DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MACHINE LEARNING FOR PREDICTING PATIENT WAITING TIMES

by

Hamed JAVADIFARD

September, 2019

İZMİR

MACHINE LEARNING FOR PREDICTING PATIENT WAITING TIMES

A Thesis Submitted to the

Graduate School of Natural and Applied Sciences of Dokuz Eylül University

In Partial Fulfillment of the Requirements for the Master of Science in Computer Engineering

by

Hamed JAVADIFARD

September, 2019

İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**MACHINE LEARNING FOR PREDICTING PATIENT WAITING TIMES**" completed by **HAMED JAVADIFARD** under the supervision of **PROF. DR. SÜLEYMAN SEVİNÇ** 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. Süleyman SEVİNÇ

Supervisor

Asst. Prof. Dr. Özlem AKTAŞ

(Jury Member)

Prof. DR. SERDAR KORUKOĞLU

(Jury Member)

Prof. Dr. Kadriye ERTEKİN

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

This thesis work would have not been possible without the help of so many people in so many ways. I would first like to thank my thesis supervisor Prof. Dr. Süleyman Sevinç for his invaluable patience, support, guidance and encouragement he managed to extend to me. Professor Sevinç was always available whenever I need help or had a question about my research or writing. He allowed me to write my own paper, but he always kept me on the right course. I am extremely grateful for the time and effort he spent with me on this work and for the helpful advice and guidance he provided. Furthermore, throughout the writing of this thesis I have received a great deal of support and assistance from other entities. In view of this fact, I would like to thank Mr. Oktay Yıldırım of Labenko Bilişim A.Ş. for introducing me to the topic as well for the support on the way. I would like to extend my thanks for your cooperation and for all of the opportunities and support you have granted to me to conduct my research and further my thesis.

My special thanks and gratitude goes to my parents and my brother for endearing conscience and support throughout my study and enabled me to come so far. I wish to express my love and gratitude to all my friends and colleagues working at Labenko Bilişim A.Ş. & Computational Medicine Group and to all my friends and class mates for their direct or indirect support in every aspect for this thesis.

Last but not least, I would like to thank everyone who has positively impacted the realization of this thesis work, as well as putting forth my apology for not being able to mention every individual one by one.

Hamed JAVADIFARD

MACHINE LEARNING FOR PREDICTING PATIENT WAITING TIMES

ABSTRACT

Accurately predicting waiting times can increase patient satisfaction and enable staff members to more accurately evaluate and respond to patient flow. In this study, we examined the applicability of the machine learning model to estimate waiting times in a blood collection unit. The blood collection units of hospitals are the places where the patient density is the highest. Because the patients who go to different outpatient clinics of the hospital come to the blood collection units to perform an examination and give blood or different tests. In this study, the patient records when he comes to the blood collection unit via a kiosk (philerobo artificial intelligence blood collection unit management system). Then they wait in the waiting room and the system calls the patient when the time comes. In our study, our model estimates the waiting time of artificial Neural Networks algorithm by considering some parameters (Arrival Time, Day of Week, Phlebotomists Count, Current Patient Duration, Waiting Patient Count with priority, Waiting Patient Count with the Same Priority, Incoming Patient Count, Outgoing Patient Count and patient priority) how long it will wait inside when the patient enters the kiosk. As a result of this, we aimed to predict how long the next patient will wait in the waiting room and to inform the patient.

Keywords: Waiting time, phlebotomy, artificial neural network, machine learning, prediction

MAKİNA ÖĞRENİMİ İLE HASTA BEKLEME SÜRELERİNİN TAHMİNİ

ÖZ

Bekleme sürelerini doğru bir şekilde tahmin edebilmek, hasta memnuniyetini artırabilir ve personel üyelerinin hasta akışını daha doğru bir şekilde değerlendirmesini ve yanıt vermesini sağlayabilir. Bu çalışmada, bir kan alma biriminde bekleme sürelerini tahmin etmek için makine öğrenimi modelini uygulanabilirliğini inceledik. Hastanelerin kan alma birimleri hasta yoğunluğunun en çok olduğu yerdir. Çünkü hastanenin farklı polikliniklerine giden hastalar, tetkik yaptırmak için kan alma birimlerine gelerek burada kan ya da farklı tetkikleri veriyorlar. Bu çalışmada, (philerobo yapay zekâ kan alma birimi yönetim sistemi) hasta bir kiosk üzerinden kan alma birimine geldiğinde kaydını yapıyor. Daha sonrada bekleme salonunda bekliyorlar ve sistem zamanı geldiğinde o hastayı çağırıyor. Yaptığımız çalışmada, modelimiz hasta kiosktan girdiği anda içerde ne kadar bekleyeceğini bazı parametreleri (geliş saati, haftanın günü, hemşire sayısı, geldiği anda kendisi ile aynı önceliği olan hasta ne kadar beklemiş, kendine eşit ve yüksek önceliği olan bekleyen hasta sayısı, aynı öncelikte olan bekleyen hasta sayısı, gelen hasta sayısı, kan alma işlemi tamamlanan hasta sayısı, Cinsiyet ve hastanın önceliği) dikkate alarak yapay Sinir Ağları algoritması bekleme süresini tahmin ediyor. Bunun sonucunda yapay zekânın öğrendiği duruma göre yeni gelecek hastanın bekleme salonunda ne kadar bekleyeceğini tahmin edilmesini ve alıp hastaya bilgi verilmesini hedefledik.

