOTE**	**IG + SMOTE**
Accuracy	0.640	0.798	0.471	0.909
Sensitivity	0.188	0.594	0.750	1.000
Specificity	0.895	0.912	0.158	0.807
Precision	0.500	0.792	0.500	0.853
F-measure	0.273	0.679	0.600	0.921

Table 4.26. Results of evaluation of SVM using colorectal microarray gene expression data

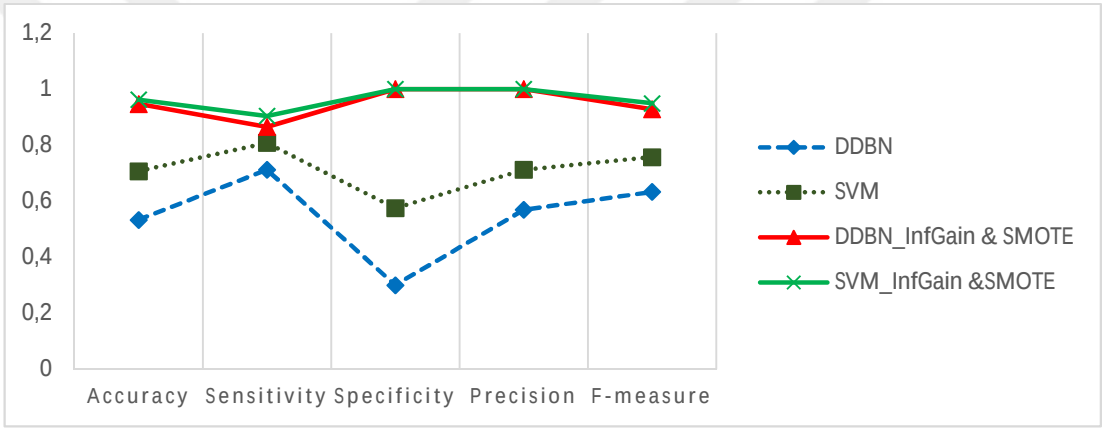
Evaluation metrics	**no preprocessing**	**IG**	**SMOTE**	**IG + SMOTE**
Accuracy	0.640	0.719	0.868	0.826
Sensitivity	0.313	0.563	0.984	0.891
Specificity	0.312	0.807	0.737	0.754
Precision	0.500	0.621	0.808	0.803
F-measure	0.385	0.590	0.887	0.844

In Figure 4.7 (a) and Figure 4.7 (b), the results obtained after application of IG and SMOTE are comparable for DDBN and SVM. In Figure 4.7. (c), performance of DDBN without preprocessing being very low in sensitivity while it is very high in

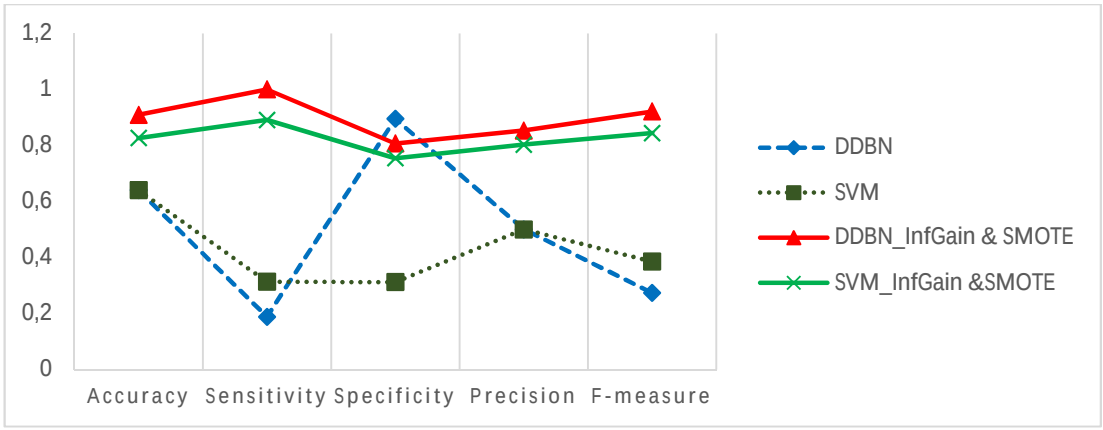
specificity caused an instability in the curve. This is because the major of the data is classified to negatives and thus positive instances are classified incorrectly – i.e. very few instances are decided to be positive. In such cases, F-measure is more reliable than using sensitivity and specificity alone to measure the overall performance since F-measure weighs positive and negative cases in a more balanced way. Although DDBN and SVM seem to produce nearly the same accuracy in colorectal data in no preprocessing category, SVM is more stable than DDBN in both detecting positive and negative cases. On colorectal gene expression data, the application of IG and SMOTE provided a considerable contribution to the improvement of results, especially to the results of DDBN.



(a)



(b)



(c)

Figure 4.7. Comparison of DDBN vs. SVM in (a) laryngeal cancer (b) bladder cancer (c) colorectal cancer datasets. The effect of SMOTE and IG preprocessing are also given for each dataset.

The layered structure of DDBN is convenient for identifying relevant genes in microarray gene expression data. As a deep model, DDBN tends to extract a relevant set of features from the input layer and then it enhances the recognition process in the next layer. In Figure 4.7, it can be seen that the application of both preprocessing steps of IG and SMOTE contributed to more stable results. In order to recognize DDBN or SVM to be successful the result in each metric should approximate to 1. Therefore we expect each curve would be as flat as possible in the alignment of 1 score without zigzags.

It should be noted that in classification tasks the performance results are very dependent on the data and in this context colorectal gene expression data is observed to be more sensitive to preprocessing steps. This is because this dataset suffers more from the imbalanced class distribution problem with relatively less number of samples in the dataset. In this dataset, DDBN has outperformed SVM in metrics of accuracy, sensitivity and specificity values of 0.909, 1.000 and 0.807, respectively. Considering all the three datasets, Table 4.27 shows that in terms of average values DDBN has outperformed SVM in all metrics.

Using *t*-test on the difference of the corresponding accuracy values of DDBN and SVM, the value of 0.842 is obtained on the t-distribution with 2 degrees of freedom. In this case the obtained *p*-value is greater than 0.10. H_0 hypothesis is not rejected which indicates the mean of the differences of the accuracies is 0. Therefore according to the analysis of the results there is not statistical significance between DDBN and SVM. This verifies that classification performance of DDBN is at least as good as that of SVM.

Table 4.27. Averages of results in each metric from three microarray gene expression data after preprocessing of IG and SMOTE

Evaluation metrics	**Averages of results**	
	DDBN	**SVM**
Accuracy	0.933	0.908
Sensitivity	0.950	0.917
Specificity	0.905	0.891
Precision	0.912	0.906
F-measure	0.931	0.909

In the first dataset where imbalanced class distribution is the most obvious, number of negatives are more than the double of number of positives, DDBN clearly outperforms SVM without any preprocessing. Another distinct feature of dataset is its number of features, being nearly half of features in other datasets. In such dataset, the performance of SVM can only be comparable to that of DDBN with preprocessing. In the other datasets, the second and the third, the class imbalance problem is slighter when compared to the first dataset. In the second dataset SVM is better without any preprocessing while the final results of the two algorithms with preprocessing are comparable. In the third one, the performances are similar without preprocessing but DDBN clearly outperforms SVM with preprocessing. Since the structural features of these two datasets are similar, different performances of DDBN and SVM suggest that their performances may vary depending on the dataset. However, in overall performance, DDBN appeared to perform better in terms of the f-measure metric which alone can safely be used to compare the overall classification performance of a classifier (Tang et al., 2009).

