DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MICRORNA TARGET PREDICTION FOR CRISPR/CAS9 SYSTEM WITH MACHINE LEARNING

by

Elif DOĞAN

October, 2019

İZMİR

MICRORNA TARGET PREDICTION FOR CRISPR/CAS9 SYSTEM WITH MACHINE LEARNING

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 Computer Engineering

by
Elif DOĞAN

October, 2019

İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**MICRORNA TARGET PREDICTION FOR CRISPR/CAS9 SYSTEM WITH MACHINE LEARNING**" completed by **ELİF DOĞAN** under supervision of **ASST. PROF. DR. ÖZLEM AKTAŞ** 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. Dr. Özlem AKTAŞ

Supervisor

.....

Assoc. Prof. Dr. Deniz KILINÇ

(Jury Member)

.....

Asst. Prof. Dr. Killa BIYANT

(Jury Member)

.....

Prof. Dr. Kadriye ERTEKİN

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to thank my supervisors Asst. Prof. Dr. Özlem AKTAŞ and from Istanbul University Asst. Prof. Dr. Tolga ENSARİ for their guidance, support and encouragement throughout the development of this project.

I would like to thank my friend Zehra AYDINLI for her key ideas about genetic science in my thesis project.

I have special thanks to my husband and my parents for their endless support all along.

Elif DOĞAN

MICRORNA TARGET PREDICTION FOR CRISPR/CAS9 SYSTEM WITH MACHINE LEARNING

ABSTRACT

Since the existence of humankind, many solutions have been investigated for the way of genetic and subsequent diseases. In the late 1900s, a groundbreaking technology, CRISPR was discovered in bacteria. After the exploration of this technique, it is supposed that incurable diseases can be healed by this invention.

The CRISPR/CAS9 system is a powerful tool for regulating damaged genome sequences. Nucleases that are damaged in their sequence are called miRNAs (micro RNAs). The miRNAs targeted by multiple promoter sgRNA (single guide RNA) are cut or regulated from RNA by the CRISPR/CAS9 method.

The sgRNAs targeted to the wrong miRNAs may cause unwanted genome distortions. To minimize these genome distortions, sgRNA target estimation was performed for CRISPR/CAS9 with deep learning in this study.

In this article, Convolutional Neural Networks (CNN), Multi-Layer Perceptron (MLP) and Bidirectional Long Short-Term (BLSTM) algorithms are used. Performance comparison of the CRISPR/CAS9 system for three algorithms was performed.

Keywords: Deep learning, neural networks, convolutional neural networks, multilayer perceptron, long-short term memory, CRISPR/CAS9

MAKİNE ÖĞRENMESİYLE CRISPR/CAS9 SİSTEMİ İÇİN MİKRORNA HEDEF TAHMİNİ

ÖZ

İnsanlığın varoluşundan beri genetik ve sonradan oluşan hastalıkların tedavisi için birçok çözüm arayışına gidilmiştir. 1900'lü yılların sonlarında bakterilerde CRISPR adında çığır açan bir teknoloji keşfedildi. Bu buluş sayesinde tedavisi bulunamayan hastalıklar tedavi edilebileceği düşünülmektedir.

CRISPR/CAS9 sistemi, hasarlı olan genom dizilişlerinin düzenlenmesinde kullanılan çok güçlü bir araçtır. Diziliminde hasar oluşan nükleazlar miRNA (micro RNA)'lar olarak adlandırılır.

Birden çok rehber sgRNA (single guide RNA) tarafından hedeflenen miRNA'lar, CRISPR/CAS9 yöntemiyle RNA'dan kesilir ya da düzenlenir. Yanlış miRNA'lara hedeflenen sgRNA'lar, istenmeyen genom mutasyonlarına sebep olabilmektedir. Bu genom bozulmalarını en aza indirgemek amacıyla, bu çalışmada derin öğrenmeyle CRISPR/CAS9 için sgRNA hedef tahmini yapılmıştır.

Bu makalede, Evrişimsel Sinir Ağları (Convolutional Neural Networks-CNN), Çok Katmanlı Algılayıcı (Multi-Layer Perceptron-MLP) ve Çift Yönlü Uzun Kısa Vadeli Hafıza (Bidirectional Long Short-Term-BLSTM) algoritmaları kullanılmıştır. Her üç algoritmanın da CRISPR/CAS9 sistemi için performans karşılaştırması gerçekleştirilmiştir.

Anahtar kelimeler: Derin öğrenme, yapay sinir ağları, evrişimli sinir ağları, çok katmanlı algılayıcı, uzun-kısa süreli hafıza ağları, CRISPR/CAS9

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

1.1 Brief Description and Goals of Thesis

Treatment methods of diseases are often discussed but rarely understood that the problem is in the origin. This means that is our genes. The latest researches show that potential diseases are available on our gene map. This means, disordered genes consist of eventual illness. If we can interpret correctly this map, we can regulate disordered genes. However, gene regulation may cause unwanted results, such as undesired gene distortions and mutations.

In this context, programmable nucleases are a major role in disordered genes. Programming perfect gene regulation is important however, the most important thing is behind the idea. This technology should be only used for the human benefit, for example, eliminate diseases, increase quality food production. However, working with genes is hazardous. If something goes wrong, other situations may cause undesirable results. The fault will be eternal and transferring the corrupted genes to other generations will be occurred. For this reason, impeccable gene regulation is obligatory.

Some of the researches show that, *in vitro* and *in vivo* techniques were used in gene regulation (Liu et al., 2015; Wang et al., 2016). However, these techniques are expensive, risky and it takes a long time to test.

Clustered Regularly Interspaced Short Palindromic Repeats-CRISPR is the latest gene regulation technique. In this research, CRISPR technique is examined. Before CRISPR, ZFN and Transcription Activator-Like Effector Nucleases (TALENs) were used in gene regulation (Klug 2010; Miller 2011). In this project, CRISPR are examined *in silico*.

Recently, several studies have investigated CRISPR using machine learning algorithms. They created new deep learning models for on/off-target prediction

(Abadi et al., 2017; Kirillov, 2017; Chuai et al., 2018). However, none of them have performed a comparison of deep learning models with each other. To finding excellent results, the comparison between other deep learning models should be performed. Within comparison, this project aims to contribute to finding an ideal solution in CRISPR gene regulation.

1.2 Brief Overview of Gene Regulation Techniques

In vivo and *in vitro* techniques means laboratory environment. *In vivo* experiment affects the whole organism of human or an animal. *In vitro* experiment is realized on a human cell, an animal cell or protista.

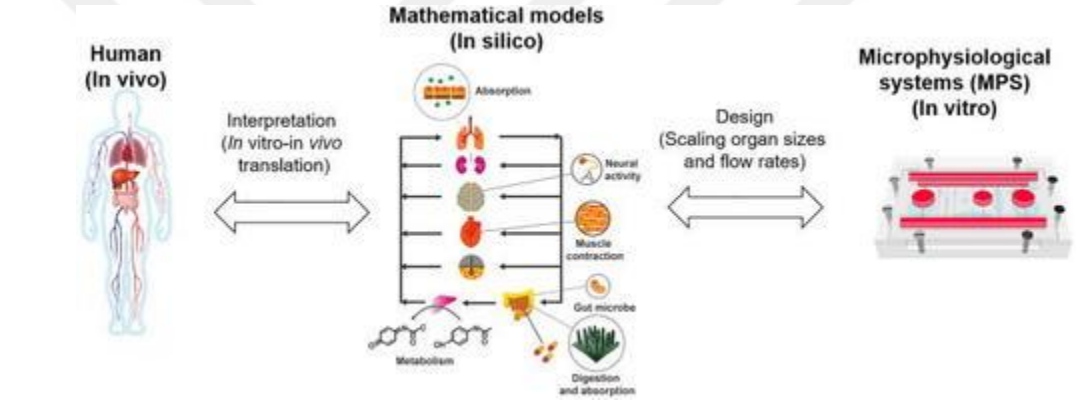


Figure 1.1 The illustration of *in vivo*, *in vitro* and *in silico* experiment platforms (Sung, 2019)

Figure 1.1 shows us *in vivo, in vitro* an *in silico* experiment environment. *In silico* means digital environment of experiment platform.

The purpose of the *in silico* technique is to increase the accuracy of disease prevention by single pointing with the help of mechanization. The large base readings provided by mechanization also contribute greatly.

1.3 CRISPR-Cas9

In Escherichia Coli (E. Coli) bacteria that investigated by *in silico* (simulation) method which is an important role nowadays has been discovered immune systems

named CRISPR (Clustered Regularly Interspaced Short Palindromic Repeats)-Cas9 (Wilkinson & Wiedenheft, 2014).

According to the system an E.Coli bacteria which is infected by any virus, add the virus DNA its memory and remember when any other virus invasion accrue. This defined virus DNA slices this virus DNA from its DNA through the Cas9 enzyme. Thus DNA repair will happen. According to recent research, the leading factor in gene organization has been observed as the "microRNA (miRNA)" targeted by the "single-guide RNA (sgRNA).

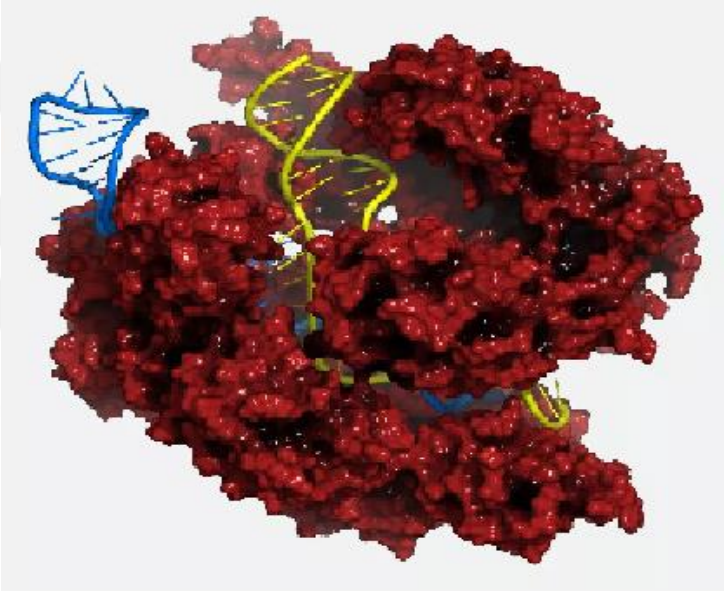


Figure 1.2 The illustration of Cas9 enzyme cuts DNA by sgRNA targeting (Plumer et al., 2019)

CRISPR/CAS9 is used to destroy targeted miRNAs in cells (John et al., 2017). It uses "guide RNA (gRNA)" as a guide to target the CRISPR/Cas9 nucleus to the DNA sequence and triggers the double-strand split at the wanted location. The cleavage and repair of these wires can produce random addition and remodeling of DNA (Kurata & Lin 2018). Figure 1.2 shows us how CRISPR/Cas9 enzyme cuts DNA into RNA to repairing. Cas9 enzyme is red, sgRNA is blue and DNA is yellow. In Figure 1.3, gene targeted by sgRNA is copied first. Behind the RNA regulation is completed, DNA cuts the disordered RNA. The fixed RNA inserted into DNA. Through this activity, gene regulation was made.

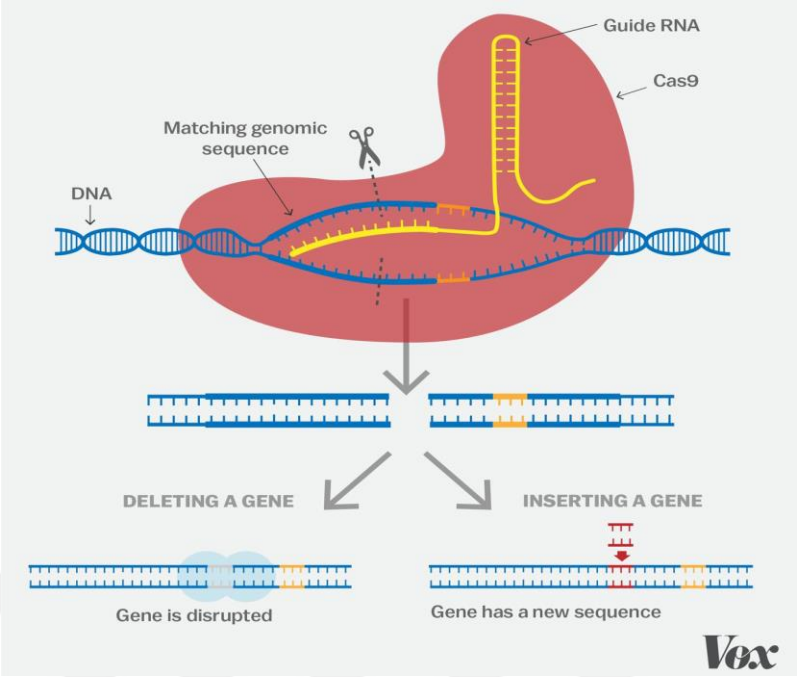


Figure 1.3 The illustration shows gene regulation by CRISPR/Cas9 enzyme (Plumer et al., 2019)

The other point is, miRNA activities are changed in each cell type (Hirosawa et al., 2017). For example, he acts uniquely in any cell. So, the miRNA activities of a human cell and a bacterial cell will not be the same. From this point, it is important to find a data set about the genomes to be studied in this respect.

The aim of this study is to assigned miRNA target estimation with machine learning algorithms. The result of wrong targeted miRNAs may cause undesirable gene mutations (Zhang et al., 2015). It is aimed to minimize the errors of miRNAs targeting by applying machine learning and deep learning algorithms. Thus, the wrong target estimate will be minimized. In this way, it will be possible to reliably repair gene damage by correctly targeting mistargeted miRNAs. Genetic disorders will be eliminated by the repair of gene deformities.

1.4 Task Distribution of Thesis

If we need to represent a flowchart about the project, obtaining comprehensive data set is the main part of the study. Data preparation was performed. To render the data set, data should be clear. So, the second task is data preprocessing in research.

After implemented the algorithm on the data set, results were examined. Also, results are displayed with related charts. The third is to determine which deep learning algorithm was performed on the data set. Last is, finding the nearest off-target. This part is the most important task of the thesis.

1.5 Development Environment of Thesis

Recommended algorithm design developed in Google Colaboratory. That is a free Jupyter notebook environment with Python programming language. Google Colaboratory supports to Tesla K80 GPU. In Colaboratory notebook setup and install are not required to run the algorithm. The process runs totally on Google Cloud (https://colab.research.google.com/). It acts as a GPU supercomputer on the cloud. In Figure 1.4, Google Colaboratory environment is available.

