DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER-BASED APPROACHES IN BIOSEQUENCE DATA ANALYSIS

by

Çağdaş KÜÇÜK

October, 2020

İZMİR

COMPUTER-BASED APPROACHES IN BIOSEQUENCE DATA ANALYSIS

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

by
Çağdaş KÜÇÜK

October, 2020
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"COMPUTER-BASED APPROACHES IN BIOSEQUENCE DATA ANALYSIS"** completed by **ÇAĞDAŞ KÜÇÜK** under supervision of **PROF. DR. ÇAĞIN KANDEMİR ÇAVAŞ** 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.

Prof. Dr. Çağın KANDEMİR ÇAVAŞ

Supervisor

Prof. Dr. Emel KURUOĞLU KANDEMİR

Jury Member

Assoc.Prof.Dr. Tarık KIŞLA

Jury Member

Prof. Dr. Özgür ÖZÇELİK

Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Prof. Dr. Çağın KANDEMİR ÇAVAŞ for her guidance and support for this study that led me to finish this study.

I would also like to thank my family and friends.

Çağdaş KÜÇÜK

COMPUTER-BASED APPROACHES IN BIOSEQUENCE DATA ANALYSIS

ABSTRACT

The structure and function of proteins are closely related. Signaling proteins take part in many biological activities like protein secretion, transportation and more. Signaling proteins are an important topic in drug discovery. For these reasons it is important to classify whether a protein is signaling or not. Machine learning solutions are a good option for this because experimental solutions are time consuming and expensive. The aim of this thesis is to classify signaling proteins using protein encoding schemes and machine learning algorithms.

1867 signaling and 3317 non-signaling proteins were downloaded. Then they were transformed using pseudo amino acid composition (PseAAC) and dipeptide composition to create two datasets. Deep neural network, random forest and support vector machine algorithms were used to classify signaling proteins.

For both datasets, the accuracy rate of all models were higher than 0.70. It shows that these models are effective in signal protein classification used with protein encoding schemes.

Keywords: Pseudo amino acid composition, dipeptide composition, support vector machine, neural network, random forest, signaling proteins

BİYOSEKANS VERİ ANALİZİNDE BİLGİSAYAR-TABANLI YAKLAŞIMLAR

ÖZ

Proteinlerin yapısı ve işlevi birbiriyle yakından ilgilidir. Sinyal proteinleri protein sekresyonu, taşınması gibi birçok biyolojik aktiviteye katılır. Sinyal proteinleri yeni ilaç bulunması alanında önemli bir konudur. Bu sebeplerden dolayı bir proteini sınıflandırıp sinyal proteini olup olmadığını bilmek önemlidir. Makine öğrenme çözümleri iyi bir seçenektir; çünkü deneysel çözümler pahalıdır ve zaman alırlar. Bu tezin amacı sinyal proteinlerini protein kodlama yöntemleri ve makine öğrenme algoritmaları kullanarak sınıflandırmaktır.

1867 sinyal ve 3317 sinyal olmayan protein indirildi. Sonra bu proteinler, psödo amino asit kompozisyonu (PseAAC) ve dipeptid kompozisyonu kullanılarak iki veri seti oluşturmak için dönüştürüldü. Derin yapay sinir ağı, rastgele orman ve destek vektör makinesi algoritmaları kullanılarak sinyal proteinleri sınıflandırıldı.

İki veri setinde de algoritmaların doğruluk oranları 0,70'den yüksek çıkmıştır ve bu modellerin protein kodlama yöntemleriyle kodlanan sinyal proteinlerinin sınıflandırılmasında etkin olduğunu göstermektedir.

Anahtar kelimeler: Psödo amino asit kompozisyonu, dipeptid kompozisyonu, destek vektör makinesi, yapay sinir ağı, rastgele orman, sinyal proteinleri

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER 1
INTRODUCTION

Signaling proteins take part in biological activities like protein transportation, secretion and many more (Nakai, 2000; Parker, 2001). For its importance in drug discovery, finding signal proteins is very crucial (Fernandez-Lozano, Cuiñas, Seoane, Fernandez-Blanco, Dorado, & Munteanu, 2015). Thus, it is advantageous to create new computational solutions. Machine learning is one of the best computational solutions. The objective of machine learning is to obtain low-cost solutions by exploiting the tolerance of imprecision, approximate reasoning and partial truth (Mitra & Hayashi, 2006). There are many studies that uses machine learning to classify protein structure in the literature like prediction of N-terminal protein sorting signals (Claros, Brunak, & von Heijne, 1997), prediction of regions prone to protein aggregation (Moreira, Philot, Lima, & Scott, 2019) and more (Jain, Garibaldi, & Hirst, 2009; Kathuria, Mehrotra, & Misra, 2018; Mitra & Hayashi, 2006; Najibi, Maadooliat, Zhou, Huang, & Gao, 2017).

Protein encoding scheme, aside from the machine learning algorithm used, is also important for good predictive results. Amino acid composition (AAC) is one of those, it is the percentage of each amino acid in a given protein sequence (Hormoz, 2013). There are many studies that use AAC to encode the proteins for topics like domain boundary prediction (Dumontier, Yao, Feldman, & Hogue, 2005), protein subcellular location prediction (Nasibov & Kandemir-Cavas, 2008), inter-domain linker prediction (Shatnawi & Zaki, 2015).

Another protein encoding scheme is pseudo-amino acid composition (PseAAC). It was proposed to add sequence information to amino acid composition by Chou (2001). Some of the studies that utilize PseAAC have topics like prediction of protein structural class (Chen, Zhou, Tian, Zou, & Cai, 2006), prediction of membrane protein types (Shen, Yang, & Chou, 2006), apoptosis protein prediction (Chen & Li, 2007), antifreeze protein prediction (Mondal & Pai, 2014), prediction of

β-Lactamase (Kumar, Srivastava, Kumari, & Kumar, 2015), prediction of protein submitochondrial locations (Qiu, et al., 2018).

Lastly, the other protein encoding scheme used in this thesis, dipeptide composition is another way for adding sequence information to amino acid composition. Some of the studies using dipeptide composition have topics like prediction of protein submitochondrial locations (Ahmad, Waris, & Hayat, 2016), classification of nuclear receptors (Bhasin & Raghava, 2004).

There are some specific signaling protein classification studies like classifying kinase conformations (McSkimming, Rasheed, & Kannan, 2017), prediction and prioritization of rare oncogenic mutations in the cancer kinome (ManChon, Talevich, Katiyar, Rasheed, & Kannan, 2014) but there seems to be only one general classification study (Fernandez-Lozano, Cuiñas, Seoane, Fernandez-Blanco, Dorado, & Munteanu, 2015). This general classification study uses star-graphs, but other methods like pseudo amino acid composition and dipeptide composition were not found in our review of the literature and we wanted to find out the effect of those methods. The purpose of this thesis is to classify signaling proteins using machine learning methods and with two protein encoding schemes; pseudo amino acid composition and dipeptide composition.

This thesis has five chapters. In the first chapter, the topic of this thesis is described and the literature is reviewed. In the second chapter, bioinformatics subjects used in the thesis are explained. In the third chapter, the machine learning algorithms and protein encoding schemes used in the thesis are explained. In the fourth chapter, how the methods that are given details in the previous chapter are used to classify signaling proteins is explained and the results are shown. In the final fifth chapter, the thesis is summarized and concluded.

CHAPTER 2

BIOINFORMATICS SUBJECTS

Bioinformatics is an interdisciplinary field that works on understanding biological data like genes and proteins using and developing software solutions. The amount of protein and gene sequences increase each year and to deal with that large amount of data software solutions are needed as it is faster. It is used in areas like proteomics, genomics, drug discovery and more. (Higgs & Attwood, 2005).

In this part, bioinformatics subjects used in this thesis are explained.

2.1 Protein

Proteins can do many functions, like enzymatic catalysis, transporting ions and molecules from one organ to another, nutrients, contractile system of muscles, tendons, cartilage, antibodies, and regulating cellular and physiological activities (Gromiha, 2010).

Proteins are made up of amino acids. There are 20 types of amino acids. Their names, one letter codes and three letter symbols are in Table 2.1 below.

Table 2.1 Amino acid names, their 1 letter and 3 letter symbols (Lide, 2005)

Amino Acid Name	1 Letter Symbol	3 Letter Symbol
Alanine	A	Ala
Cysteine	C	Cys
Aspartic acid	D	Asp
Glutamic acid	E	Glu
Phenylalanine	F	Phe
Glycine	G	Gly
Histidine	H	His
Isoleucine	I	Ile
Lysine	K	Lys
Leucine	L	Leu
Methionine	M	Met
Asparagine	N	Asn
Proline	P	Pro
Glutamine	Q	Gln
Arginine	R	Arg
Serine	S	Ser
Threonine	T	Thr
Valine	V	Trp
Tryptophan	W	Tyr
Tyrosine	Y	Val

Side chain is the difference between amino acids. Amino acids have a carbon atom in the middle and that atom has four connections; a hydrogen atom, an amino group (NH_2), a carboxyl group (COOH) and the last one is the side chain (Gromiha, 2010). A few amino acids and their representations are in Figure 2.1.

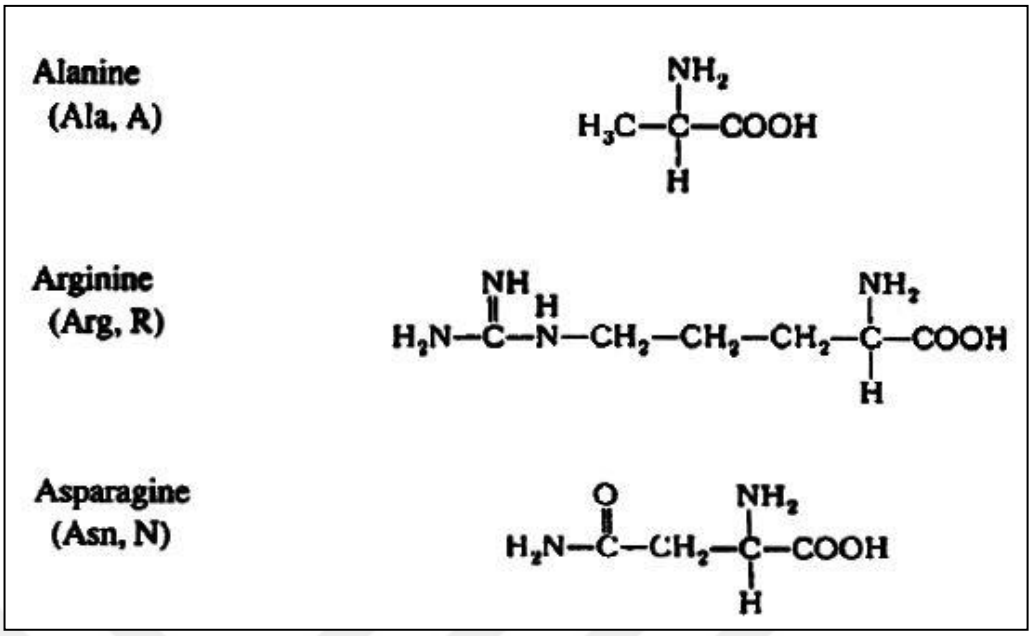

Figure 2.1 Some amino acids and their representations (Lide, 2005)

Proteins have four levels of structures: primary, secondary, tertiary, and quaternary (Gromiha, 2010). They are shown in Figure 2.2 below.