As a result, empirical results of the study suggest that DDBN has the potential as a decision support system in gene expression data based cancer diagnosis. With the proposed model that used IG and SMOTE, the DDBN algorithm can be generalized to perform adequately in other datasets of cancer, as well. By use of preprocessing, the processing time is reduced and the performance can be improved. The overall diagnosis performance of DDBN is 93.3% accuracy, considering the datasets of this study which appeared to be adequate for most of the cases.

4.6. Probability Estimation of DDBN for Risk of Recurrence of Laryngeal Cancer

In laryngeal dataset we have class values as the disease of laryngeal cancer is recurred or not. If the disease recurred its label is 1, therefore ideal probability of recurrence for such a sample is 1. Similarly if the disease is not recurred then its label is 0, meaning that ideal probability of recurrence for such a sample is 1. DDBN is a probabilistic generative model which can be employed in discriminative tasks. The

output of DDBN model represents the probability of the relevant label. For the case of laryngeal cancer data, due to the fact that we have recurrence information in labels, DDBN model is able to produce the probabilities of recurrence or no recurrence of a given new patient data sample. Figure 4.8 represents ideal probability values of reccurrence of disease for samples in laryngeal cancer data.

In the previous section we investigated the effect of IG and SMOTE on classification performance of DDBN and SVM by observing various evaluation metrics. In this section we figure out this effect by representing the improvement in probability estimates of DDBN. In Figure 4.9 we see that these estimation values are irregular, not obeying the arrangement in Figure 4.8. Note that these values are derived from the evaluation that we gained the accuracy, sensitivity and specificity values of 0.716, 0.500 and 0.813 respectively (See Table 4.20 in previous section).

The effect of preprocessing steps can be observed in Figure 4.10. Comparing to Figure 4.9 the estimation results are more arranged regarding Figure 4.8. Note that these values are derived from the evaluation that we gained the accuracy, sensitivity and specificity values of 0.944, 0.985 and 0.907 respectively (See Table 4.20 in previous section).

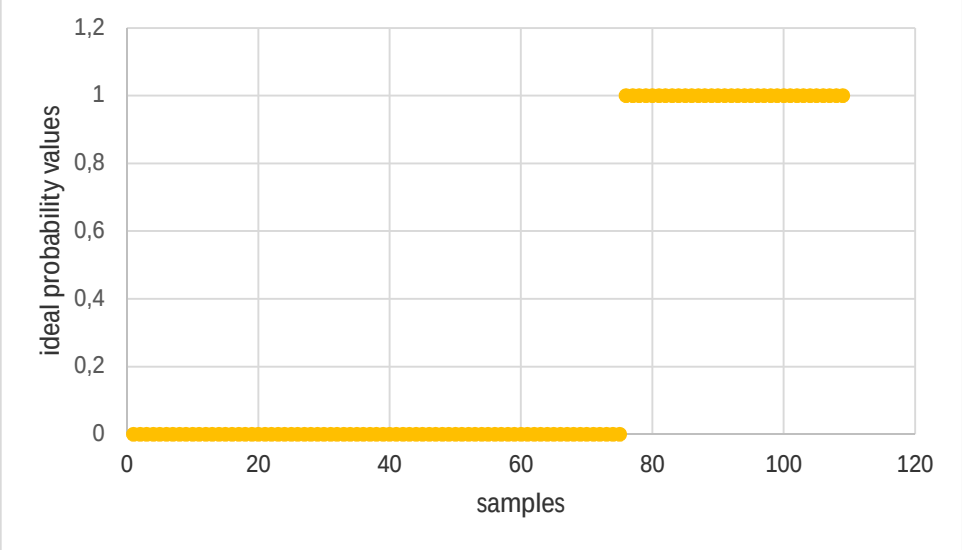


Figure 4.8. Distribution of laryngeal cancer class labels holding the ideal probabilities of recurrence of the disease

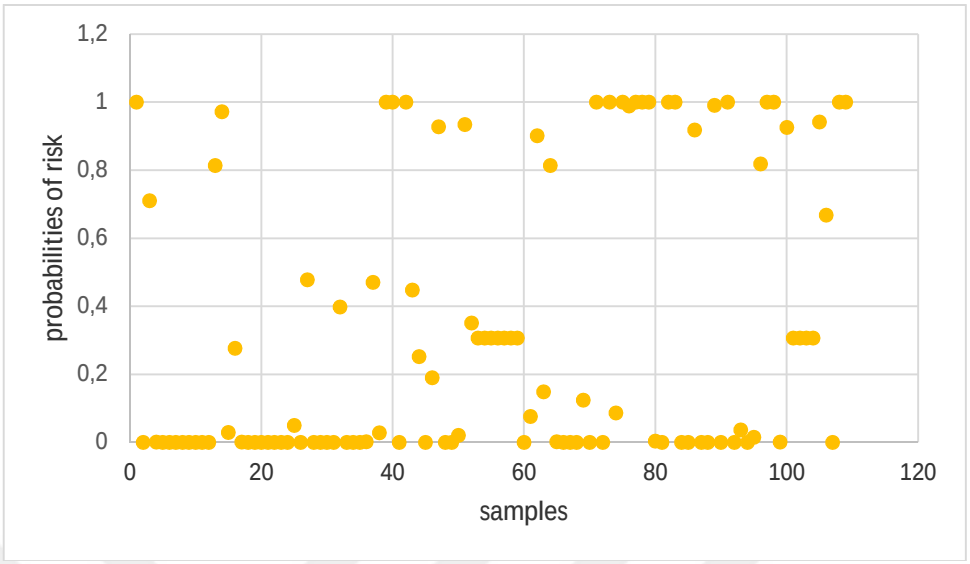


Figure 4.9. Probability estimation of DDBN as output decision without preprocessing steps of IG and SMOTE.