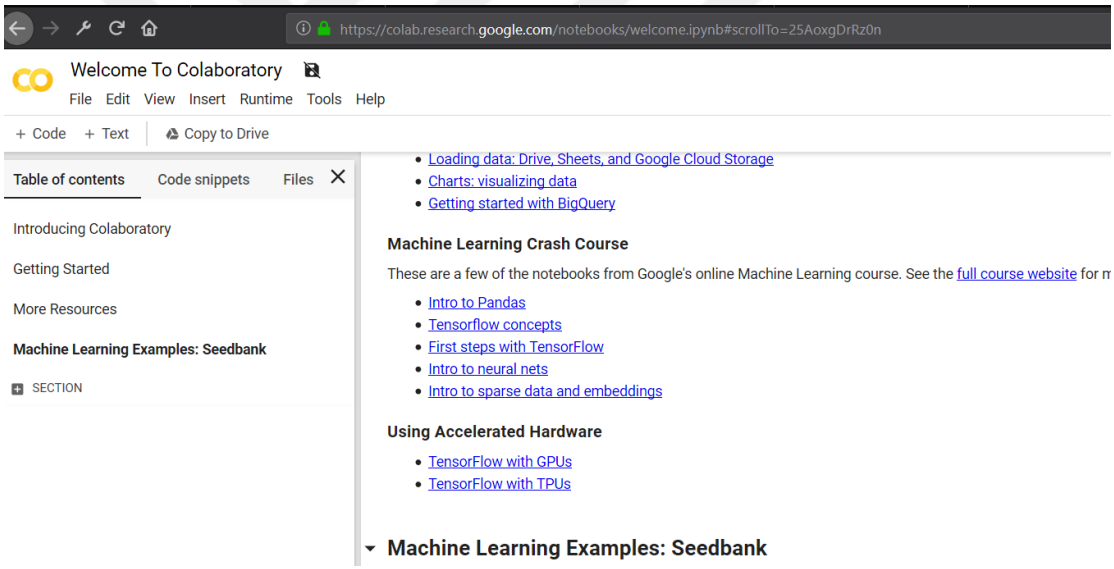


Figure 1.4 Environment of Google Colaboratory

A GPU supercomputer is a networked combination of computers with several Graphic Processing Units. GPU supercomputers allow more agile processing of tasks because of the essentially parallel nature of GPUs. By shader cores on GPU that allow multiple pixels to be rendered and multiple streams of data processing at the same time. It also can similarly process multiple streams of data at the same time. It can be managing the enormous workloads.

In this thesis, Python programming language with Keras and Tensorflow library were used. Keras is an open-source neural network algorithm library. Either we can generate our own model or use a prepared model. We have generated our own deep learning models in this thesis.

1.6 Organization of Thesis

This research is separated into 5 parts and 1 appendices. At first, a short explanation of the research, definition and work distribution are presented in Chapter 1. Stages within the thesis are shortly explained in Chapter 2. Previous academic studies on related topics are discussed in Chapter 3. Used deep learning algorithms and data set preparation, creating a new model are described in Chapter 4. Lastly, an outline of the complete research is given in Chapter 5.

CHAPTER TWO
TASK DEFINITION

In this section deep learning and neural networks, algorithms are shortly described. Activation functions, optimization functions, loss functions are analyzed.

2.1 Neural Networks

An artificial neural network is similar to a real neuron. In Figure 2.1 shows us dendrite take signals from other neurons. They are the input layer of the network. Soma (cell body) calculates and sums of the neurons. This represents the hidden layers of the neural network. Axon transmits the signals to axon terminals, which are the output layer of the neural network.

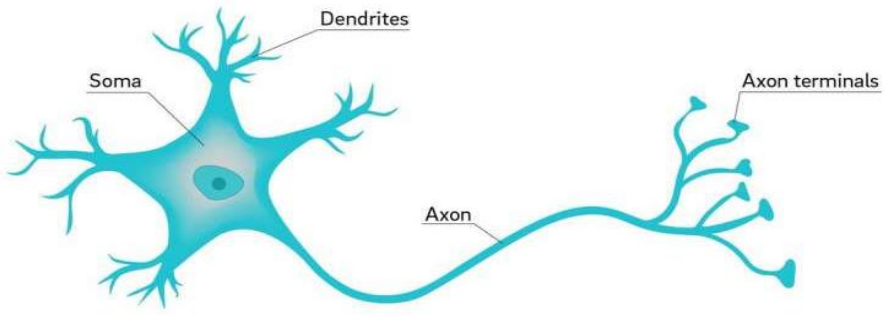


Figure 2.1 A real neuron (Puppo et al., 2018)

In Figure 2.2 represents layered directed graph which is fully connected neural network. One input layer and one output layer is connected to hidden layers.

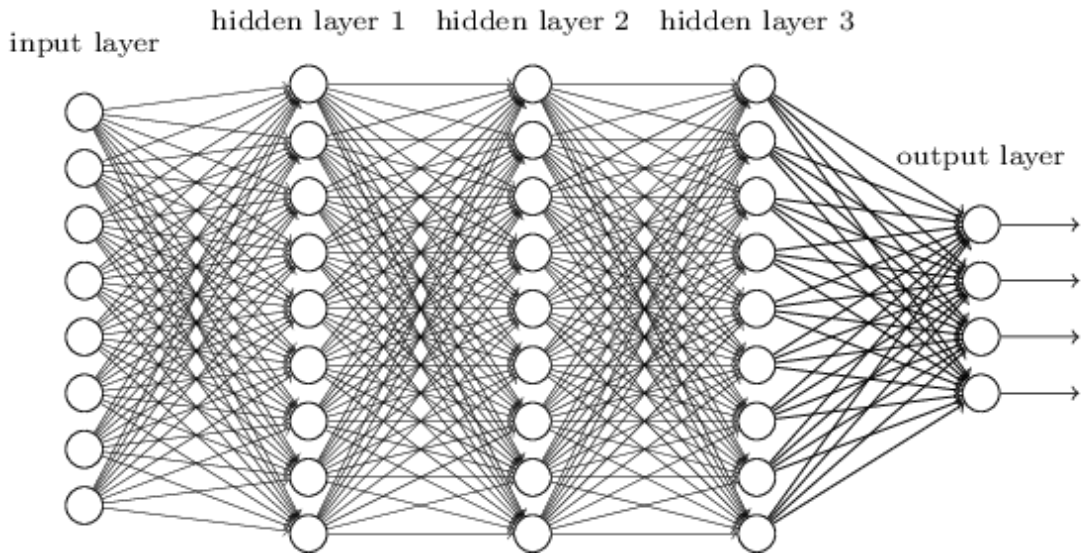


Figure 2.2 Fully connected neural network

Figure 2.3 shows us the formula of Convolutional Neural Networks. x_0 is the input layer of the formula. w_0 is weight vector and w_0x_0, $w_1x_1 \ldots w_ix_i$ are dendrites. Calculation with dot product in cell body formula is given below. (*w*: weight, *x*:input, *b*: bias value, *f*: activation function)

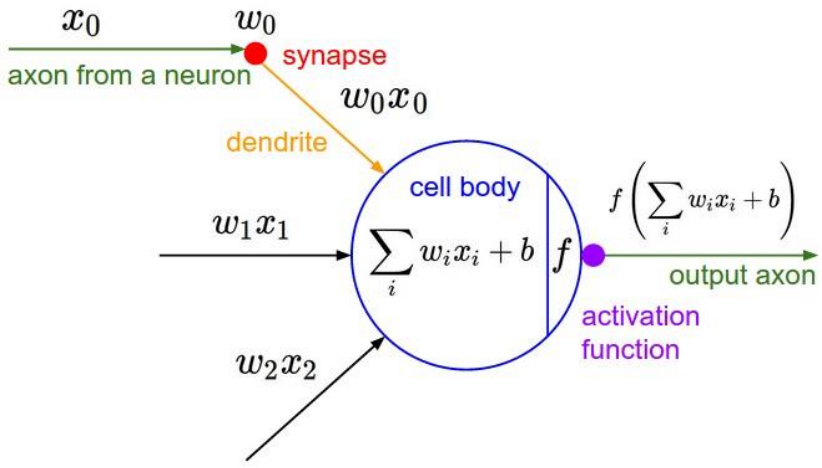


Figure 2.3 Fully connected neural network formula

2.2 Activation Functions

Activation functions are necessary for a neural network model learning. Without activation functions, deep learning models seem to be linear classifiers. The purpose of the activation function is to fix model non-linear and turn in the input signal into a sensible output signal. Activation functions are given below.

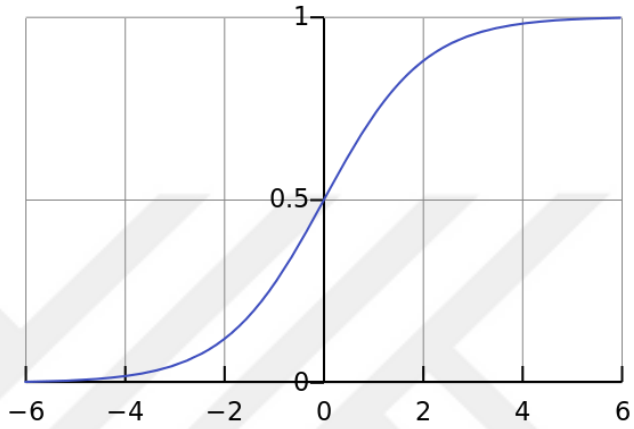


Figure 2.4 Sigmoid activation function

Sigmoid or logistic activation function formula is given in formula 2.1. (*f(x)*: logistic/sigmoid function, *e*: exponential)

$$f(x) = \frac{1}{1+e^{-x}} \tag{2.1}$$

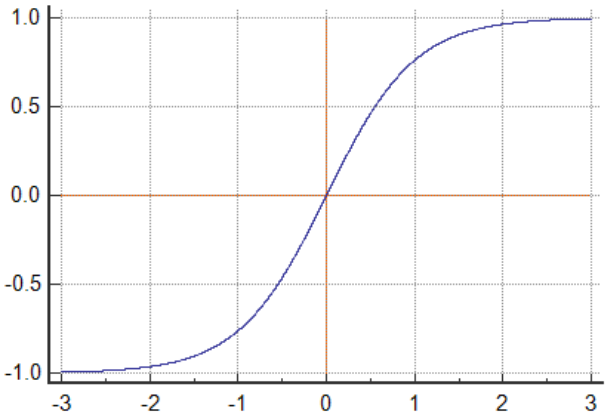


Figure 2.5 Hyperbolic Tangent activation function

In Figure 2.5 hyperbolic tangent activation function is shown.

$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1 \qquad (2.2)$$

In 2.2 formula hyperbolic tangent activation function is shown. (*e:* exponential)

According to Hahnloser et al. (2000), the most successful result is given by Rectified Linear Unit (ReLU) activation function. ReLU activation function is shown in Figure 2.6.

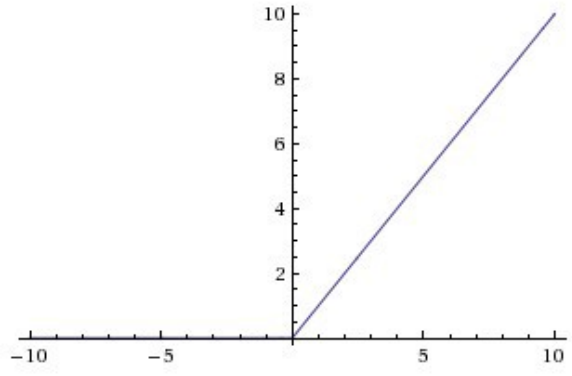


Figure 2.6 Rectified Linear Unit (ReLU) activation function

$$f(x) = \max(0, x) \qquad (2.3)$$

The formula of ReLU is shown in formula 2.3. According to formula if the output is greater than 0 then the result is x, otherwise, the result is 0. (*f(x)*: activation function, *x*: output)

The last primary activation function used in neural networks is softmax. Softmax formula is given in formula 2.4.(*x*: features, *j*: weight)

$$f_i(\vec{x}) = \frac{e^{x_i}}{\sum_{j=1}^{J} e^{x_j}} \qquad for\ i = 1,2 \ldots, J \qquad (2.4)$$

According to Dunne & Campbell (1997), using softmax function with categorical crossentropy loss function in neural networks model gives more accurate results.

In Table 2.1 other activation functions is given. (Salman, 2018)

Table 2.1 Activation functions (Salman, 2018)

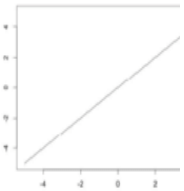
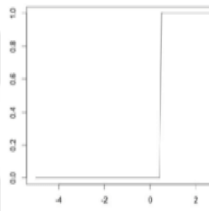
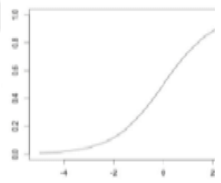
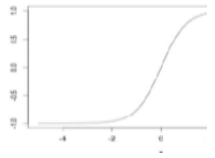
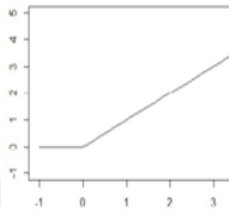
Activation Function	**Description**	**Equation**	**Implementation**
Linear	It does not change the output.	$\emptyset(x) = x$	
Step/Treshold	It returns true for values that are over the defined threshold.	$\emptyset(x) = \begin{cases} 1, if\ x \geq 0.5 \\ 0, otherwise \end{cases}$	
Sigmoid/Logistic	The output is just a positive number. It assures that values stay within a nearly small range.	$\emptyset(x) = \frac{1}{1+e^{-x}}$	

Table 2.2 Continuous activation functions (Salman, 2018)

Hyperbolic Tangent	Output values in the range between -1 and 1.	$\emptyset(x) = \tanh(x)$	
Rectified Linear Unit (ReLU)	It is a linear unsaturated function. ReLU does not saturate to -1,0 or 1. The activation function runs towards and finally finds a value.	$\emptyset(x) = \max(0, x)$	

The major activation functions, that are used in neural network models are examined above. There are many others exist. However, according to Sibi et al. (2013), there is a slice different between activation functions on the neural network model. The most important things about the neural network model are training algorithm, network sizing, and learning parameters.

2.3 Loss Functions

The loss function is used for parameter estimation. The loss function is determinative for estimation quality. In deep learning with the optimization algorithms, error function or loss function is used. The further output layer of the model should be relevant to loss function. Using this, reduce the loss rate in the next period. The varieties of loss function is given the following. There are two groups of loss functions are used. The first group is used in classification algorithms: Log loss, Focal loss, Exponential loss, Hinge loss, Binary Cross Entropy, Multi Class Cross Entropy, Kullback Leibler Divergence Loss, Sparse Multi Class Cross Entropy. The second group is used in regression algorithms: Mean Square Error/Quadratic Loss (MSE), Mean Squared Logarithmic Error, Mean Absolute Error (MAE).

In Formula 2.7 and 2.8 MSE and MAE which are convenient for regression algorithms formula is given below. (*n*: number of training examples, *i*: training example number in data set, y_i: ground truth, $\widehat{y_i}$: prediction of i'th training)

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{n} \tag{2.7}$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \widehat{y_i}|}{n} \tag{2.8}$$

In Formula 2.9, Cross Entropy Loss function which is convenient for classification algorithms is given below. (i: training example number in data set, y_i: ground truth, $\widehat{y_i}$: prediction of i'th training)

$$Cross\ Entropy\ Loss = -(y_i \log(\widehat{y_i}) + (1 - y_i)\log(1 - \widehat{y_i})) \tag{2.9}$$

2.4 Optimization Functions

Optimization methods allow neural network models to learn. Whereas, some of them are faster than the others. The major optimization techniques in deep learning are Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam). SGD implementation is easy. Also, it is fine-tuned to feature scaling. Adam measures

singular learning rates for various parameters. In Formula of SGD is given in 2.10. (*X*: total data set, *x*: a sample from *X*, *l(x,w)*: the loss computed for each sample)

$$SGD = \frac{1}{|X|} \sum_{x \in X} l(x, w) \tag{2.10}$$

In Figure 2.7, we can see the performance of SGD optimization with various loss functions.