Anahtar kelimeler: Bekleme süresi, kan alma birimi, yapay sinir ağları, makine öğrenme, tahmin

CONTENTS

Page

M.Sc THESIS EXAMINATION RESULT FORM ii

ACKNOWLEDGEMENTS iii

ABSTRACT iv

ÖZ v

LIST OF FIGURES vii

LIST OF TABLES ix

CHAPTER ONE – INTRODUCTION 1

1.1 Motivation 1

1.2 Problem Definition 2

1.3 Organization of the Thesis 3

CHAPTER TWO – LITERATURE REVIEW 4

2.1 Uncertainty of Waiting Time 4

2.2 The Effects of Waiting in Healthcare 5

2.3 Wait Time Studies 6

2.4 Artificial Neural Network 7

2.4.1 The Explanation of Artificial Neural Network 7

2.4.2 Learning in Artificial Neural Network 11

2.4.3 Types of Learning in Artificial Neural Network 11

2.4.4 How Artificial Neural Network Learn 12

2.4.5 Advantages and Disadvantages of Artificial Neural Network 13

2.4.6 Neural Network Structure 14

2.5 Novelty of the Proposed Study 16

CHAPTER THREE – METHOD 17

3.1 System Overview 17

3.2 Utilized Programs.. 18

3.2.1 Python 18

3.2.2 Library 18

3.2.2.1 Keras 19

3.2.2.2 Json.. 19

3.2.2.3 Datetime 19

3.2.2.4 Matplotlib.. 19

3.2.2.5 Scinit-Learn 20

3.2.2.6 Numpy.. 20

3.2.2.7 Pandas 20

3.3 Data Preparation 21

3.3.1 Arrival Time 22

3.3.2 Patient Priority.. 23

3.3.3 Waiting Patient Count with Priority 23

3.3.4 Waiting Patient Count with the same Priority 23

3.3.5 Current Patient Duration.. 24

3.3.6 Phlebotomist Count 24

3.3.7 Incoming Patient Count 24

3.3.8 Outgoing Patient Count 25

3.3.9 Day of Week.. 25

3.3.10 Waiting Time.. 25

3.4 One-Hot Vector 25

3.5 Layer and Model 27

3.5.1 Input Layer... 28

3.5.2 Hidden Layer 29

3.5.2.1 Activation Function.. 30

3.5.3 Output Layer... 31

3.6 Compile 32

3.7 Fit 34

3.8 Evaluation 35

CHAPTER FOUR – RESULT **36**

4.1 Comparison of Neural Network Result With Different Number of Hidden Layers and Neurons 38

4.2 Comparison of Number Different Number of Class 41

CHAPTER FIVE – CONCLUSION AND FUTURE WORK **42**

REFERENCES.. **43**

LIST OF FIGURES

Page

Figure 2.1 Structure of the neural network. The model consists of 2 input and 1 output. F(x1,x2) = y1 9

Figure 2.2 Feed-forward Neural network 10

Figure 2.3 Feed-Back Neural network 10

Figure 2.4 Type of Machine Learning 12

Figure 2.5 Biological Nervous system 14

Figure 2.6 The basic ANN structure is similar to biological nervous system 15

Figure 3.1 The flow of the proposed model 17

Figure 3.2 Derived JSON data from the patient 22

Figure 3.3 A simple example of the One-hot vector 27

Figure 3.4 Structure if the proposed model of Neural Network 28

Figure 3.5 Neural Network with activation function 30

Figure 3.6 Sigmoid curve 31

Figure 3.7 Graph of the proposed model of Neural Network 32

Figure 4.1 Line graph of resulting ANN model with 8 classification (Test 2 in Table 4.1) 39

Figure 4.2 Line graph of resulting ANN model with 6 classification (Test 5 in Table 4.2) 40

Figure 4.3 Line graph of resulting ANN model with 5 classification (Test 3 in Table 4.3) 40

Figure 4.4 Overfitting of ANN model 41

LIST OF TABLES

Page

Table 4.1 Result of 8 classifications Neural Network test 36

Table 4.2 Result of 6 classifications Neural Network test 36

Table 4.3 Result of 5 classifications Neural Network test 37

Table 4.4 Confusion matrix result of 8 classifications 37

Table 4.5 Confusion matrix result of 6 classifications 38

Table 4.6 Confusion matrix result of 6 classifications 38

CHAPTER ONE
INTRODUCTON

1.1 Motivation

Hospitals' blood collection units have the highest congestion of patients. Because patients who go to different outpatient clinics of the hospital (There are approximately 20-30 different polyclinics in hospitals), when doctors make a test request from patients, a large part of the patients (amount of up to 60% in some hospitals) come to the blood collection units where they give blood or different tests. That is why blood collection units are the most intensive and waiting time units in hospitals. Generally, people are standing in this unit as are hungry before giving blood, because most of the tests are done on an empty stomach. Here, of course, waiting patients as hungry is automatically affecting badly.