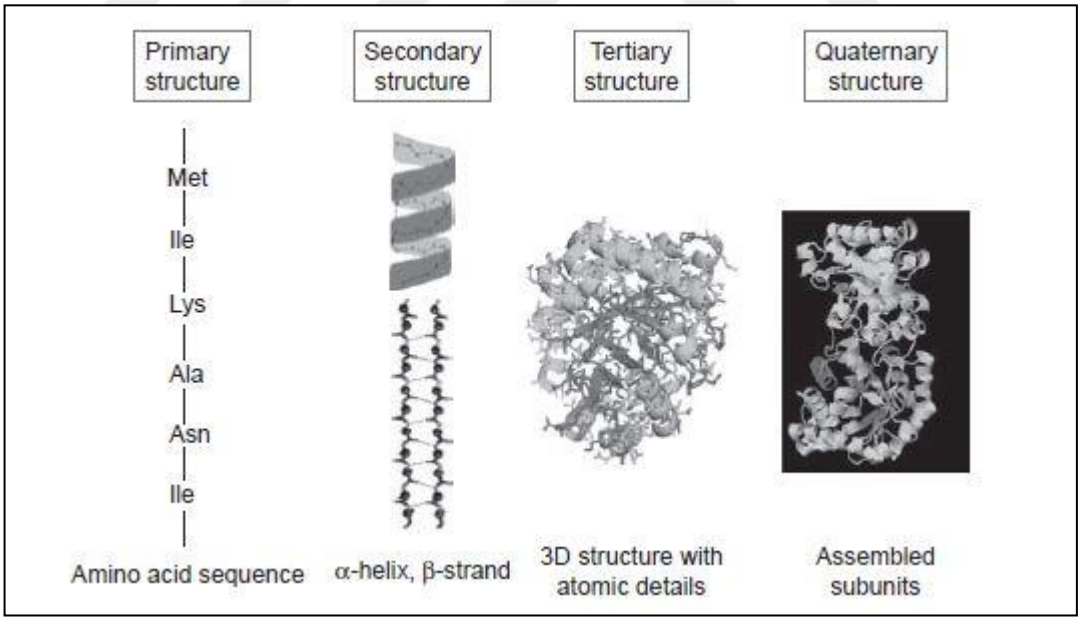

Figure 2.2 Four level of protein structures (Gromiha, 2010)

2.2 Fasta Format

A FASTA format contains amino acid sequences of proteins. Each amino acid is denoted by their 1-letter symbol. First line starts with a ">" symbol and the descriptions of the proteins. After that comes the sequence of amino acids. A FASTA sequence example is in Figure 2.3.

Figure 2.3 A FASTA format protein sequence example

2.3 Protein Databases

To store large amounts of biological data, there many databases. For proteins, there are sequence databases and structure databases. Structure databases hold 3D protein structures obtained with techniques like X-ray crystallography, NMR spectroscopy, neutron diffraction, electron microscopy (Gromiha, 2010).

UniProt is a protein sequence database. It is the combination of three databases; Protein Information Resource (PIR), SWISS-PROT and TrEMBL. It has three components; UniProt Knowledgebase (UniProt) that holds protein sequences and information, UniProt Non-redundant Reference (UniRef) that combines associated protein together to fasten the search process and UniProt Archive (UniParc) that holds sequence history for proteins (Higgs & Attwood, 2005).UniProt database site is shown in Figure 2.4 below.

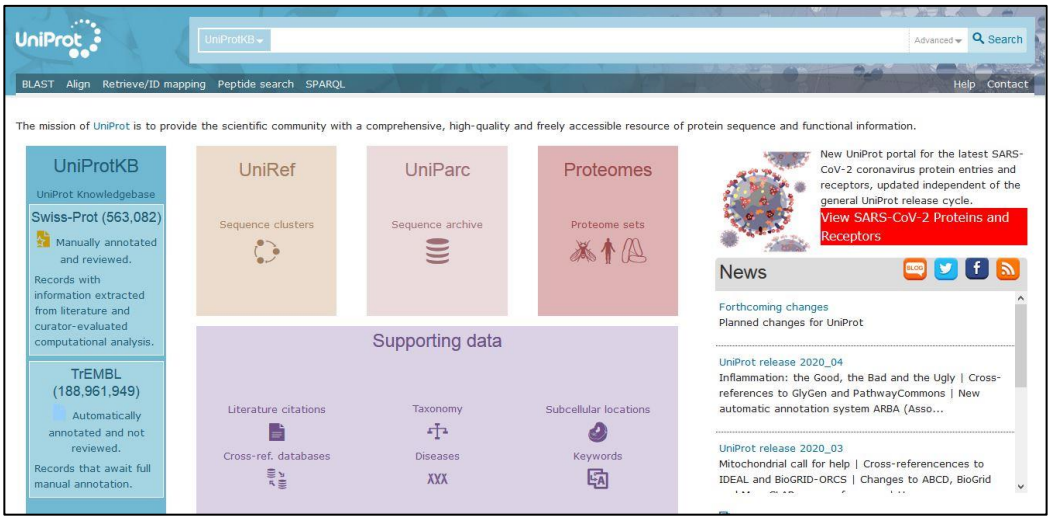

Figure 2.4 UniProt website (The UniProt Consortium, 2018)

On the UniProt site, there is a search bar and from there proteins and genes can be found with advanced search options like function, subcellular location, structure, pathology and many more. In Figure 2.5, some advanced search options are shown.

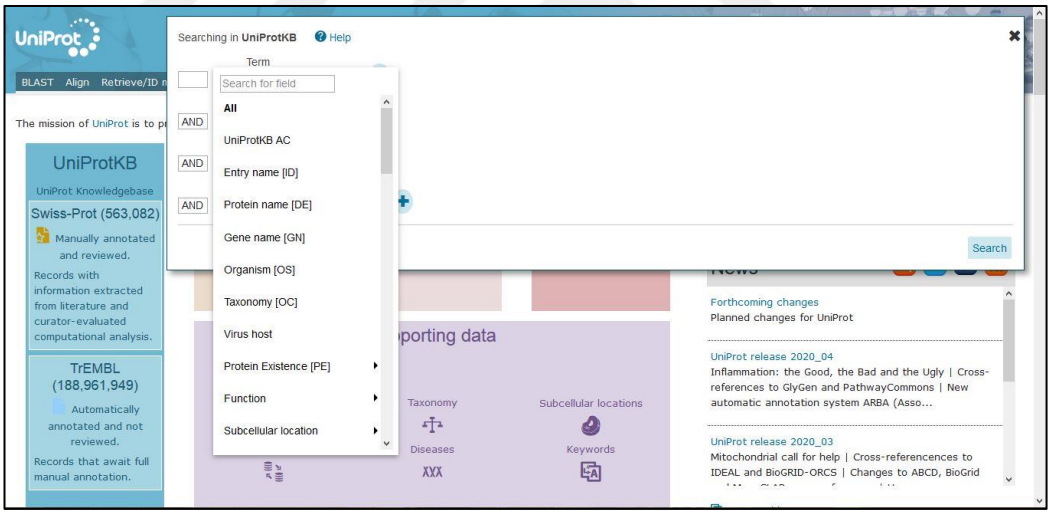

Figure 2.5 UniProt advanced search demonstration (The UniProt Consortium, 2018)

Protein Data Bank (PDB) is a protein structure database. It holds information like atomic coordinates, partial bond connectivity, temperature factor, how the structure was obtained, crystallographic conditions and more (Gromiha, 2010).Advanced search on PDB site is shown in Figure 2.6.

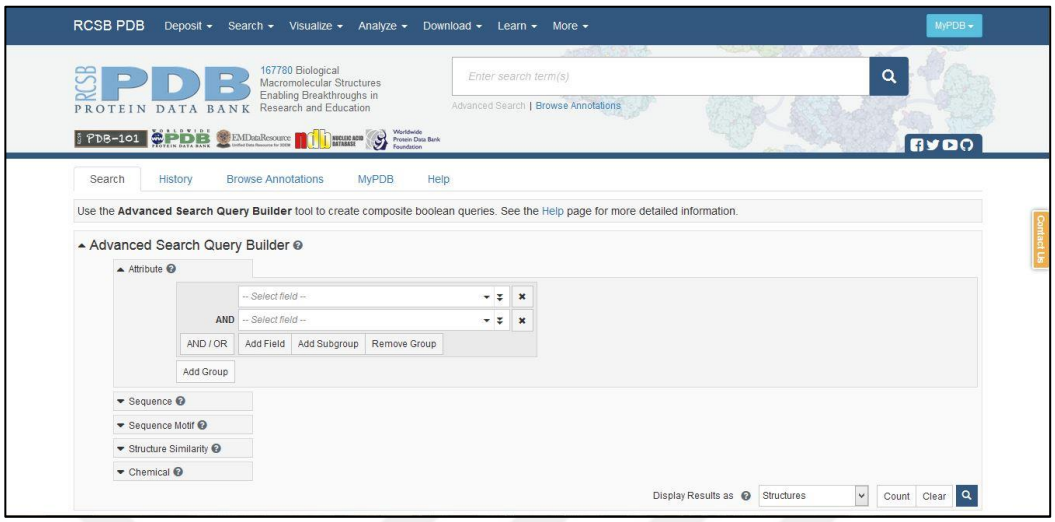

Figure 2.6 PDB advanced search demonstration (Berman, et al., 2000)

PISCES and ASTRAL are nonredundant protein structure databases. Proteins with similar structures affects studies so nonredundant databases are necessary. They have options like cutoff value, sequence identity, number of amino acid residues, resolution, R-factor and more (Gromiha, 2010).On the PISCES site, premade lists with specific conditions can be downloaded. An example would be cullpdb_pc25_res1.6_R0.25_d200820_chains4782.gz file where pc25 represents percentage identity cutoff 25%, res1.6 represents resolution 1.6, R0.25 represents R-factor cutoff 0.25, d200820 represents August 20 2020 and chains 4782 represents the number of chains. The page showing premade lists on the PISCES site is in Figure 2.7 below.

