DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PREDICTION OF SYNERGISTIC DRUG COMBINATIONS BY USING MACHINE-LEARNING METHODS

by
Ali CÜVİTOĞLU

July, 2017
İZMİR

PREDICTION OF SYNERGISTIC DRUG COMBINATIONS BY USING MACHINE-LEARNING METHODS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Master of
Science in Computer Engineering

by
Ali CÜVİTOĞLU

July, 2017
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"PREDICTION OF SYNERGISTIC DRUG COMBINATIONS BY USING MACHINE-LEARNING METHODS"** completed by **ALİ CÜVİTOĞLU** under supervision of **ASST. PROF. ZERRİN IŞIK** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Zerrin IŞIK

Supervisor

Asst. Prof. Dr. Mustafa Özerşal

(Jury Member)

Assoc. Prof. Dr. Atil Alptekin

(Jury Member)

Prof. Dr. Emine İlknur CÖCEN
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Asst. Prof. Zerrin IŞIK, for her supervision, endeavor and guidance throughout this study.

And, I would like to express my sincere gratitude to my wife, my brother and parents for their support, patience and love. I could not be able to complete this thesis without their precious support.

Ali CÜVİTOĞLU

PREDICTION OF SYNERGISTIC DRUG COMBINATIONS BY USING MACHINE-LEARNING METHODS

ABSTRACT

Cancer is still one of the challenging diseases to develop new therapies due to the late diagnosis and its complex progression nature. There is an urgent need for new therapy regimes for cancer patients having late stage diagnosis or recurrence. New computational approaches can help to identify more effective drug combinations as new treatment options for cancer. For this purpose, we developed a classification model to identify more effective drug pairs out of all possible combinations by using gene expression of single drug treatment and biological network data. Three different machine-learning methods which are Artificial Neural Network (ANN), Random Forest (RF) and Support Vector Machine (SVM) were trained with six features derived by using different biological data. The model was evaluated on two different drug treatment data sets that contain both positive (more effective) and negative (not effective) drug combinations. The proposed model has achieved successful results in the test case to find promising features and a machine-learning method, which might be suitable for the prediction of more effective drug combinations.

Keywords: Bioinformatics, Drug combinations, Gene expression, SVM, ANN, RF.

MAKİNE ÖĞRENMESİ YÖNTEMLERİYLE ETKİLEŞEN İLAÇ KOMBİNASYONLARININ TAHMİN EDİLMESİ

ÖZ

Kanser geç teşhis edilmesi ve karmaşık ilerlemesinden dolayı hala yeni tedavilerin geliştirilmesi için ilgi çeken hastalıklardan birisidir. İleri safhada teşhis edilen veya kanseri nükseden hastalar için yeni tedavi yöntemlerine acil olarak ihtiyaç duyulmaktadır. Yeni hesaplamalı yaklaşımlar, daha etkili ilaç birleşimlerini yeni tedavi seçenekleri olarak tanımlamada yardımcı olabilirler. Bu amaçla, daha etkili ilaç ikililerini olası tüm kombinasyonlar içinden tanımlayabilmek için, ilaç uygulanmış gen ifadesi ve biyolojik ağ verilerini kullanan bir sınıflandırma yöntemi geliştirdik. Farklı biyolojik verilerden türetilmiş altı öznitelik ile üç farklı makine öğrenmesi yöntemi, Yapay Sinir Ağları (YSA), Destek Vektör Makineleri (DVM) ve Rasgele Orman (RO), eğitildi. Bu yöntem, içinde hem etkili hem de etkisiz ilaç birleşimlerini bulunduran iki farklı ilaç verisi üzerinde sınandı. Önerilen yöntem, umut verici öznitelikleri ve daha etkili ilaç birleşimlerinin kestiriminde uygun olabilecek makine öğrenmesi yöntemini bulmak için sınama verisi üzerinde başarılı sonuçlar elde etmiştir.

Anahtar Kelimeler: Biyoenformatik, İlaç birleşimleri, Gen ifadesi, DVM, YSA, RO.

CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

1.1 Motivation

Cancer is a malignant disease that is formed by the irregular division and proliferation of cells in a tissue or organ. All organs are composed of cells, which are the smallest building blocks of our body. Healthy body cells (excluding muscle and nerve cells) have the ability to divide. They can divide to regenerate dead cells and repair injured tissues. But they cannot make division infinitely. Throughout its life, number of divisibility is determined for every healthy cell. However, the cancer cells lose this conscious and cannot control the division. Cancer cells get together to form tumors, which compress normal tissues, infiltrate or destroy them. There are stages of cancer. If cancer is diagnosed in early stages, it can be treated quite efficiently and the patient can continue daily life. However most of the cancer cases are usually diagnosed in the later stages due to limited symptoms of cancers in early stages. The cancer treatment is very costly. Therefore, a lower cost and effective treatment method is needed for the late stages of cancers.

There are many researches for the cancer treatment. One of the most remarkable studies has been the recognition of the effects of non-cancer treatment drugs on cancer treatment (Boguski, Mandl, & Sukhatme, 2009; Gupta, Sung, Prasad, Webb, & Aggarwal, 2013). Especially, combinations of these drugs (as pair or triple etc.) emerge as new treatment method. However, testing all combinations of drugs to find synergistic drug combinations, which mean they have positive effects on cancer treatment, in wet-lab experiments is laborious and very expensive. Therefore, our motivation is to find new computational methods to suggest chemically more synergistic (effective) drug combinations that can be easily verified by wet-lab experiments.

Nowadays, machine-learning methods are applied different application domains. In the last decades, these methods have been used in medical applications and they

provided many improvements in different levels. Specifically, various machine-learning methods have been proposed to understand the disease developments. We also aim to use machine learning methods in the classification of drug combinations.

The motivation of this thesis is to apply machine-learning methods on predicting synergistic drug combinations for cancer treatments. We developed a classification method that aims to identify effective drug pairs out of all possible combinations by using single drug treatment and biological network data. Six metrics (i.e., features) have been computed on two different drug treatment data sets to train three different machine-learning methods to classify drug pairs into combination classes (i.e., positive or negative).

1.2 Problem Definition

The prediction of more effective drug combinations for cancer treatment is a challenging problem for several reasons. We proposed several hypotheses to approach this problem with the computational methods. The main data source of this study is microarray experiments that measure gene expression levels of cells before and after drug treatment. Gene expression data can be significant to understand the relations between two drugs. So, we will investigate whether gene expression data is one of the appropriate data for predicting drug combinations.

Another important factor is what kind of features we should compute to train machine-learning model. There might be various metrics to represent the relations between two drugs. However, not all metrics can fit machine-learning model well. We should identify the best combination of features that will be the input of machine-learning models. Can this combination change from one machine-learning model to another? At that point, finding the most successful machine learning model with best fitting features is very critical. Therefore, we will experiment different machine-learning models with these features.

Machine-learning methods should be well trained by using enough data samples. As much as the need for enough samples, the samples of in each target class must be balanced as well. It is very difficult to find such a balanced data for drug combinations for cancer treatment. Moreover, another challenge is related with cross validation methods. So, we should use the appropriate data partitioning method to train the system well with the unbalanced data set.

1.3 Contribution

We have tested the proposed method with two different data sets to measure the consistency. One of our contributions is the generation of the second data set after obtaining the first data set from NCI-DREAM consortium (Bansal et al., 2014). Another novelty of the study is the extraction of Drug Perturbation Network (DPN) of each drug before calculation of drug combinations.

Two of six features which are named as GO Term Similarity and Overlap of DPN are calculated in this study for the first time. Another contribution is the application of a special cross-validation method to use in imbalanced data sets.

1.4 Organization of the Thesis

This thesis includes five chapters and the rest of the thesis is organised as follows:

Chapter 2 explains the main data sources of the study. Analysis of microarray data have been reviewed. Protein-protein interaction (PPI) networks have been explained. We have discussed the differences of our method from other synergy prediction methods. Furthermore, we have reviewed related studies about SVM, ANN and RF.

Chapter 3 gives an overview about our system. Two different drug treatment data sets have been described. We have explained why and how we used the DEMAND algorithm. Six features with their formulas have been explained. After that, machine learning methods have been disscussed with their specific parameters. Then, two

cross-validation methods have been discussed. Lastly, we have introduced evaluation methods used in the study.

Chapter 4 presents a detailed discussion of the results

Chapter 5 covers the conclusion and future works.

CHAPTER TWO
LITERATURE REVIEW

2.1 Microarray Data and Analysis

The rapid development of computer technology in parallel with molecular biology has brought two disciplines closer to each other. Thus, a gene chip (microarray), one of the endpoints that biotechnology can arrive conceptually, has emerged. In traditional methods in molecular biology, expression of a gene at a time is focused. This means that it is difficult to see all of the gene functions at the same time with conventional methods such as “reporter gene”, “northern blotting”, “southern blotting” etc. Because these methods are focused on specific genes or proteins at a time. Gene chip technology is met with great interest because it allows the whole genome to be visualized on a simple chip, allowing scientists to see the interactions of thousands of genes at the same time.

Microarray is a kind of microscopic DNA spot that is formed in an array by attaching to a solid surface like glass, plastic, or silicon chip. In a microarray, there can be tens of thousands of these spots. The DNA fragments attached to the surface (usually 20-100 nucleotides in length) are defined as probes. Microarray technology is derived from the "Southern Blotting" technique, in which DNA can be identified by binding a substrate and probing with a known gene or fragment.

2.1.1 How are microarrays produced and how do they work?

A variety of methods can be used in the production of microarrays: fine-tipped needle printing on glass slides, photolithography with pre-prepared masking, photolithography with dynamic micro devices, ink-jet printing, electrochemistry in microelectrode arrays etc.

Gene array experiments are typically aimed at determining the level of gene expression in different tissues or conditions for different time points. For this purpose,

RNA extraction is performed based on different tissues, conditions or times according to the subject studied. These RNA samples are diluted to ensure that each sample is of equal density. For each mRNA molecule in the original RNA population, a single-stranded labeled cDNA (complementary fragment of mRNA) is produced. As the density of a particular mRNA increases, the amount of cDNA increases. Probes are produced by reverse transcription of mRNA into single stranded cDNA in the presence of labeled nucleotides. For this reason, the labeled probe is actually a population of cDNA molecules representing the mRNA population. Generally, the labeled nucleotides are labeled with fluorescent markers such as Cy3 and Cy5 or digoxigenin (DIG), which can be detected by chemical luminescent detection. Probes are hybridized with filters containing cDNAs spotted in a two-dimensional array (Figure 2.1). The amount of hybridization in a given gene corresponds to the amount of mRNA found for the gene of interest. Filter arrays are incubated with probe and washed as in Southern or Northern blotting. The hybridized probe is detected by chemical fluorescence for DIG labeling and direct UV fluorescence for microarrays. The sequence density of each spot is measured with a CCD camera and the data is acquired as a TIF image.