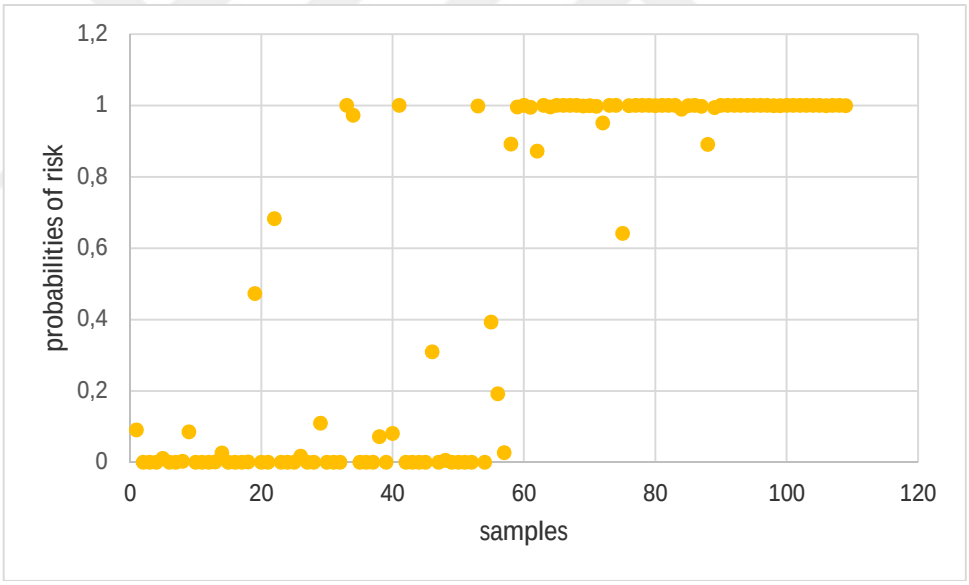
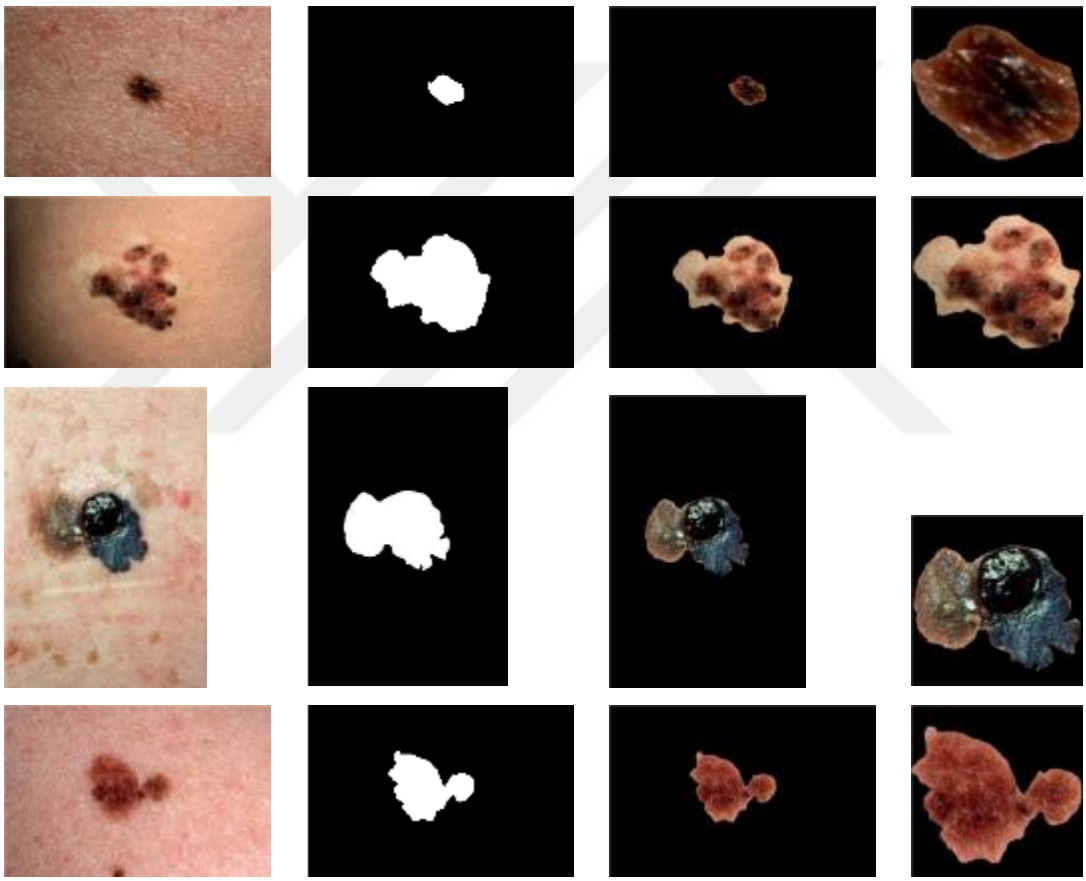


Figure 4.10. Probability estimation of DDBN as output decision after using preprocessing steps of IG and SMOTE.

4.7. Texture analysis of Melanoma Images for Diagnosis using Convolutional Neural Networks

In this section a comparative analysis of SVM versus the CNN for melanoma data classifications is presented. For experiments in classification of Convolutional Neural Network (CNN) we used Caffe environment. Caffe (Jia, 2003) is a deep

learning framework developed by the Berkeley Vision and Learning Center (BVLC) with the support of community contributors. Our CNN model is trained in a supervised fashion in the Caffe deep learning framework. And for experiments in classification of SVM we used MATLAB. There are a total of 206 images, 150 of them selected randomly and used for training of the CNN and SVM models. The remaining 56 images are used for testing the classification performance of each of the models.



(a) (b) (c) (d)

Figure 4.11. (a) Example instances from melanoma dataset (b) Contours for segmentation (c) Segmented images (d) Cropped images in size of 350x350.

Figure 4.11 represents the phases an image passes in preprocessing steps. The second and third images are malignant skin lesions and other two are benign. Figure 4.11 (a) represents the original image taken by the camera. Figure 4.11 (b) images are segmentation contours provided by collectors of the dataset (Amelard et al.,

2015). Figure 4.11 (c) images are the ones we extracted out the lesion on the skin from the original images by using the segmentation contours.

In order to classify images in Caffe and also in MATLAB, we should give same size of images to the model. The image sizes are very variable in dataset, for example 1640x1043 or 357x550 are sizes of two instance images. We cropped the images according to minimum melanoma image in dataset, therefore all images become 350x350 in size as represented in Figure 4.11 (d). Figure 4.12 and Figure 4.13 are confusion matrices of classification results of CNN and SVM on melanoma data respectively. In melanoma data there isn't the problem of imbalance. Thus according to these figures stable and similar results are gained both from DDBN and SVM. DDBN classified a total of 16 + 26 = 42 instances correctly, and it is 12 + 28 = 40 for SVM.

		Classified As	
		Status=0	Status=1
Actual Class Value	Status=0	16	8
	Status=1	6	26

Figure 4.12. Confusion matrix of CNN classification

		Classified As	
		Status=0	Status=1
Actual Class Value	Status=0	12	12
	Status=1	4	28