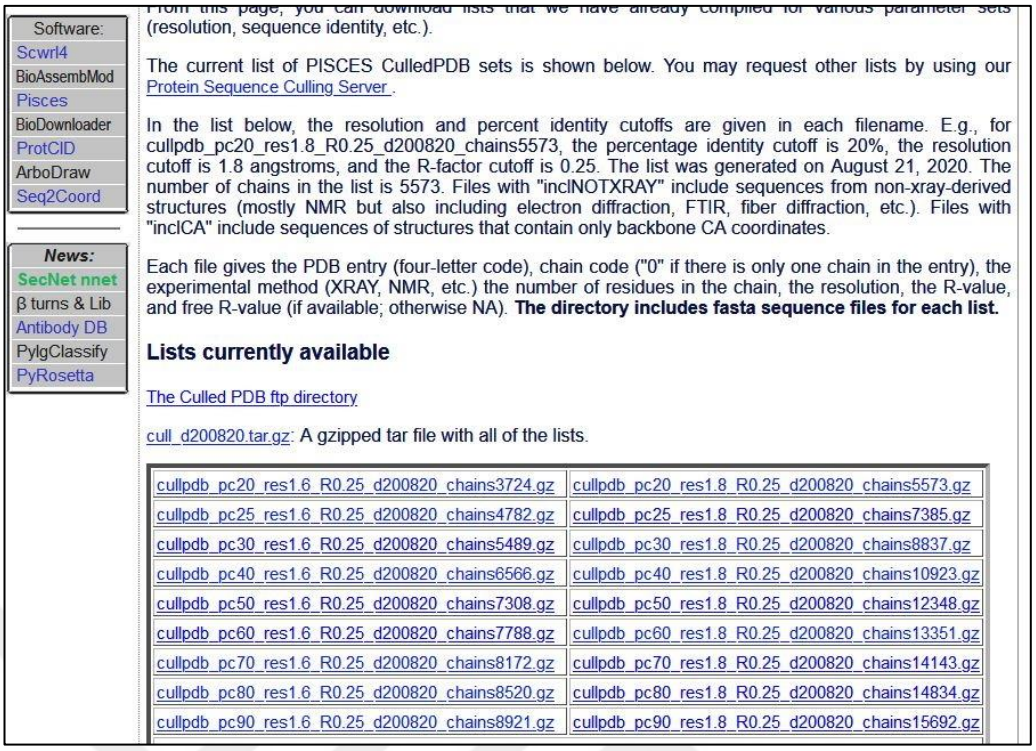

Figure 2.7 PISCES premade lists page (Wang & Dunbrack Jr, 2003)

It is also possible to download custom lists of proteins. Page for this task on the PISCES site is shown in Figure 2.8.

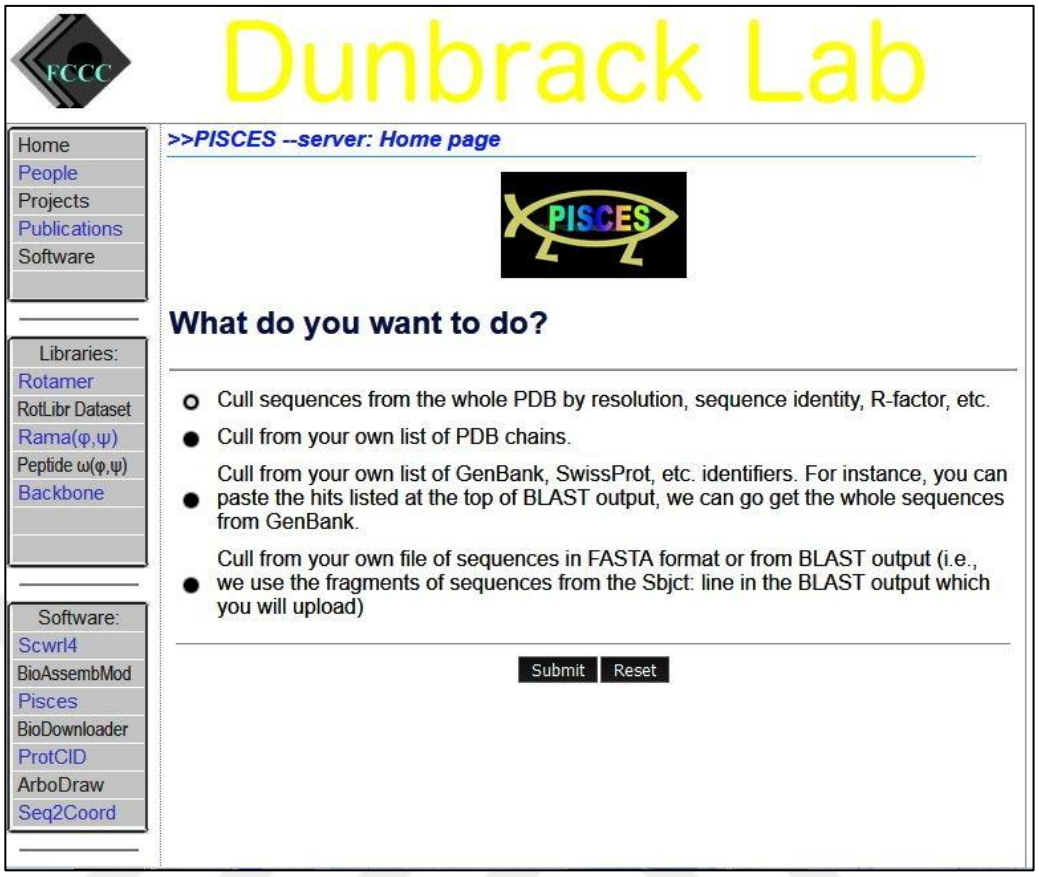

Figure 2.8 PISCES custom list generation page (Wang & Dunbrack Jr, 2003)

2.4 Cell Signaling

Cells interact with each other through signals to start or end biological processes. There are four types of cell signaling: through direct contact, paracrine (short distance), endocrine (long distance) and synaptic (neurons). A cell can also send a signal to itself and it is called autocrine signaling. For endocrine signaling, molecules called hormones are used which can travel long distances to deliver the signal (Johnson & Raven, 2002). Types of cell signaling can be seen in Figure 2.9.

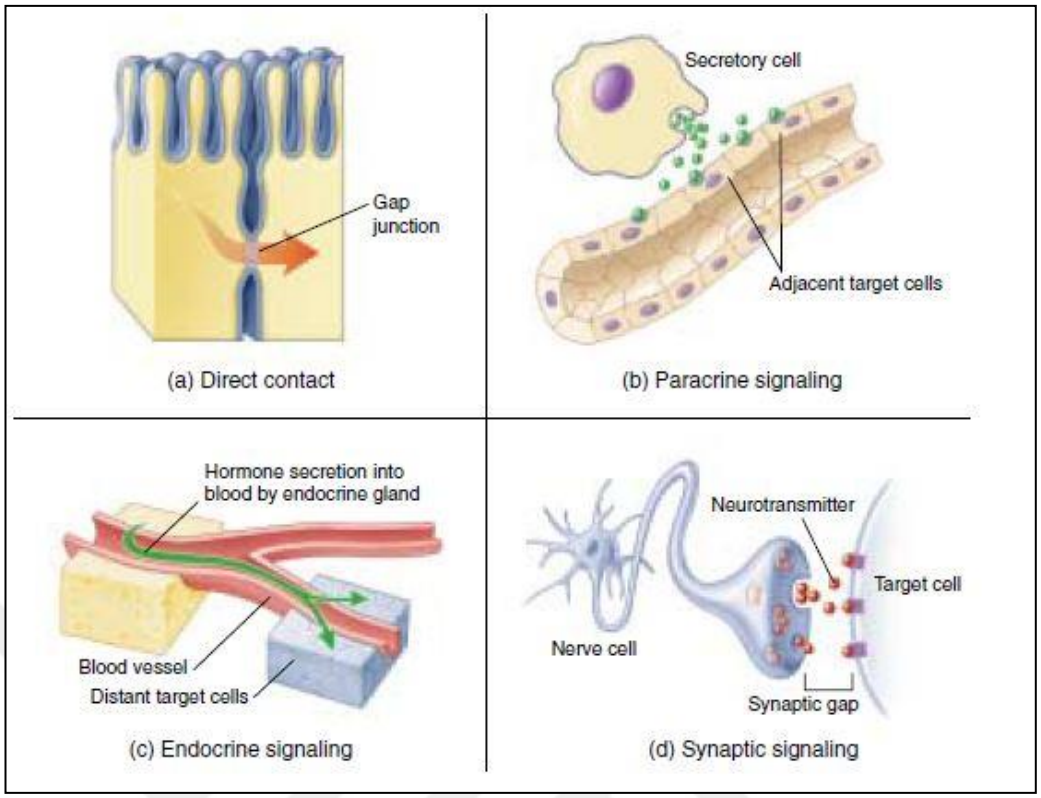

Figure 2.9 Types of cell signaling (Johnson & Raven, 2002)

Cells have receptors that capture the signal and start the signal process. Receptors can be inside of the cell (intracellular) or on the membrane of the cell. Intracellular receptors are for smaller molecules that can pass the membrane of the cell. Receptors that are on the surface on the cell are for larger molecules like hormones that are too big to pass through the membrane. There are 3 types of receptors that are on the surface of the cell: G-protein linked receptors, enzymic receptors and chemically gated ion-channels. These receptors capture the outer signal and their parts in the cell convert it to an internal one (Johnson & Raven, 2002). Receptors are shown in Figure 2.10 below.