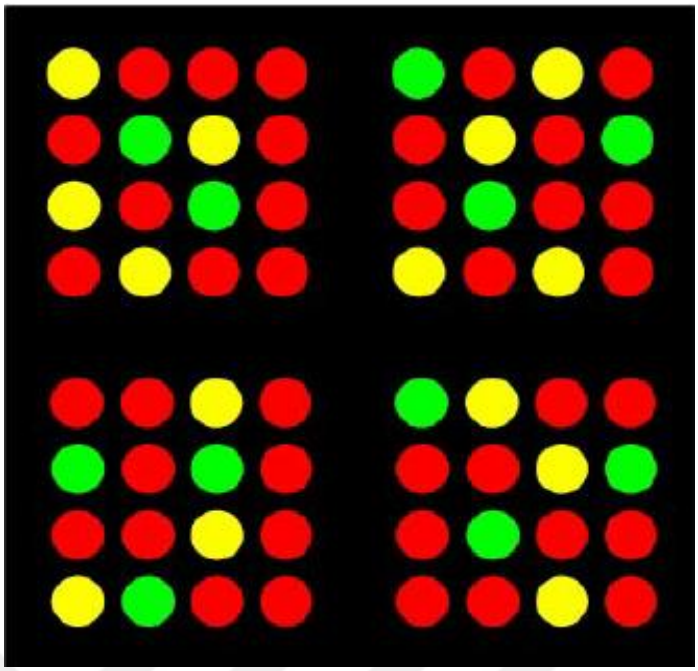


Figure 2.1 A microarray image. Red spots are marked only with Cy3 whereas green spots are marked only with Cy5. Yellow spots are marked with both Cy3 and Cy5. (Figure is adapted from (Bajcsy, Liu, & Band, 2014)).

2.1.2 Related Works

Microarrays are used in many areas. Some of the best-known areas are gene expression analysis, genetic and mutation analysis, environmental research, as diagnostic tools and identifying antimicrobial genes (Kumar, Goel, Fehrenbach, Puniya, & Singh, 2005).

As gene expression analysis, using a cDNA microarray, Escherichia coli bacteria were exposed to a large number of toxic chemicals, and gene expression levels were characterized and differentiated (Kim & Gu, 2007). After generating subsets of the Escherichia coli genome, DNA microarray technology was also used in detection of differential transcription profiles these subsets (Oh & Liao, 2000). The different gene expression of Oral Gingival Epithelium (OGE) and Epithelial Rests of Malassez (ERM) cells are analyzed using a DNA microarray technique (Kurashige et al., 2008). Cassone M. et al. used DNA microarrays to detect genetic elements carrying

glycopeptide resistance clusters in Enterococcus (Cassone, Del Grosso, Pantosti, Giordano, & Pozzi, 2008). DNA microarrays were also used to study on microorganisms which can't be cultured such as nitrification, methanogenesis and denitrification (Saleh-Lakha et al., 2005). Another study was about Mycobacterium spp detection using DNA microarrays with real-time PCR. Not only detection but also species identification of Mycobacterium spp was studied. (Tobler, Pfunder, Herzog, Frey, & Altwegg, 2006).

There may be a change in gene expression after drug treatment. Microarrays can be used to monitor these changes (Debouck & Goodfellow, 1999). Profiling the action of large numbers of chemicals when biological targets used is a challenging but can be solved by using chemical microarrays (Ma & Horiuchi, 2006).

2.1.3 Gene Expression Profiles from Microarrays

One of the most known usages of the microarray technique is to measure the differences in gene expression. All genes transcribed from genomic DNA are called transcriptomic or gene expression profiles. The phenotype and function of a cell is determined by its transcriptome. Although the genome is fixed from the cell to another, the gene expression profile can rapidly change according to the conditions in which the cell is located. Under various conditions, changes in the expression levels of genes can be analyzed to extract important information about the function of the proteins encoded by these genes. Microarrays are used to monitor changes in gene expression patterns in various processes. For example, the microarray can be used in a highly comprehensive manner with the characterization of gene expression differentiation in cancer cells and also in the diagnosis of other diseases.

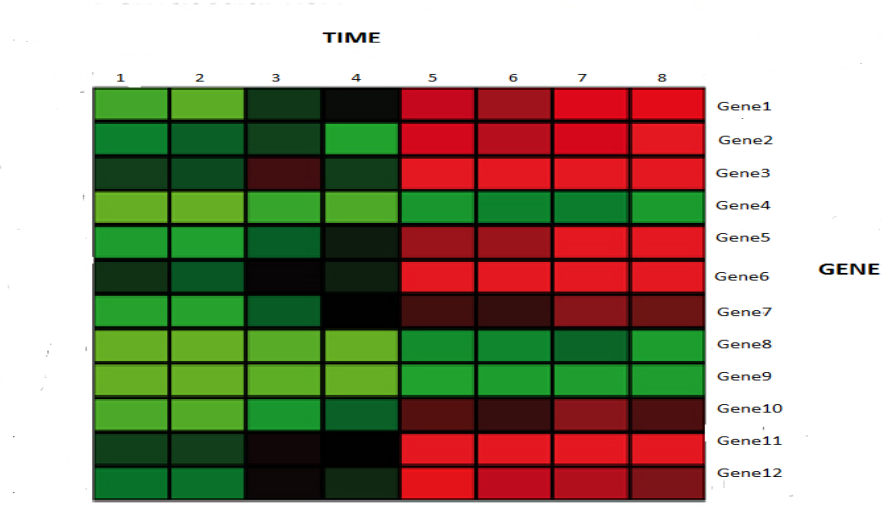


Figure 2.2 Gene expression value is measured at different times. Columns show time and rows show genes. Reduced expression is indicated with green and increased expression is indicated with red.

Statistically significant gene expressions are called as differential gene expression that is computed by comparing two experimental conditions. So, a gene may have different values at different times (Figure 2.2) or experimental setup. Some of the best-known methods of measuring differential gene expression are CuffDiff (Trapnell et al., 2012), DESeq (Anders & Huber, 2010), edgeR (Robinson, McCarthy, & Smyth, 2010).

2.2 Protein Interaction Networks

The biological activities in the cells take place when the proteins interact with each other or with other molecules. For example, some molecules in the environment of cell are recognized by proteins in the cell membrane, and this interaction turns out a signal that the same protein interacts with other cell proteins that this molecule exists. The relevant units in the cell that receive this signal adjust their functions accordingly. For example, the presence of a dangerous substance or nutrient in the outside of a cell will send a signal through signal transduction, consequently either cell defense mechanisms will be active or physiological events will start in case of food. Another example is that some proteins bind to another protein to form a pair of proteins and transport this formed protein complex to the required region of the cell. To perform such cellular

tasks, proteins should form either physical or functional interactions with other proteins.

Studies have been done to find protein interactions on different organisms (Gavin et al., 2006; Giot, 2003; Li, 2004). As the interactions between the proteins were found, these interactions began to be represented as a graph structure. These topological representations are called as Protein-Protein Interaction (PPI) networks. In PPI networks, proteins are represented with nodes. Interactions observed between two individual proteins are represented with edges. (Figure 2.3). The type of edge can vary depending on the type of interaction. For example, the binding of two proteins is shown as an undirected edge. Another example is that If a gene expression is regulated by a transcription factor (TF) there will be a directed edge from TF to the gene (Cho, Kim, & Przytycka, 2012).

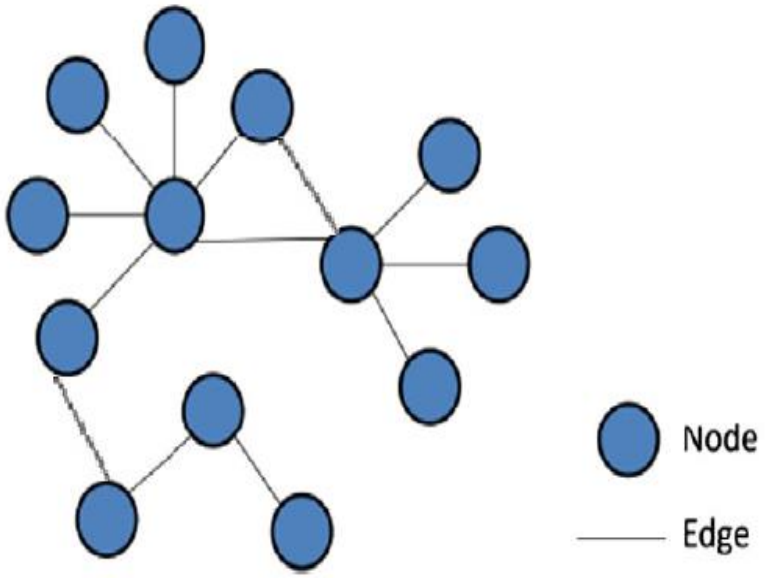


Figure 2.3 Simple structure of a graph. Nodes represent proteins whereas edges represent interactions.

The earliest known PPI networks were built by using yeast-two-hybrid (Y2H) (Fields & Song, 1989), protein complementation assays (PCA) (Tarassov et al., 2008) and affinity purification followed by mass spectrometry (AP-MS) (Gavin, Maeda, & Kuhner, 2011) technologies. Y2H and PCA are based on direct interaction between

proteins which are called physical interaction networks. AP-MS establishes a link between proteins that are physically interacting due to a common function in a cell.

Many interactions between proteins are still not known. Efforts are being made to find unknown interactions and also to test the correctness of known interactions (von Mering et al., 2002). Integrating the different networks is also an important development for the future of unknown interactions. These integrated PPI networks have also different purposes, e.g., use in disease classification or survival time prediction (Dao et al., 2010; Lee, Chuang, Kim, Ideker, & Lee, 2008). Some of the highly used integrated PPI networks are; STRING (D. Szklarczyk et al., 2015), GeneMANIA (Zuberi et al., 2013), mentha (Calderone, Castagnoli, & Cesareni, 2013). These global integrated PPI networks have a wide range of uses such as in understanding complex diseases (Cho et al., 2012), system biomedicine (Sevimoglu & Arga, 2014) etc.

2.3 Synergistic Drug Combinations

Synergistic Drug Combinations mean that drugs have positive effects in the treatment when these drugs are combined. Here, we have reviewed previous works made on synergistic drug combinations.

Different diseases can share common molecular pathways or targets in a cell. For this reason, a drug can be used for another purpose. Repurposing non-cancer drugs as an anti-cancer treatment offers a new opportunity, notably when drugs are used in combination (Boguski, Mandl, & Sukhatme, 2009; Gupta, Sung, Prasad, Webb, & Aggarwal, 2013). Identification of ultimate drug combinations (e.g., pairs, triples etc.) as a new therapy regime is very expensive and time-consuming procedure even for the cell-line experiments. Therefore, new in-silico methods have been proposed to suggest chemically more effective drug combinations for later wet-lab experiments.

Recently, systems biology approaches have made promising contributions to identify better (synergistic) drug combinations as new treatment regimes (Chen, Liu,

Yang, Yang, & Lu, 2015; Ryall & Tan, 2015). Huang *et al.* developed the DrugComboRanker algorithm that ranks potential drug combinations by choosing drugs with high overlap in the disease network and affecting multiple signalling pathways (Huang et al., 2014). Another study proposed the DIGRE model to predict drug combination effects by modelling drug response dynamics and gene expression changes after individual drug treatments (Yang et al., 2015). The DEMAND algorithm is another recent study, which developed a regulatory network-based approach that elucidates genome-wide drug mechanisms after drug treatments (Woo, Shimoni, Yang, Subramaniam, Iyer, Nicoletti, Rodríguez Martínez, et al., 2015).

Sun *et al.* developed the Ranking-system of Anti-Cancer Synergy (RACS) to rank drug combinations from the most synergistic to non-synergistic (Sun et al., 2015). To do this, they combinied features of targeting networks and transcriptome profiles. They focused on three types of cancer. RACS is a semi-supervised learning model which addresses the limited positive/labelled samples and a set of unlabelled combinations.