Figure 4.13. Confusion matrix of SVM classification

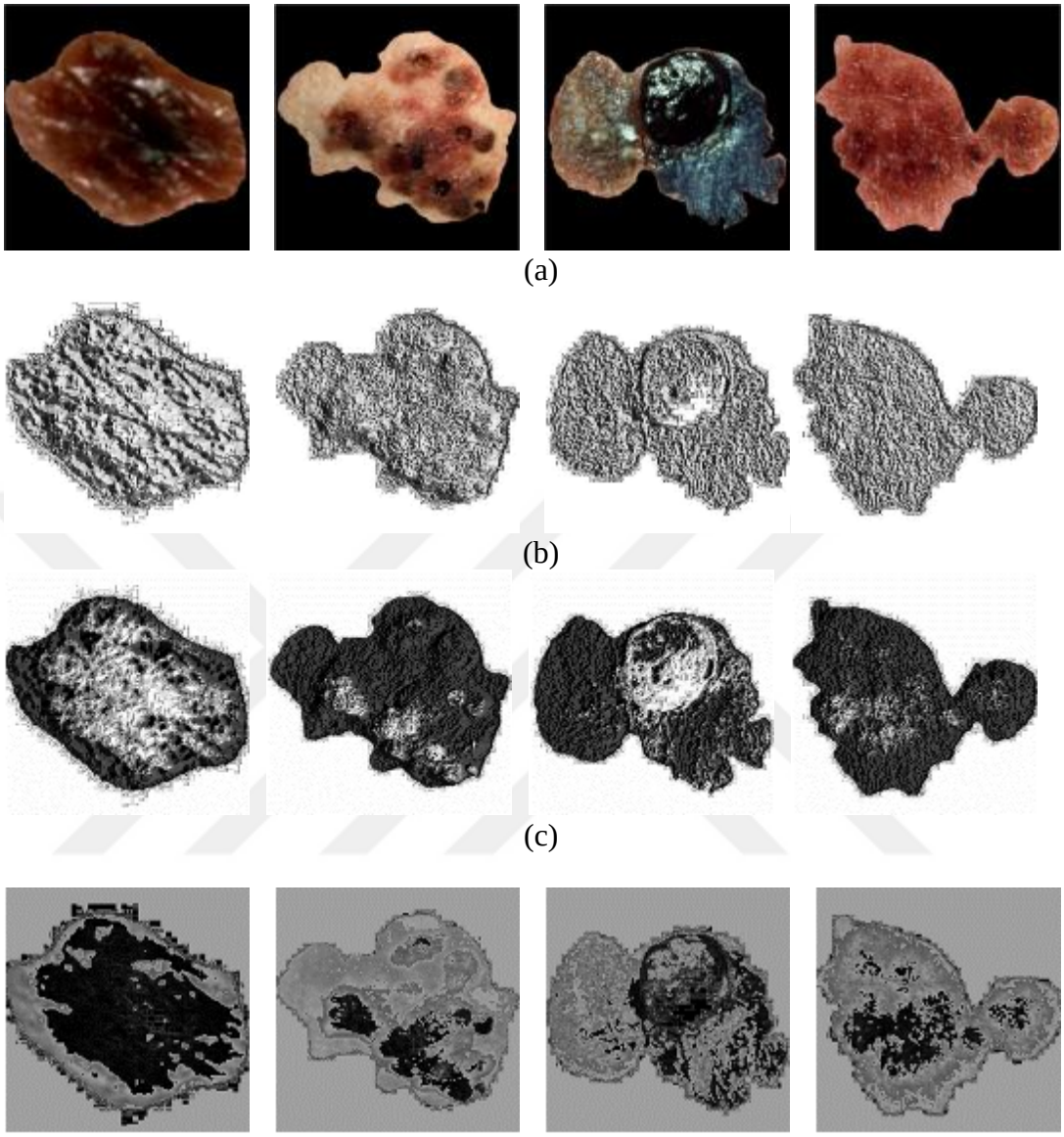


Figure 4.14. Images of skin lesions after texture analysis (a) Some skin lesions from melanoma images (b) LBPs of some skin lesions from melanoma image data for P=8, R=1, (c) LBPs of some skin lesions from melanoma image data for P=16, R=2 (d) BDIPs of some skin lesions from melanoma image data

We analyzed the melanoma data with and without texture analysis approaches. Texture analysis is employed as preprocessing steps before classification with the expectation of improvement in classification metrics. Local Binary Patterns (LBP) is a widely used texture analysis method having two important parameters; number of neighbors of the pixel to use in extracting the patterns, abbreviated by P,

and the constant distance of these parameters of which layout is in a particular manner, abbreviated by R. Two LBP analysis is achieved, in the first one P=8 and R=1, and in the second one P = 16 and R = 2. The third texture analysis is implemented by using BDIP. Figure 4.14 represents the resultant images from the texture analysis methods.

Table 4.28. Results of CNN vs SVM without preprocessing, and by using LBP with P=8 and R=1

Evaluation metrics	**no preprocessing**		**$LBP_{8,1}$**	
	CNN	**SVM**	**CNN**	**SVM**
Accuracy	0.750	0.714	0.696	0.589
Sensitivity	0.813	0.875	0.719	0.781
Specificity	0.666	0.500	0.667	0.333
Precision	0.765	0.700	0.742	0.610
F-measure	0.788	0.778	0.730	0.685

Table 4.29. Results of CNN vs SVM by using LBP with P=18 and R=2, and BDIP

Evaluation metrics	**$LBP_{16,2}$**		**BDIP**	
	CNN	**SVM**	**CNN**	**SVM**
Accuracy	0.714	0.714	0.642	0.554
Sensitivity	0.719	0.875	0.719	0.875
Specificity	0.708	0.500	0.542	0.125
Precision	0.767	0.700	0.676	0.691
F-measure	0.742	0.778	0.697	0.331

As Table 4.28 and Table 4.29 summarize the effect of texture analysis on melanoma images, any improvement is not observed in classification of neither LBP transforms of images nor BDIP transforms. However images may not necessarily have a particular texture property in a homogenous form for LBP to contribute for improvement in evaluation metrics. For CNN all the texture analysis decreased the evaluation values, also for SVM except LBP with P = 16, R = 2, it produced the same results as SVM with no preprocessing. For example we observe a very low value of specificity in SVM classification of BDIP preprocessed images. This means it predicts very few of actual negatives accurately. Structure of data is important for convenience of preprocessing steps, any compatibility is not found for LBP and BDIP. Figure 4.15 and Figure 4.16 shows the effect of LBP on CNN and SVM for different values of P and R.

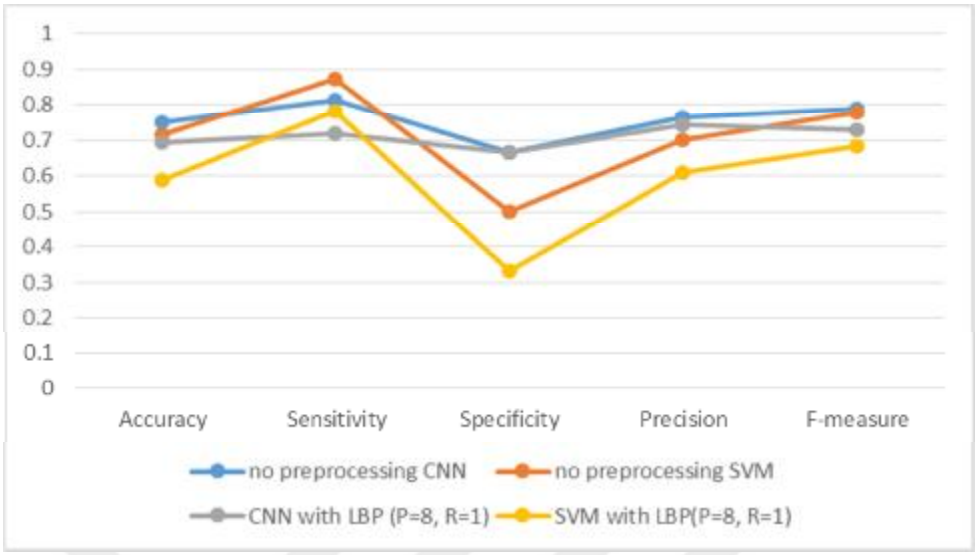


Figure 4.15. Representation of results of CNN and SVM with no preprcessing and LBP (P=8, R=1)

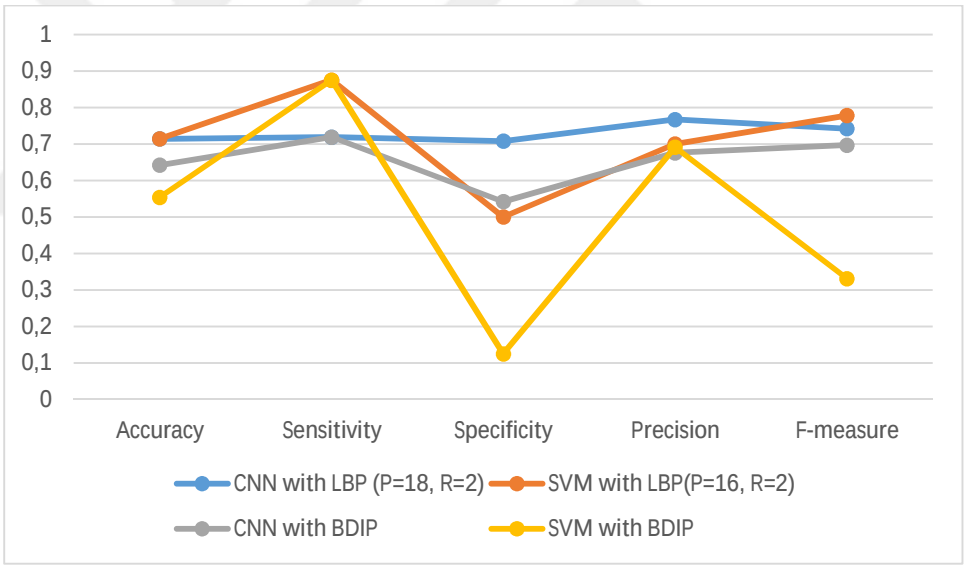


Figure 4.16. Representation of results of CNN and SVM with LBP (P=8, R=1) and BDIP preprocessing