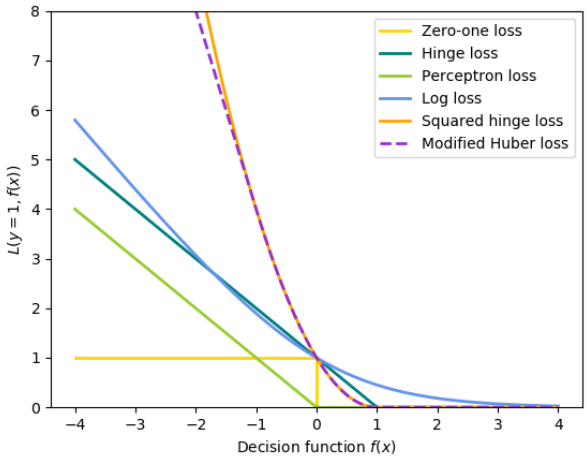


Figure 2.7 SGD on various loss functions (Pedregosa et al., 2011)

2.5 Off-target Prediction with Deep Learning

Figure 2.8 shows the workflow of this thesis. At first, data is prepared. Second, the algorithm applied to the dataset one by one. Then activation functions, loss functions are applied to the algorithm. After, optimization algorithm is applied. Further, the model fitting on data and machine learning part is conducted. Last, the results are compared. Accuracy and loss ratio is defined.

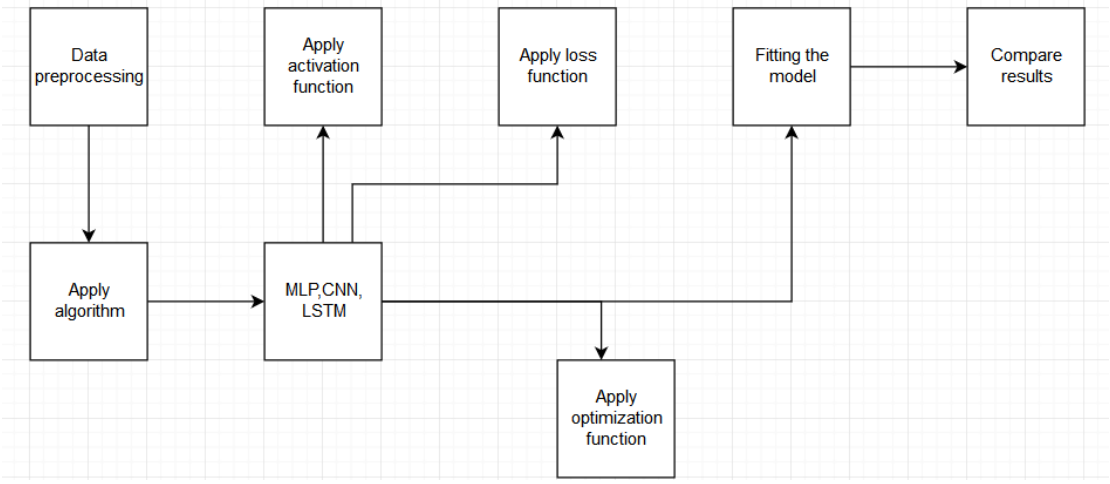


Figure 2.8 Workflow of thesis

CHAPTER THREE
PREVIOUS WORK

This part of the research consists of past studies about the role of deep learning in gene regulation. Some of the researches applied MLP, while the others worked with CNN, DCNN, LSTM/RNN etc. Some other researchers analyzed CPF1 enzyme, while the others examined CAS9 enzyme. Following the study, these researches are analyzed via complete literature research performing.

3.1 Cutting Frequency Determination

Studies declared, Cutting Frequency Determination (CFD) measures the off-target effect on sgRNA. Some of the researchers asserted that CFD is better for measure mismatches than other calculation techniques. While the other researchers stated that their algorithms are better than CFD. In this research, CFD and machine learning algorithms are combined.

Doench et al. (2016) proposed to CFD scale the potential of off-target activity. According to their research, targeting the genome H2-D they measure off-target activity by CFD score and the other off-target metrics, such as CCtop and Hsu-Zhang (MIT Score). CFD score provided the best performance compared to others. In the situation that, more than one mismatch CCtop and Hsu-Zhang (MIT Score) metrics presented low performance. In their research, CFD scores were analyzed with the experimental tool named Guide-Seq to interpret CFD in-depth. Guide-Seq results were compared to calculated off-target scores. The result of this, the best Pearson correlation was provided by CFD and the second one was CCtop and the last one was Hsu-Zhang (MIT Score).

In Figure 3.1 off-target activity performance comparison is given. This figure shows us Area Under Curve (AUC) and Pearson correlation coefficient related to CFD. Also, it shows us the behavior of the scoring off-target activities performance by increasing the number of mismatches.

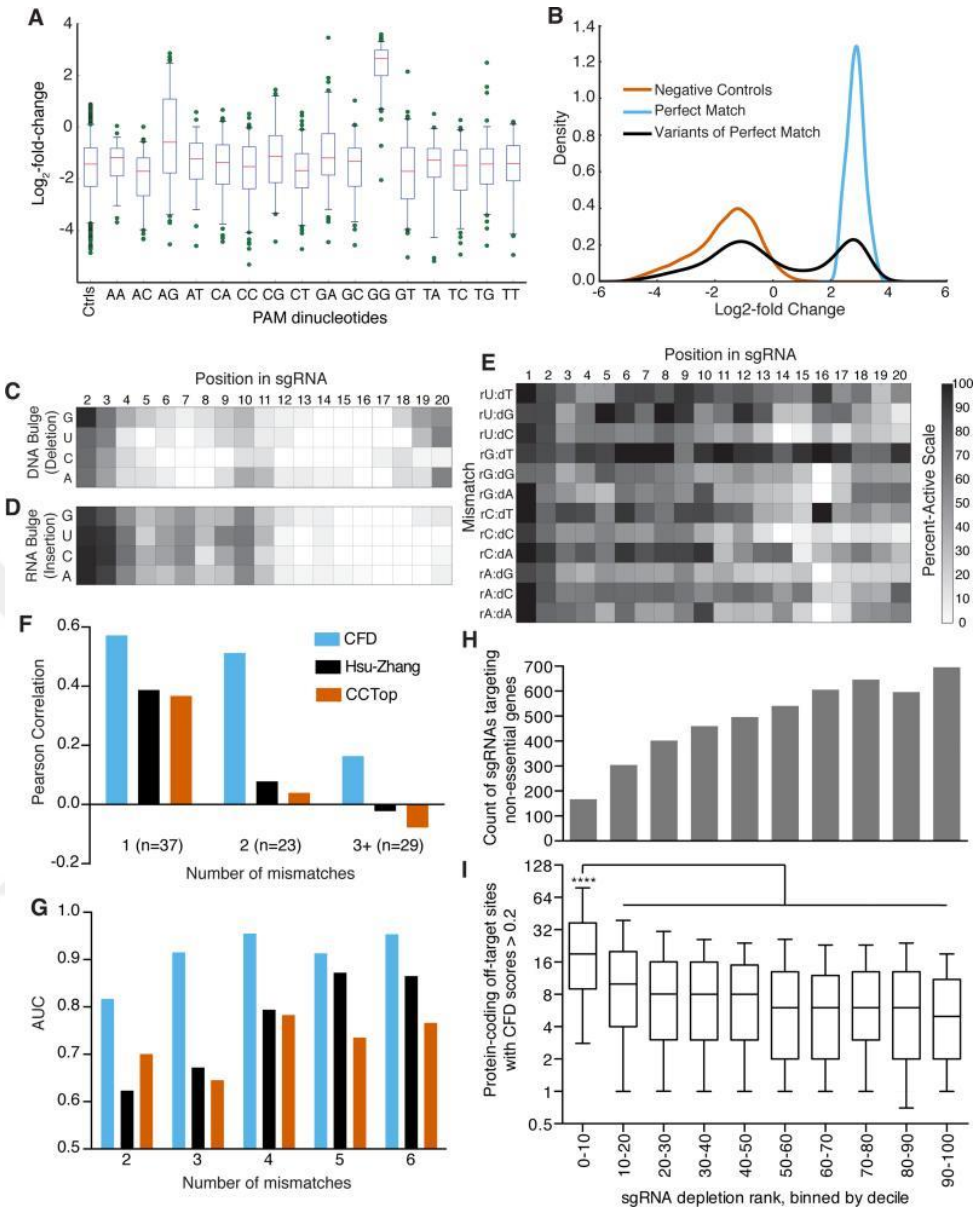


Figure 3.1 CFD performance (Doench et al., 2016)

Haeussler et al. (2016) investigated and developed a web-based tool named CRISPOR. They also examined four groups of off-target activity algorithm. The measurement points to, the best performance between the algorithms was CFD. Figure 3.2 shows us these four algorithms. As we see, CFD algorithm's AUC score is the highest. It is 0.91, while MIT Score AUC is 0.87, MIT Website AUC is 0.73, Cropit AUC score is 0.81 and CCtop Score AUC is 0.77 etc. They also stated that off-target scores are measured by the position on guide RNA sequence of mismatches.

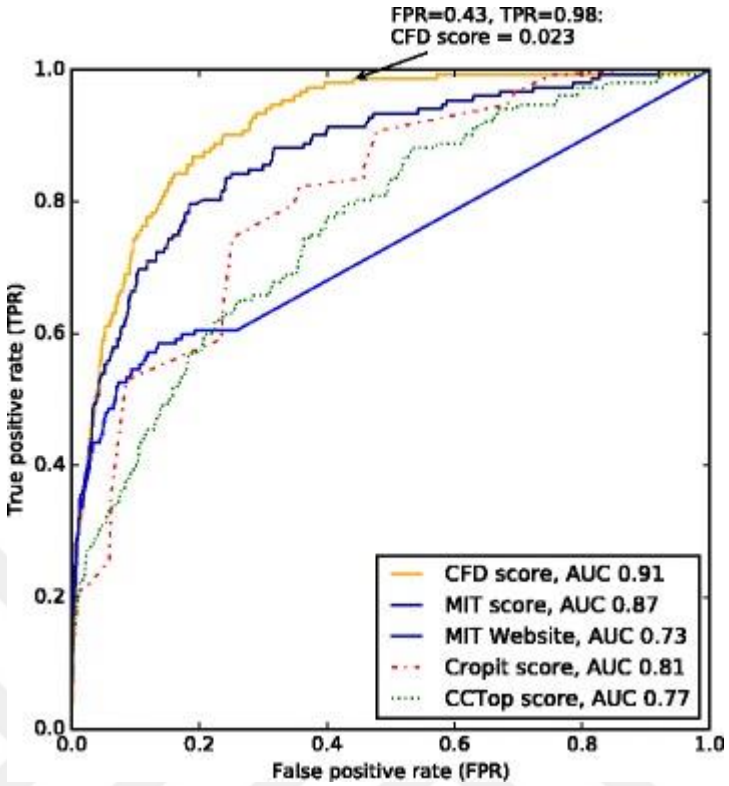


Figure 3.2 CFD performance (Haeussler et al., 2016)

Listgarten (2017) proposed that in CRISPR/Cas9 for 1 mismatch causes effect the 100 sites. 2 mismatch effects 1.000 sites and 3 mismatch effects 100.000 sites. Also, she states that a wrong target may cause block suppressed cancer genes. To avoid these unwanted results, the right targeting algorithm choosing is essential.

Lin & Wong (2018) analyzed four off-target prediction algorithms and deep learning algorithms. The off-target prediction methods are CFD, MIT, CROP-IT and CCtop and deep learning algorithms. The best performance of the methods was CFD. However, they finally indicated that their deep learning algorithm is better than CFD. The AUC value of CFD is 0.793 and the AUC value of CNN is 0.881.

Listgarten et al. (2018) stated that the formula of CFD. They indicated that CFD is similar to the Naïve Bayes algorithm. CFD measuring is given in Formula 3.1. ($Y=1$: gRNA target is active, $Y=0$: gRNA target is inactive, X_i: mismatch occurrence, i: mismatch number)

$$CFD = \prod_{i \in \{i|X_{i=1}\}} P\,(Y = 1|X_i = 1) \quad (3.1)$$

They also stated that if features are independent, then the Naïve Bayes algorithm simplified like in Formula 3.2. (*Y=1*: gRNA target is active, *Y=0*: gRNA target is inactive, X_i: mismatch occurrence, i: mismatch number)

$$Naïve\ Bayes = \prod_i P(Y = 1|X_i) \quad (3.2)$$

3.2 One-hot Encoding and DNA Sequence Mapping

One-hot encoding is a technique for converting categorical data into binary form to let reasonable for machine learning algorithms. These input features converted into 0's and 1's. The studies stated that DNA sequence mapping is handled by the binary representation of nucleotides. The bases transformed into numbers in a meaningful form such as 1's and 0's.

Damasevicius (2008) stated that methods of classification problems in DNA encoding. He summarized the DNA sequence mapping rules in binary forms. The rules table is given in Figure 3.3. These mapping rules are binary feature mappings that consist of the various number of vectors. Furtherly, he denoted all types of nucleotides such as strong nucleotides, weak nucleotides, amines, ketones, purines, pyrimidines. Also, Figure 3.3 shows us mapping rules are individualistic from each other. Since they are related to the different parts of the DNA molecules. He additionally declared that binary mapping rules increase the quality of the classification in machine learning algorithms. (*A*: Adenine, *C*: Cytosine, *G*: Guanine, *T*: Thymine, *S*: strong nucleotides, *W*: weak nucleotides, *K*: ketones, *M*: amines, *R*: purines, *Y*: pyrimidines)

Rule type	Rule name	Symbol: feature	Rule	Feature size
Binary	Orthogonal	1:4	$\langle A \to (0,0,0,1), C \to (0,0,1,0), G \to (0,1,0,0), T \to (1,0,0,0) \rangle$	$4N$
	Binary 1	1:2	$\langle A \to (0,0), C \to (0,1), G \to (1,0), T \to (1,1) \rangle$	$2N$
	Binary 2	1:2	$\langle A \to (0,0), C \to (0,1), G \to (1,1), T \to (1,0) \rangle$	
	Binary 3	1:2	$\langle A \to (0,0), C \to (1,1), G \to (0,1), T \to (1,0) \rangle$	
Single-letter	A	1:1	$\langle A \to 1, B \to 0 \rangle, B = \{C, G, T\}$	N
	C	1:1	$\langle C \to 1, D \to 0 \rangle, D = \{A, G, T\}$	
	G	1:1	$\langle G \to 1, H \to 0 \rangle, H = \{A, C, T\}$	
	T	1:1	$\langle T \to 1, V \to 0 \rangle, V = \{A, C, G\}$	
Grouping	SW	2:1	$\langle S \to 1, W \to 0 \rangle, S = \{A, T\}, W = \{C, G\}$	N
	KM	2:1	$\langle K \to 1, M \to 0 \rangle, K = \{A, C\}, M = \{G, T\}$	
	RY	2:1	$\langle R \to 1, Y \to 0 \rangle, R = \{A, G\}, Y = \{C, T\}$	

Figure 3.3 DNA sequence mapping rules (Damasevicius, 2008)

Listgarten (2017) recommended that each of the bases has a binary meaning in machine learning. In Table 3.1 all of the DNA bases has binary equivalent. All of the bases converted as binary numbers. So the DNA sequence in machine learning should be like that ACGTAATGT is 000100100100100000010001100001001000.