2.4 Artificial Neural Networks

With the inspiration of the biological nervous system, artificial neural networks (ANN) were developed. Biological nerve cells communicate with each other through synapses. A nerve cell sends information to the other cells via its axons (Figure 2.4). Similarly, the artificial nerve cells generate an output by passing the information from the outside through an aggregation function and an activation function, and send it to the other cells (process elements) over the network connections.

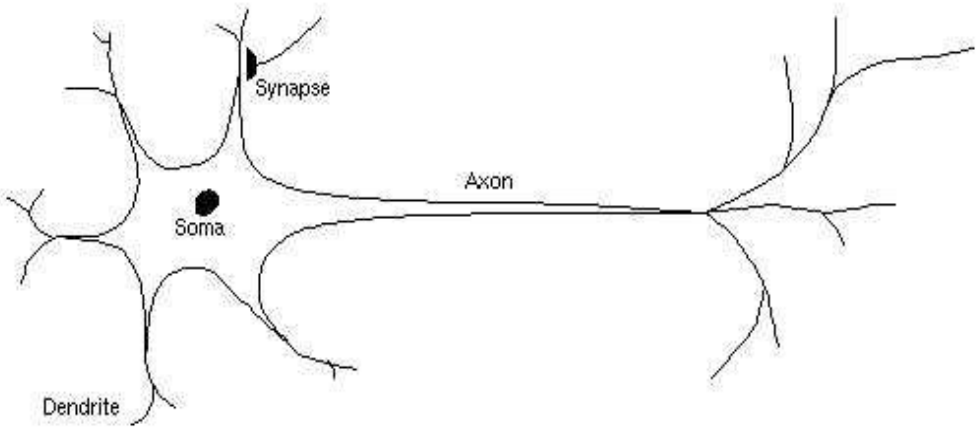


Figure 2.4 Biological nervous system. (Figure is adapted from ("Neural Networks - Neuron," n.d.)

When transforming Figure 2.4 to Figure 2.5, soma becomes neuron, dendrites become inputs, axon becomes output and synapses become weights. ANN with single neuron is called perceptron.

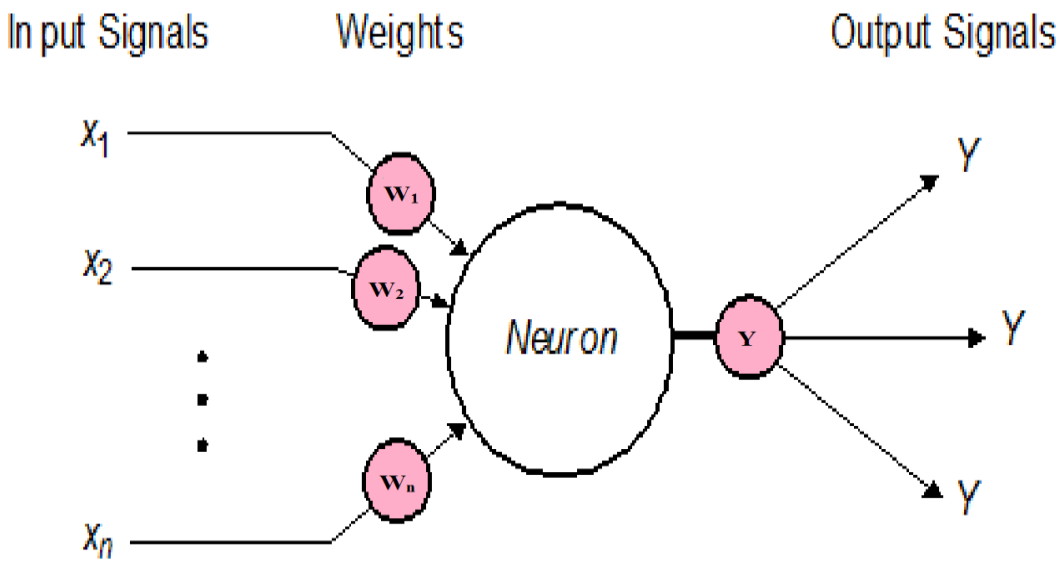


Figure 2.5 Basic ANN structure resembles biological nervous system.

The values of the links connecting artificial neural networks to one another are called weight values. The process elements form a network with three parallel layers; input layer, intermediate layer (hidden layers), output layer (Figure 2.6).

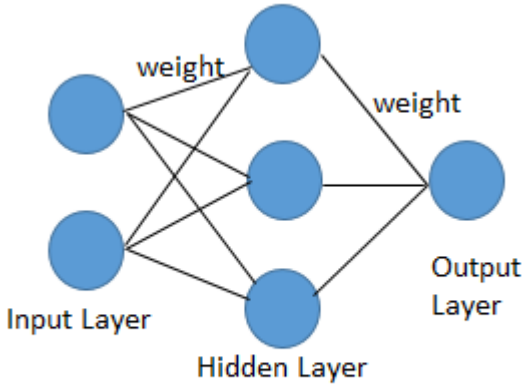


Figure 2.6 Layers of ANN

Information is transmitted to the network through the input layer. They are processed in intermediate layers and the results are sent to the output layer. The network weight values are used in this process. At the beginning, these values are assigned randomly. As the network is trained with training data, the weights will reach the optimum values. Although what individual weights mean is not known, it can be said that the network's intelligence about the inputs is given by using these weights. It is possible for the network to learn an event well by choosing the most accurate artificial neural network model for that event. This can be accomplished by determining the number of hidden layers in the intermediate layers and the corresponding weight values.

2.4.1 Layers and Neurons

A sufficient number of hidden layers must be used to make a better classification. We can draw a decision plane according to chosen ANN’s layer size and its total neurons in each layer (Figure 2.7).

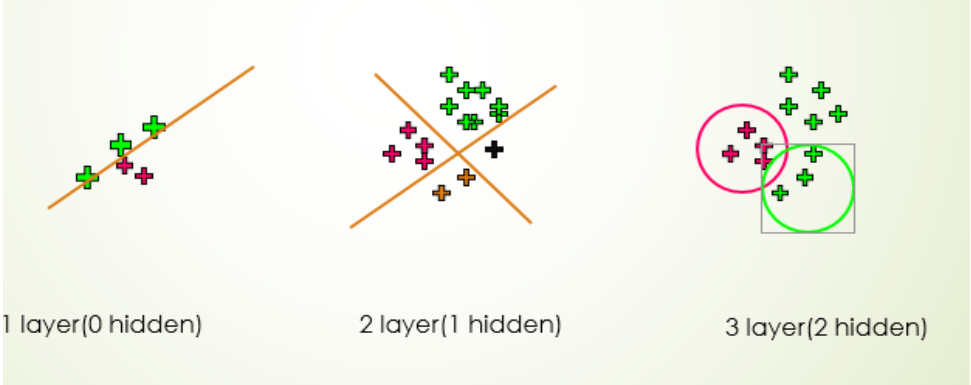


Figure 2.7 Significance of hidden layers.

Addition to the importance of hidden layers in the data classification, there is a crucial role of number of neurons used in hidden layers. ANNs with more hidden neurons can express more complicated functions as classifiers (Figure 2.8). However, over-fitting might occur when a model with high capacity fits the noise in the training data.

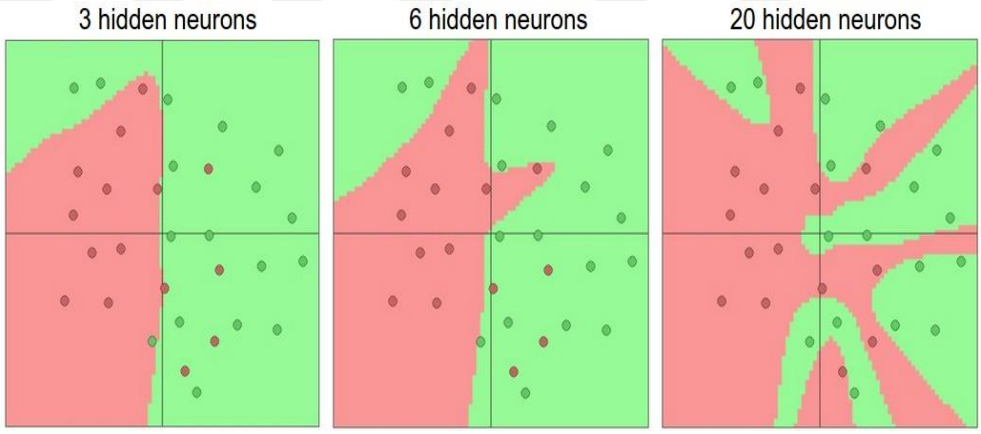


Figure 2.8 Significance of neurons in a hidden layer. (Figure is adapted from ("CS231n Convolutional Neural Networks for Visual Recognition," n.d.)

In fact, there is no method to set the exact number of hidden layers and the number of neurons in each layer to represent the training data in the best way. These can be found by experimentally. Each of the neurons in the input layer is connected to all of

the neurons in the hidden layer. And all of the neurons in the hidden layer are connected to all of the neurons in the next hidden or the output layer.

2.4.2 ANN Types

Basically two different ANN methods can be mentioned. The first one is a feed-forward neural network, which doesn't have any loop-back to previous layers or to the same layer (Figure 2.9).

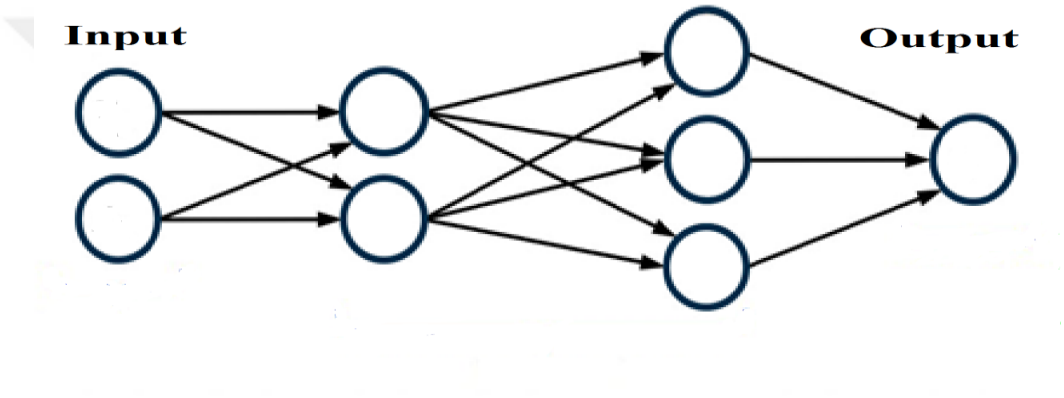


Figure 2.9 Feed-forward neural network.

The second type of ANN is feed-back neural networks (Figure 2.10), which have some loops returning to previous layers or to the same layer.

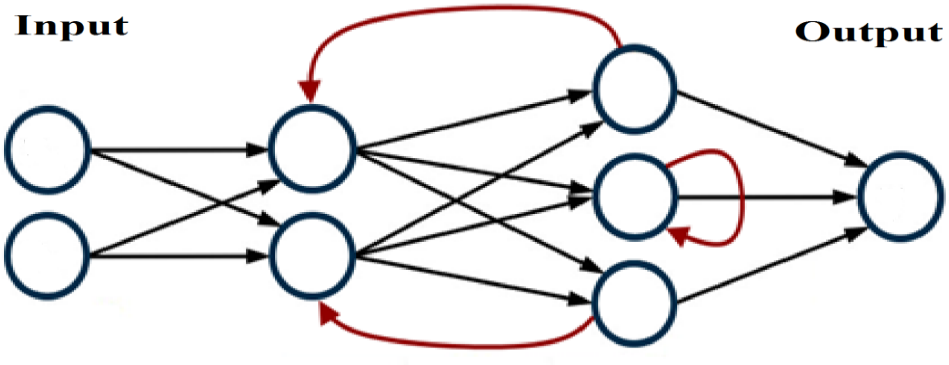


Figure 2.10 Feed-back neural network.