Regarding the results without texture analysis preprocessing the best results are obtained by CNN. In fact CNN outperformed SVM very slightly, due to the essence of considering all metrics. Although the accuracy of CNN is 0.750, the less value of sensitivity indicates that it is not able to decide positives better than SVM. In such a case F-measure is more informative in comparison of classifiers. CNN and SVM produced 0.788 and 0.778 f-values respectively. Therefore CNN is of one percent ahead from SVM

5. CONCLUSIONS

In this study, it is pointed out that deep learning approaches can be successfully applied to clinical decision making if there are sufficient data for training the deep model. Publicly available real-life biomedical datasets are used in the experiments. DDBN is employed for text-based biomedical data or microarray gene expression data evaluation, and CNN is employed for medical image data evaluation.

In LDD and HDD, evaluating the performance of DDBN is analyzed by comparing the results with the results of baseline machine learning methods - Naive Bayes, k-Nearest Neighbors, Support Vector Machine and Random Forest - which are successfully applied to predictive data mining in biomedicine in literature. Experiments are presented in two phase for each of LDD and HDD. At first phase comparisons are provided by experiments on different architectures of DDBN having different number of hidden layers. At second phase the best results of first phase are compared to other powerful methods of; Naive Bayes, k-NN, SVM and Random Forest. For LDD, the best results are achieved by using one-hidden layer, but an approximately average performance of other methods is obtained. As for HDD, adding an additional hidden layer improved discriminatory effectiveness. In average values DDBN produced the best results in MAE, TPR and accuracy metrics for HDD.

Emphysema recognition from HRCT scans requires high level anatomical knowledge of a physician that must be supported by automated computer recognition. In literature emphysema is researched in detail and many computer based solutions are presented. But the result of direct classification don’t take place in most of them, a model is used together with texture analysis preprocessing. Differently, we directly processed raw HRCT images in order to emphasize the classification performance of CNN. To our knowledge, the best results are obtained, comparing to this kind of studies. The results lead to expectation of more success by incorporating such texture analysis preprocessing methods as future work.

Another investigation was carried out on cancer data discrimination by deep approaches to contribute the studies in this field. In literature cancer researches show that diagnosis by using microarray gene expression data is more effective compared to the traditional methods. An extensive evaluation of DDBN in cancer data analysis is presented, motivated by its recent successful applications in the field of artificial intelligence. We included test results of the algorithm conducted over microarray gene expression data of laryngeal, bladder and colorectal cancer. High dimensionality and imbalanced class distribution are two main problems inherent in the gene expression data. To deal with them, two preprocessing steps are applied; IG for selection of predictive genes, and SMOTE for oversampling the minority class samples. All the results are compared with the corresponding results of SVM which has previously been proved to be robust by machine learning studies. Our experiments showed that in microarray gene expression data analysis both DDBN and SVM achieved promising results after eliminating the effect of redundant and inconsistent features by use of preprocessing steps of IG and SMOTE. In terms of average values DDBN has outperformed SVM in all metrics with accuracy, sensitivity and specificity values of 0.933, 0.950 and 0.905, respectively. Additionally IG and SMOTE preprocessing steps when applied together in laryngeal cancer data have improved the risk estimation of DDBN in issue of recurrence of the disease.

For the aim of computer-aided diagnosis of melanoma skin cancer CNN and SVM are employed, which are two prominent methods in two dimensional data classification in literature. We investigated the effect of texture analysis methods on classification. The best results are obtained by CNN without using any texture analysis method. According to these results sensitivity and specificity values of 0.813 and 0.666 are obtained respectively, which means that CNN is better in prediction of malignant ones in melanoma skin cancer images.

Last of all, other deep learning methods besides DDBN and CNN are worth investigating for not only other types of cancers but also in different areas of biomedicine.

REFERENCES

ACKLEY, D. H., HINTON, G. E., and SEJNOWSKI, T. J., 1985. A learning algorithm for Boltzmann machines. Cognitive science, 9(1): 147-169.

AMELARD, R., GLAISTER, J., WONG, A., and CLAUSI, D. A. , 2015. High-level intuitive features (HLIFs) for intuitive skin lesion description. Biomedical Engineering, IEEE Transactions on, 62(3): 820-831.

AREL, I., ROSE, D. C., and KARNOWSKI, T. P., 2010. Deep machine learning-a new frontier in artificial intelligence research [research frontier]. Computational Intelligence Magazine, IEEE, 5(4): 13-18.

AZIM, T., and NIRANJAN, M., 2014. Texture classification with Fisher kernel extracted from the continuous models of RBM. In Computer Vision Theory and Applications (VISAPP), 2014 International Conference on, Vol. 2, pp. 684-690, IEEE.

BECKONERT, O., BOLLARD, M. E., EBBELS, T. M., KEUN, H. C., ANTTI, H., HOLMES, E., LINDON, J. C., and NICHOLSON, J. K., 2003. NMR-based metabonomic toxicity classification: hierarchical cluster analysis and k-nearest-neighbour approaches. Analytica Chimica Acta, 490(1): 3-15.

BENGIO, Y., 2009. Learning deep architectures for AI. Foundations and trends® in Machine Learning, 2(1): 1-127.

BENGIO, Y., LAMBLIN, P., POPOVICI, D., and LAROCHELLE, H., 2007. Greedy layer-wise training of deep networks. Advances in Neural Information Processing Systems, 19: 153.

BERTHONNAUD, E., DIMNET, J., ROUSSOULY, P., and LABELLE, H., 2005. Analysis of the sagittal balance of the spine and pelvis using shape and orientation parameters. Journal of Spinal Disorders & Techniques, 18(1): 40-47.

BLAGUS, R., and LUSA, L., 2012. Evaluation of smote for high-dimensional class-imbalanced microarray data. In Machine learning and applications (icmla), 2012 11th International Conference on, pp. 89-94, IEEE.

BOLÓN-CANEDO, V., SÁNCHEZ-MAROÑO, N., and ALONSO-BETANZOS, A., 2015. Distributed feature selection: An application to microarray data classification. Applied Soft Computing, 30: 136-150.

BOULESTEIX, A. L., JANITZA, S., KRUPPA, J., and KÖNIG, I. R., 2012. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2(6): 493-507.

BREIMAN, L. (2001). Random forests. Machine Learning, 45(1): 5-32.

BROSCH, T., TAM, R., and Alzheimer's Disease Neuroimaging Initiative, 2013. Manifold learning of brain MRIs by deep learning. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013, pp. 633-640, Springer Berlin Heidelberg.

BROWN, M. P., GRUNDY, W. N., LIN, D., CRISTIANINI, N., SUGNET, C. W., FUREY, T. S., ARES, M., and HAUSSLER, D., 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proceedings of the National Academy of Sciences, 97(1): 262-267.

CHAWLA, N. V., 2003. C4.5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In Proceedings of the ICML (Vol. 3).

CHAWLA, N. V., BOWYER, K. W., HALL, L. O., and KEGELMEYER, W. P., 2002. SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 321-357.