Table 3.1 The meaning of the bases

A	0001
C	0010
G	0100
T	1000

Tahir et al. (2019) stated that genome sequences as consisting of 0's and 1's. The form of the sequence converted into binary representation is in Figure 3.4. They denoted as one dimensional vector with nucleotides as four channels. Utilizing this technique, input features are eligible for the machine to learning.

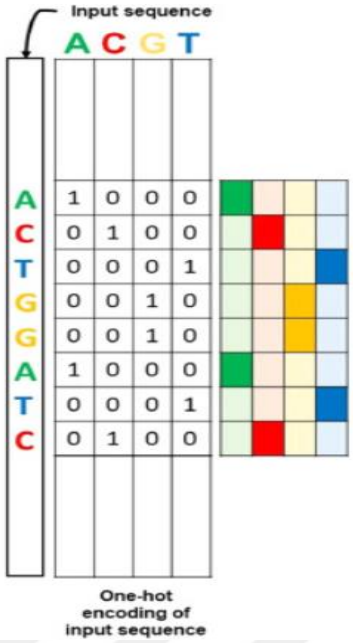


Figure 3.4 One-hot encoding in genome sequences (Tahir et al., 2019)

They took into account 4 channels vector in Figure 3.4. After that, these input vectors are included in deep learning algorithms.

3.3 Machine Learning Approaches in sgRNA Targeting

Montague et al. (2014) developed a CRISPR tool with machine learning algorithms. The tool named CHOPCHOP. The tool accepts as input DNA sequences in FASTA format. This format is a representation of nucleotides as a text form. First, Lipman et al. (1985) proposed FASTA format is given in Figure 3.5.

```
>P01013 GENE X PROTEIN (OVALBUMIN-RELATED)
QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTREMPFHVTKQESKPVQMMCMNNSFNVATLPAE
KMKILELPFASGDLSMLVLLPDEVSDLERIEKTINFEKLTEWTNPNTMEKRRVKVYLPQMKIEEKYNLTS
VLMALGMTDLFIPSANLTGISSAESLKISQAVHGAFMELSEDGIEMAGSTGVIEDIKHSPESEQFRADHP
FLFLIKHNPTNTIVYFGRYWSP
```

Figure 3.5 FASTA format (Lipman et al., 1985)

Abadi et al. (2017) proposed that a machine learning tool named CRISTA. This machine learning tool predicted genome segmentation tendency with a specific sgRNA. They claimed that the highest accuracy is provided by their prediction tool. In Figure 3.6 the algorithm flow is given according to three separate databases i.e.

GUIDE-Seq (Kleinstiver et al., 2016) with 19 sgRNAs and 502 sites, BLESS (Ran et al., 2015) with 4 sgRNAs and 37 sites, HTGTS (Frock et al., 2014) with 9 sgRNAs and 176 sites. They denoted that in this algorithm, they measured the performance of each sgRNA sequence. The AUC of the algorithm CRISTA is 0.96 (96%).

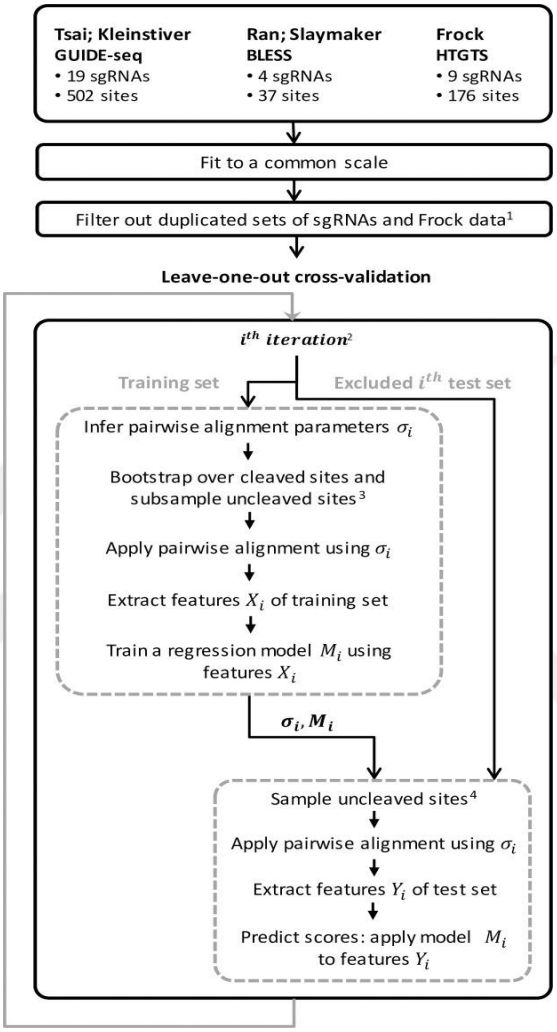


Figure 3.6 Machine learning algorithm sgRNA target detection (Abadi et al., 2017)

Listgarten et al. (2018) proposed that a new machine learning approach in Figure 3.7. According to this model, 2 mismatches are divided into two separate mismatches. Each of the single layers' scores calculated. Then, these separated scores merged in the second layer. These calculations accepted as input features. Then these input features are run in a model. False Positive Rate (FPR) of the proposed algorithm found 0.98 (98%).

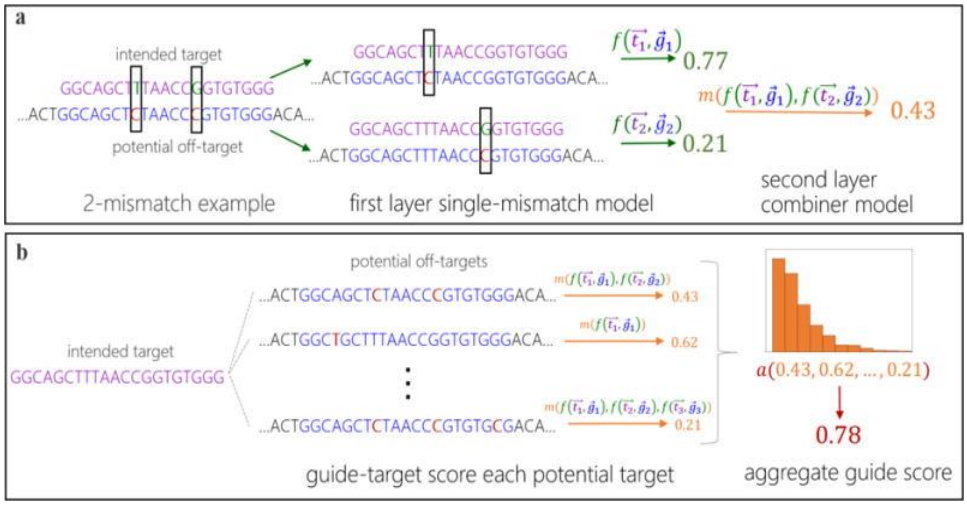


Figure 3.7 gRNA targeting machine learning approach (Listgarten et al., 2018)

According to Figure 3.7, the highest CFD score means the lower mismatch numbers for sgRNA sequences. For this reason, in this research, CFD score is used as a target prediction in Neural Network models. This subject is in the next section.

3.4 Deep Learning Algorithms

In research, three main neural network algorithms are examined. These are Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN) and Long-Short Term Memory (LSTM). The three algorithms using areas are researched. Most of the studies are researched deep learning algorithms on image processing area. However, they also stated that surprisingly deep learning algorithms provided high-performance in text data.

3.4.1 Multi-layer Perceptron

Svozil et al. (1997) stated multi-layer feed-forward perceptron. According to research MLP algorithms implementations are formed generally 3 layers. Input layers, hidden layers, and output layers. In Figure 3.8 multi-layer perceptron layers are given.

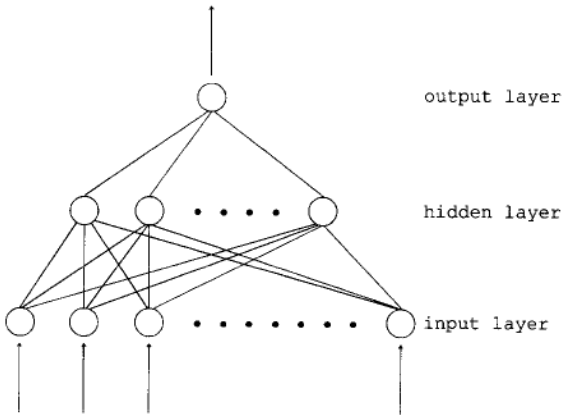


Figure 3.8 Multi-layer feed forward neural networks (Svozil et al., 1997)

Zhang et al. (1998) examined MLP on face recognizing. They researched that emotion detection from image data. They developed two-layer perceptron and achieved 92.3% accuracy.

Hapudeniya (2010) stated that machine learning algorithms such as MLP is used in Bioinformatics. He examined that MLP is able to classify the data set to splits the data into separated sections. Workflow diagram of machine learning algorithm in Bioinformatics is given in Figure 3.9.

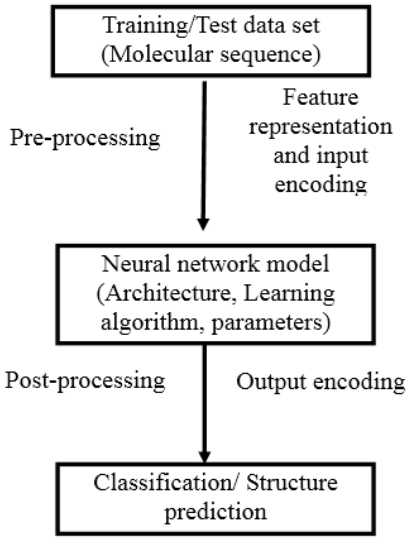


Figure 3.9 Machine learning algorithms in Bioinformatics workflow diagram (Hapudeniya, 2010)

Kökver et al. (2014) investigated the affecting factors of hypertension with machine learning algorithms. They used classification algorithms such as Naïve Bayes. Also, they used neural network algorithms such as MLP. Precision of the

Naïve Bayes algorithm found 81% and MLP was 75%. The accuracy found 91.66%. with Naïve Bayes and 86.11% with MLP.

Timuş, Oğuz & Kıyak (2015) analyzed that sleep apnea detection with ECG signals by using MLP. They found the accuracy of MLP algorithm 75.29%, sensitivity 75.09% specificity 75.53%. They decided to analyze other machine learning algorithms.

Marrtin, Vedat & İmal (2015) proposed that using MLP algorithm in the intrusion detection system. The known attacks finding accuracy rate is 92.63%. Whereas the unknowns accuracy is 72.57%. They stated that accuracy of the detection average is 83.11%.

Adem et al. (2016) examined market research and discount with MLP algorithm. They practiced 676 samples. That samples are separated into testing and validating. Test samples were 430 and validation was 123. The average accuracy they found was 96.97%.

Arslan, Mustafa & Kalinli (2016) examined that statistical algorithms and machine learning algorithms including MLP on microarray cancer data. They run algorithms on 34 sample, 857 genes and 2 classes (cancer, non-cancer). Finally, they obtained machine learning algorithms that are better than statistical algorithms. The highest accuracy over machine learning algorithms was MLP with a 97.06% ratio.

Voyant et al. (2017) stated that Levenberg-Marquardt (LM) algorithm is implemented with MLP for forecasting. The first stage of the implementation is started with randomly and then continued with probability distributions. They proposed Prunning MLP (pMLP) normalized Root Mean Squared Error (nRMSE) is better for early stages. However, continue of the implementation the results not changed considerably.

Portharaju, Prasad & Sreedevi (2018) stated that K-Means clustering algorithm used by MLP algorithms. All clusters are determined by Elbow method. They examined the minimum RMSE is provided by MLP is the best cluster is selected. Utilizing this, they proposed the minimize recollection waste.

Güçkıran et al. (2019) examined that DNA microarray gene expression and classification by Support Vector Machine (SVM), MLP and Random Forest (RF) algorithms. They compared the performance of each algorithm. The model they performed is given in Figure 3.10 is given. They used 2 hidden layers with ReLU and output layer with Softmax. They fit the model with SGD optimization with learning rate 0.005 and momentum as 0.9.

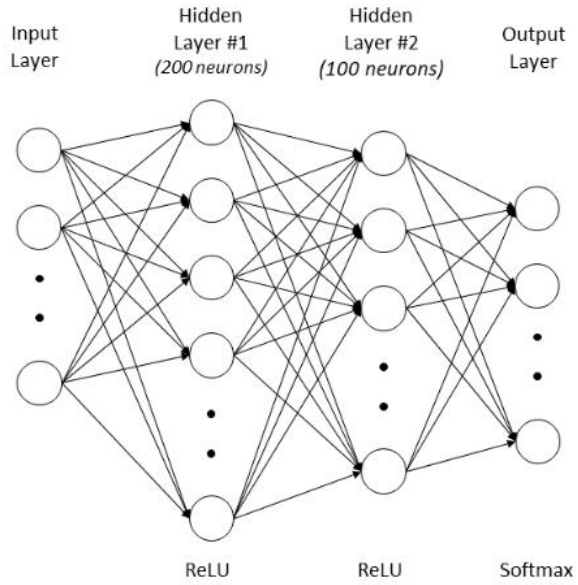


Figure 3.10 MLP model (Güçkıran et al., 2019)

3.4.2 Convolutional Neural Networks

Simard et al. (2003) stated that comparing three neural networks algorithms such as MLP, CNN and SVM. They applied algorithms on MNIST data. This data consists of English handwritten digit images.

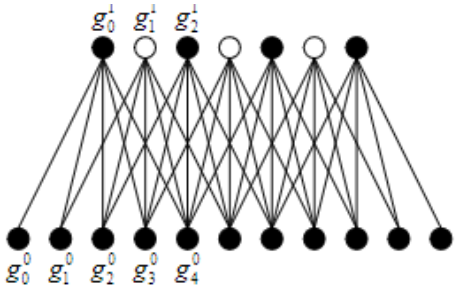


Figure 3.11 One layered CNN

Figure 3.11 shows the one layered CNN. Diverse numbers of outgoing links and incoming links are fixed numbers. Comparing the other algorithms, they attained the highest performance with CNN algorithm.

Kim (2014) analyzed that sentence level word classification using CNN. He generated a CNN model with max pooling and softmax activation functions. In Figure 3.12 the model is given. The inputs are the all words of the sentence.