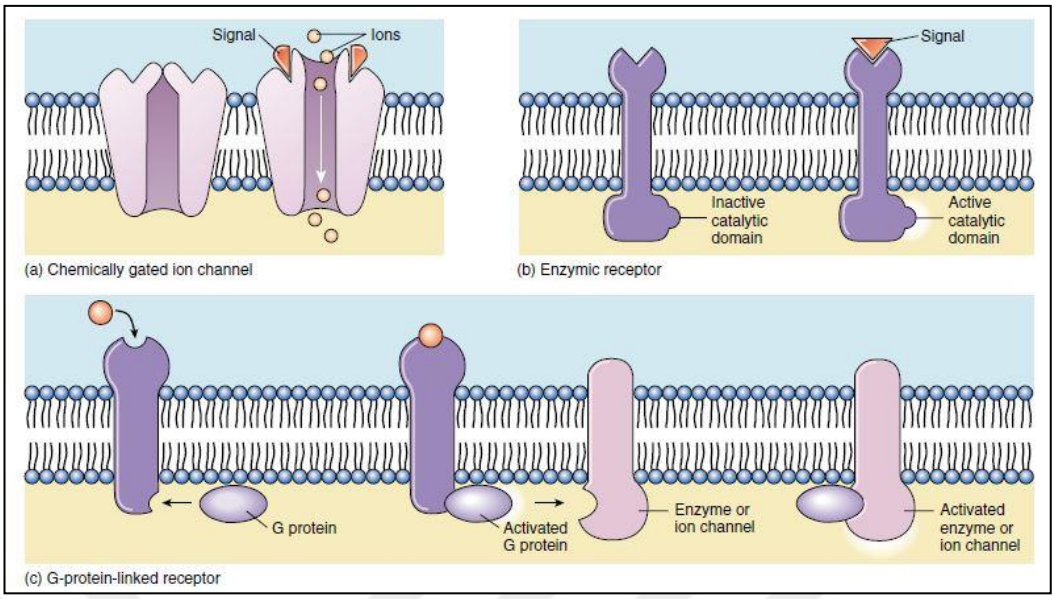

Figure 2.10 Types of receptors (Johnson & Raven, 2002)

Some receptors use smaller molecules called second messenger to send the signal inside the cell. Cyclic adenosine monophosphate (cAMP) and calcium are examples of second messengers (Johnson & Raven, 2002). They can be seen in Figure 2.11.

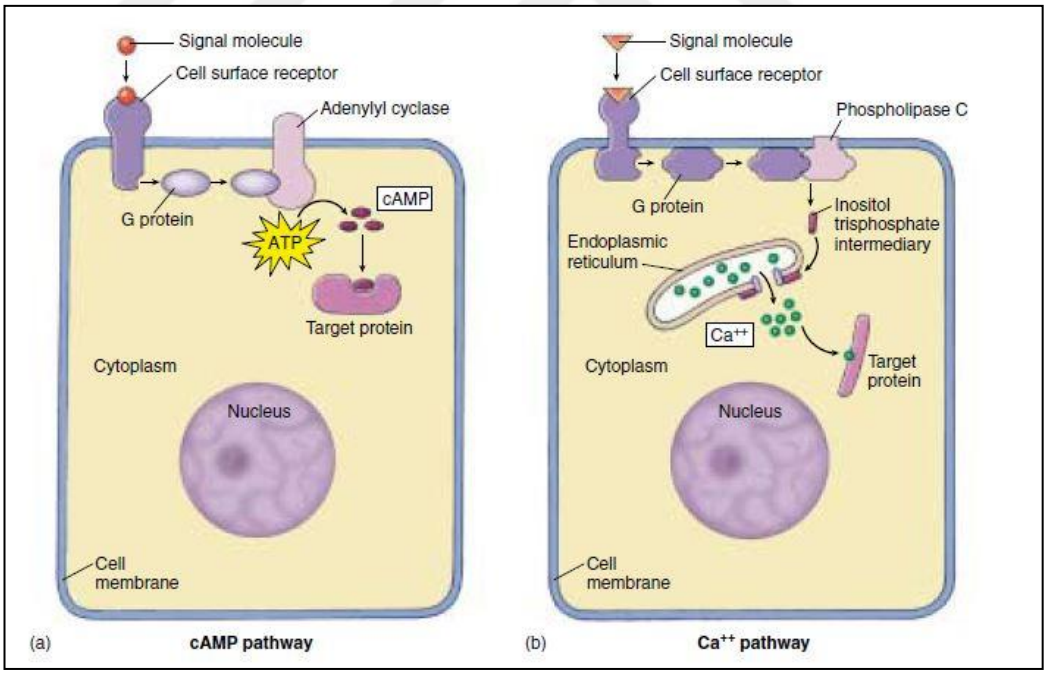

Figure 2.11 Second messenger examples (Johnson & Raven, 2002)

CHAPTER 3

MATERIALS AND METHODS

Machine learning is an approach that uses data to solve problems. An example would be handwritten digit recognition. To write a program to recognize digits, one would need to find rules to differentiate them. Machine learning solution would use handwritten digits themselves to learn and recognize the digits instead (Kim, 2017). Machine learning has three types, supervised, unsupervised and reinforcement learning.

In supervised learning, training dataset has inputs and outputs called label. The algorithm learns from training set examples and predicts new data labels. If the labels are discrete variables (categorical) then it is called classification. If the labels are continuous variables (numerical) then it is called regression (Bishop, 2006).

In unsupervised learning, training dataset has only inputs with no corresponding outputs. A type of unsupervised learning is clustering where the algorithm groups the inputs based on their similarities (Bishop, 2006).

In reinforcement learning, finding the optimal steps to reach the maximum reward is the goal. Rather than training sets, the problems is solved by trial and error. Steps are usually connected and each step affects the next steps.

Some of the classification algorithms are listed below:

- Support Vector Machine (SVM)
- Artificial Neural Network (ANN)
- K-nearest Neighbor
- Decision Trees
- Naive Bayes

Machine learning is important in many areas and has many applications. An example of it in bioinformatics can be integration of a deep learning classifier with a random forest approach for predicting malonylation sites study. In this study, they built a long short-term memory with word embedding classifier and a random forest classifier with enhanced amino acid composition and then used them together to predict malonylation sites. For random forest classifier, they choose 1000 trees and $\sqrt{N}$ variables at each split where N is the dimension of the input feature vector (Zhen, Ningning, Yu, Wen, Xuhan, & Lei, 2018). A scheme of their process can be seen in Figure 3.1 below.

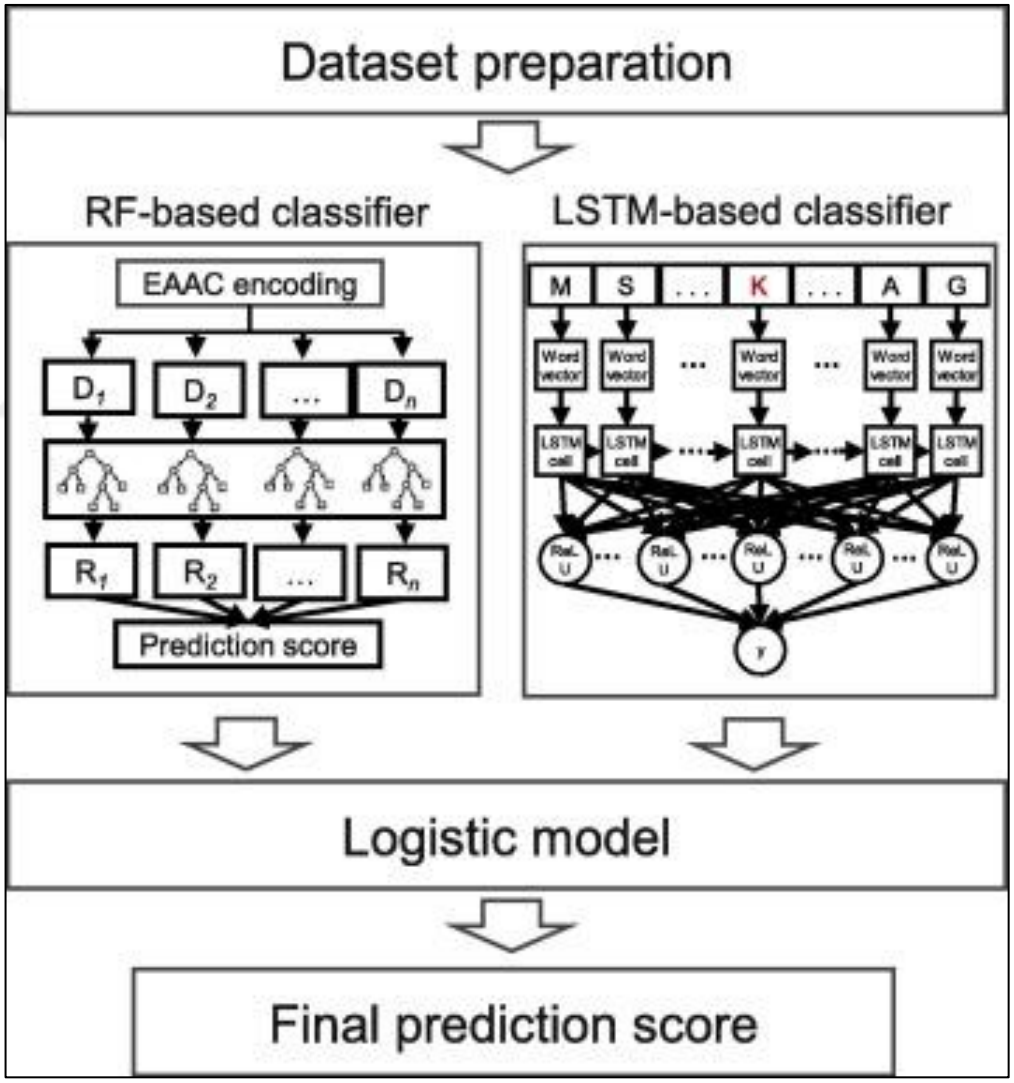

Figure 3.1 Scheme of integration of a deep learning classifier with a random forest approach for predicting malonylation study (Zhen, Ningning, Yu, Wen, Xuhan, & Lei, 2018)

Another example study is hybrid method based on information gain and support vector machine for gene selection in cancer classification. They used information gain first to remove unrelated genes and then a support vector machine was used to further the removal process. Finally another support vector machine was used from the remaining genes to predict cancer. (Lingyun, Mingquan, Xiaojie, & Daobin, 2017). Information gain has the following formula in the eq. 3.1.

$$g(Y,X) = H(Y) - H(Y|X) \tag{3.1}$$

$H(Y)$ is the entropy of the dataset Y, $H(Y|X)$ is the conditional entropy based on the known variable X and p in the following formulas eq. 3.2 and eq. 3.3 is probability distribution.

$$H(Y) = -\sum p(y)\log p(y) \tag{3.2}$$

$$H(Y|X) = \sum_{x \in X} p(x)\, H(Y|X = x) \tag{3.3}$$

From the result of information gain values, the higher ranking ones are chosen. A scheme of their process is shown in Figure 3.2.