2.4.3 Training of ANN

ANNs are commonly designed as multi-layer perceptron (MLP) instead of single perceptron. Due to MLP networks work in a supervised manner, both inputs and outputs are provided to the network during the training. The philosophy of the learning is that the difference (error) between the output produced by the network during training and the expected output is distributed to the weights of the network to reduce this learning error in time (Figure 2.11).

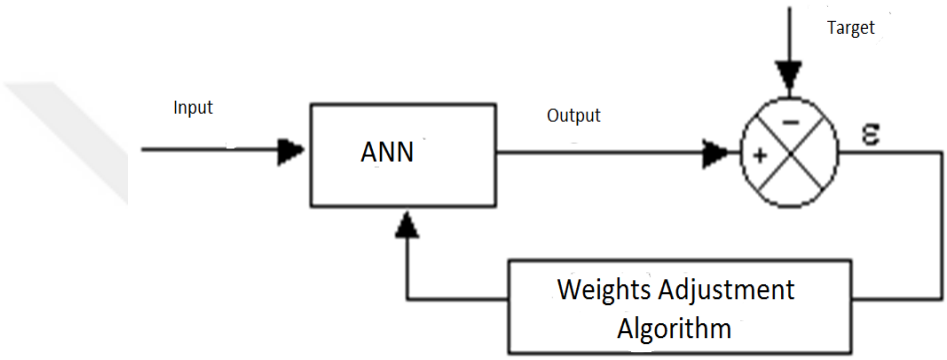


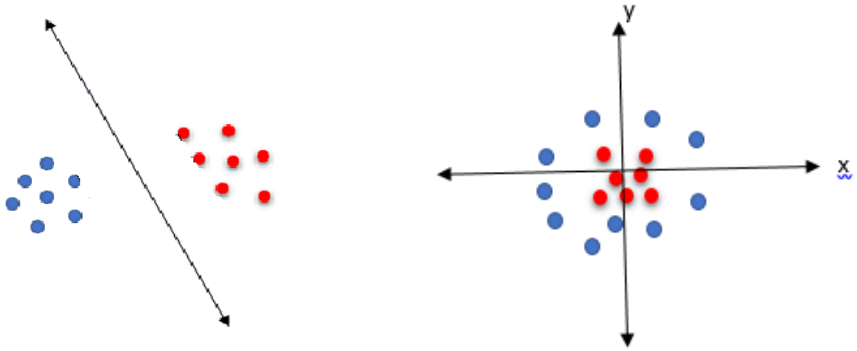
Figure 2.11 Training ANN using the error of the result.

The distribution to the weights are adjusted by multiplication of a specific coefficient. This coefficient is called learning rate.

2.5 Support Vector Machines

Support Vector Machine (SVM) is a machine learning method proposed for classification and regression problems in data sets where the inter-variable patterns are unknown. SVM aims to make accurate estimation and generalization of new data by firstly learning in training data. SVM is a nonparametric classifier, i.e., there is no presupposition assumption about the distribution of the data. Inputs and outputs are matched in training sets.

SVM is basically divided into two categories according to the linearity of the data set. If the data can be separated linearly, it is called linear SVM (Figure 2.12, (a)). When the data are not linearly separable, it is called non-linear SVM (Figure 2.12, (b)).



(a) (b)

Figure 2.12 (a) Linear SVM versus (b) Non-linear SVM.

2.5.1 Linear SVM

If two classes are to be classified, an infinite number of planes can pass between these classes (Figure 2.13). The aim of the SVM is to find the hyper-plane that maximizes the distance between the nearest samples of two classes. As seen in Figure 2.14, SVM makes calculations to find two support vectors with the highest margin value.

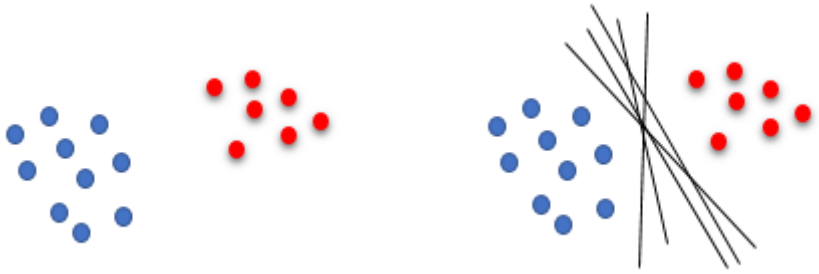


Figure 2.13 Infinite number of planes pass between two classes.

When classes are seperated without noise, it is called hard-margin.

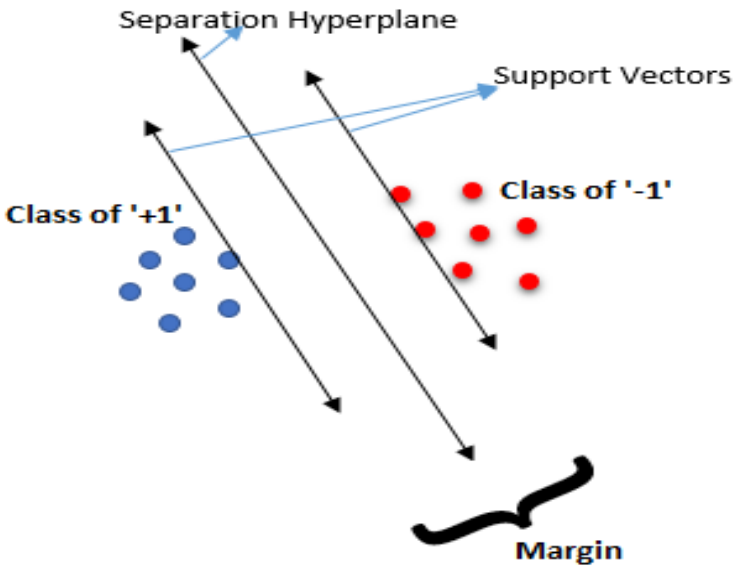


Figure 2.14 Support vectors with maximum margin

Suppose that the blue points represent +1 class and the red points represent -1 class in Figure 2.14. If we call the support vector passing through the +1 class *SP1* and the other is *SP2*, then,

$$SP1 = < w, x_1 > + b \geq 1, \quad (2.1)$$

$$SP2 = < w, x_2 > + b \leq -1 \quad (2.2)$$

where, w is weight vector, b is bias, x_1 and x_2 represent the sample points. Taking the inequalities 2.1 and 2.2 into consideration, *SP1* plane with **w** normal and perpendicular distance |1-b | / ||w|| from origin and *SP2* plane with **w** normal and perpendicular distance |-1-b| / ||w|| from origin are parallel planes. *SP1* and *SP2* boundary planes are therefore equidistant from the separation hyper-plane. There are no training samples between the *SP1* and *SP2* boundary planes. However, the training examples existing on the planes become the support vectors that are the training examples closest to the hyper-planes. The separation hyper-plane is the plane passing through the center of the

boundary, which maximizes the distance between the support vectors of both classes. After support vectors are determined, new test samples can be given to separation hyper-plane which is known as the plane passing through the middle of the support vectors;

$$SH = \langle w, x_n \rangle + b = 0 \quad (2.3)$$

When the data can be separated by some noisy samples, it is called soft-margin. In soft margin, calculations are done in the similar way as in the hard-margin. The data can be separated by a few noises at the same dimension.

2.5.2 Non-Linear SVM

Nonlinear SVMs can be used when data samples cannot be separated by a linear function. In real life problems it is often not possible to linearly separate a data set with a hyper-plane. Therefore, the separation of classes is possible by estimating a separation curve. However, in practice it is difficult to predict such a curve (Figure 2.15).

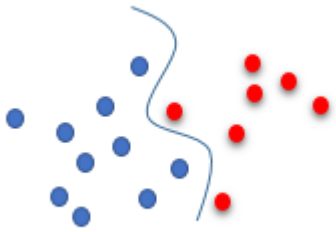


Figure 2.15 Separation of non-linear SVM with a curve

In this case, it is necessary to map the data to a higher dimensional space. If a data set cannot be linearly separated in an n-dimensional space, the data points are mapped to an upper dimension by using the kernel trick as shown in Figure 2.16.

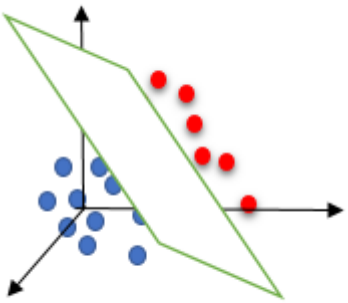


Figure 2.16 Non-linear SVM in an upper dimension

Kernel trick is used to map data from input space to a higher dimensional space which is called feature space. The choice of kernel function has an important role in the performance of the classification. The most commonly used kernel functions are:

-linear function: $K(x, x') = x^T x'$

-polynomial function: $K(x, x') = (1 + x^T x')^d$

-radial basis function: $K(x, x') = \exp(-\|x - x'\|^2 / 2\sigma^2)$

-sigmoid function: $K(x, x') = \tanh(\eta \, x \, x' + \theta)$

After obtaining future space, SVM try to find a hyper-plane with maximum margin. To do this, decision function uses dot product between feature vectors of samples. Instead of symbolizing the space clearly for ideal hyper-plane, kernel functions apply a dot product in the future space. (Vapnik, 1998)

2.6 Random Forests

Random Forest (RF) is a combination of multiple decision trees. Instead of producing a single decision tree, RF aims to combine the decisions that result in the training of multiple and multivariate trees with different training clusters. It generates multiple classifiers instead of only one, and then classifies new data with votes from their estimates.

The algorithm begins with the division of the training data into a predetermined number of subsets. The data consist the records, features and class information of each record (Table 2.1). Subsets do not have to be completely different from each other, i.e., subsets can contain overlapping records (Figure 2.17). This method is called bootstrap aggregating (i.e., bagging).

Table 2.1 Data for RF consist records, features and class information

	Feature 1(F1)	F 2	F 3	F 4	F 5	F 6	F 7	F 8	F 9	Cl ass
Record 1(R1)										
R2										
…										
R90										

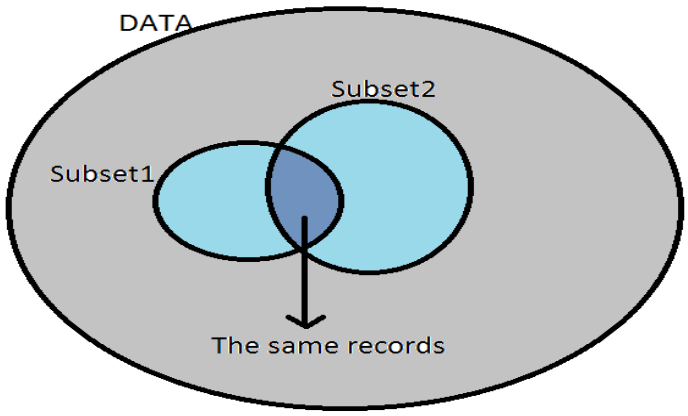


Figure 2.17 Overlap of subsets

The next step of the algorithm is to obtain feature subsets from each subset (Table 2.2). There are many combinations of feature selection. The algorithm finds different combinations of features by running a few iterations. The best feature subsets of each

subset can be found by using misclassification rate (Equation 2.4) after running of all iterations. To calculate misclassification rate, confusion matrix is calculated by using an unseen data when a subset is obtained (Table 2.3, Figure 2.18).