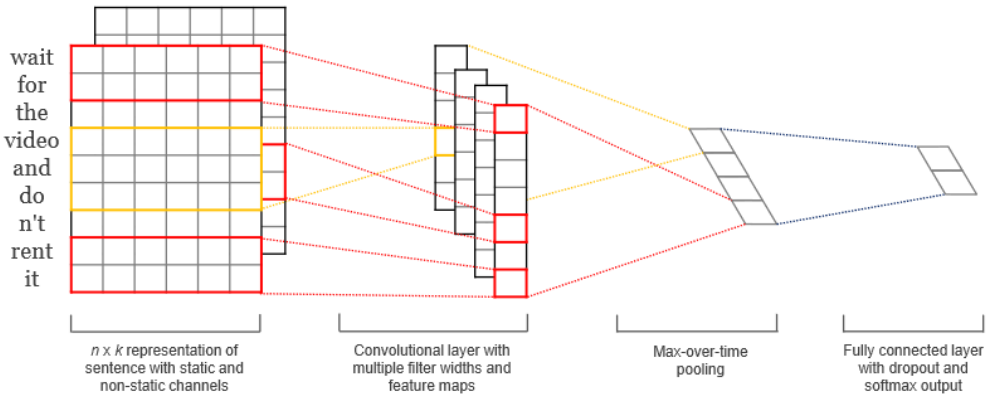


Figure 3.12 CNN model with two layer (Kim, 2014)

Parkhi et al. (2015) explored that face recognition with CNNs. They applied their algorithm on 2.6 million of face data. After comparing the other deep learning algorithms such as, DeepID3, DeepFace and Fisher Vector Faces, they found 97.3% accuracy rate. ROC is given in Figure 3.13.

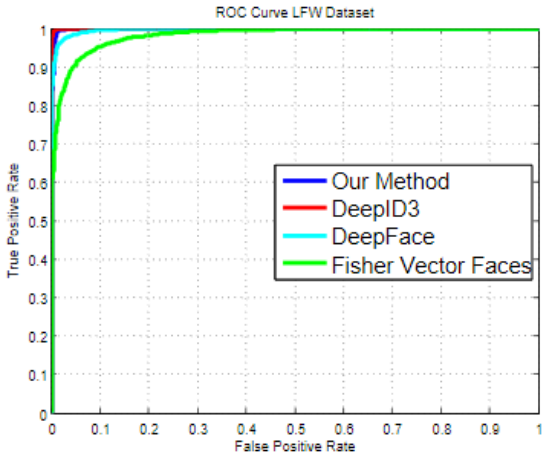


Figure 3.13 ROC curve of the algorithm (Parkhi et al., 2015)

The another method for face recognition is stated by Ensari (2017). He stated that classification algorithms applied with projected gradient descent nonnegative matrix factorization (NMF-PGD) in order to face recognition. The result of the study is the accuracy of the face recognition changed according to k-low value. This study shows a way of pattern recognition problems solving with NMF-PGD.

Schroff et al. (2015) explored that face recognition applied CNN. The algorithm named as FaceNet. They normalized the loss function. They stated that the most important part of the face recognition and verification is loss function.

Figure 3.14 Loss normalization

As in the Figure 3.14, Anchor, Negative and Positive variables are given. The distance between anchor, which is current image, positive is the nearest right target image and the negative is the nearest wrong target image.

$$\sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+ \qquad (3.3)$$

In Formula 3.3 they normalized the loss function. (x_i^a: Anchor image of a person. x_i^p: The nearest positive image of anchor. x_i^n: The nearest negative image.)

Aoki, Genta & Sakakibara (2018) examined that CNN classification non-coding RNA sequences pattern detection. They used both one-hot encoding and word2vec techniques. Word2vec technique first discovered by Mikolov et al. (2013). This technique is thinking every word as a vector. In Figure 3.15 word2vec model is given.

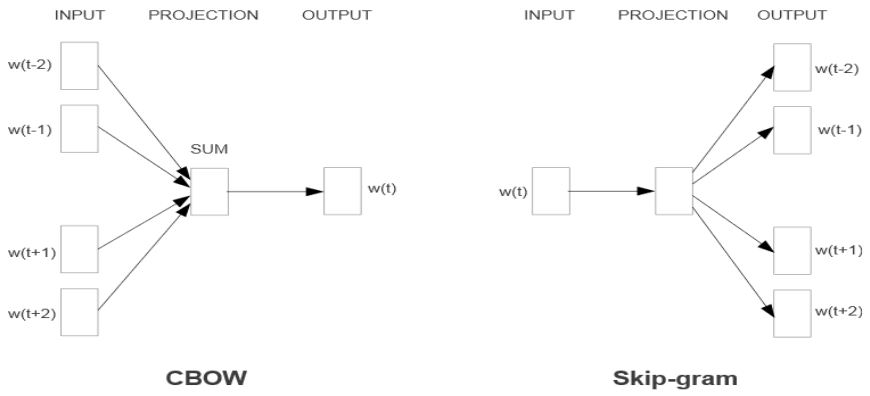


Figure 3.15 Compare of CBOW and Skip-gram (Mikolov et al., 2013)

In Figure 3.15 CBOW means Continues Bag of Words. This is word prediction from the given context. The skip-gram finds the all possible words of the given word. This technique is used in neural networks and machine learning algorithms.

RNA sequence pattern detection accuracy is calculated in Figure 3.16 below. (*TP*: True positive, *FP*: False Positive).

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

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

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

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

Figure 3.16 Accuracy, Precision, Recall and F-value (Aoki, Genta & Sakakibara, 2018)

The accuracy with CNN algorithm and one-hot encoding technique found 0.97. Whereas, the accuracy of CNN algorithm with word2vec found 0.98. The other way of this model is Natural Language Processing (NLP). Aktaş, Özlem & Çebi (2013) stated that, Rule Based Sentence Detection Method for Turkish sentences. Instead of CBOW, using RBSDM is another strong technique for classifying sentences. The success rate of this algorithm found 99.78%.

3.4.3 Recurrent Neural Networks and Bidirectional Long-short Term Memory (RNN and BLSTM)

Zhang et al. (2016) stated that used RNN-BLSTM for speech recognition area of multi-pitch estimation. They compared with DNN model with their model. According to RNN-BLSTM precision rate found 83.42% whereas precision rate found with DNN 73.55%.

Xue, Shaofei & Yan (2017) analyzed that online speech recognition using Latency Controlled-BLSTM. They proposed two new models for speech recognition. The Figure 3.17 shows LSTM network with memory cell. Since BLSTM is not enough for recognition that, wait for the whole sentence is completed. One of them is Forward Approximation Backward DNN Initialization. The other of them is Forward

Approximation Backward Simple RNN. They implement these models. As a result, they found 0.6% MAE. They increased the speed from 24% to 61%.

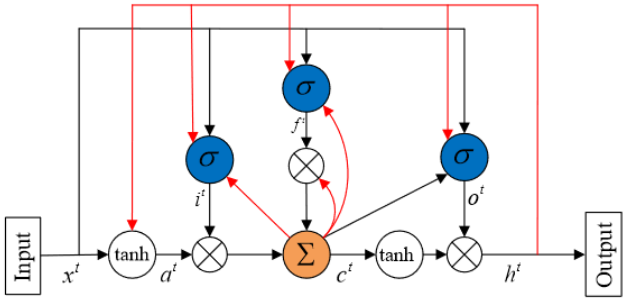


Figure 3.17 LSTM model

Mousa, Amr & Schuller (2017) examined that contextual BLSTM on sentiment analysis. They collect data set from IMDB. In Figure 3.18 BLSTM model is given. They found accuracy BLSTM model with binary classifier 90.15%. They found accuracy better with Contextual BLSTM and binary classifier than BLSTM, which is 92.83%.

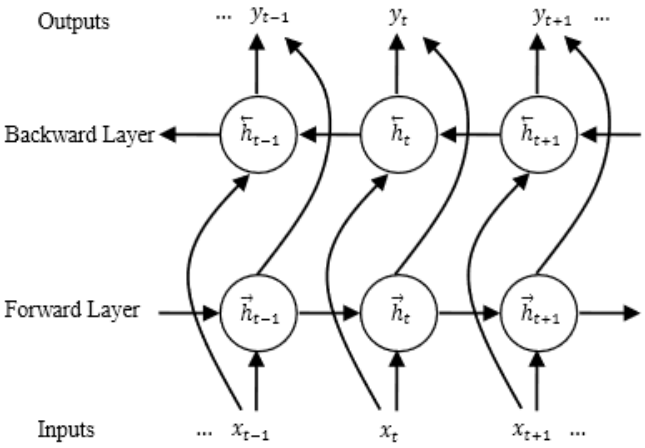


Figure 3.18 BLSTM model (Mousa, Amr & Schuller, 2017)

Yorulmuş et al. (2018) analyzed that forecasting electricity prices by means of using RNN and LSTM. They used electricity consumption and production values as data. The result of the prediction was in terms of MAE and RMSE metrics. They found 17.2TL MAE value.

Yin et al. (2018) stated that sentiment analysis applying on BLSTM and CNN fusion. In Figure 3.19 the architecture of the model is given. Data set consist of 2000 positive and 2000 negative texts. They used softmax activation function, binary crossentropy loss function. Dropout ratio set to 0.5. They found accuracy with only BLSTM was 85.8. However, they found BLSTM with CNN fusion model accuracy was 87.3%.

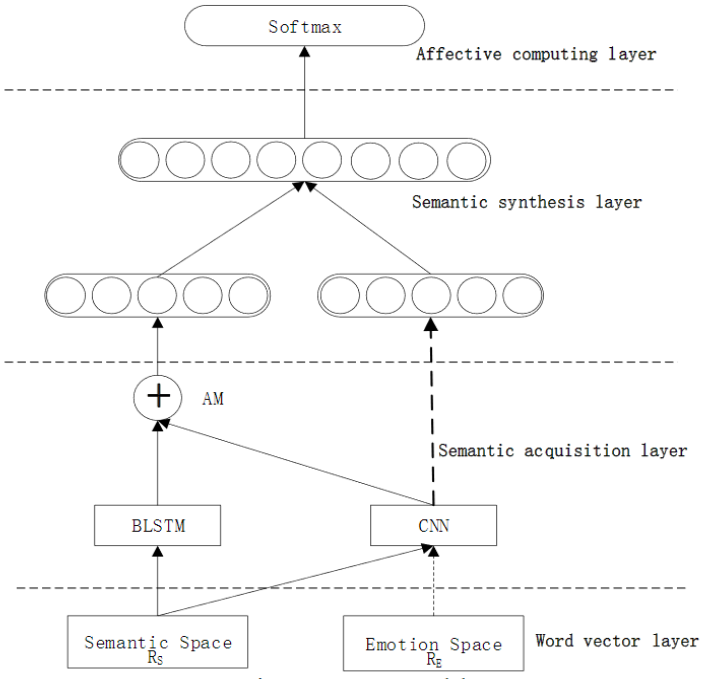


Figure 3.19 BLSTM with CNN fusion (Yin et al., 2018)

Süzen (2019) analyzed that predictig numbers of mathematical exam questions according to their subjects using LSTM. He used 931 questions of data. He divided data into 80% as train and 20% as test. MinMaxScalar normalization function and sigmoid activation function used. Learning rate set to 0.001, epoch number set to 100. The other algorithms such as Decision Tree, Logistic Regression, Poisson Regression, MLP are compared with CNN. The accuracy rate of algorithms are: Decision Tree is 73.20%, Logistic Regression is 89.54%, Poisson Regression is 82.60%, MLP is 86.82. The best result was obtained by CNN. The accuracy found with CNN is 98.42%.

Kızrak, Ayyüce & Bolat (2019) explored the predictive maintenance of aircraft engine health with LSTM. They used NASA Turbofan Engine Corruption

Simulation data set used. They used Adam optimization algorithm. They stated that Adam optimization is suitable for huge numbers of data. They compared the LSTM results with other neural network algorithms. According to Logistic Regression they found accuracy 92%, MAE 26, according to MLP accuracy 92.667%, MAE 17.139 and according to LSTM accuracy 96.8%, MAE 1.343. In Figure 3.20 shows loss rate decreasing for each epoch.

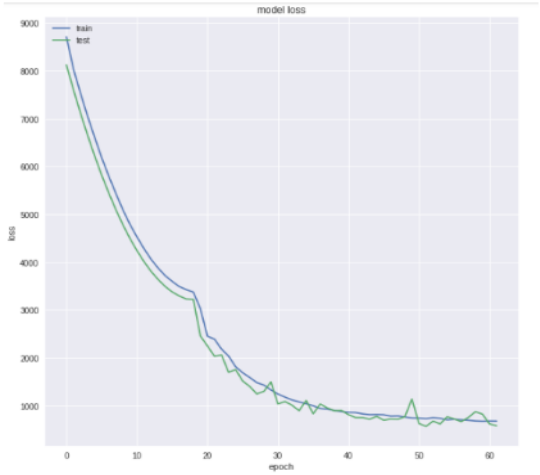


Figure 3.20 Loss function LSTM (Kızrak, Ayyüce & Bolat, 2019)

CHAPTER FOUR
CRISPR/CAS9 TARGET PREDICTION

4.1 CRISPR Data set

Applied *in silico* research and review, BLAST (Basic Local Alignment Search Tool) was used for CRISPR (Altschul et al., 1997). BLAST is a search tool that analyses the amino acids and DNA sequences of proteins and finds similarities between them. Besides BLAST, the data set resources have been used such as National Human Genome Research Institute (NHGRI) (Welter et al., 2014), miRBase (Griffiths et al., 2008), GenomeCrispr (Rauscher et al., 2017), CrisprInc (Cohen, 2017), ENSEMBL (Yates et al., 2016), ENCODE (Feingold et al., 2004), CRISPRz (Varshney et al., 2016), CRISPOR (Haeussler et al., 2016), CRISPR Local (Sun et al., 2018). In the algorithm studies performed with these data sets, estimation tools such as mirWalk (Dweep et al., 2011), TargetScan (ID2 PPI analysis network) (Shi et al., 2017), miRanda (Enright et al., 2003), mirBase (Griffiths-Jones et al., 2008), mirTarget (Ritchie et al., 2015), TarBase (Sethupathy et al., 2006) have been developed.

In this study, CRISPR Local data set has been used (Liu, 2018). The source of CRISPR Local data set is ENSEMBL Plants. There are approximately 854.610 lines of CRISPR data in the original. There are 11 column features in this data set which are examples of "Cyanidioschyzon merolae" alga. In Figure 4.1 sample data set from CRISPR-Local is shown.

CGTTGAAGTAGCCATCAGCG+AGG	0.7909	AT1G06430	1:+1962511	CGATGAAGCAGCCATCTGCA+CAG	4	AT1G01010.1.exon1;	0.0214	
TGATGGCTACTTCAACGTCG+CGG	0.7724	AT1G01740	1:+274664	CTTTGGCTACTTCAACATCG+CAG	4	AT1G01010.1.exon1;		0.0932
GTTGAGGTCAAGGACCAGTG+GGG	0.7487	AT1G51035	1:+18917578	GTTCAGTTCATGAACCAGTG+CAG	4	AT1G01010.1.exon2;	0.0223	
TTCACCGTGTTGGTGGATGG+AGG	0.7036	AT1G31080	1:+11092019	GTCACCGTCTTGGTGGATCC+CGG	4	AT1G01010.1.exon6;	0.0990	
GCTTACCGGAGAATCTGTTG+AGG	0.7031	AT1G04680	1:+1307049	GCTAACCGGAGAAACCGTTA+GAG	4	AT1G01010.1.exon2;	0.0478	
CAGAGAGCGAGAGAGATCGA+CGG	0.7005	AT1G30540	1:+10817050	AAGAGAGAGAGAGAGAGA+GAG	4	AT1G01010.1.exon1;		0.0107

Figure 4.1 CRISPR-Local sample data set (Liu, 2018)

In Figure 4.2 columns explanation is given.

```
The sequence of sgRNA(23nt).
The on-target score of the sgRNA.
The name of off-target gene with the highest CFD score.
The chromosome and the coordinate of the start position of the off-target site with the highest CFD score.
The sequence of off-target site.
The number of mismatches between sgRNA and off-target site.
The name of exon where the sgRNA located(split by ;).
The highest CFD score between sgRNA and all off-target sites.
```

Figure 4.2 CRISPR-Local columns (Liu, 2018)

This features; the gene in which the sgRNA, on target estimated chromosome and its coordinate, sgRNA sequence with 23'nt., on-target prediction score, off-target prediction gene which has the greatest CFD score, the chromosome on target prediction, its coordinate and beginning position, off-target prediction sequence, the number of sgRNA and mismatch on the off-target sequence, axon name, axon start position, all off-target and sgRNA having the highest CFD score.