CHEN, Y., and ZHAO, Y., 2008. A novel ensemble of classifiers for microarray data classification. Applied soft computing, 8(4): 1664-1669.

CHUN, Y. D., SEO, S. Y., and KIM, N. C., 2003. Image retrieval using BDIP and BVLC moments. Circuits and Systems for Video Technology, IEEE Transactions on, 13(9): 951-957.

COLLOBERT, R., and WESTON, J., 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine learning, pp. 160-167, ACM.

CORTES, C., and VAPNIK, V., 1995. Support-vector networks. Machine learning, 20(3): 273-297

DANZIGER, S.A., BARONIO, R., HO, L., HALL, L., SALMON, K., HATFIELD, G.W., KAISER, P., and LATHROP, R.H., 2009. Predicting Positive p53 Cancer Rescue Regions Using Most Informative Positive (MIP) Active Learning, PLOS Computational Biology, 5(9): e1000498.

DENAS, O., 2014. Deep Models for Gene Regulation. PhD Thesis. Emory University.

DENG, J., DONG, W., SOCHER, R., LI, L. J., LI, K., & FEI-FEI, L., 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 248-255). IEEE.

DEPEURSINGE, A., FONCUBIERTA–RODRIGUEZ, A., VAN DE VILLE, D., and MÜLLER, H., 2012. Multiscale lung texture signature learning using the Riesz transform. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012, pp. 517-524, Springer Berlin Heidelberg.

DERMIS Dermatology Information System, http://www.dermis.net, 2016, Accessed: 25 Jan 2016.

DETRANO, R., JANOSI, A., STEINBRUNN, W., PFISTERER, M., SCHMID, J., SANDHU, S., GUPPY, K., LEE, S., and FROELICHER, V., 1989. International application of a new probability algorithm for the diagnosis of coronary artery disease. American Journal of Cardiology, 64(5): 304-310.

DÍAZ-URIARTE, R., and DE ANDRES, S. A., 2006. Gene selection and classification of microarray data using random forest. BMC bioinformatics, 7(1): 1.

DUDA, R. O., and HART, P. E., 1973. Pattern classification and scene analysis (Vol. 3). New York: Wiley.

DUDOIT, S., FRIDLYAND, J., and SPEED, P., 2002. Comparison of discrimination methods for classification of tumors using gene expression data. Journal of the American statistical association, 97(457): 77-87.

ELTER, M., SCHULZ-WENDTLAND, R., and WITTENBERG, T., 2007. The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process. Medical Physics, 34(11): 4164-4172.

ESTABROOKS, A., JO, T., and JAPKOWICZ, N., 2004. A multiple resampling method for learning from imbalanced data sets. Computational Intelligence, 20(1): 18-36.

FAKOOR, R., LADHAK, F., NAZI, A., and HUBER, M., 2013. Using deep learning to enhance cancer diagnosis and classification. In Proceedings of the ICML Workshop on the Role of Machine Learning in Transforming Healthcare. Atlanta, Georgia: JMLR: W&CP.

FERNÁNDEZ-NAVARRO, F., HERVÁS-MARTÍNEZ, C., RUIZ, R., and RIQUELME, J. C., 2012. Evolutionary generalized radial basis function neural networks for improving prediction accuracy in gene classification using feature selection. Applied Soft Computing, 12(6): 1787-1800.

FOUNTZILAS, E., KOTOULA, V., ANGOURIDAKIS, N., KARASMANIS, I., WIRTZ, R. M., ELEFTHERAKI, A. G., VELTRUP, E., MARKOU, K., NIKOLAOU, A., PECTASIDES, D., AND FOUNTZILAS, G., 2013. Identification and validation of a multigene predictor of recurrence in primary laryngeal cancer. PloS one, 8(8): e70429.

FREUND, Y., and HAUSSLER, D., 1994. Unsupervised learning of distributions of binary vectors using two layer networks. Computer Research Laboratory, University of California, Santa Cruz.

FRIMAN, O., BORGA, M., LUNDBERG, M., TYLÉN, U., and KNUTSSON, H., 2002. Recognizing emphysema—a neural network approach. In: Proceedings of sixteenth international conference on pattern recognition, Quebec, Canada, pp 1–4.

FUREY, T. S., CRISTIANINI, N., DUFFY, N., BEDNARSKI, D. W., SCHUMMER, M., and HAUSSLER, D., 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics, 16(10): 906-914.

GALDERMA, S. A., 2014. DermQuest Image Library. http://www.dermquest.com, Accessed: 25 Jan 2016

GANGEH, M. J., SØRENSEN, L., SHAKER, S. B., KAMEL, M. S., DE BRUIJNE, M., and LOOG, M., 2010. A texton-based approach for the classification of lung parenchyma in CT images. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010, pp. 595-602, Springer Berlin Heidelberg.

GLAISTER, J. L., 2013. "Automatic segmentation of skin lesions from dermatological photographs." MSc dissertation, Dept. Systems Design Eng., University of Waterloo, Ontario, Canada.

GOLUB, T.R., SLONIM, D. K., TAMAYO, P., HUARD, C., GAASENBEEK, M., MESIROV, J. P., COLLER, H., LOH, M., DOWNING, J.R., CALIGIURI, M. A., BLOOMFIELD, C. D., LANDER, E. S., 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science, 286: 531–537.

GORDON, G. J., JENSEN, R. V., HSIAO, L. L., GULLANS, S. R., BLUMENSTOCK, J. E., RAMASWAMY, S., RICHARDS, W. G., SUGARBAKER, D. J., and BUENO, R., 2002. Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. Cancer Research, 62(17): 4963-4967.

GUPTA, A., WANG, H., and GANAPATHIRAJU, M., 2015. Learning structure in gene expression data using deep architectures, with an application to gene clustering. In Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on, pp. 1328-1335, IEEE.

HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., and WITTEN, I. H., 2009. The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1): 10-18.

HAN, J., KAMBER, M., 2000. Data Mining Concepts and Techniques. Morgan Kaufmann Publishers, 1st Ed., San Francisco, USA.

HAYKIN, S., 1999. Neural Networks: A Comprehensive Foundation, Prentice Hall, USA, 842 p.

HHS US Department of Health and Human Services. Chronic obstructive pulmonary disease. Washington, DC: USDHHS; 2003. Data fact sheet.

HINTON, G. E., 2002. Training products of experts by minimizing contrastive divergence. Neural Computation, 14(8): 1771-1800.

HINTON, G. E., and SALAKHUTDINOV, R. R., 2006. Reducing the dimensionality of data with neural networks. Science, 313(5786): 504-507.

HINTON, G. E., OSINDERO, S., and TEH, Y. W., 2006. A fast learning algorithm for deep belief nets. Neural Computation, 18(7): 1527-1554.

HÖRSTER, E., and LIENHART, R., 2008. Deep networks for image retrieval on large-scale databases. In Proceedings of the 16th ACM International Conference on Multimedia, pp. 643-646, ACM.

HUA, K. L., HSU, C. H., HIDAYATI, S. C., CHENG, W. H., and CHEN, Y. J., 2015. Computer-aided classification of lung nodules on computed tomography images via deep learning technique. OncoTargets and therapy, 8.

HUANG, G. B., LEE, H., and LEARNED-MILLER, E., 2012. Learning hierarchical representations for face verification with convolutional deep belief networks. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 2518-2525, IEEE.

IBRAHIM, M., and MUKUNDAN, R., 2014. Multi-fractal techniques for emphysema classification in lung tissue images. Int. Proc. Chem. Environ. Eng, 78: 115-119.