Table 2.2 A record subset of data

	F1	F5	F8	Class
R1				
…				
R35				
…				
R75				
Total Record=60				

Table 2.3 Confusion matrix

	Predicted NO	Predicted YES
Actual NO	TN	FP
Actual YES	FN	TP

$$Misclassification\ Rate = \frac{FN+FP}{Total} \tag{2.4}$$

where *TN* is true negative, *FP* is false positive, *FN* is false negative and *TP* is true positive.

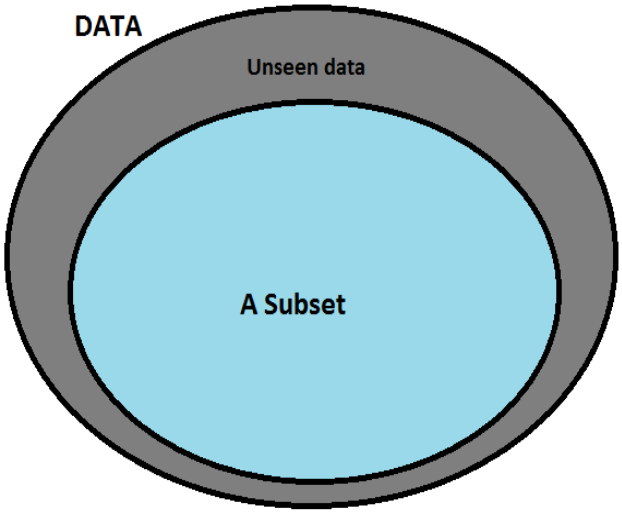


Figure 2.18 Unseen data is the rest of data after extracting a subset

The best feature subset is the one, which has the lowest misclassification rate. After finding the best feature subset, these subsets are transformed into decision trees.

For instance, the data set contains 90 records with 9 features and target class information. To generate 500 decision trees, 500 subsets must be obtained from the data. Each subset contains 60 records of the data, the remaining 30 records will be used in the calculation of the performance by a confusion matrix. For each subset, feature subsets will be generated. Feature subsets include 3 features for 60 records as seen in Table 2.2. After calculating the confusion matrix and misclassification rate for each feature subset, a feature subset with the smallest misclassification rate is represented as a decision tree. Thus, 500 decision trees are obtained from 500 different subsets. In this case, there is now a trained RF. For a new record whose class is unknown, RF runs all decision trees. Whichever class label is generated mostly from the 500 decision trees, that class is assigned as the target class for the new record. Continuing the previous example, supposing that there are 2 classes, which are A and B. For a new record, if 251 decision trees generated class A as a result and 249 decision trees generated class B, the target class of new record will be decided as class A.

2.7 Novelty of the Proposed Study

We have reviewed some studies made about finding synergistic drug combinations in Section 2.3. These studies developed tools to rank drug combinations calculating features or tried to predict drug combination effects.

We used similar drug data and calculated some similar features to classify drug combination effects using well-known machine-learning algorithms. We developed a classifier of drug combination effects. Our model is currently able to work on gene expression data. As a novelty, Drug Perturbation Networks (DPN) are extracted instead of only using gene expression data for feature calculation. We compared the features to find better combinations and machine-learning methods to reach the best performing classifier.

Novelty of this study is building a new classifier which produces the best feature combinations out 6 with a compatible machine-learning method out of three methods using DPNs of each drug extracted from gene expression data. After finding the most efficient classifier, we expect to find synergistic drug combinations for later wet-lab experiments.

CHAPTER THREE
METHOD

3.1 System Overview

We developed a classification tool that aims to identify effective drug pairs out of all combinations. When we look at the overall flow of the system it is possible to explain it in five steps (Figure 3.1). In the first step, Drug Perturbation Network (DPN) was extracted from a single drug treatment data for each drug (see Section 3.2 and 3.3). As a second step, six features were calculated for all combinations of drugs (see Section 3.4). Thirdly, these combinations were divided into a test set and a training set with a predetermined percentage. Three different standard machine-learning methods, which are ANN, SVM and RF, are trained using only the training data in the fourth step. Finally, we used an evaluation method to assess the performance of trained models on the test set.

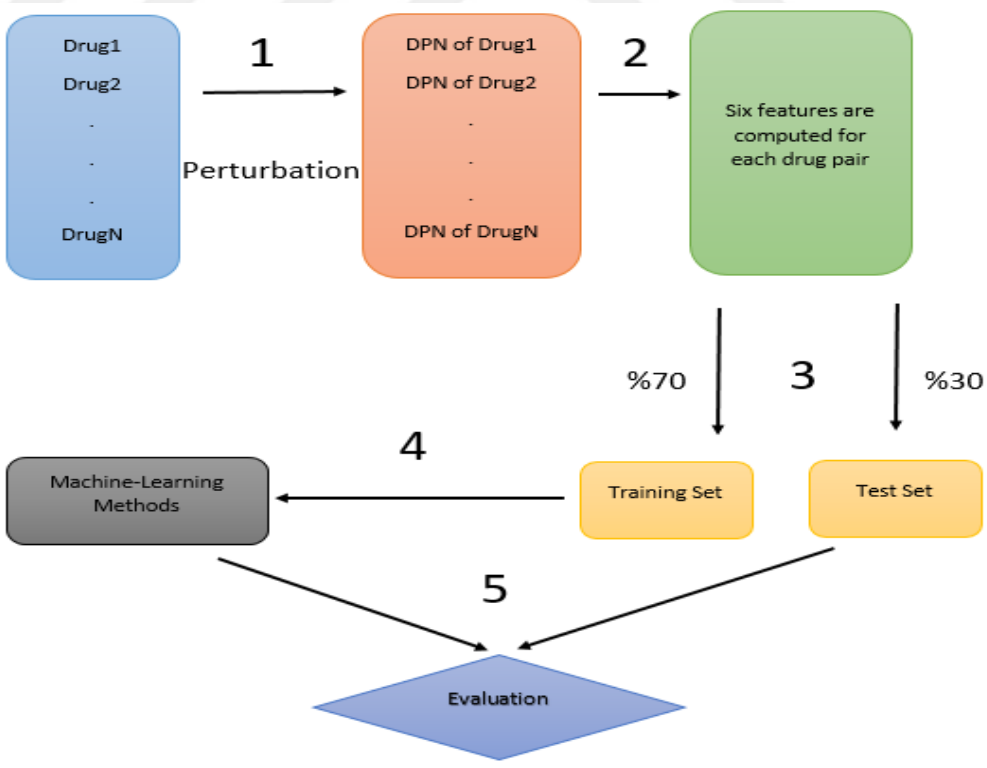


Figure 3.1 The pipeline of the proposed model.

3.2 Drug Treatment Data

It is quite challenging situation to obtain a drug data set that contains the best and worst drug combinations information. However, we managed to find out two different data. They contain different number and types of drugs, but they both contain gene expression samples for each drug.

The first drug treatment data we used is provided from drug synergy prediction competition conducted by the NCI-DREAM Consortium (Bansal et al., 2014). It contains gene expression data for fourteen FDA-approved drugs before and after drug treatment on B-cell lymphoma cancer cells. Each drug experiment contains gene expression values, which are measured after six hours, twelve hours and twenty-four hours of drug treatment. Untreated (control) samples are also measured. There were three replicates for each measurement. The NCI-DREAM Consortium also provided labels for pairwise drug combinations: positive or negative. Fourteen drugs present ninety-one pairwise combinations but we removed five of these combinations as a result of division by zero due to the ratio of shared Gene Ontology (GO)) terms by two drugs in the third feature (Mutual Information on Biological Process) which is explained in Section 3.4, Eighty-six drug pairs are labelled in this dataset: seventeen drug pairs are positive (synergistic) samples, sixty-nine of them are negative ones. We call this data as *DREAM* data in the rest of the thesis.

As a second data set, we aimed to obtain drug data used in different cancer treatments. Therefore, we searched pairwise drug combinations used in treatment of breast (MCF7), leukemia (HL60) and prostate (PC3) cancer cells in Drug Combination Database (DCDB) (Liu et al., 2014). We identified seventeen drug pairs out of twenty different drugs. We considered these seventeen drug pairs as positive and the rest of combinations of twenty drugs became our negative set. We removed some combinations in results of some calculations as it happened at *DREAM* data. As a result, seventeen positive and one hundred eighty-seven negative pairs formed our second data. Gene expression data of these drugs are collected from CMap (Lamb, 2007) and GEO (“Home - GEO - NCBI,” n.d.). Drug data from CMap contain fold

change (FC) value and *p*-value for each gene affected due to the drug treatment. So we arranged the drug data from GEO as there will be FC value and p-value for each gene. This data set is named as *Mixture* data.

3.3 Extracting Drug Perturbation Networks (DPN)

We aimed to design more effective features for the classification of drug combinations. The biological network (i.e., drug perturbation network - DPN), which is affected due to the drug treatment, might be beneficial to predict better drug combinations.

We applied a new algorithm – DEMAND – to compute a drug perturbation network (DPN) for each drug in *DREAM* data (Woo, Shimoni, Yang, Subramaniam, Iyer, Nicoletti, Rodriguez Martinez, et al., 2015). The DEMAND algorithm requires both control and treated samples of a compound with a PPI network as an input. The DEMAND algorithm generates a network by using the Gaussian Kernel method to calculate interaction probability density. And it uses KL-divergence to evaluate probability density difference of control and treated samples. Then, the DEMAND algorithm computes *p*-value according to dysregulation of each gene. Gene expression samples for control, 12 and 24 hours were given as the input of the DEMAND algorithm. The other input parameter is the biological network that is a publicly available protein-protein interaction (PPI) network: STRING (D. Szklarczyk et al., 2015). The DEMAND algorithm computes a DPN for each drug. We applied 0.05 *p*-value threshold for the selection of significant genes of each DPN (Figure 3.2).

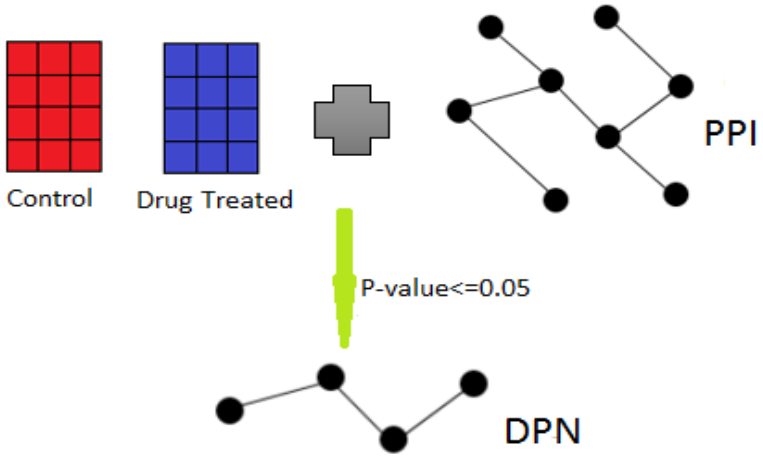


Figure 3.2 Summary of DEMAND algorithm.