	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	...	s15	s16	s17	s18	s19	s20	s21	s22	s23	onScore
0	2	4	4	1	1	4	4	8	8	2	...	8	2	8	4	4	4	2	2	2	0.617110
1	8	8	2	2	4	8	8	1	1	8	...	8	8	4	4	2	1	1	2	2	0.555640
2	1	2	4	4	1	1	4	4	8	8	...	1	8	2	8	4	4	4	2	2	0.552625
3	8	4	4	4	2	2	2	1	4	1	...	4	1	1	2	2	8	8	2	2	0.550106
4	1	4	2	8	8	4	4	4	2	2	...	8	1	1	2	4	1	1	2	2	0.511023
5	2	4	2	1	4	1	2	1	8	2	...	8	4	2	2	1	1	8	2	2	0.505702
6	8	1	8	2	8	4	4	4	2	2	...	2	8	4	1	4	4	4	2	2	0.489717
7	2	1	4	1	2	4	4	2	2	2	...	2	8	8	4	4	4	2	2	2	0.488317
8	4	2	1	4	1	2	4	4	2	2	...	4	2	8	8	4	4	4	2	2	0.456005
9	2	4	4	1	1	4	4	8	8	2	...	8	2	8	4	4	4	2	2	2	0.617110

Figure 4.3 Sample Data set

The sequences having 4 channels like Adenine (A), Cytosine (C), Guanine (G) and Thymine (T) are used as [1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1]. According these channels, the process has been realized with converted RNA sequence to binary system. These sequences were used as binary in the data set. In this example, each base in the sequence is considered as a separate column and feature. The 23nt. sgRNA sequence, on-target prediction score features were used.

Figure 4.3 shows an illustration of the sample data set. The binary representation of nucleotides converted into an integer number. This technique is called integer encoding.

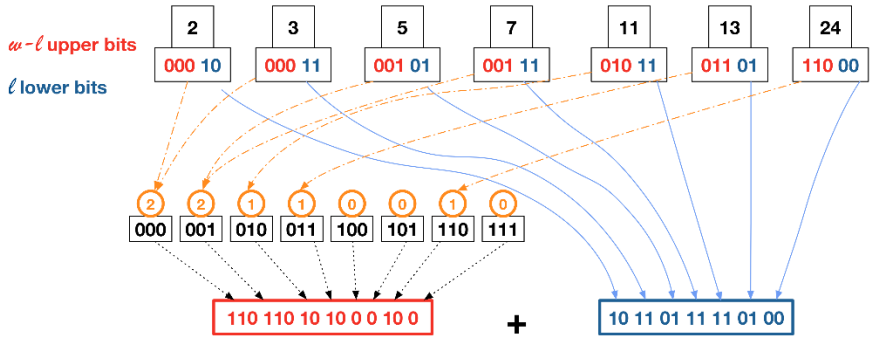


Figure 4.4 Integer encoding (Mailla et al., 2019)

This representation is given in Figure 4.4 by Mailla et al. 2019. Likewise this technique in this study, nucleotides represented as A (1000) is 8, C (0100) is 4, G (0010) is 2 and T (0001) is 1. Data set consist 34.200 lines of data.

4.1 Multi-Layer Perceptron (MLP)

In this study Multi layer perceptron-MLP, Convolutional Neural Networks-CNN and Bidirectional Long Short-Term-BLSTM algorithms were used. In the MLP model a fully connected structure of dense layers was formed. Information on the model used is shown in Figure 4.5.

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 2)	50
dense_10 (Dense)	(None, 50)	150
activation_5 (Activation)	(None, 50)	0
dropout_3 (Dropout)	(None, 50)	0
dense_11 (Dense)	(None, 64)	3264
activation_6 (Activation)	(None, 64)	0
dense_12 (Dense)	(None, 1)	65

Total params: 3,529
Trainable params: 3,529
Non-trainable params: 0

Figure 4.5 MLP Model

MLP model summary is given in Figure 4.6. The layers of the MLP model is represented with 4 dense layers. 2 of them are hidden layers. First one is input and last one is output layer. In the input layer, inputs are 24. 23's of the inputs are sgRNA sequences. Each nucleotide is described separately. The remaining 1 is CFD

score. Data set was separated, 40% as test 60% as train. In the middle of the model, softmax activation function is used. In the output layer, sigmoid activation function used. Model started with 24 input and ended with 1 output.

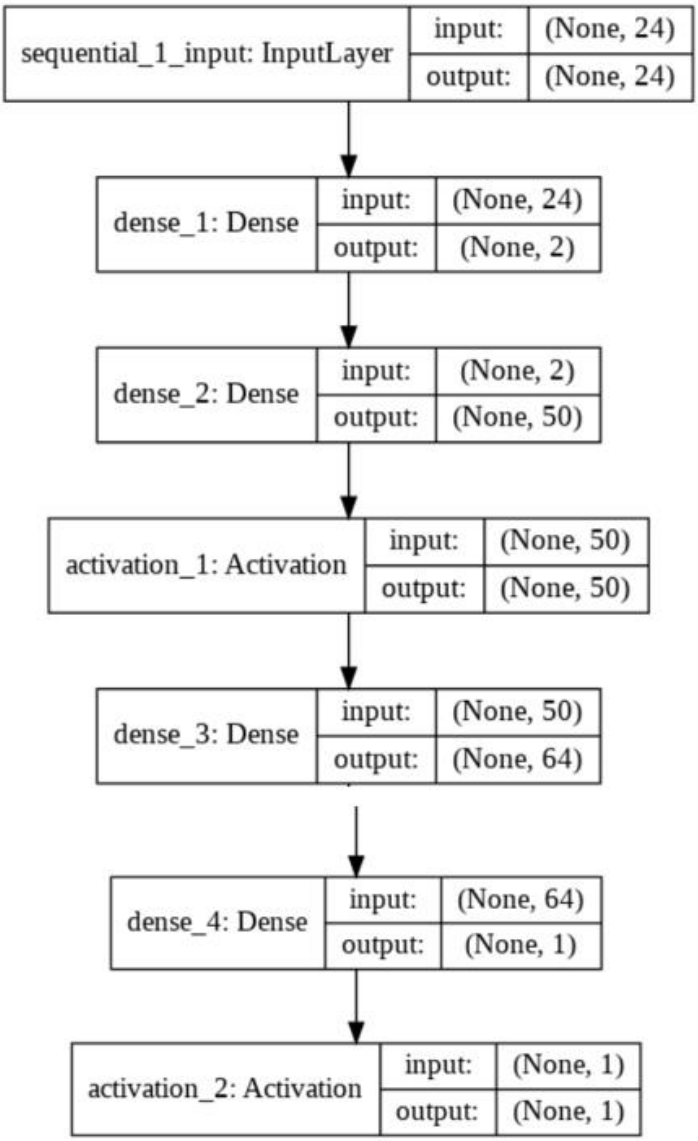


Figure 4.6 Model summary of MLP

In Figure 4.7 model is compiled with Stochasic Gradient Descent optimization method and binary crossentropy loss function is used. Learning rate of the optimization method is set to 0.0005.

```
sgd = SGD(lr=0.0005)
model.compile(loss='binary_crossentropy', optimizer=sgd, metrics=['accuracy'])
```

Figure 4.7 Model compile

The rates of logistic regression and accuracy according to the MLP model are as shown in Figure 4.8 and Figure 4.9. Accuracy was found 81.16% according to MLP model. Loss function value is 0.2468%.

```
Accuracy = 0.8116
```

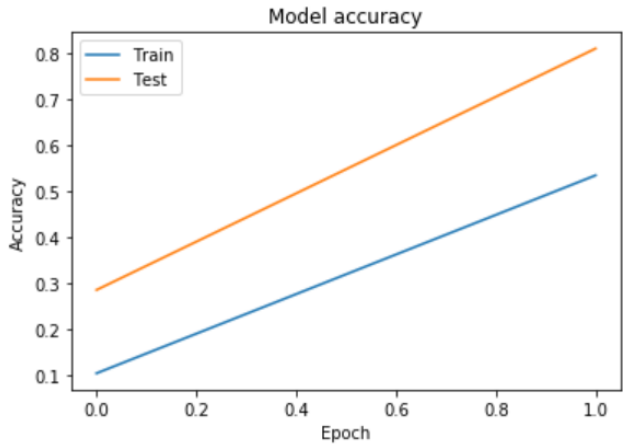


Figure 4.8 Model accuracy

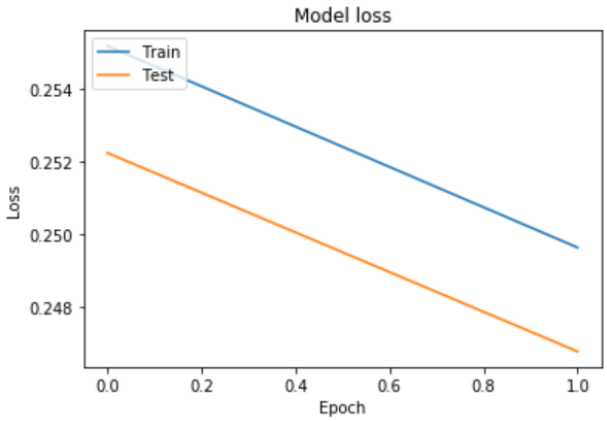


Figure 4.9 Model loss

As a result of MLP model accuracy, precision, recall, F1 measure metrics are represented in Figure 4.10.

```
Precision: 0.803821
Recall: 0.803874
Accuracy: 0.803874
F1 score: 0.891275
```

Figure 4.10 Metrics

4.2 Convolutional Neural Network (CNN)

Convolutional Neural Networks are a model of artificial neutral network which is used successfully in image processing, bioinformatics, robotics, data mining, finance and many other areas. However, except for image analysis, surprisingly high accuracy ratio was obtained in emotion analysis, text classification and question answering applications.

According to this model, it is applied to nxn matrix with nxn filtering method (dot product), with acceptation of n>m. Thus, it allows the identification and classification of properties. As shown in Figure 4.11 3x3 matrix as a result of the intrinsic product of a 5x5 matrix and filtering was obtained.

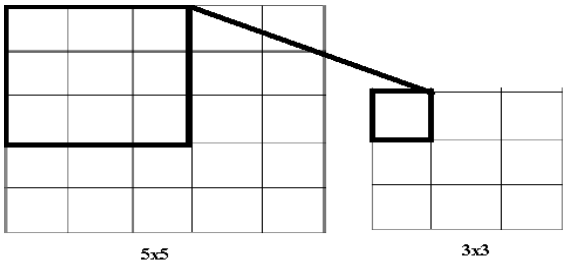
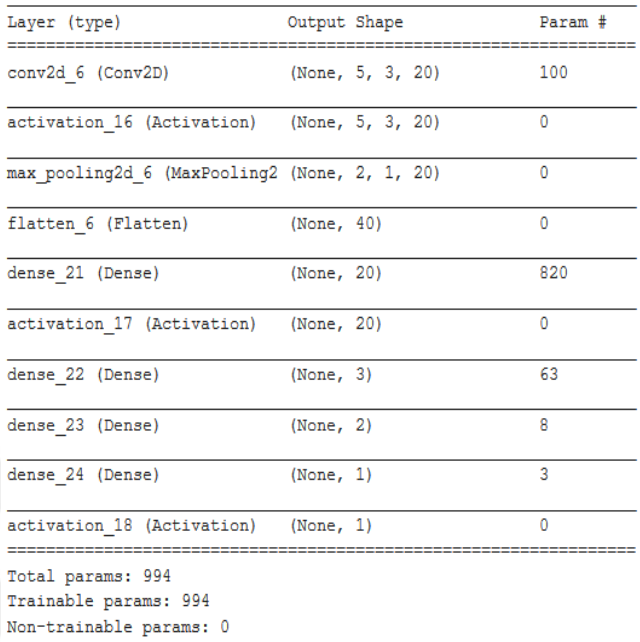


Figure 4.11 Convolution

In this study, data set was separated two group one of training 60%, the other test 40%. The model is being fixed up to non-linear by using the tangent and sigmoid activation functions. Convolution network is used to clarify the properties. The convolution network helps to create a new matrix with the results of the multiplication of the matrices. In order to prevent over fitting, maxpooling layer was used. It selects the elements with the maximum value from the matrix pool of the specified size in the maxpooling layer.

Accordingly, the information obtained when the CNN model is generated can be seen in Figure 4.12.



Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 5, 3, 20)	100
activation_16 (Activation)	(None, 5, 3, 20)	0
max_pooling2d_6 (MaxPooling2	(None, 2, 1, 20)	0
flatten_6 (Flatten)	(None, 40)	0
dense_21 (Dense)	(None, 20)	820
activation_17 (Activation)	(None, 20)	0
dense_22 (Dense)	(None, 3)	63
dense_23 (Dense)	(None, 2)	8
dense_24 (Dense)	(None, 1)	3
activation_18 (Activation)	(None, 1)	0

Total params: 994
Trainable params: 994
Non-trainable params: 0

Figure 4.12 CNN Model

CNN model summary is given in Figure 4.13 and Figure 4.14. The layers of the CNN model is represented with 7 dense layers. 5 of them are hidden layers. First one is input and last one is output layer. In the input layer, inputs are 24. 23's of the inputs are sgRNA sequences. Each nucleotide is described separately. The remaining 1 is CFD score. Data set was separated, 40% as test 60% as train. In the starting of the model, ReLU activation function is used. In the middle of the layer softmax activation function is used. In the output layer, sigmoid activation function used. Model started with 24 input and ended with 1 output.