IBRAHIM, R., YOUSRI, N. A., ISMAIL, M. A., and EL-MAKKY, N. M., 2014. Multi-level gene/MiRNA feature selection using deep belief nets and active learning. In 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 3957-3960. IEEE.

JAIN, K.A., MAO, J., and MOHUIDDIN, K.M., 1996. Artificial Neural Networks: A Tutorial. Theme Feature, 29: 31-44.

JERANT, A. F., JOHNSON, J. T., SHERIDAN, C., and CAFFREY, T. J., 2000. Early detection and treatment of skin cancer. American family physician, 62(2): 357-386.

JIA, Y., 2013. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe. berkeleyvision.org/

KARABULUT, E. M., and IBRIKCI, T., 2014. Effective automated prediction of vertebral column pathologies based on logistic model tree with SMOTE preprocessing. Journal of medical systems, 38(5): 1-9. DOI:10.1007/s10916-014-0050-0.

KARABULUT, E. M. and IBRIKCI, T., 2015. Emphysema discrimination from raw HRCT images by convolutional neural networks. In: 2015 9th International Conference on Electrical and Electronics Engineering (ELECO). IEEE. p. 705-708.

KARABULUT, E. M., and IBRIKCI, T., 2016. Texture analysis of Melanoma Images for Computer-aided Diagnosis. International Conference on Intelligent Computing, Computer Science and Information Systems (ICCSIS-16), Pattaya, Thailand, pp.158-161.

KARABULUT, E. M., and IBRIKCI, T., 2017. Discriminative Deep Belief Networks for Microarray Based Cancer Classification. Biomedical Research, 28(2), (In Press).

KHAN, J., WEI, J. S, RINGNER, M., SAAL, L. H., LADANYI, M., WESTERMANN, F., BERTHOLD, F., SCHWAB, M., ANTONESCU, C. R., PETERSON, C. and MELTZER, P. S., 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature medicine, 7(6): 673-679.

KINSELLA, M., MÜLLER, N. L., ABBOUD, R. T., MORRISON, N. J., and DYBUNCIO, A., 1990. Quantitation of emphysema by computed tomography using a density mask program and correlation with pulmonary function tests. CHEST Journal, 97(2): 315-321.

KONONENKO, I., 2001. Machine learning for medical diagnosis: history, state of the art and perspective. Artificial Intelligence in medicine, 23(1): 89-109.

KRIZHEVSKY, A., SUTSKEVER, I., and HINTON, G. E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems, pp. 1097-1105.

KUMAR, A. P., and VALSALA, P., 2013. Feature Selection for high Dimensional DNA Microarray data using hybrid approaches. Bioinformation, 9(16): 824.

LAROCHELLE, H., ERHAN, D., COURVILLE, A., BERGSTRA, J., AND BENGIO, Y., 2007. An empirical evaluation of deep architectures on problems with many factors of variation. In Proceedings of the 24th International Conference on Machine Learning, pp. 473-480. ACM.

LECUN, Y., BOTTOU, L., BENGIO, Y., and HAFFNER, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11): 2278-2324.

LEE, J. W., LEE, J. B., PARK, M., SONG, and S. H., 2005. An extensive comparison of recent classification tools applied to microarray data. Computational Statistics & Data Analysis, 48(4): 869-885.

LEE, T. S., and MUMFORD, D., 2003. Hierarchical Bayesian inference in the visual cortex. JOSA A, 20(7): 1434-1448.

LI, H., LI, X., RAMANATHAN, M., and ZHANG, A., 2014. Identifying informative risk factors and predicting bone disease progression via deep belief networks. Methods, 69(3): 257-265.

LI, J. and LIU, H., 2002. Kent ridge bio-medical data set repository. http://citeseer.ist.psu.edu/liu95chi.html.

LI, K., LI, X., ZHANG, Y., and ZHANG, A., 2013. Affective state recognition from EEG with deep belief networks. In Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on, pp. 305-310, IEEE.

LICHMAN, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

LITTLE, M. A., MCSHARRY, P. E., ROBERTS, S. J., COSTELLO, D. A., and MOROZ, I. M., 2007. Exploiting nonlinear recurrence and fractal scaling

properties for voice disorder detection. BioMedical Engineering OnLine, 6(1): 23.

LU, Y. and HAN, J., 2003. Cancer classification using gene expression data. Information Systems, 28(4): 243-268.

MENDOZA, C. S., WASHKO, G. R., ROSS, J. C., DIAZ, A. A., LYNCH, D. A., CRAPO, J. D., SILVERMAN, E. K., ACHA, B., SERRANO, C., and ESTEPAR, R. S. J., 2012. Emphysema quantification in a multi-scanner HRCT cohort using local intensity distributions. In Biomedical Imaging (ISBI), 2012 9th IEEE International Symposium on, pp. 474-477, IEEE.

MEUNIER, D., LAMBIOTTE, R., FORNITO, A., ERSCHE, K. D., and BULLMORE, E. T., 2010. Hierarchical modularity in human brain functional networks. Hierarchy and dynamics in neural networks, 1, 2.

MOHAMED, A. R., DAHL, G. E., and HINTON, G., 2012. Acoustic modeling using deep belief networks. Audio, Speech, and Language Processing, IEEE Transactions on, 20(1): 14-22.

MUKHERJEE, S., 2003. Classifying microarray data using support vector machines. A practical approach to microarray data analysis, 1: 166-185.

MÜLLER, N. L., STAPLES, C. A., MILLER, R. R., and ABBOUD, R. T., 1988. "Density mask": an objective method to quantitate emphysema using computed tomography. Chest, 94(4): 782-787.

NACHBAR, F., STOLZ, W., MERKLE, T., COGNETTA, A. B., VOGT, T., LANDTHALER, M., BILEK, P., BRAUN-FALCO, O., and PLEWIG, G., 1994. The ABCD rule of dermatoscopy: high prospective value in the diagnosis of doubtful melanocytic skin lesions. Journal of the American Academy of Dermatology, 30(4): 551-559.

NEWMAN, D.J., HEITTECH, S., BLAKE, C.L., MERZ, C.J., 1998. UCI Repository of machine learning databases. University California Irvine, Department of Information and Computer Science.

OJALA, T., PIETIKÄINEN, M., and HARWOOD, D., 1996. A comparative study of texture measures with classification based on featured distributions. Pattern recognition, 29(1): 51-59.

OTHMAN, A. A., TIZHOOSH, H. R., and KHALVATI, F., 2014. EFIS—Evolving fuzzy image segmentation. Fuzzy Systems, IEEE Transactions on, 22(1): 72-82.

PENG, Y., WU, Z., and JIANG, J., 2010. A novel feature selection approach for biomedical data classification. Journal of Biomedical Informatics, 43(1): 15-23.

PETRICOIN, E. F., ARDEKANI, A. M., HITT, B. A., LEVINE, P. J., FUSARO, V. A., STEINBERG, S. M., MILLS, G. B., SIMONE, C., FISHMAN, D. A., KOHN, E. C., and LIOTTA, L. A., 2002. Use of proteomic patterns in serum to identify ovarian cancer. The Lancet, 359(9306): 572-577.

PIROOZNIA, M., YANG, J. Y., YANG, M. Q., and DENG, Y., 2008. A comparative study of different machine learning methods on microarray gene expression data. BMC Genomics, 9(1): 1.

PLIS, S. M., HJELM, D. R., SALAKHUTDINOV, R., and CALHOUN, V. D., 2013. Deep learning for neuroimaging: a validation study. arXiv preprint arXiv:1312.5847.