For the Mixture data, we computed DPNs for each drug using the FC value and the *p*-value that we mentioned in the previous section. We selected the same *p*-value applied for DREAM data, while FC value has been greater than 1,25.

3.4 Computation of Features

After obtaining a DPN for each drug for both data sets, we calculated six features as the input for machine-learning methods.

3.4.1 Shortest Distance of Two Drugs

For this feature we used the known target proteins of each drug. A drug target is a protein or enzyme, which is affected by the designed drug, and its original function in the cell is changed or corrupted after binding of the given drug to its binding pocket. We utilized the STITCH database for finding the targets of drugs. STITCH is a searchable database that coordinates data obtained from metabolic pathways, crystal structures and drug–target connections (Damian Szklarczyk et al., 2016). The distance between two proteins (one target of different two drugs) is the shortest path length between these proteins in the PPI network. Each drug generally has more than one

protein target. So, in order to normalize shortest path lengths for all possible pair of drug targets, we take the average of summation of all possible shortest path lengths. Assume that *i* and *j* are the index of drugs; the *x* and *y* indexes represent the targets of drug *i* and drug *j*, respectively.

$$Distance = \frac{1}{M.N} \sum_{x=1}^{M} \sum_{y=1}^{N} SP(Ty, Tx) \quad (3.1)$$

where *M* is total number of targets of drug *i*, *N* is the total number of targets of drug *j*, *SP* is the shortest path of length of two proteins in the STRING - PPI network.

3.4.2 GO Term Similarity

Gene Ontology (GO) is one of the well-established databases for the functional analysis of a given protein set. There are three fundamental categories available in GO: Biological Process (BP), Molecular Function (MF) and Cellular Component (CC). We only focused on the BP annotations. To find GO term similarity of BP annotations of proteins covered in DPN, we used the "GOSemSim" package in R-Bioconductor. The 'clusterSim' function which uses "Resnik" method to compute the similarity of two given proteins based on their BP annotations.

$$Resnik(c1, c2) = IC\ (\ LCS\ (c1, c2)) \quad (3.2)$$

where *LCS* is the lowest common subsumer which is a concept in lexical taxonomy that defines the shortest distance of two clusters and *IC* is the information content which is the logarithm of the probability of finding the concept in a given corpus.

3.4.3 Mutual Information of Biological Processes

Only the cancer related BP annotations from Gene Ontology for target proteins of two drugs are considered for this metric.

$$MI(i,j) = P(i,j) \ * \ log \ \frac{P(i,j)}{P(i)*p(j)} \tag{3.3}$$

where *P(i)* and *P(j)* are the ratio of cancer related BP annotations for the targets of drug *i* and *j*; *P(i, j)* is the ratio of these GO terms shared between targets of drug *i* and *j*.

3.4.4 *Overlap of DPN*

Each DPN network covers several proteins that are affected after the application of a drug to the cells. To observe the similarity between two drugs, we computed the Jaccard index by using proteins covered in the individual DPNs of two drugs. So, assume that *Dx* and *Dy* contain the set of proteins in the DPN of drug *x* and *y*, respectively.

$$JI(Dx, Dy) = \frac{Dx \cap Dy}{Dx \cup Dy} \tag{3.4}$$

3.4.5 Efficacy based on Degree

Degree indicates how many edges (neighbors) a node has in a graph. In this case, the ratio of sum of degrees of known protein targets of a drug in DPN to sum of degrees of known protein targets of the drug in STRING-PPI network generates our fifth feature.

$$Efficacy \ Degree = \frac{\sum degree(i) \ in \ DPN}{\sum degree(i) \ in \ STRING} \tag{3.5}$$

where *i* represents total protein targets of both drugs whereas DPN is the combination of DPNs of both drugs.

3.4.6 Efficacy based on Betweenness

Betweenness gives the number of occurrences in the shortest distances between every two nodes in a graph. Similar to fifth feature, we calculated betweenness of all protein targets of a drug in DPN and in STRING-PPI and the ratio of them generates the sixth feature.

$$Efficacy\ Betweenness = \frac{\sum betweenness(i)\ in\ DPN}{\sum betweenness(i)\ in\ STRING} \quad (3.6)$$

where *i* represents total protein targets of both drugs whereas DPN is the combination of DPNs of both drugs.

3.5 Machine-Learning Methods

We have selected three different machine-learning methods to test data sets with these features. We mentioned the algorithms of these methods in Chapter 2. Here we will explain the parameters we chose in the implementations of these methods.

We used “neuralnet” R package for the implementation of ANNs. The structure has had one hidden layer with 3 neurons. Learning rate was 0.3 and “backpropagation” has been chosen as the learning algorithm.

To implement SVM, “e1071” package is utilized in R-Bioconductor. Basically, three parameters have to be set. The first one is choosing the kernel function. We decided to run SVM with two different kernel functions to select the best one. The kernel functions we tested are “sigmoid” and “radial”. For the rest two parameters, we used a function called “svm.tune” from the same package, which produces optimum ‘cost’ and ‘gamma’ values for kernel functions.

RF algorithm only needs a parameter, which is the number of trees that it creates. For RF, we used a package called “randomForest” in R-Bioconductor.

3.6 Cross-Validation

When a data set contains limited number of samples, it is more convenient to test the performance of machine learning methods by applying a cross-validation (CV) scheme. We used two different CV methods in our study. The first one is Monte Carlo simulation. We implemented a different idea for data as a second approach.

In Monte Carlo simulation, training data are chosen randomly with a predetermined percentage and the rest of data becomes the test data (Figure 3.3). We applied 10-fold cross validation to get consistent results.

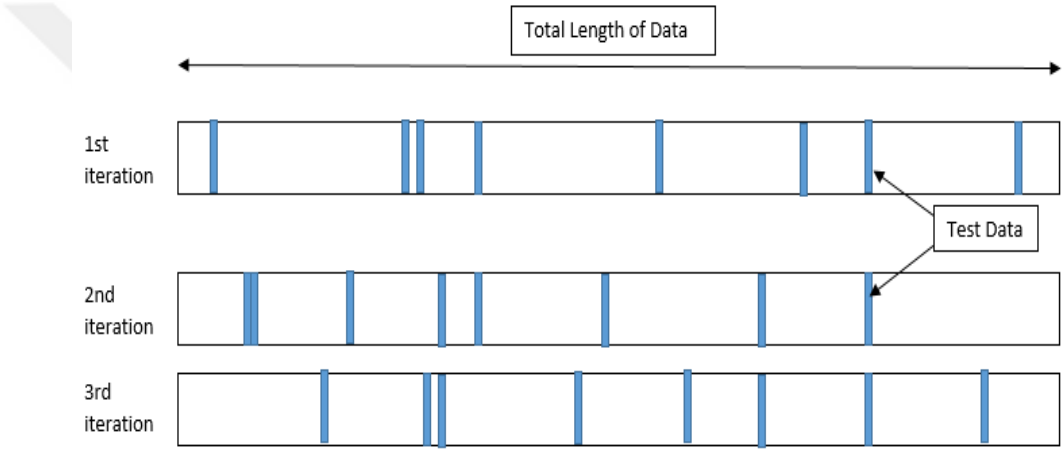


Figure 3.3 Presentation of Monte Carlo simulation

As a second approach, we partitioned the negative sets so that each part of the partition would be equal to the size of the positive sets. Then, we combined each negative part with positive set and applied 10-fold Monte Carlo simulation to each part. Finally, we computed average performance over all parts. For *DREAM* data, we got 4 negative parts in equal size to positive set and for the second data, we got 11 negative parts in equal size to positive set in *Mixture* data.

3.7 Evaluation

We used some evaluation methods to measure the performance of the system. All of these metrics are based on the true and false estimates mentioned in Table 2.3

(Section 2.6). To remember the variables used in Table 2.3, we constructed Figure 3.4 that is a confusion matrix.

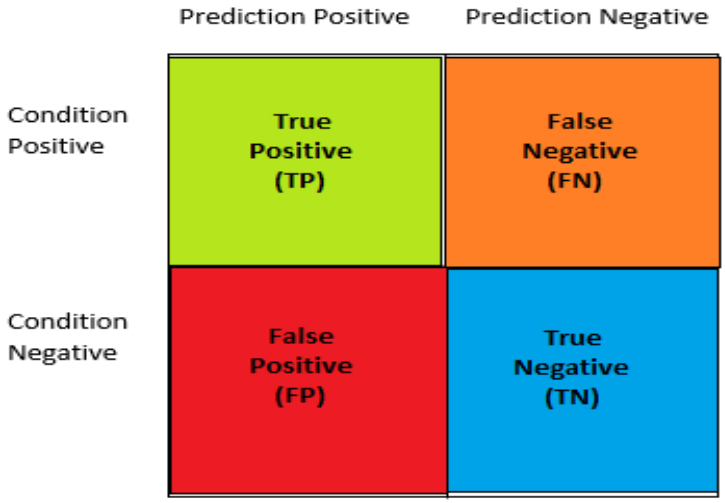


Figure 3.4 Confusion matrix based on conditions and predictions

TP is true positive prediction in which the method predicted the label of drug pair as positive that is also positive in the original data. *TN* is true negative, true label of the drug pair is predicted by the method as negative, which is the same in the original data. *FP* is false positive whose true label is negative but the method predicted as positive. Similarly, *FN* is false negative, whose true label is positive but the method assigned as negative.

3.7.1 Accuracy

Accuracy gives the percentage of correct estimated results. Classification algorithms are used to classify all classes correctly. In this case, as well as the *TP*, the performance of the system is affected by the *TN*.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{3.7}$$

3.7.2 Precision and Recall

Precision is the measure of certainty or quality while recall is the measure of completeness or quantity. For example, assume that there are 10 letter of A and 6 letter of B in a text. A method identifies 8 letter of A but actually 5 of them is letter A, precision is 5/8 while recall is 5/11.

$$Precision = \frac{TP}{TP+FP} \tag{3.8}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.9}$$

When we consider the class of positive as a relevant element in our problem, if precision value is high, results of the algorithm are more relevant than irrelevant ones whereas high recall value is an indicator of that most of the results are relevant.

3.7.3 F-measure

This measure is a harmonic mean of precision and recall. This measure can be considered as an average of these two measurements.

$$F_1 = \frac{2*Precision*Recall}{Precision+Recall} \tag{3.10}$$

This measure is also called F_1 measure because recall and precision are evenly weighted in this formula. $F_{0.5}$ and F_2 are kinds of this measure where weight of recall higher than precision in F_2 while precision's weight is higher than recall in $F_{0.5}$. We used F1 measure in our method.

CHAPTER FOUR
RESULTS

The proposed supervised model aims to classify drug pairs into positive (more effective) and negative (not useful combinations) classes. Six new features were implemented to incorporate various biological information for the improvement of the solution. Machine-learning methods were trained by using six features with two different drug data sets.

Different configurations were experimented to analyse the performance of the machine-learning models in this challenging classification problem. We first compared the kernel functions to determine which kernel to use in next experiments. Another configuration is the effect of partitioning the data into training and test sets in different sizes. After deciding percentages of partitioning, we tested individual performance of features and the best combinations of features. Then, the effects of data on the method have been discussed. Finally, we compared the methods of machine learning with each other.