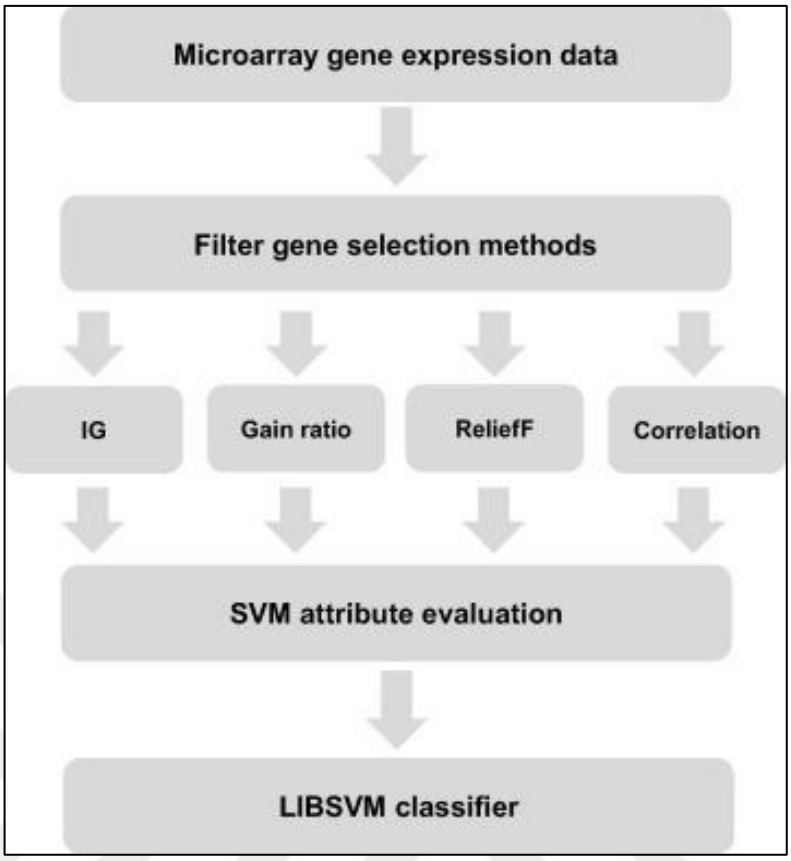

Figure 3.2 Scheme of hybrid method based on information gain and support vector machine for gene selection in cancer classification (Lingyun, Mingquan, Xiaojie, & Daobin, 2017)

Final example study is predicting the subcellular localization of human proteins using machine learning and exploratory data analysis. They used naive bayes, multilayer perceptron, support vector machine and J48 decision tree models for classification (George, Sonia, & Chittibabu, 2006). Decision tree from the study is in the Figure 3.3 below.

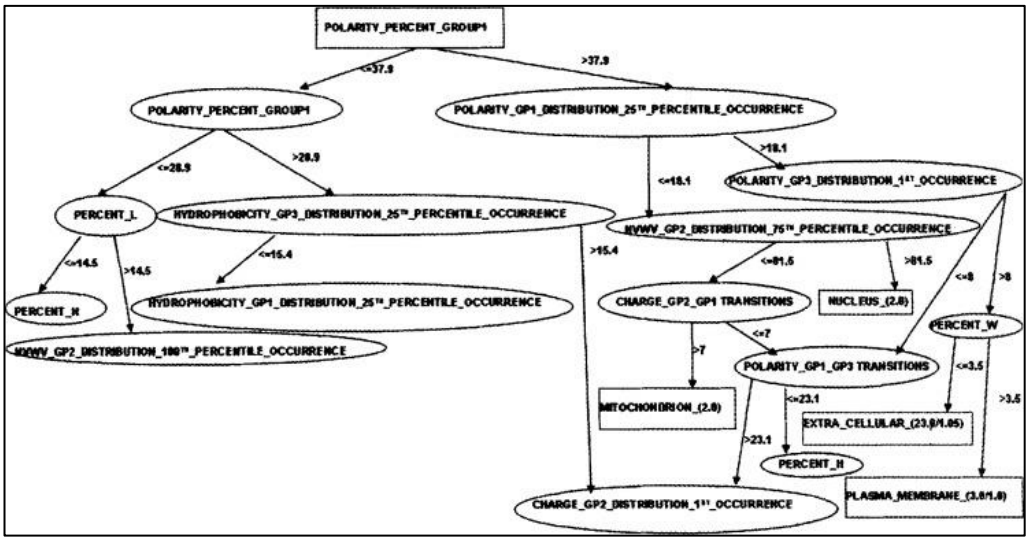

Figure 3.3 Decision tree from predicting the subcellular localization of human proteins using machine learning and exploratory data analysis (George, Sonia, & Chittibabu, 2006)

These are just some of the machine learning applications in bioinformatics.

In this part, machine learning methods and protein encoding schemes using in this thesis are explained.

3.1 Amino Acid Composition

Amino acid composition is calculated by counting each type of amino acid in a protein sequence and then normalizing the values (Bhasin & Raghava, 2004). The formula is in the eq. 3.4.

$$Comp(i) = \sum n_i * 100/N \tag{3.4}$$

i is for the current amino acid, n_i is for the number of i in the sequence while N is the total number of amino acids in the sequence.

Amino acid composition is used to transform the sequence so that it can be used in machine learning algorithms. The resulting dataset becomes a 20 attribute dataset with each column holding the frequency of that type of amino acid in the sequence.

3.2 Pseudo Amino Acid Composition

In the previous section, amino acid composition was explained. Amino acid composition has only the frequencies of amino acids as attributes but not the sequence in which they are connected. Chou (2001), to add sequence information to the amino acid composition, proposed a method called pseudo amino acid composition (PseAAC). The formula is in the eq. 3.5 below where λ is the value that decides how many extra attributes are added and w is the weight factor which decides how much it is going to affect the x_u.

$$x_u = \begin{cases} \dfrac{f_u}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{\lambda}\theta_j}, (1 \le u \le 20) \\ \dfrac{w\theta_{u-20}}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{\lambda}\theta_j}, (20+1 \le u \le 20+\lambda) \end{cases} \tag{3.5}$$

The i-th tier correlation factor θ_i has the sequence order correlations of all the amino acids in the protein sequence and is calculated in eq. 3.6 where L is the number of amino acids in the protein chain and R_i is the i-th amino acid in a sequence like $R_1, R_2, \dots, R_L$.

$$\theta_\lambda = \frac{1}{L-\lambda}\sum_{\mathrm{i}=1}^{\mathrm{L}-\lambda} \Theta(\mathrm{R_i}, \mathrm{R_{i+\lambda}})(\lambda < L) \tag{3.6}$$

$\Theta(\mathrm{R_i}, \mathrm{R_j})$is the correlation function which is calculated using the standardized hydrophobicity ($\mathrm{H_1}$), hydrophilicity ($\mathrm{H_2}$) andside-mass chain (M) values in eq. 3.7**.**

$$\begin{aligned} \Theta(\mathrm{R_i}, \mathrm{R_j}) = \frac{1}{3}\{&\left[\mathrm{H_1}(\mathrm{R_j}) - \mathrm{H_1}(\mathrm{R_i})\right]^2 + \left[\mathrm{H_2}(\mathrm{R_j}) - \mathrm{H_2}(\mathrm{R_i})\right]^2 \\ &+ \left[\mathrm{M}(\mathrm{R_j}) - \mathrm{M}(\mathrm{R_i})\right]^2\} \end{aligned} \tag{3.7}$$

Names, 1-letter symbols, hydrophobicity ($\mathrm{H_1}$), hydrophilicity ($\mathrm{H_2}$) and side-mass chain (M) values for the amino acids are given in Table 3.1 (Eisenberg, 1984; Hopp & Woods, 1981; Lide, 2005).

Table 3.1 Hydrophobicity, hydrophilicity and side-chain mass values

Amino Acid Names	1 Letter Symbols	Hydrophobicity(H_1)	Hydrophilicity (H_2)	Side-Chain Mass (M)
Alanine	A	0.62	-0.5	15.0
Cysteine	C	0.29	-1.0	47.0
Aspartic acid	D	-0.90	3.0	59.0
Glutamic acid	E	-0.74	3.0	73.0
Phenylalanine	F	1.19	-2.5	91.0
Glycine	G	0.48	0.0	1.0
Histidine	H	-0.40	-0.5	82.0
Isoleucine	I	1.38	-1.8	57.0
Lysine	K	-1.50	3.0	73.0
Leucine	L	1.06	-1.8	57.0
Methionine	M	0.64	-1.3	75.0
Asparagine	N	-0.78	0.2	58.0
Proline	P	0.12	0.0	42.0
Glutamine	Q	-0.85	0.2	72.0
Arginine	R	-2.53	3.0	101.0
Serine	S	-0.18	0.3	31.0
Threonine	T	-0.05	-0.4	45.0
Valine	V	1.08	-1.5	43.0
Tryptophan	W	0.81	-3.4	130.0
Tyrosine	Y	0.26	-2.3	107.0

H_1, H_2 and M values need to be standardized using the formulas in eq. 3.8 where H_1^0, H_2^0 and M^0 are the original values before they can be used in the PseAAC.

$$\begin{cases} H_1(i) = \dfrac{H_1^0(i) - \sum_{i=1}^{20} \frac{H_1^0(i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20}\left[H_1^0(i) - \sum_{i=1}^{20} \frac{H_1^0(i)}{20}\right]^2}{20}}} \\ H_2(i) = \dfrac{H_2^0(i) - \sum_{i=1}^{20} \frac{H_2^0(i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20}\left[H_2^0(i) - \sum_{i=1}^{20} \frac{H_2^0(i)}{20}\right]^2}{20}}} \\ M(i) = \dfrac{M^0(i) - \sum_{i=1}^{20} \frac{M^0(i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20}\left[M^0(i) - \sum_{i=1}^{20} \frac{M^0(i)}{20}\right]^2}{20}}} \end{cases} \quad (3.8)$$

3.3 Dipeptide Composition

Dipeptide composition is another way to transform protein sequences. In this method, the number dipeptides are calculated and divided by the total possible number of dipeptides in the sequence (Bhasin & Raghava, 2004). The formula is in the eq. 3.9 below.

$$fraction\ of\ dep(i) = \frac{total\ number\ of\ dep(i)}{total\ number\ of\ all\ possible\ dipeptides} \quad (3.9)$$

First 20 attributes are the same as other methods, frequencies of amino acids. After that there are 400 more attributes which are all possible dipeptide combinations as there are 20 types of amino acids. It goes like this; AA, AC, ..., AY, BA, ..., YY.

3.4 Random Forest

Random forest is an ensemble algorithm created by Breiman (2001). It is made up of several decision trees.

Random forest algorithm trains like this. A specified number of trees are created and for each tree a bootstrap sample of the dataset is created which means sampling with replacement. For each split, a specified number of attributes are chosen at random from all attributes. The best one of those attributes is selected for splitting (Hastie, Tibshirani, & Friedman, 2009).

After training, the prediction happens by each tree giving a result. Those results are treated as votes and all the votes are aggregated and the class with the highest vote is chosen as predicted (Hastie, Tibshirani, & Friedman, 2009).