RANZATO, M. C. P., CHOPRA, S., and LECUN, Y., 2007. Efficient learning of sparse representations with an energy-based model. In Proceedings of NIPS.

RYOO, Y. J., and KIM, N. C., 1988. Valley operator for extracting sketch features: DIP. Electronics Letters, 24(8): 461-463.

SAINATH, T. N., KINGSBURY, B., RAMABHADRAN, B., FOUSEK, P., NOVAK, P., and MOHAMED, A. R., 2011. Making deep belief networks effective for large vocabulary continuous speech recognition. In Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on, pp. 30-35, IEEE.

SAMARASINGHE, S. and CHAIBOONCHOE, A., 2015. Neural Networks and Fuzzy Clustering Methods for Assessing the Efficacy of Microarray Based Intrinsic Gene Signatures in Breast Cancer Classification and the Character and Relations of Identified Subtypes. Artificial Neural Networks, 285-317.

SALEM, S. A., SALEM, N. M., and NANDI, A. K., 2009. Augmentation of a nearest neighbour clustering algorithm with a partial supervision strategy for biomedical data classification. Expert Systems, 26(1): 8-21.

SARIKAYA, R., HINTON, G. E., and RAMABHADRAN, B., 2011. Deep belief nets for natural language call-routing. In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pp. 5680-5683, IEEE.

SATOH, K., KOBAYASHI, T., MISAO, T., HITANI, Y., YAMAMOTO, Y., NISHIYAMA, Y., and OHKAWA, M., 2001. CT assessment of subtypes of pulmonary emphysema in smokers. CHEST Journal, 120(3): 725-729.

SEJNOWSKI, T. J., and TESAURO, G., 1989. The Hebb rule for synaptic plasticity: algorithms and implementations. Neural models of plasticity: Experimental and theoretical approaches, 94-103.

SERRE, T., KREIMAN, G., KOUH, M., CADIEU, C., KNOBLICH, U., and POGGIO, T., 2007. A quantitative theory of immediate visual recognition. Progress in brain research, 165: 33-56.

SHAKER, S. B., VON WACHENFELDT, K. A., LARSSON, S., MILE, I., PERSDOTTER, S., DAHLBÄCK, M., BROBERG, P., STOEL, B., BACH, K. S., HESTAD, M., FEHNIGER, T. E., and DIRKSEN, A., 2008. Identification of patients with chronic obstructive pulmonary disease (COPD) by measurement of plasma biomarkers. The clinical respiratory journal, 2(1): 17-25.

SINISCALCHI, S. M., YU, D., DENG, L., and LEE, C. H., 2013. Exploiting deep neural networks for detection-based speech recognition. Neurocomputing, 106: 148-157.

SØRENSEN, L., SHAKER, S. B., and DE BRUIJNE, M., 2010. Quantitative analysis of pulmonary emphysema using local binary patterns. Medical Imaging, IEEE Transactions on, 29(2): 559-569.

STATNIKOV, A., ALIFERIS, C. F., TSAMARDINOS, I., HARDIN, D., and LEVY, S., 2005. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics, 21(5): 631-643.

SU, Z., HONG, H., PERKINS, R., SHAO, X., CAI, W., and TONG, W., 2007. Consensus analysis of multiple classifiers using non-repetitive variables: diagnostic application to microarray gene expression data. Computational Biology and Chemistry, 31(1): 48-56.

TAMILSELVAN, P., and WANG, P., 2013. Failure diagnosis using deep belief learning based health state classification. Reliability Engineering & System Safety, 115:124-135.

TANG, Y., ZHANG, Y. Q., CHAWLA, N. V., and KRASSER, S., 2009. SVMs modeling for highly imbalanced classification. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 39(1): 281-288.

TARCA, A. L., ROMERO, R., and DRAGHICI, S., 2006. Analysis of microarray experiments of gene expression profiling. American journal of obstetrics and gynecology, 195(2): 373-388.

URQUIDI, V., GOODISON, S., CAI, Y., SUN, Y., and ROSSER, C. J., 2012. A candidate molecular biomarker panel for the detection of bladder cancer. Cancer Epidemiology Biomarkers & Prevention, 21(12): 2149-2158.

USCS Working Group - United States Cancer Statistics Working Group, 2013. United States cancer statistics: 1999–2010 incidence and mortality web-based report. Atlanta: US Department of Health and Human Services, Centers for Disease Control and Prevention and National Cancer Institute.

VAN'T VEER, L. J., DAI, H., VAN DE VIJVER, M. J., HE, Y. D., HART, A. A., MAO, M., PETERSE, H. L., VAN DER KOOY, K., MARTON, M. J., WITTEVEEN, A. T., SCHREIBER, G. J., KERKHOVEN, R. M., ROBERTS, C., LINSLEY, P. S., BERNARDS, R., and FRIEND, S. H., 2002. Gene expression profiling predicts clinical outcome of breast cancer. Nature, 415(6871): 530-536.

VINYALS, O., and RAVURI, S. V., 2011. Comparing multilayer perceptron to deep belief network tandem features for robust ASR. In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pp. 4596-4599, IEEE.

VOS, T., FLAXMAN, A. D., NAGHAVI, M., LOZANO, R., MICHAUD, C., EZZATI, M., SHIBUYA, K., SALOMON, ABDALLA, S., ABOYANS, V., ABRAHAM, J., J. A., ACKERMAN, I., AGGARWAL, R., AHN, S. Y., ALI, M. K., ALMAZROA, M. A., ALVARADO, M., and ANDERSON, H. R., 2013. Years lived with disability (YLDs) for 1160 sequelae of 289 diseases and injuries 1990–2010: a systematic analysis for the Global Burden of Disease Study 2010. The Lancet, 380(9859): 2163-2196.

VURAL, H., and SUBAŞI, A., 2015. Data-Mining Techniques to Classify Microarray Gene Expression Data Using Gene Selection by SVD and Information Gain. Modeling of Artificial Intelligence, 6(2): 171-182

WANG, H., and RAJ, B., 2015. A Survey: Time Travel in Deep Learning Space: An Introduction to Deep Learning Models and How Deep Learning Models Evolved from the Initial Ideas. arXiv preprint arXiv:1510.04781.

WATANABE, T., KOBUNAI, T., TANAKA, T., ISHIHARA, S., MATSUDA, K., and NAGAWA, H., 2009. Gene expression signature and the prediction of lymph node metastasis in colorectal cancer by DNA microarray. Diseases of the Colon & Rectum, 52(12): 1941-1948.

WOLBERG, W. H., and MANGASARIAN, O. L., 1990. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proceedings of the National Academy of Sciences, 87(23): 9193-9196.

WU, C., OROZCO, C., BOYER, J., LEGLISE, M., GOODALE, J., BATALOV, S., HODGE, C.L., HAASE, J., JANES, J., and HUSS, J. W., 2009. BioGPS: an extensible and customizable portal for querying and organizing gene annotation resources. Genome Biol. 10(11):R130. http://www.biogps.org/

YANG, S., and NAIMAN, D. Q., 2014. Multiclass cancer classification based on gene expression comparison. Statistical applications in genetics and molecular biology, 13(4): 477-496.

CURRICULUM VITAE

She was born on June 20th, 1980 in Gaziantep, Turkey. She was graduated from Karadeniz Technical University in 2002 with a B.S.in Computer Engineering. In 2012 she received her M.S. degree in Electrical and Electronics Engineering Department from Cukurova University. She worked on machine learning techniques for computer based decision support systems. In PhD studies, she focused on deep learning approaches for biomedical data classification. In Gaziantep University Vocational High School she is working as an instructor at Computer Programming Department.