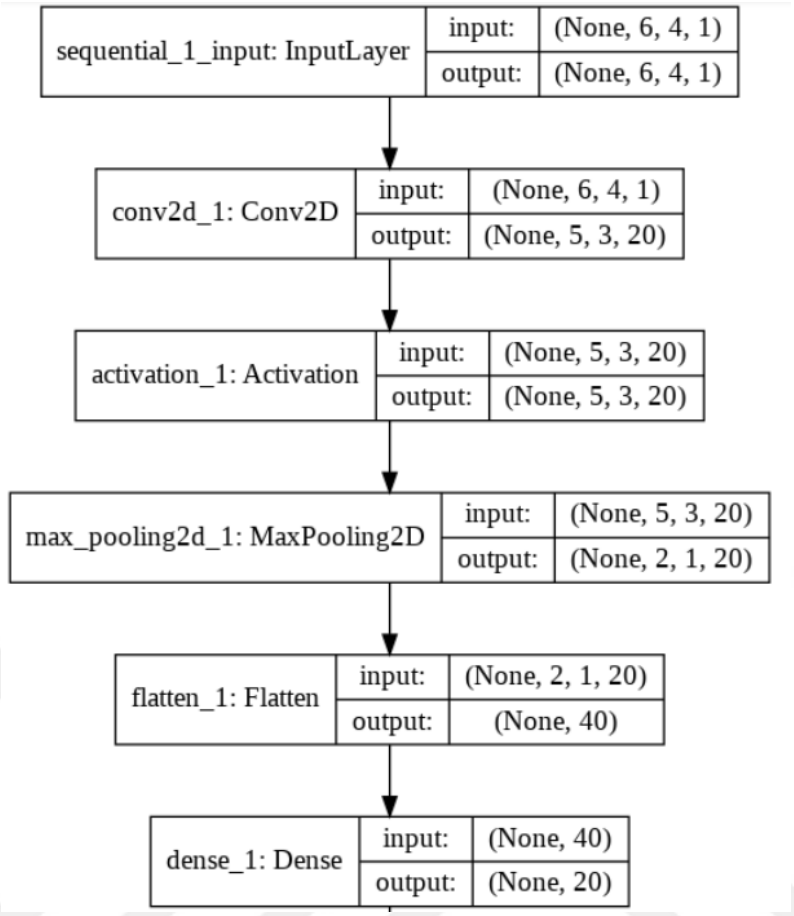


Figure 4.13 Model summary of CNN

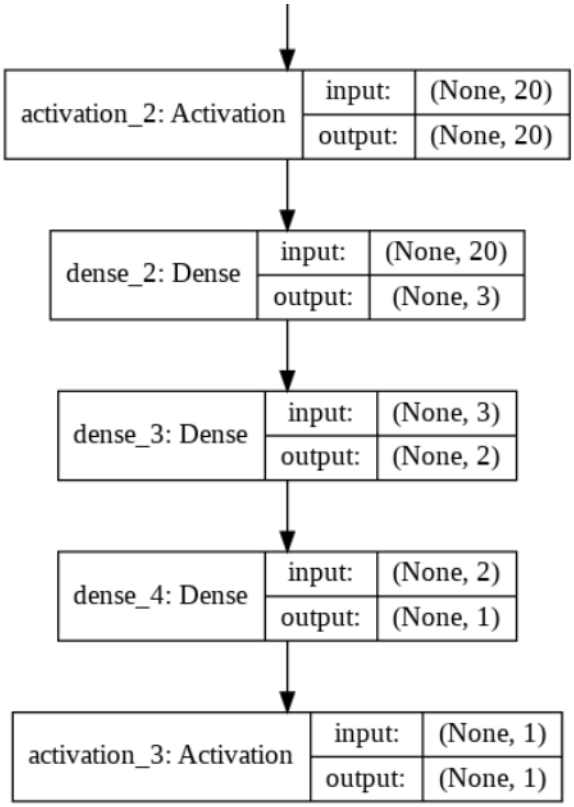


Figure 4.14 Model summary of CNN

The rate of loss and accuracy according to the CNN model is shown in Figure 4.15 and Figure 4.16. Loss function value is 0.0505 %.

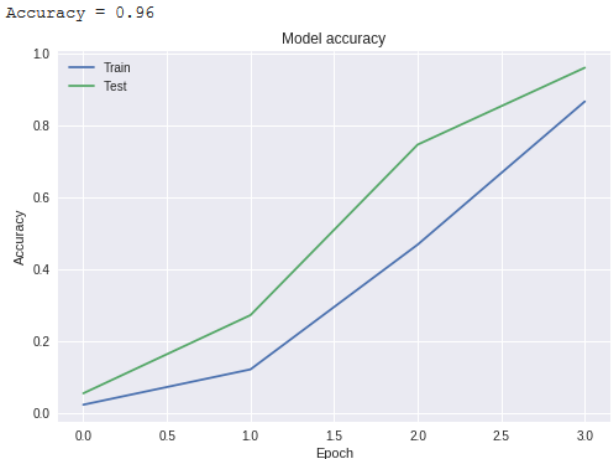


Figure 4.15 Model accuracy

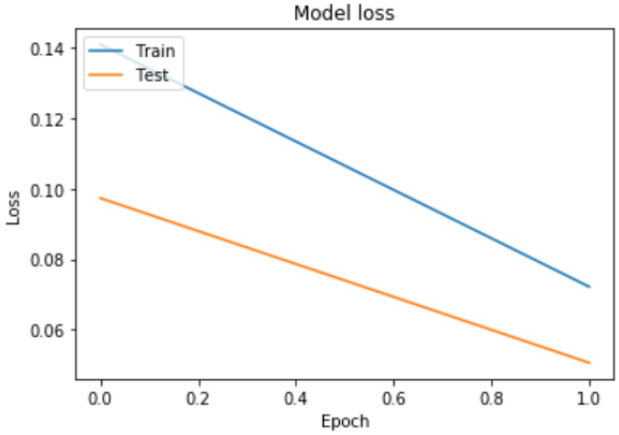


Figure 4.16 Model loss

As a result of CNN model accuracy, precision, recall, F1 measure metrics are represented in Figure 4.17.

```
Precision:  0.967909
Recall: 0.967909
Accuracy: 0.967909
F1 score: 0.983693
```

Figure 4.17 Metrics

4.2 Bidirectional Long-Short Term Memory (BLSTM)

Bidirectional LSTM is different from other feed forward models in neural networks, he has feedback system. Accordingly, the information obtained when the bidirectional LSTM model is generated can be seen in Figure 4.18.

Layer (type)	Output Shape	Param #
embedding_29 (Embedding)	(None, 24, 32)	3200
bidirectional_7 (Bidirection	(None, 200)	106400
dropout_16 (Dropout)	(None, 200)	0
dense_94 (Dense)	(None, 1)	201
activation_39 (Activation)	(None, 1)	0

Figure 4.18 BLSTM Model

BLSTM model summary is given in Figure 4.19. The layers of the BLSTM model is represented with 3 layers. 1 of them is hidden layer. First one is input with Embedding layer and last one is output with Dense layer. In the input layer, inputs are 24. 23's of the inputs are sgRNA sequences. Each nucleotide is described as a

separate feature. The remaining 1 is CFD score. Data set was separated 40% as test 60% as train. In the starting of the model, Embedding layer is used with 100 inputs. Embedding layers feature max review length is set to 24 and embedding vector length is set to 32. This layer let BLSTM layer to words description as a dense vector. In order to avoid overfitting (memorization of data) Dropout layer is used. In the output layer, sigmoid activation function used. Model started with 24 input and ended with 1 output.

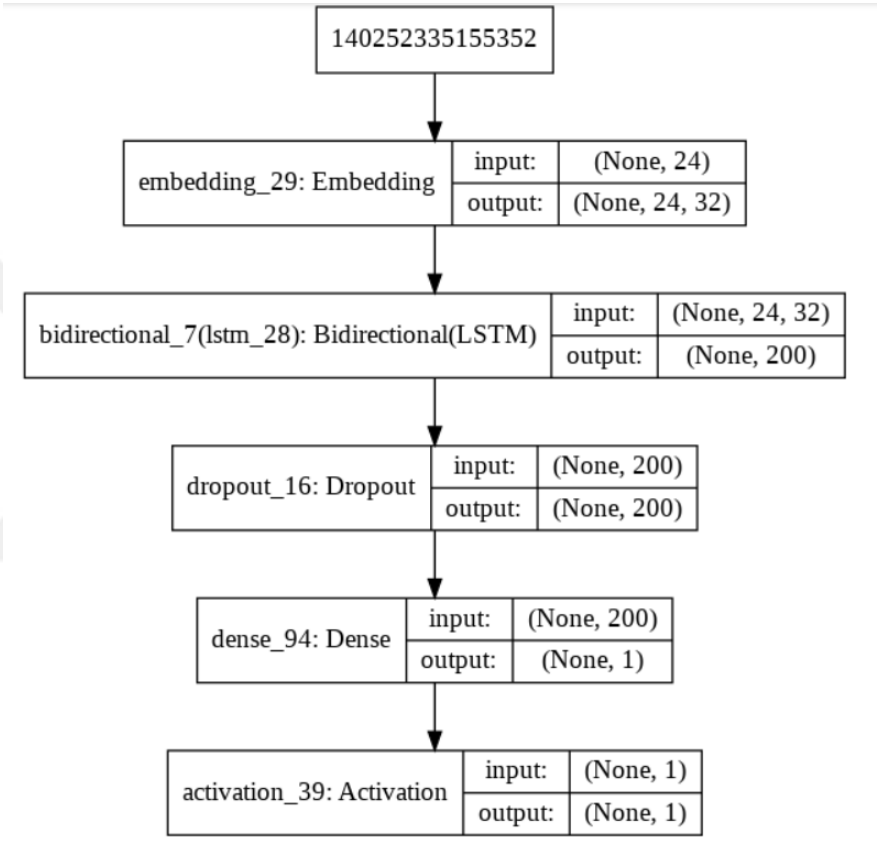


Figure 4.19 Model summary of BLSTM

Model compiled with binary crossentropy loss function and used Stochastic Gradient Descent optimization function. Optimization function learning rate was 0.00001. Figure 4.20 shows mode compilation.

```
model.add(Embedding(100, embedding_vecor_length, input_length=max_review_length))
model.add(Bidirectional(LSTM(100)))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))
sgd = SGD(lr=0.00001)
model.compile(loss='binary_crossentropy', optimizer=sgd, metrics=['accuracy'])
```

Figure 4.20 BLSTM Model compile

The rates of logistic regression and accuracy according to the BLSTM model are as shown in Figure 4.21. Accuracy was found 80.88% according to bidirectional LSTM model. Loss function value is 0.6918.

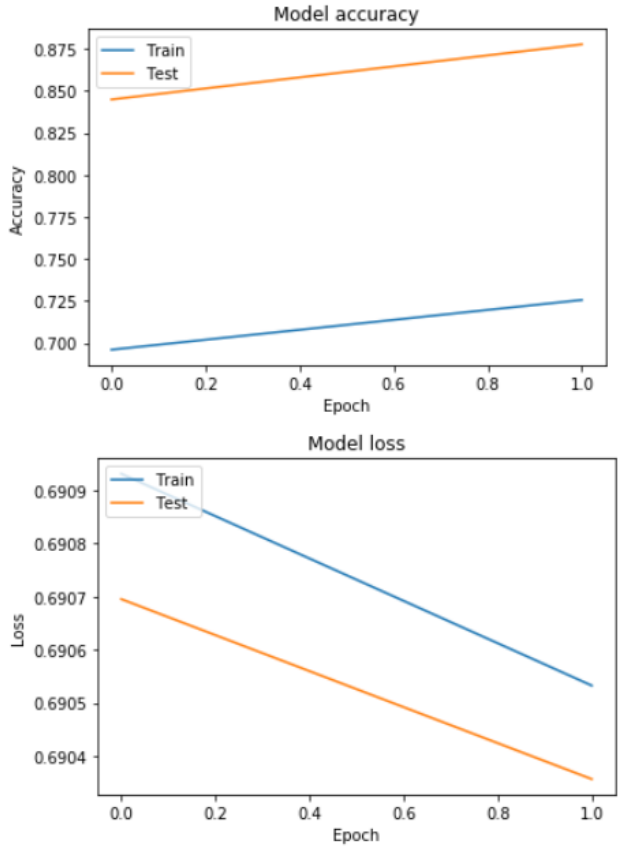


Figure 4.21 Model accuracy and model loss

As a result of BLSTM model accuracy, precision, recall, F1 measure metrics are represented in Figure 4.22.

```
Precision: 0.876168
Recall: 0.876170
Accuracy: 0.876170
F1 score: 0.933998
```

Figure 4.22 Metrics

Table 4.1 shows the MLP, CNN and bidirectional LSTM model accuracy rates.

Table 4.1 MLP, CNN and BLSTM results

	Convolutional Neural Network-CNN	Multilayer Perceptron-MLP	Bidirectional Long Short-Term - BLSTM
Accuracy	96.79%	80.38%	87.62%
Loss	0.0505	0.2468	0.6904
Precision	96.79 %	80.38 %	87.61 %
Recall	96.79%	80.38%	87.61%
F1 score	98.36%	89.12%	93.39%

According to Zhu & Liang (2018) explored that CRISPR-CPF1 sgRNA targeting using machine learning algorithms. The source of study was Ensembl Plants. They used SVM algorithm. According to their result the accuracy rate was 87% they found.

Table 4.2 Comparing with other studies

	Zhu & Liang (2018) (SVM)	Our research (CNN)
Accuracy	87%	96.7%

Table 4.2 shows us comparing results of this study and the last studies.

CHAPTER FIVE
CONCLUSION

5.1 Results and Evaluation

In this study, the algorithms of Multilayer Perceptron-MLP, Convolutional Neural Networks-CNN and Bidirectional Long Short-Term Memory-BLSTM have been compared with use of CRISPR data set. As a result, according to this data set, the accuracy rate in the MLP model was 81.12% and Bidirectional LSTM model was 80.88% whereas for CNN this result was found to be 96%. According to the results, a higher accuracy rate was obtained with the CNN model than MLP and BLSTM. In the CNN model, revised CRISPR has reached up to 7 layers according to the ENSEMBL Plants data set and 4 layers have been formed in MLP and 2 layers have been formed in BSLTM.

Comparing other algorithms with our algorithm is better performance according to results. Research that used with SVM algorithm performed 87% accuracy result. However, our model achieved 96.7%. This result is more reliable performance than the research used SVM algorithm. Any mistargeted position causes unwanted genome distortions. For this reason, the accuracy rate is urgent in sgRNA targeting.

5.2 Future Enhancement

This study represented a new way of sgRNA targeting. Cyanidioschyzon merolae alga nucleotides and CFD scores are run with deep learning algorithms. Three algorithms are compared with each other. The most successful result is provided utilizing applying CNN algorithm. This study proposes a way of deciding between deep learning algorithms by comparing the performances of the algorithms.

This research tends to offer an insight into other future analyses according to sgRNA targeting in the CRISPR-CAS9 system. In this study CFD sequence

predicting with deep learning algorithms. In the data set each sgRNA sequence has CFD scores.

Integer encoding for DNA sequence is used in this study. FASTA formatting is a technique for DNA sequencing and one-hot encoding is another technique for DNA sequencing. Most of the studies used one-hot encoding and FASTA format for the whole of the DNA sequence. However, in this research, we did not need an entire genome sequence. We only need for sgRNA sequence encoding which is 23bp.

One hot encoding is used by each nucleotide as a vector. They converted into an array. However, each nucleotide is considered separate features of inputs in this study. Utilizing this study integer encoding is also used and researched. Integer encoding is likewise one-hot encoding. Integer encoding is a bit differ from one-hot encoding This study represented a way of DNA sequencing encoding technique which is integer encoding.