3.5 Support Vector Machine

Support vector machine (SVM) is a machine learning algorithm that uses maximum margin to fit the data (Aggarwal, 2014). There are infinite hyperplanes that separate the classes of data as shown in Figure 3.4.

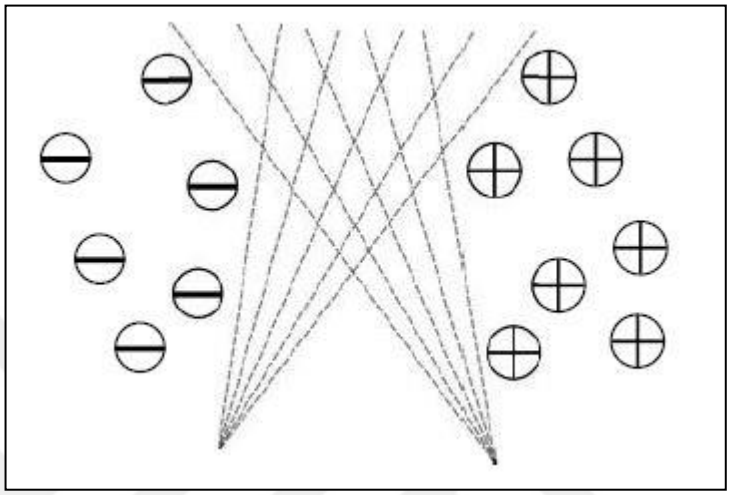

Figure 3.4 Infinite hyperplanes that seperate two classes of data (Aggarwal, 2014)

The distance between the points from each class that are the closest to the hyperplane is called the margin. As stated above there are infinite possibilities of hyperplanes, the one with the maximum margin is the one the algorithm seeks. The closest data points are called support vectors (Aggarwal, 2014).

Some data points might be on the wrong side of the hyperplane. To choose how much of that error is acceptable, there is a hyper-parameter c. When c is higher the acceptance of error is lower.

SVM assumes that the data is linearly separable but it is not always the case in real world data. To fix this, there is a solution called kernel trick. Kernel trick is changing the dimension of the data using a kernel function so that the data becomes linearly separable. Polynomial function, sigmoid function, radial basis function are some examples of kernel functions.

3.6 Artificial Neural Network

Artificial neural network (ANN) is a machine learning algorithm that tries to mimic the neurons in the human brain. It has many components that needs explaining but before that we need to look at the biological neuron first. It is shown in the Figure 3.5 below.

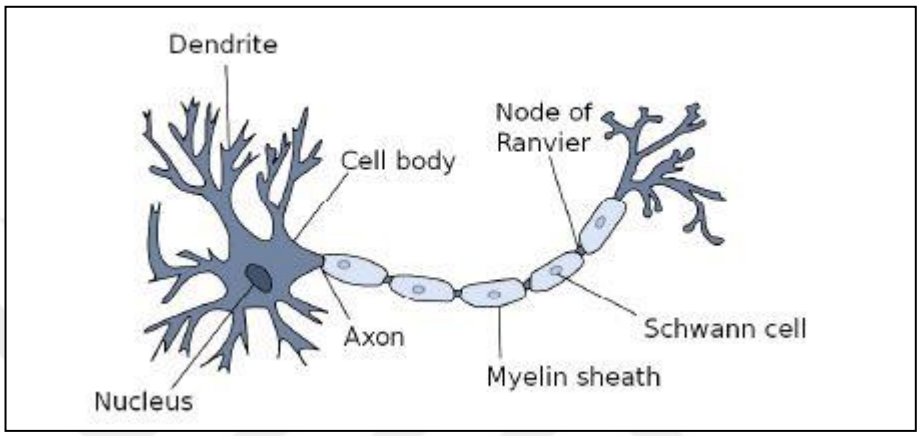

Figure 3.5 A biological neuron (Kriesel, 2007)

A neuron has dendrites that receive information from other neurons through connection in between the neurons called synapses. It goes to the nucleus of the neuron, processed and then goes through the axon to the next neuron (Kriesel, 2007).

An artificial neural network has layers, weights and nodes (neurons). An artificial neuron example is in Figure 3.6 below.

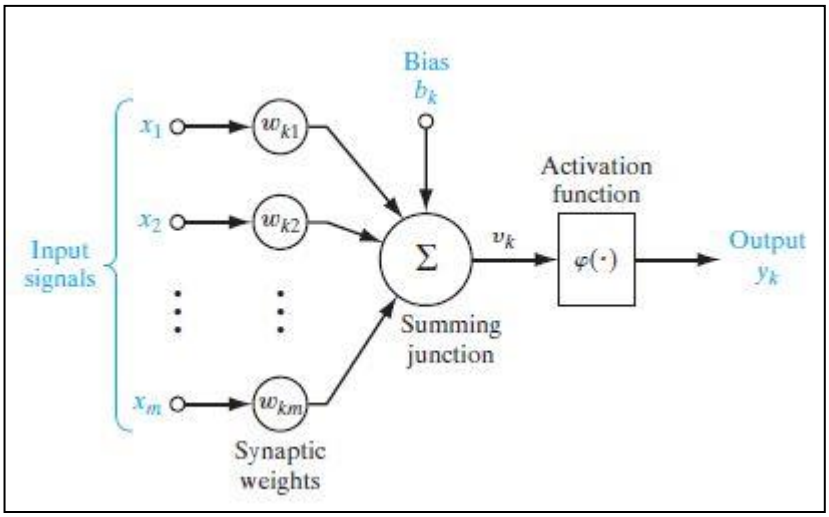

Figure 3.6 An artificial neuron (Haykin, 2009)

A neuron has connections from inputs or other neurons called weights, a summing junction, an activation function and it generates an output. Inputs are multiplied and then added together like in eq. 3.10.

$$u_k = \sum_{i=1}^{m} w_{ki} x_i \tag{3.10}$$

After that the summed result and bias b_k is added together if there is any bias. It is in the eq. 3.11 as follow,

$$v_k = u_k + b_k \tag{3.11}$$

That result then goes through the activation function which is in the eq. 3.12 below.

$$y_k = \varphi(v_k) \tag{3.12}$$

y_k is the output of the neuron that obtained from the activation function.

There are many activation functions. Some of the are threshold, sigmoid (logistic) and hyperbolic tangent function (Haykin, 2009).

Threshold function is in the eq. 3.13 below.

$$\begin{cases} 1 \; if \; v_k \geq 0 \\ 0 \; if \; v_k < 0 \end{cases} \tag{3.13}$$

Sigmoid function is shown in eq. 3.14.

$$\frac{1}{1 + e^{-v_k}} \tag{3.14}$$

Hyperbolic tangent function is in eq. 3.15.

$$\tan v_k \tag{3.15}$$

There is also the rectified linear unit (ReLU) function which is currently used in deep learning models (Goodfellow, Bengio, & Courville, 2016). It is shown in eq. 3.16.

$$\max (0, v_k) \tag{3.16}$$

There are many types of neural networks. Some of them are multilayer perceptron, learning vector quantization, Hopfield network, recurrent network, self organizing map and radial basis function network (Kriesel, 2007). In this thesis a multilayer perceptron neural network is used and in the next section it will be explained.

3.6.1 Multilayer Perceptron

A multilayer perceptron has one or more hidden layers, each neuron has a nonlinear activation function and it is fully connected. An example of a multilayer perceptron is in Figure 3.7 below.

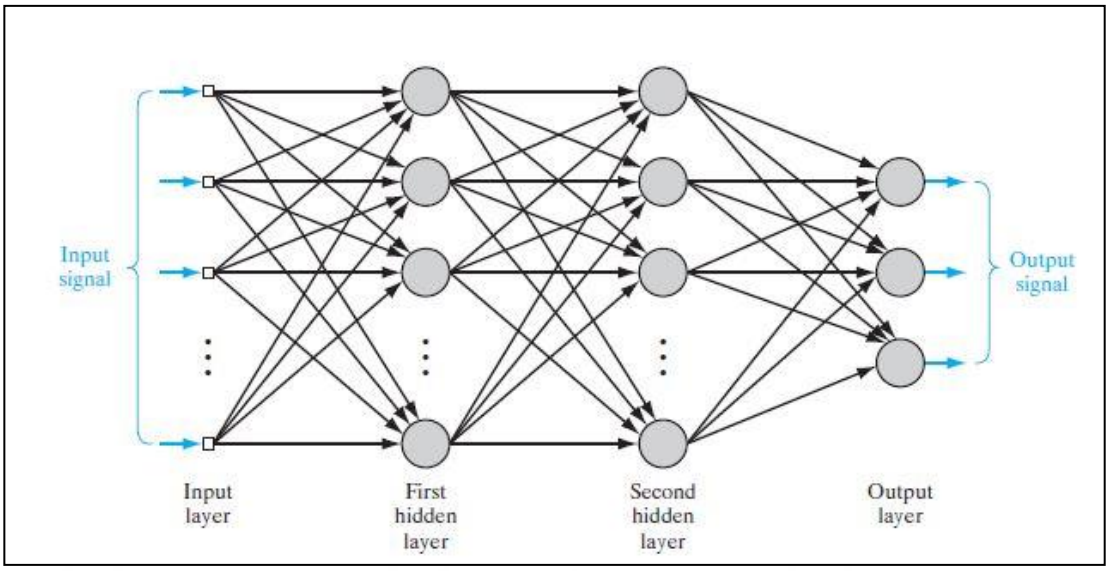

Figure 3.7 A multilayer perceptron (Haykin, 2009)

The training of these networks is two parts, forward phase and a backward phase.

In the forward phase, the inputs go through the weights and neurons of each layer all the way to the output layer and one or more output is acquired. The weights does not change in this phase (Haykin, 2009).

In the backward phase, the error from those outputs are calculated with a loss function and weights are changed with backpropagation (Rumelhart, Hinton, & Williams, 1986). Forward and backward phases happening once is called an iteration and these happening on the whole training set is called an epoch. Error adjustment can happen when a single line of input passes through the network (online training) or a group of inputs passes (batch training).