4.1 Comparison of Kernel Functions for SVM

The performance of SVMs can change due to various parameters of the model. We compared two kernel functions that are radial and sigmoid functions using two different data sets. For this purpose, we applied the Monte Carlo simulation in which the *DREAM* data was partitioned as training (70%) data and test data (30%) samples. Monte Carlo simulation selects 70% of data as training set randomly and the rest 30% becomes test set. The training accuracy of SVM with radial kernel reached 100% for *DREAM* data while sigmoid kernel was 72%. The test performance was 80% when radial function is used for *DREAM* data, while sigmoid function's accuracy was only 67% (Table 4.1).

Table 4.1 The performance comparison of kernel functions in the DREAM data.

	Accuracy	
Kernel Functions	**Train**	**Test**
Sigmoid	0.720	0.670
Radial	1.000	0.800

We tested both kernels on the *Mixture* data by applying the second cross-validation method. The second type cross-validation divides negative set into parts which have the same size of positive set and then combines each negative part with positive set. After using Monte Carlo simulation for all parts, average of all parts are calculated. The same partitioning of training and test set was applied. Sigmoid kernel forcibly passed 50% accuracy for both training and test set (Table 4.2). Radial kernel reached 99% accuracy in training set and 72% for test set.

Table 4.2 The performance comparison of kernel functions in the Mixture data.

	Accuracy	
Kernel Functions	**Train**	**Test**
Sigmoid	0.550	0.580
Radial	0.990	0.720

These performances were calculated as the average accuracy of the 10 fold cross-validation using six features together. The results show that the radial kernel function has a more successful classification performance than sigmoid one. As a result of these experiments, we decided to use the radial function in the rest of SVM computations.

4.2 The Effects of Data Partitioning

We aimed to find the most efficient partitioning percentage while separating data samples into training and test set. Three types of partitioning have been tested: 80%-20%, 70%-30% and 60%-40% as training and test set, respectively. RF and SVM with

radial kernel were run for each partition. The *Mixture* data were used with the second type cross-validation and all features together.

Table 4.3 SVM results in different partitioning using all features as input

	Accuracy		*Precision*		*Recall*		*F1*	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
%60-40	0.999	0.696	1	0.753	0.998	0.694	0.999	0.708
%70-30	0.999	0.718	1	**0.787**	0.997	0.722	0.999	**0.738**
%80-20	0.999	**0.723**	0.999	0.785	0.998	**0.72**	0.999	**0.744**

The results in Table 4.3 show that 60%-40% partitioning is behind the others. The partitioning of 80%-20% has the best accuracy on test set with a small difference. Precision and recall values give information about the success of predicting true positive values. Partition of 70%-30% and 80%-20% are very close to each other while they are more preferable than the partition of 60%-40% at the test set. F1 measure, which balances precision and recall, also supports these results. Results for training set don't differ from each other.

Table 4.4 RF results in different partitioning using all features as input

	Accuracy		*Precision*		*Recall*		*F1*	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
%60-40	0.692	0.671	0.659	0.739	0.707	0.67	0.678	0.691
%70-30	0.723	0.736	0.686	0.771	0.742	0.762	0.71	0.744
%80-20	0.736	0.73	0.705	0.782	0.753	0.76	0.726	0.737

When we check RF results in Table 4.4, it has similar results with SVM. The partition of 80%-20% and 70%-30% are similar while the partition of 60%-40% are behind them as a result of all evaluation metrics.

We have observed better performances by separating our data by 80%-20% and 70%-30% based on these experiments. For further implementations, we considered these results.

4.3 Comparison of Features

In this section, we have discussed features with their single performance as an input to machine-learning methods. Firstly, we split the *DREAM* data by 70%-30% and gave each feature separately to the RF and SVM. For *DREAM* data, we applied the Monte Carlo simulation.

Table 4.5 Accuracy of single feature performance as RF input using DREAM data

	Train	Test
Shortest Path of Two Drugs	0.672	0.68
Mutual Information of BP	0.688	0.68
GO Term Similarity	0.786	0.72
Overlap of DPN	0.704	0.64
Efficacy based on Betweenness	0.77	0.52
Efficacy based on Degree	0.655	0.76

The features *Efficacy Degree* and *GO term similarity* have reached 76% and 72% accuracy, respectively. These results can be misleading, due to outnumbered samples in negative class (i.e., for 70%-30% partitioning, 20 negative samples and 5 positive samples are used for testing). Here, we have to check precision and recall values but they are not provided in Table 4.5 due to less than 50% performance. Using the *DREAM* data with Monte Carlo simulation and 70%-30% partitioning, *Efficacy Degree* and *GO term similarity* features performed more accurately on classifying negative class.

Table 4.6 Accuracy of single feature performance as SVM input using DREAM data

	Train	Test
Shortest Path of Two Drugs	0.808	0.788
Mutual Information of BP	0.829	0.788
GO Term Similarity	0.803	0.8
Overlap of DPN	0.803	0.8
Efficacy based on Betweenness	0.847	0.76
Efficacy based on Degree	0.844	0.784

With the same condition, the performance of each feature on SVM was very close to each other (Table 4.6). However, it is quite difficult to compare features as it happened in RF calculations for the *DREAM* data using Monte Carlo simulation.

Due to this challenging situation, we used the *Mixture* data with the second type of cross-validation. With the same setup, the partition of 70%-30% is used for all machine learning methods.

Table 4.7 Single feature performance as ANN input using Mixture data

	Accuracy		**Precision**		**Recall**		**F1**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
Shortest Path of Two Drugs	0.51	0.52	0.49	0.49	0.57	0.6	0.53	0.56
Mutual Information of BP	0.62	0.58	0.59	0.58	0.64	0.61	0.63	0.59
GO Term Similarity	0.56	0.49	0.53	0.45	0.57	0.48	0.58	0.52
Overlap of DPN	0.53	0.48	0.58	0.52	0.53	0.48	0.59	0.55
Efficacy based on Betweenness	0.64	0.54	0.63	0.53	0.67	0.53	0.63	0.55
Efficacy based on Degree	0.79	**0.72**	0.73	**0.64**	0.85	**0.83**	0.8	**0.69**

According to ANN results in Table 4.8, *Efficacy Degree* is clearly outperforming others. We check the precision and recall values to avoid having the same problem we faced in the *DREAM* data. Precision is 0.638 while recall is 0.833; these results show *Efficacy Degree* is even ahead of other features for classifying positive samples. The feature of *Mutual Information on BP* (*MI on BP*) is as the second best feature (Table 4.7).

Table 4.8 Single feature performance as RF input using Mixture data

	Accuracy		**Precision**		**Recall**		**F1**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
Shortest Path of Two Drugs	0.5	0.49	0.48	0.61	0.5	0.49	0.49	0.54
Mutual Information of BP	0.53	0.55	0.54	0.56	0.53	0.56	0.53	0.55
GO Term Similarity	0.62	**0.57**	0.67	**0.69**	0.61	**0.57**	0.63	**0.6**
Overlap of DPN	0.51	0.49	0.52	0.56	0.51	0.48	0.51	0.51
Efficacy based on Betweenness	0.52	0.52	0.5	0.55	0.52	0.53	0.51	0.52
Efficacy based on Degree	0.78	**0.79**	0.75	**0.76**	0.8	**0.86**	0.77	**0.78**

The feature of *Efficacy Degree* has the highest performance in RF calculations (Table 4.8). In accuracy, *GO term similarity* and *MI on BP* comes after the feature of *Efficacy Degree*. In contrast to results obtained in ANN, the feature of *GO term similarity* is performed higher values than *MI on BP* according to not only accuracy but also F1 measure.

Table 4.9 Single feature performance as SVM input using Mixture data

	Accuracy		**Precision**		**Recall**		**F1**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
Shortest Path of Two Drugs	0.89	0.56	0.85	0.57	0.92	0.6	0.88	0.55
Mutual Information of BP	0.88	0.56	0.85	0.54	0.91	0.59	0.87	0.55
GO Term Similarity	0.9	**0.66**	0.94	**0.68**	0.87	**0.68**	0.9	**0.66**
Overlap of DPN	0.86	0.54	0.85	0.57	0.87	0.54	0.85	0.55
Efficacy based on Betweenness	0.89	0.52	0.9	0.58	0.89	0.51	0.89	0.56
Efficacy based on Degree	0.9	**0.79**	0.84	**0.7**	0.96	**0.89**	0.89	**0.76**

The feature of *Efficacy Degree* showed that it is the best performing one for all methods even with unbalanced data. In SVM, the feature of *GO term similarity* became the second best feature with a better result than it performed in RF (Table 4.9).

In addition to these results, the results of the partition of 80% -20% were obtained to compare features. As we discussed in the previous chapter, there was not much difference between partitioning 80% -20% and 70% -20%.

As an addition to this chapter, we combined two features, which provided the highest performances. We calculated all the calculations by feeding these two features. We combined the features of *Efficacy Degree* and *GO term similarity* as the input of RF and SVM in which these features provided the best performance (Table 4.10). The experiments showed that combining these features improved the performance in SVM but decreased in RF.

Table 4.10 Results of combined two features, Efficacy Degree and GO term similarity

	Accuracy		**Precision**		**Recall**		**F1**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
RF(70%-30%)	0.779	0.76	0.737	0.722	0.808	0.798	0.769	0.738
SVM(70%-30%)	0.963	0.8	0.934	0.747	0.992	0.879	0.959	0.783

4.4 The Effects of Data on the Method

The *DREAM* data set has seventeen positive with sixty-nine negative pairs while the *Mixture* data has seventeen positive pairs with one hundred eighty-seven negative pairs. When the Monte Carlo method is used directly, it is evident that the system classifies negative pairs better than positive ones due to outnumbered negative samples. For this reason, the second cross-validation method is chosen for both data. All features have been calculated for both data sets and given as input to RF. Training set forms 70% of the data while the rests become test data for all iterations.

Table 4.11 The DREAM data versus the Mixture data using the second type cross-validation

	Accuracy		Precision		Recall		F1	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
The DREAM data	0.551	0.605	0.573	0.62	0.549	0.605	0.558	0.597
The Mixture data	0.723	0.736	0.686	0.771	0.742	0.762	0.71	0.744

According to second type cross-validation, the negative sample for *DREAM* data have been divided to four parts, while the *Mixture* data have eleven parts. Results in Table 4.11 shows the average of all parts belong to their own data. The *Mixture* data under the same conditions is more promising than the *DREAM* data. The main reason of this result is the size of data samples.

4.5 Comparison of Machine-Learning Methods

Machine learning methods constitute one of the most crucial point of our method. Because it is critical to choose a machine-learning method that is compatible with these kind of data and features. We ran the method with the three different machine-learning methods (ANN, RF, and SVM) to compare with each other.

We kept all variables constant out of these three methods to make an appropriate comparison. The *Mixture* data with the second type cross-validation have been selected with the combination all features. Training set have been selected as 70% of all samples and test set became the rest for each iteration.