REFERENCES

Abadi, S., Yan, W., X., Amar, D. & Mayrose, I. (2017). A machine learning approach for predicting CRISPR-Cas9 cleavage efficiencies and patterns underlying its mechanism of action. *PLoS Computational Biology, 13,* 1-4.

Adem, K., Zengin, N., Hekim, M. & Karaca, S. (2016). Prediction of the relationship between the bist 100 index and advanced stock market indices using artificial neural network. *Journal of New Theory, 13,* 86-95.

Aktaş, Ö. & Çebi, Y. (2013). Rule-based sentence detection method (RBSM) for Turkish. *International Journal of Language and Linguistics, 1,* 1-6.

Altschul, S., F., Madden, T., L., Schäffer, A., A., Zhang, J., Zhang, Z. & Miller, W., et al. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research, 25,* 3389-3402.

Aoki, G. & Sakakibara, Y. (2018). Convolutional neural networks for classification of alignments of non-coding RNA sequences. *Bioinformatics, 34,* 237–244.

Arslan, M., T. & Kalinli, A. (2016). A Comparative Study of Statistical and Artificial Intelligence based Classification Algorithms on Central Nervous System Cancer Microarray Gene Expression Data. *International Journal of Intelligent Systems and Applications in Engineering, 4,* 78-81.

Bolser D., Staines D.M., Pritchard E. & Kersey P. (2016) *Ensembl Plants: Integrating Tools for Visualizing, Mining, and Analyzing Plant Genomics Data. In: Edwards D. (eds) Plant Bioinformatics*. Methods in Molecular Biology, *1374.* New York: Humana Press.

Chuai, G., Ma, H., Yan, J., Chen, M., Hong, N. & Xue, D. (2018). DeepCRISPR: optimized CRISPR guide RNA design by deep learning. *Genome Biology, 19,* 1474-1489.

Cohen, J. (2017). The birth of CRISPR inc, *Science*, *355*, 680-684.

Damasevicius, R. (2008). Analysis of binary feature mapping rules for promoter recognition in imbalanced DNA sequence datasets using Support Vector Machine. *4th International IEEE Conference Intelligent Systems, 3*, 11-20.

Doench, J., G., Fusi, N., Sullender, M., Hedge, M., Vaimberg, E., W. & Donovan, K., F. (2016). Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. *Nature Biotechnology*, *34*, 184-191.

Dunne, R., A. & Campbell, N., A. (1997). On The Pairing Of The Softmax Activation And Cross-Entropy Penalty Functions And The Derivation Of The Softmax Activation Function. *In Proceedings of the Australian Conference on Neural Networks*, *1*, 181-185.

Dweep, H., Sticht, C., Pandey, P. & Gretz, N. (2011). MiRWalk - Database: Prediction of possible miRNA binding sites by "walking" the genes of three genomes. *Journal of Biomedical Informatics*, *44*, 838-841.

Enright, A. J., John, B., Gaul, U., Tuschl, T., Sander, C. & Marks, D. S. (2003). MicroRNA targets in Drosophila. *Genome Biology*, *5*, 101-110.

Ensari, T. (2017). Pattern recognition from face images. *Journal of Naval Sciences and Engineering*, *13*, 14-20.

Feingold, E. A., Good, P. J., Guyer, M. S., Kamholz, S., Liefer, L. & Wetterstrand, K. (2004). The ENCODE (ENCyclopedia of DNA elements) project. *Science*, *306*, 636-640.

Frock, R., L., Hu, J., Meyers, R., M., Ho, Y., Kii, E. & Alt, F., W. (2014). Genome-wide detection of DNA double-stranded breaks induced by engineered nucleases. *Nature Biotechnology, 33*, 179–186.

Griffiths-Jones, S., Saini, H. K., van Dongen, S. & Enright, A. J. (2008). miRBase: tools for microRNA genomics. *Nucleic Acids Research, 36*, 154–158.

Güçkıran, K., Cantürk, İ. & Özyılmaz, L. (2019). DNA microarray gene expression data classification using SVM, MLP, and RF with feature selection methods relief and LASSO. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 23*, 126-132.

Haeussler, M., Schönig, K., Eckert, H., Eschstruth, A., Mianné, J. & Renaud, J., B. (2016). Evaluation of off-target and on-target scoring algorithms and integration into the guide RNA selection tool CRISPOR. *Genome Biology, 17*, 138-148.

Hahnloser, R., H., R., Sapeshkar, R., Mahowald, M., A., Douglas, R., J. & Seung, H., S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature, 405*, 947–951.

Hapudeniya, M. (2010). Artificial neural networks in bioinformatics. *Sri Lanka Journal of Bio-Medical Informatics, 1*, 104–111.

Hirosawa, M., Fujita, Y., Parr, C., J., C., Hayashi, K., Kashida, S. & Hotta, A., et al. (2017). Cell-type-specific genome editing with a microRNA-responsive CRISPR-Cas9 switch. *Nucleic Acids Research, 45*, 13-40.

Karpathy, A., (2019). *Convolutional neural networks (CNNs/ConvNets)*. Retrieved September 17, 2019 from http://cs231n.github.io/convolutional-networks/

Kızrak, M. & Bolat, B. (2019). Predictive maintenance of aircraft motor health with Long-Short Term Memory method. *Bilişim Teknolojileri Dergisi*, *12*, 103-109.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, *4*, 65-87.

Kleinstiver, B., P., Pattanayak, V., Prew M., S., Tsai, S., Q., Nguyen, N., T. & Zheng, Z., et al. (2016). High-fidelity CRISPR–Cas9 nucleases with no detectable genome-wide off-target effects. *Nature Research*, *529*, 490–495.

Kökver, Y., Barışçı, N., Çiftçi, A. & Ekmekçi, Y. (2014). Data mining classification on hypertension database. *E-Journal of New World Sciences Academy*, *9*, 15-25.

Lin, J. & Wong K., C. (2018). Off-target predictions in CRISPR-Cas9 gene editing using deep learning. *Bioinformatics*, *34*, 656–663.

Lipman, D., J. & Pearson, W., R. (1985). Rapid and sensitive protein similarity searches. *Science*, *227*, 1435–1441.

Listgarten, J. (2017). *Machine-learning-based CRISPR guide design; models, inference and algorithms meeting; Broad Institute.* Retrieved September 17, 2019 from https://www.broadinstitute.org/videos/machine-learning-based-crispr-guide-design

Listgarten, J., Weinstein, M., Kleinstiver, B., P., Sousa, A., A., Joung, J., K. & Crawford, J., et al. (2018). Prediction of off-target activities for the end-to-end design of CRISPR guide RNAs. *Nature Biomedical Engineering*, *2*, 38-47.

Liu, H. (2018). CRISPR-Local: a local single-guide RNA (sgRNA) design tool for nonreference plant genomes. *Bioinformatics*, *35*, 105-128.

Liu, Y., Tao, W., Wen, S., Li, Z., Yang, A. & Deng, Z., et al. (2015). In vitro CRISPR/Cas9 system for efficient targeted DNA editing. *mBio*, *6*, 1-8.

Mallia, A., Siedlaczek, M., Suel, T. & Zahran, M. (2019). GPU-accelerated decoding of integer lists. *In Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, *5*, 152-180.

Marttin, V. & İmal, N. (2015). Using neural network, in computer networks intrusion detection system and study of achievement. *Gaziosmanpaşa Journal of Scientific Research,11*, 21-40.

Medcalc, (2019). *TANH function*. Retrieved September 17, 2019 from https://www.medcalc.org/manual/tanh_function.php

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient estimation of word representations in vector space. *In Proceedings of Workshop at International Conference on Learning Representations*, *2*, 1-12.

Miller, J., C., Tan, S., Qiao, G., Barlow, K., A., Wang, J. & Xia, F., D., et al. (2011). A TALE nuclease architecture for efficient genome editing. *Nature Biotechnology*, *29*, 143–148.

Montague, T. G., Cruz, J. M., Gagnon, J. A., Church, G. M. & Valen, E. (2014). CHOPCHOP: a CRISPR/Cas9 and TALEN web tool for genome editing. *Nucleic Acids Research*, *42*, 401–407.

Mousa, A. & Schuller, B. (2017). Contextual bidirectional long short-term memory recurrent neural network language models: A Generative Approach to Sentiment Analysis. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, *1*, 1023–1032.

Parkhi, O.M., Vedaldi, A. & Zisserman, A. (2015). Deep face recognition. *In Proceedings of the British Machine Vision Conference, 41*, 1-12.

Parmar, R. (2018). *Common loss functions in machine learning.* Retrieved September 2, 2018 from https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V. & Thirion, B. (2011). Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825-2830.

Plumer, B., Barclay, E., Belluz, J., Irfan, U. & Zarracina, J. (2018). *A simple guide to CRISPR, one of the biggest science stories of the decade*. Retrieved December 27, 2018 from https://www.vox.com/2018/7/23/17594864/crispr-cas9-gene-editing

Potharaju, S., P. & Marriboyina, S. (2018). An unsupervised approach for selection of candidate feature set using filter based techniques. *Gazi University Journal of Science*, *31*, 789-799.

Puppo, F., George, V. & Silva, G., A. (2018). An optimized structure-function design principle underlies efficient signaling dynamics in neurons. *Scientific Reports*, *8*, 104-109.

Ran, F., A., Cong, L., Yan, W., X., Scott, D., A., Gootenberg, J., S. & Kriz, A., J., et al. (2015). In vivo genome editing using Staphylococcus aureus Cas9. *Nature*, *5* 510- 520.

Rauscher, B., Heigwer, F., Breinig, M., Winter, J. & Boutros, M. (2017). GenomeCRISPR - a database for high-throughput CRISPR/Cas9 screens. *Nucleic Acids Research, 8*, 103-112.

Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W. & Shi, W., et al. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research, 43*, 47-58.

Sainath, T., Mohamed, A., Kingsbury, B. & Ramabhadran, B. (2013). Deep convolutional neural networks for LVCSR. *IBM Thomas Watson Research Center, 5*, 204-260.

Salman, H., M., A. (2018). Effect of successive convolution layers to detect gender. *Iraqi Journal of Science, 59*, 1717-1732.

Schroff, F., Kalenichenko, D. & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition, 9*, 815-823.

Sethupathy, P., Corda, B. & Hatzigeorgiou, A. G. (2006). TarBase: A comprehensive database of experimentally supported animal microRNA targets. *RNA, 12*, 192–197.

Sharma, A. (2017). *Understanding activation functions in neural networks.* Retrieved March 30, 2017 from https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0

Sharma, S. (2017). *Activation functions in neural networks.* Retrieved September 6, 2017 from https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Shi, Y., Yang, F., Wei, S. & Xu, G. (2017). Identification of key genes affecting results of hyperthermia in osteosarcoma based on integrative CHIP-SEQ/Targetscan analysis. *Medical Science Monitor, 23*, 2042-2048.

Sibi, P., Jones, S., A. & Siddarth, P. (2013). Analysis of different activation functions using back propagation neural networks. *Journal of Theoretical and Applied Information Technology*, *47*, 3-12.

Simard, P., Y., Steinkraus, D. & Platt, J., C. (2003). Best practices for convolutional neural networks applied to visual document analysis. *Seventh International Conference on Document Analysis and Recognition. Proceedings*, *4*, 958-963.

Sualp, M. & Can, T. (2011). Using network context as a filter for miRNA target prediction. *Biosystems*, *105*, 201-209.

Subramanian, V. (2017). *Deep learning at scale, accurate, large mini batch SGD.* Retrieved August 9, 2017 from https://towardsdatascience.com/deep-learning-at-scale-accurate-large-mini-batch-sgd-8207d54bfe02

Sung J., H., Wang Y. & Shuler M., L. (2019). Strategies for using mathematical modeling approaches to design and interpret multi-organ microphysiological systems (MPS) featured. *APL Bioengineering*, *3*, 420-455.

Süzen, A. A. (2019). Estimation of mathematics question numbers in the university entrance exam by lstm deep neural networks. *Engineering Sciences*, *14*, 112–120.

Svozil, D., Kvasnicka, V. & Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems, 39*, 43-62.

Tahir, M., Tayara, H. & Chong, K., T. (2019). iRNA-PseKNC(2methyl): Identify RNA 2'-O-methylation sites by convolution neural network and Chou's pseudo components. *Journal of Theoretical Biology*, *465*, 1-6.

Tekbulut, M., T. (2006). *MicroRNA target prediction by constraint programming*, Master Thesis, Sabancı University, İstanbul.

Timuş, O. & Kıyak, E. (2016). Optimizing MLP classifier and ECG features for sleep apnea detection. *Journal of Naval Sciences and Engineering, 11,* 1-18.

Ugurlu, U., Taş, O. & Yorulmus, H. (2018). A long short term memory application on the Turkish intraday electricity price forecasting. *Pressacademia, 7,* 126-130.

Varshney, G. K., Zhang, S., Pei, W., Adomako-Ankomah, A., Fohtung, J. & Schaffer, K. (2016). CRISPRz: a database of zebrafish validated sgRNAs. *Nucleic Acids Research, 44,* 822–826.

Voyant, C., Paoli, C., Nivet, M., L., Notton, G., Fouilloy, A. & Motte, F. (2017). Multi-layer perceptron and pruning. *Turkish Journal of Physics, 1,* 1-6.

Wang, L., Yang, Y., White, J., Deirdre, M., Bell, P. & Wilson, J., M. (2016). Crispr/Cas9-Mediated ın vivo gene targeting corrects haemostasis in newborn and adult FIX-KO mice. *Blood, 128,* 1168-1174.

Welter, D., MacArthur, J., Morales, J., Burdett, T., Hall, P. & Junkins, H., et al. (2014). The NHGRI GWAS catalog, a curated resource of SNP-trait associations. *Nucleic Acids Research, 42,* 1001–1006.

Winter, J., M., Gildea, D., E., Andreas, J., P., Thibodeau, S., N., Churchill, G., A. & Crawford, N., P., S. (2017). Mapping complex traits in a diversity outbred f1 mouse population identifies germline modifiers of metastasis in human prostate cancer. *Cell Systems, 4,* 31-45.

Xue, S. & Yan, Z. (2017). Improving latency-controlled BLSTM acoustic models for online speech recognition. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, 8,* 5340-5344.

Yates, A., Akanni, W., Amode, M., R., Barrell, D., Billis, K. & Carvalho-Silva, D. (2016). Ensembl 2016. *Nucleic Acids Research, 44*, 710–716.

Yin, L., Ye, X. & Yao, J. (2018). A sentiment analysis method based on BLSTM and CNN fusion. *Journal of Physics, Conference Series, 5*, 1087-1095.

Zhang, J., Tang, J. & Dai, L. (2016). RNN-BLSTM based multi-pitch estimation, *Proceedings Interspeech 2016, 10*, 1785-1789.

Zhang, Z., Lyons, M., Schuster, M. & Akamatsu, S. (1998). Comparison between geometry-based and Gabor wavelets-based facial expression recognition using multi-layer perceptron. *In Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition, 4*, 454-459.

Zhu, H. & Liang, C.(2019) CRISPR-DT: designing gRNAs for the CRISPR-Cpf1 system with improved target efficiency and specificity. *Bioinformatics, 35*, 2783–2789.

APPENDICES

APPENDIX 1:

Table A.1 Activation functions

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$