The error of neuron is calculated in eq. 3.17.

$$e_j(n) = d_j(n) - y_j(n) \tag{3.17}$$

$e_j(n)$ is the error of the j-th node of the output layer, $d_j(n)$ is the expected output and $y_j(n)$ is the actual output of the neuron. The general error of the network is calculated in the eq. 3.18 below.

$$\mathrm{E}(n) = \frac{1}{2}\sum_{j \in C} e_j^2(n) \tag{3.18}$$

C is the set of all of the neurons in the output layer. To find the correction of weight $\Delta w_{ij}(n)$ and apply it to $w_{ij}(n)$, $\frac{\partial \mathrm{E}(n)}{\partial w_{ij}(n)}$ needs to be found first. Using chain rule, eq. 3.19 is acquired.

$$\frac{\partial \mathrm{E}(n)}{\partial w_{ij}(n)} = \frac{\partial \mathrm{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ij}(n)} \tag{3.19}$$

By calculating and putting everything in the equation above we get eq. 3.20.

$$\frac{\partial \mathrm{E}(n)}{\partial w_{ij}(n)} = -e_j(n)\varphi'\Big(v_j(n)\Big) y_i(n) \tag{3.20}$$

By putting the equation above into the eq. 3.21 below we get $\Delta w_{ij}(n)$.

$$\Delta w_{ij}(n) = -\eta \frac{\partial \mathrm{E}(n)}{\partial w_{ij}(n)} \tag{3.21}$$

η is the learning rate of the optimizer. Changing the η affects how fast or slow the learning happens. By putting $\frac{\partial \mathrm{E}(n)}{\partial w_{ij}(n)}$ in we get the following eq. 3.22.

$$\Delta w_{ij}(n) = \eta \delta_j(n) y_i(n) \tag{3.22}$$

$\delta_j(n)$ is the local gradient. It is in eq. 3.23.

$$\delta_j(n) = e_j(n)\varphi'\Big(v_j(n)\Big) \tag{3.23}$$

Training is finished when a stopping criteria is reached or the specified number of epochs are reached (Haykin, 2009). To control the learning process, optimizers like stochastic gradient descent, RMSProp, Adam, AdaGrad are used (Goodfellow, Bengio, & Courville, 2016).

Neural networks have existed for a while and evolved with time. Currently they are called deep neural networks. With activation functions like ReLU, the vanishing gradient problem can be solved and with this, the network can have many layers but more layers can cause overfitting. To overcome this problem, methods like dropout can be used. Dropout method, uses a probability to whether drop the output of a node or not (Goodfellow, Bengio, & Courville, 2016).

CHAPTER 4

APPLICATION

4.1 Dataset

Gathering of signaling and non-signaling proteins is like in the work of Fernandez-Lozano et al (2015). To download signaling proteins, steps required are like the following:

- From PDB, choose "Advanced Search"
- then "Biological Process Browser"
- then choose "signaling (GO ID:23052)" as a criterion
- after that select "Protein" and "Representative Structures 30%"

1867 signaling proteins were downloaded from PDB (Berman, et al., 2000) by following those steps.

To download signaling proteins, steps required are like the following:

- From PISCES CulledPDB, choose "percent identity cutoff" less than 20%, resolution 1.6 Â and R-factor 0.25

3404 proteins were downloaded from PISCES CulledPDB (Wang & Dunbrack Jr, 2003), and were checked to remove any signaling protein. After that the dataset had 1867 signaling and 3317 non-signaling proteins.

The downloaded proteins were put together and PseAAC was used with $\lambda = 3$ and $w = 0.05$ values to turn the data into a 23 attribute and 1 class attribute dataset and dipeptide composition was used to turn the data into a 420 attribute and 1 class attribute dataset. For PseAAC, first 20 attributes are 20 amino acids' frequencies, the other 3 are about the sequence order and the last attribute showing whether that protein is a signaling protein or not. For dipeptide composition, first 20 attributes are

also 20 amino acids' frequencies, the later 400 attributes are dipeptide composition and the last attribute is the class attribute. A scheme of our process is in the Figure 4.1 below.

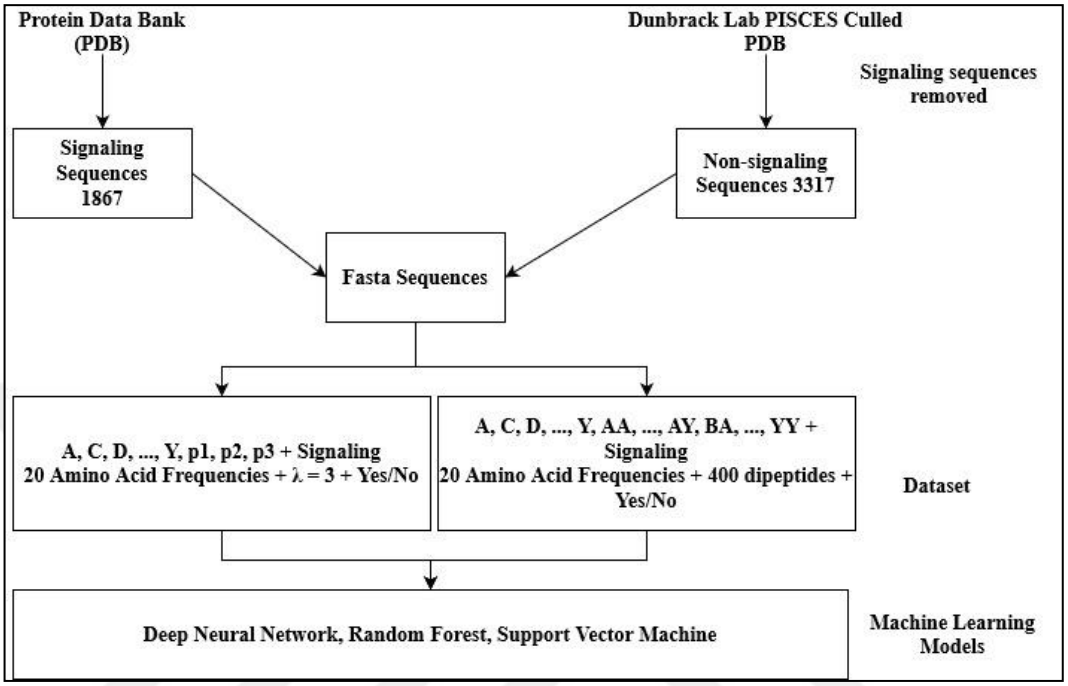

Figure 4.1 Scheme of our process

Some proteins transformed using PseAAC is shown in Figure 4.2.

V	W	Y	ps1	ps2	ps3	signaling
0.0977166	0.00000000	0.01542890	0.0720690	0.0722132	0.0739849	yes
0.0323283	0.01077610	0.03232830	0.0803338	0.0939212	0.0714187	yes
0.0282311	0.00651486	0.03257430	0.0737516	0.0838329	0.0823481	yes
0.0491635	0.00000000	0.03277570	0.0709092	0.0691863	0.0732880	yes
0.0915480	0.00000000	0.00704216	0.0663218	0.0584099	0.0654204	no
0.0605414	0.00605414	0.01210830	0.0755165	0.0839966	0.0655564	no
0.0250783	0.00000000	0.00000000	0.0569413	0.0713135	0.0692400	no
0.0619346	0.00750723	0.02064490	0.0827858	0.0820924	0.0843990	no
0.0711884	0.00000000	0.00000000	0.0482884	0.0585187	0.0507970	no

Figure 4.2 Some proteins transformed by PseAAC

4.2 Model Parameters

Dataset was split into two sets, training and test set. Training set contained the 75% of the proteins while the test set contained the remaining 25%. Initial attempts

at training a classifier showed that the imbalance between classes made the classifier lean on the majority class. To solve this problem a technique called Synthetic Minority Over-Sampling Technique (SMOTE) was used (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). Training set which was originally made of 1400 signaling and 2487 non-signaling samples with the help of SMOTE turned into 2800 signaling 3500 non-signaling samples. Test set contains 467 signaling and 830 non-signaling samples.

For neural network models, the following parameters were tried:

- Number of layers: 1-5
- Number of nodes: 16-512
- Dropout: 0.1-0.5
- Loss function: binary cross entropy, mean squared error, mean absolute error
- Optimizer: stochastic gradient descent, RMSprop, Adam, Adadelta
- Batch size: 64-512

For random forest models, the following parameters were tried:

- mtry: Random search between 5-number of attributes in the dataset
- ntree: 100-1000

For support vector machines, the following parameters were tried:

- cost: $2^5 - 2^{15}$
- gamma: $2^{-15} - 2^{-5}$

Among the several parameters tried, the following ones are chosen because they gave the best results.

For the dataset with PseAAC transformation, the parameters chosen are described below.

RF has number of trees (ntree) and number of attributes selected randomly at each split (mtry) hyper-parameters that needed tuning. After tuning it was found that the best values for mtry and ntree were 5 and 500 respectively.

For SVM, γ and c hyper-parameters needed tuning.0.04347826 for γ and 1for c were the best resulting ones. Radial basis function was the best resulting kernel for SVM.

DNN model for this work has 5 layers with each layer having 256 nodes and 0.5 dropout optimization is used after each layer. For hidden layers, ReLU and for output layer, sigmoid activation functions were used. Binary cross entropy was chosen for the loss function. For learning, Adam optimizer was used with the default parameters. Batch size was set to 512 and epoch was set to 100.

For the dataset with dipeptide composition transformation, the parameters chosen are described below.

For RF, mtry was 20 and ntree was 500.

For SVM, c was 32 and γ was 0.001953125.

For DNN, the settings were the same as the PseAAC model.

4.3 Results and Discussion

Many hyper-parameters were tried for each model. The best results of each model for the PseAAC transformed dataset is in Table 4.1.

Table 4.1 Results of proteins transformed with PseAAC

Models	Accuracy	Precision	Recall	MCC	Specificity	F1-Score	AUROC
SVM	0.749	0.636	0.713	0.472	0.770	0.672	0.742
RF	0.748	0.629	0.732	0.476	0.757	0.677	0.745
DNN	0.763	0.673	0.664	0.483	0.818	0.668	0.741

DNN has the highest accuracy, precision, Matthews Correlation Coefficient (MCC) and specificity. Yet on recall and f1-score it has the lowest values. RF has the best f1-score and AUROC values.

The best results of each model for the dipeptide composition transformed dataset is in Table 4.2.

Table 4.2 Results of proteins transformed with dipeptide composition

Models	Accuracy	Precision	Recall	MCC	Specificity	F1-Score	AUROC
SVM	0.747	0.637	0.692	0.462	0.778	0.663	0.735
RF	0.783	0.785	0.546	0.511	0.916	0.644	0.732
DNN	0.766	0.665	0.704	0.499	0.800	0.684	0.752

RF has the best accuracy, precision, MCC and specificity values yet it is having the lowest recall, f1-score and AUROC values. DNN has the best AUROC and f1-score values.

Given the results, it is difficult to choose which protein encoding scheme or machine learning model itself improved the outcome. The results of all the models used in this study are very close, yet DNN with dipeptide composition has the best f1-score and AUROC values so it is the best model.

CHAPTER 5

CONCLUSION

Cells send and receive many signals in order to start or end biological processes like protein secretion, transportation, apoptosis. Signaling proteins take part in cell signaling and identifying them is important to find new drugs that intervene these processes to fight diseases. For this purpose, finding new methods to identify signaling proteins that are cost effective solutions like machine learning is necessary.