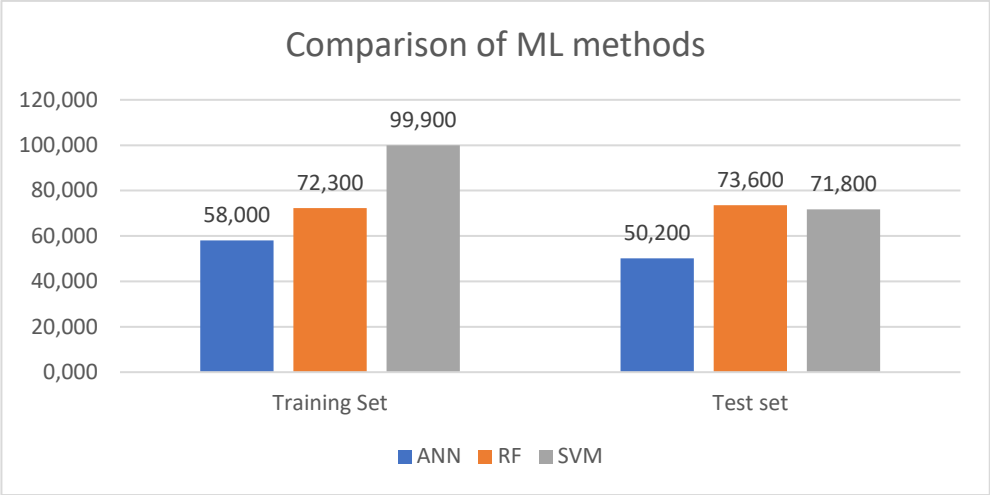


Figure 4.1 Comparison of Machine-Learning methods in training set and test set

When the accuracy of training sets is compared, SVM shows almost perfect result with 99%. However, the accuracy of test set for SVM is not as successful as the training set (Figure 4.1). RF has consistent results in both training and test sets. ANN shows 50% accuracy in test set which means ANN is not effective with this method.

Table 4.12 Comparison of machine-learning methods using the Mixture data

	Accuracy		Precision		Recall		F1	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
ANN	0.58	0.502	0.578	0.547	0.669	0.508	0.597	0.655
RF	0.723	0.736	0.686	0.771	0.742	0.762	0.71	0.744
SVM	0.999	0.718	1	0.787	0.997	0.722	0.999	0.738

RF and SVM have similar results in all evaluation methods for the test set. Although SVM is the best for training set, RF has slightly better results for test set (Table 4.12). SVM was one step ahead of RF (Table 4.10). In general terms, while RF shows balanced results, SVM draws up the results with better feature combinations.

CHAPTER FIVE
CONCLUSION AND FUTURE WORK

The prediction of more effective drug combinations is challenging problem even for the wet-lab experiments. In this study, we proposed a supervised model to classify better and useless drug combinations by implementing six features, testing two different data sets, and running three different machine-learning methods. The type of cross-validation method has a significant role in this model due to imbalanced data. Sometimes, the size of the data samples may not be sufficient to train the model. Although these challenging problems, the model has achieved very successful results in the test case to find prospective promising features and a machine-learning method which may be appropriate for this purpose.

Combination of better features carries up the success of machine-learning methods. Finding more effective features based on gene expression of drugs will contribute to the development of this model. Another important contribution to model is to find cross-validation method which suits imbalanced data. Unsupervised learning method can be an option for the improvement in prediction. Another important point in this method is the step of extracting the DPN. A new algorithm to extract DPN may be more useful. Eventually, another improvement for this study is to generate synergistic drug combination.

REFERENCES

Anders, S., & Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, *11*(10), R106.

Bajcsy, P., Liu, L., & Band, M. (2007). DNA Microarray Image Processing. *DNA Array Image Analysis: Nuts & Bolts (Nuts & Bolts Series)*, 1–77. Retrieved from http://isda.ncsa.illinois.edu/peter/publications/books/DNAPressChapterContent_v9.pdf

Bansal, M., Yang, J., Karan, C., Menden, M. P., Costello, J. C., Tang, H., et al. (2014). A community computational challenge to predict the activity of pairs of compounds. *Nature Biotechnology*, *32*(12), 1213–1222.

Boguski, M. S., Mandl, K. D., & Sukhatme, V. P. (2009). Repurposing with a Difference. *Science*, *324*(5933), 1394–1395.

Calderone, A., Castagnoli, L., & Cesareni, G. (2013). mentha: a resource for browsing integrated protein-interaction networks. *Nature Methods*, *10*(8), 690–691.

Cassone, M., Del Grosso, M., Pantosti, A., Giordano, A., & Pozzi, G. (2008). Detection of genetic elements carrying glycopeptide resistance clusters in Enterococcus by DNA microarrays. *Molecular and Cellular Probes*, *22*(3), 162–167.

Chen, D., Liu, X., Yang, Y., Yang, H., & Lu, P. (2015). Systematic synergy modeling: understanding drug synergy from a systems biology perspective. *BMC Systems Biology*, *9*(1), 56.

Cho, D.-Y., Kim, Y.-A., & Przytycka, T. M. (2012). Chapter 5: Network Biology Approach to Complex Diseases. *PLoS Computational Biology*, *8*(12), 1–11.

CS231n Convolutional Neural Networks for Visual Recognition. (n.d.). Retrieved June 21, 2017, from http://cs231n.github.io/neural-networks-1/

Dao, P., Colak, R., Salari, R., Moser, F., Davicioni, E., Ester, M. et al. (2010). Inferring cancer subnetwork markers using density-constrained biclustering. *Bioinformatics*, *26*(18), 625–631.

Debouck, C., & Goodfellow, P. N. (1999). DNA microarrays in drug discovery and development. *Nature Genetics*, *21*, 48–50.

Fields, S., & Song, O. (1989). A novel genetic system to detect protein-protein interactions. *Nature*, *340*(6230), 245–246.

Gavin, A.-C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M., et al.. (2006). Proteome survey reveals modularity of the yeast cell machinery. *Nature*, *440*(7084), 631–636.

Gavin, A.-C., Maeda, K., & Kühner, S. (2011). Recent advances in charting protein-protein interaction: mass spectrometry-based approaches. *Current Opinion in Biotechnology*, *22*(1), 42–49.

Giot, L. (2003). A Protein Interaction Map of Drosophila melanogaster. *Science*, *302*(5651), 1727–1736.

Gupta, S. C., Sung, B., Prasad, S., Webb, L. J., & Aggarwal, B. B. (2013). Cancer drug discovery by repurposing: teaching new tricks to old dogs. *Trends in Pharmacological Sciences*, *34*(9), 508–517.

Home - GEO - NCBI. (n.d.). Retrieved June 21, 2017, from https://www.ncbi.nlm.nih.gov/geo/

Huang, L., Li, F., Sheng, J., Xia, X., Ma, J., Zhan, M., & Wong, S. T. C. (2014). DrugComboRanker: drug combination discovery based on target network analysis. *Bioinformatics*, *30*(12), 228–236.

Kim, B. C., & Gu, M. B. (2007). Discrimination of toxic impacts of various chemicals using chemical?gene expression profiling of Escherichia coli DNA microarray. *Process Biochemistry*, *42*(3), 392–400.

Kumar, A., Goel, G., Fehrenbach, E., Puniya, A. K., & Singh, K. (2005). Microarrays: The Technology, Analysis and Application. *Engineering in Life Sciences*, *5*(3), 215–222.

Kurashige, Y., Saitoh, M., Nishimura, M., Noro, D., Kaku, T., Igarashi, S., et al. (2008). Profiling of differentially expressed genes in porcine epithelial cells derived from periodontal ligament and gingiva by DNA microarray. *Archives of Oral Biology*, *53*(5), 437–442.

Lamb, J. (2007). The Connectivity Map: a new tool for biomedical research. *Nature Reviews Cancer ,* 54-60.

Lee, E., Chuang, H.-Y., Kim, J.-W., Ideker, T., & Lee, D. (2008). Inferring Pathway Activity toward Precise Disease Classification. *PLoS Computational Biology*, *4*(11), e1000217.

Li, S. (2004). A Map of the Interactome Network of the Metazoan C. elegans. *Science*, *303*(5657), 540–543.

Liu, Y., Wei, Q., Yu, G., Gai, W., Li, Y., & Chen, X. (2014). DCDB 2.0: a major update of the drug combination database. *Database, 2014*, 124-124.

Ma, H., & Horiuchi, K. Y. (2006). Chemical microarray: a new tool for drug screening and discovery. *Drug Discovery Today*, *11*(13–14), 661–668.

Neural Networks - Neuron. (n.d.). Retrieved June 21, 2017, from https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Neuron/index.html

Oh, M.-K., & Liao, J. C. (2000). Gene Expression Profiling by DNA Microarrays and Metabolic Fluxes in Escherichia coli. *Biotechnology Progress*, *16*(2), 278–286.

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139–140.

Ryall, K. A., & Tan, A. (2015). Systems biology approaches for advancing the discovery of effective drug combinations. *Journal of Cheminformatics*, *7*(1), 7.

Saleh-Lakha, S., Miller, M., Campbell, R. G., Schneider, K., Elahimanesh, P., Hart, M. M., et al. (2005). Microbial gene expression in soil: methods, applications and challenges. *Journal of Microbiological Methods*, *63*(1), 1–19.

Sevimoglu, T., & Arga, K. Y. (2014). The role of protein interaction networks in systems biomedicine. *Computational and Structural Biotechnology Journal*, *11*(18), 22–27.

Sun, Y., Sheng, Z., Ma, C., Tang, K., Zhu, R., Wu, Z.,et al. (2015). Combining genomic and network characteristics for extended capability in predicting synergistic drugs for cancer. *Nature Communications*, *6*, 8481.

Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., et al. (2015). STRING v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Research*, *43*(D1), D447–D452.

Szklarczyk, D., Santos, A., von Mering, C., Jensen, L. J., Bork, P., & Kuhn, M. (2016). STITCH 5: augmenting protein?chemical interaction networks with tissue and affinity data. *Nucleic Acids Research*, *44*(D1), D380–D384.

Tarassov, K., Messier, V., Landry, C. R., Radinovic, S., Molina, M. M. S., Shames, I., et al. (2008). An in Vivo Map of the Yeast Protein Interactome. *Science*, *320*(5882), 1465–1470.

Tobler, N. E., Pfunder, M., Herzog, K., Frey, J. E., & Altwegg, M. (2006). Rapid detection and species identification of Mycobacterium spp. using real-time PCR and DNA-Microarray. *Journal of Microbiological Methods*, *66*(1), 116–124.

Trapnell, C., Hendrickson, D. G., Sauvageau, M., Goff, L., Rinn, J. L., & Pachter, L. (2012). Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nature Biotechnology*, *31*(1), 46–53.

Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley

von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S., & Bork, P. (2002). Comparative assessment of large-scale data sets of protein?protein interactions. *Nature*, 399–403.

Woo, J., Shimoni, Y., Yang, W., Subramaniam, P., Iyer, A., Nicoletti, P., et al. (2015). Elucidating Compound Mechanism of Action by Network Perturbation Analysis. *Cell*, *162*(2), 441–451.

Woo, J., Shimoni, Y., Yang, W., Subramaniam, P., Iyer, A., Nicoletti, P.,et al. (2015). Elucidating Compound Mechanism of Action by Network Perturbation Analysis. *Cell*, *162*(2), 441–451.

Yang, J., Tang, H., Li, Y., Zhong, R., Wang, T., Wong, S., et al. (2015). DIGRE: Drug-Induced Genomic Residual Effect Model for Successful Prediction of Multidrug Effects. *CPT: Pharmacometrics & Systems Pharmacology*, *4*(2), 91–97.

Zuberi, K., Franz, M., Rodriguez, H., Montojo, J., Lopes, C. T., Bader, G. D., et al. (2013). GeneMANIA Prediction Server 2013 Update. *Nucleic Acids Research*, *41*(W1), 115–122.