In this thesis, pseudo amino acid composition and dipeptide composition combined with machine learning were used to classify signaling proteins. Results from the dataset that used pseudo amino acid composition were obtained early and they were presented (Küçük & Kandemir Çavaş, 2020b) and published (Küçük & Kandemir Çavaş, 2020a). 5184 proteins were downloaded and encoded with protein encoding schemes to create 2 datasets. The dataset encoded with pseudo amino acid composition had 23 attributes while the dataset encoded with dipeptide composition had 420 attributes. Both datasets were split, creating training and test sets containing 75% and 25% of the proteins respectively. Training sets were imbalanced, ratio of non-signaling proteins to signaling proteins were almost 2 to 1, making the algorithms lean on the non-signaling side. To fix this problem, Synthetic Minority Over-Sampling Technique was applied to them. Deep neural network, random forest and support vector machine algorithms were trained for classification of signaling proteins. Support vector machine and random forest algorithms gave a better result than deep neural network on pseudo amino acid encoded dataset yet on dipeptide composition encoded dataset, they gave a worse one. From these results it cannot be concluded that dipeptide composition or pseudo amino acid composition is the better encoding scheme for this particular problem. Other than that the results were similar yet deep neural network combined with dipeptide composition gave the best result of f1-score 0.684 and AUROC 0.752. These results show that to classify the signaling proteins deep neural network, random forest and support vector machine can be chosen as satisfying and robust models.

Each year the amount of biological data stored increases. To analyze these data, software solutions are needed more and more because it is faster and cheaper. This thesis shows that machine learning combined with pseudo amino acid composition or dipeptide composition can give good results for classifying signaling proteins. In the future, more encoding schemes or different machine learning models can be tried or a more general model for predicting protein function can be created.

REFERENCES

Aggarwal, C. (2014). Support Vector Machines. In *Data Classification: Algorithms and Applications* (1st ed.)(187-202). Boca Raton: Chapman & Hall/CRC.

Ahmad, K., Waris, M., & Hayat, M. (2016). Prediction of protein submitochondrial locations by incorporating dipeptide composition into Chou's general pseudo amino acid composition. *The Journal of Membrane Biology , 249* (3), 293-304.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., et al. (2000). The protein data bank. *Nucleic Acids Research , 28* (1), 235-242.

Bhasin, M., & Raghava, G. P. (2004). Classification of nuclear receptors based on amino acid composition and dipeptide composition. *Journal of Biological Chemistry , 279* (22), 23262-23266.

Bishop, C. M. (2006). *Pattern recognition and machine learning.* Singapore: Springer.

Breiman, L. (2001). Random forests. *Machine Learning , 45* (1), 5-32.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research , 16*, 321-357.

Chen, C., Zhou, X., Tian, Y., Zou, X., & Cai, P. (2006). Predicting protein structural class with pseudo-amino acid composition and support vector machine fusion network. *Analytical Biochemistry, 357* (1), 116-121.

Chen, Y., & Li, Q. (2007). Prediction of apoptosis protein subcellular location using improved hybrid approach and pseudo-amino acid composition. *Journal of Theoretical Biology , 248* (2), 377-381.

Chou, K.-C. (2001). Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins: Structure, Function, and Bioinformatics , 43* (3), 246-255.

Claros, M., Brunak, S., & von Heijne, G. (1997). Prediction of N-terminal protein sorting signals. *Current Opinion in Structural Biology , 7* (3), 394–398.

Dumontier, M., Yao, R., Feldman, H., & Hogue, C. (2005). Armadillo: domain boundary prediction by amino acid composition. *Journal of Molecular Biology , 350* (5), 1061-1073.

Eisenberg, D. (1984). Three-dimensional structure of membrane and surface proteins. *Annual Review of Biochemistry , 53* (1), 595-623.

Fernandez-Lozano, C., Cuiñas, R. F., Seoane, J. A., Fernandez-Blanco, E., Dorado, J., & Munteanu, C. R. (2015). Classification of signaling proteins based on molecular star graph descriptors using Machine Learning models. *Journal of Theoretical Biology , 384*, 50-58.

George, K. A.-M., Sonia, M. L., & Chittibabu, G. (2006). Predicting the subcellular localization of human proteins using machine learning and exploratory data analysis. *Genomics, Proteomics & Bioinformatics , 4* (2), 120-133.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Feedforward Networks. In *Deep Learning* (1st ed.)(163-217). Cambridge: The MIT Press.

Gromiha, M. M. (2010). *Protein Bioinformatics: From Sequence to Function.* India: Academic Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Random Forests. In *The Elements of Statistical Learning* (2nd ed)(587-604). New York: Springer.

Haykin, S. (2009). *Neural Networks and Learning Machines.* New Jersey: Pearson.

Higgs, P. G., & Attwood, T. K. (2005). *Bioinformatics and Molecular Evolution.* United Kingdom: Blackwell Publishing Ltd.

Hopp, T., & Woods, K. (1981). Prediction of protein antigenic determinants from amino acid sequences. *Proceedings of the National Academy of Sciences , 78* (6), 3824-3828.

Hormoz, S. (2013). Amino acid composition of proteins reduces deleterious impact of mutations. *Scientific Reports , 3*, 2919.

Jain, P., Garibaldi, J., & Hirst, J. (2009). Supervised machine learning algorithms for protein structure classification. *Computational Biology and Chemistry , 3*, 216-223.

Johnson, G., & Raven, P. (2002). Cell-Cell Interactions. In *Biology* (123-140). USA: McGraw-Hill.

Kathuria, C., Mehrotra, D., & Misra, N. (2018). Predicting the protein structure using random forest approach. *Procedia Computer Science , 132*, 1654-1662.

Kim, P. (2017). *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence.* California: Springer.

Kriesel, D. (2007). A Brief Introduction to Neural Networks. available at http://www.dkriesel.com.

Kumar, R., Srivastava, A., Kumari, B., & Kumar, M. (2015). Prediction of β-lactamase and its class by Chou's pseudo-amino acid composition and support vector machine . *Journal of Theoretical Biology , 365*, 96-103.

Küçük, Ç., & Kandemir Çavaş, Ç. (2020a). Classification of signaling proteins using computer-based methods. *Euroasia Journal of Mathematics-Engineering Natural & Medical Sciences*, 53-62.

Küçük, Ç., & Kandemir Çavaş, Ç. (2020b). Sinyal proteinlerinin bilgisayar-temelli yöntemlerle sınıflandırılması. *Ankara II. Uluslararası Bilimsel Araştırmalar Kongresi*, 292-293. Ankara: IKSAD Yayınevi.

Lide, D. (2005). *CRC handbook of chemistry and physics* (85th ed ed.). Boca Raton: CRC press.

Lingyun, G., Mingquan, Y., Xiaojie, L., & Daobin, H. (2017). Hybrid method based on information gain and support vector machine for gene selection in cancer classification. *Genomics, Proteomics & Bioinformatics , 15* (6), 389-395.

ManChon, U., Talevich, E., Katiyar, S., Rasheed, K., & Kannan, N. (2014). Prediction and prioritization of rare oncogenic mutations in the cancer kinome using novel features and multiple classifiers. *PLoS Comput Biol , 10* (4), e1003545.

McSkimming, D. I., Rasheed, K., & Kannan, N. (2017). Classifying kinase conformations using a machine learning approach. *BMC Bioinformatics , 18* (1), 86.

Mitra, S., & Hayashi, Y. (2006). Bioinformatics with soft computing. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) , 36* (5), 616-635.

Mondal, S., & Pai, P. (2014). Chou′ s pseudo amino acid composition improves sequence-based antifreeze protein prediction. *Journal of Theoretical Biology , 356*, 30-35.

Moreira, C., Philot, E., Lima, A., & Scott, A. (2019). Predicting regions prone to protein aggregation based on SVM algorithm. *Applied Mathematics and Computation , 359*, 502-511.

Najibi, S., Maadooliat, M., Zhou, L., Huang, J., & Gao, X. (2017). Protein structure classification and loop modeling using multiple Ramachandran distributions. *Computational and Structural Biotechnology Journal , 15*, 243-254.

Nakai, K. (2000). Protein sorting signals and prediction of subcellular localization. *Advances in Protein Chemistry* , 277–344.

Nasibov, E., & Kandemir-Cavas, C. (2008). Protein subcellular location prediction using optimally weighted fuzzy k-NN algorithm. *Computational Biology and Chemistry , 32* (6), 448-451.

Parker, J. (2001). *Encyclopedia of Genetics.* Cambridge, Massachusetts: Academic Press.

Qiu, W., Li, S., Cui, X., Yu, Z., Wang, M., Du, J., et al. (2018). Predicting protein submitochondrial locations by incorporating the pseudo-position specific scoring

matrix into the general Chou's pseudo-amino acid composition. *Journal of Theoretical Biology , 450*, 86-103.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning representations by back-propagating errors. *Nature , 323* (6088), 533-536.

Shatnawi, M., & Zaki, N. (2015). Inter-domain linker prediction using amino acid compositional index. *Computational Biology and Chemistry , 55*, 23-30.

Shen, H., Yang, J., & Chou, K. (2006). Fuzzy KNN for predicting membrane protein types from pseudo-amino acid composition. *Journal of Theoretical Biology , 240*, 9-13.

The UniProt Consortium. (2018). UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research , 47* (D1), D506-D515.

Wang, G., & Dunbrack Jr, R. L. (2003). PISCES: a protein sequence culling server. *Bioinformatics , 19* (12), 1589-1591.

Zhen, C., Ningning, H., Yu, H., Wen, T. Q., Xuhan, L., & Lei, L. (2018). Integration of a deep learning classifier with a random forest approach for predicting malonylation sites. *Genomics, Proteomics & Bioinformatics , 16* (6), 451-459.

APPENDICES

APPENDIX 1: Proteins transformed using dipeptide composition

YM	YN	YP	YQ	YR	YS	YT	YV	YW	YY	signaling
0.00000000	0.00000000	0.00000000	0.00649351	0.00649351	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	yes
0.00000000	0.00000000	0.00000000	0.00000000	0.01388890	0.01388890	0.00000000	0.00000000	0.00000000	0.00000000	yes
0.00549451	0.00274725	0.00000000	0.00000000	0.00000000	0.00274725	0.00274725	0.00000000	0.00000000	0.00549451	yes
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.01010100	0.01010100	0.00000000	0.00000000	yes
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	no
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00775194	0.00000000	0.00000000	no
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	no
0.00000000	0.00000000	0.00000000	0.00000000	0.00731707	0.00000000	0.00243902	0.00243902	0.00000000	0.00000000	no
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	no