This also has a medical effect, so patients have a negative effect on the outcome of the test or on their results. Both the patient is standing on feet and has an effect on the medical results standing on feet due to exhausted stress and hormones are changing. Waiting time in large hospitals can take almost half an hour - one hour, and patients are waiting to stand in order not to lose their order in the queue. This queue is usually a two-stage, before the patients enter a queue for the preparation of the tubes, and after waiting for their turn to give the blood to the nurse. In the study of the PHLEROBO Al-Optimized Blood Collection Unit, the waiting time in the blood collection units was optimized and the waiting period was taken down.

Another feature of the application is to provide information about how long to wait for the patients who arrive according to the time they optimize, so that patients can not only do the time planning but also do not stand on feet for all waiting time. Because there is such a question in the minds of the patients that when their process will come. Everybody wants the process to happen as soon as possible. If we can give an estimate to the patient at that point, then the patient will be relieved if there is some amount of time. In systems where people are standing up in the queue, people actually predict how long they will wait for the length of the queue, for example, if the line is short they expect to wait shortly, or if the line is long, they expect to wait too long. In places where the numerator system is used, people are trying to predict when the number is going to take place.

However, the estimated time in both processes does not reflect the truth. Because, it hasn’t been known how fast the line moves. For example, it can be a very long queue, but many nurses can work inside so it can progress quickly or may be a short one, but there are few working nurses, so the queue is progressing slowly and can wait for a long time. The numerator systems can give sequence numbers from different series according to priority or internal systematic (e.g. in banks). For example, a number starts from 100, another one starts from 700, for example, the first patient has a 102 number in his hand and the other may get a 723 sequence number. Which number will be called first? For example, the patient has a 102 number in his hand, but 723, 724 and 725 are called in sequence, but the sequence number 102 is not called (we meet in the bank). Here, in fact, the patient does not have a clear knowledge of how long to wait. They are just trying to predict when to pick up the number according to the size of his number. Patients can wait comfortably when given Patient prediction wait time.

1.2 Problem Definition

The prediction of waiting times is challenging task due to several reasons. Firstly, collecting patient data is a very difficult and time-consuming process. Besides, the knowledge required to predict the patients' waiting time is not only the knowledge of the patients. Determining the factors that affect patients' waiting time is a difficult problem in itself.

Another challenge to predict waiting times in blood collection units is that many different polyclinics send their patients to blood collection units. Because different polyclinic patients have different priorities. Such priority queues are itself difficult problem to solve. In addition, the number of nurses working in the blood collection unit is also of great importance. To handle with these challenges, it is also important to choose the best Machine Learning (ML) algorithm. Another challenging task is to decide whether classification or regression will be applied.

By collecting and processing data from the Phlebotomy Unit of Izmir University of Health Sciences Tepecik Training And Research Hospital, we decided to use Artificial Neural Networks (ANN) to do classification to handle challenges mentioned above. We utilized KERAS library on Python.

1.3 Organization of Thesis

There are five chapters in this thesis and the rest of the thesis is organized as follows:

Chapter 2 reviews Uncertainty of Waiting Time and related studies emphasize the importance of predicting waiting times. Then, we will mention about the effects of waiting in healthcare. Waiting time studies are reviewed. Furthermore, we have reviewed related studies about ANN.

Chapter 3 gives an overview about our system. Then, we explained utilized programs in sequence. Data preparation and Features are explained in detail. Lastly, Neural Networks and selected parameters are mentioned.

Chapter 4 presents a detailed discussion of the results

Chapter 5 covers the conclusion and future works.

CHAPTER TWO
LITERATURE REVIEW

2.1 Uncertainty of Waiting Time

Queues unavoidably form when an operation takes longer than the time for people to arrive for that operation. Just as the wait times at banks or supermarkets where people need to queue after one another, queues form before patients see a doctor and after the doctor visit when they need to be attended for certain tests. There are several different factors affecting the queue time. For example, the number of attending doctors, amount of patients coming to the unit, the duration of test and operations in a polyclinic included, many different factors are relevant when discussing unit wait times. These operations could even be something that is requested by many different polyclinics, like phlebotomy. When queues are unavoidable and unpreventable, minimizing queue times is of high importance. As a result predicting waiting times while taking the hospital equipment, employees and the tests required for the patient into consideration gains importance. Waiting times could have negative consequences in many different aspects (psychological, social, financial or physical) on the person who's doing the waiting, especially when the reason for the wait is unknown. In many cases, patients have to wait while standing. In hospitals, sometimes they need to be hungry before the tests.

Queues could come up anywhere and are universal. Even waiting for the light to turn green in traffic is queuing. However, with new technologies even the time for a traffic light to turn are starting to display the time to wait. Taking this into consideration, predicting the waiting times in hospitals and informing patients with an approximate value could have positive effects. Before the positive effects, the greatest priority is to not have negative effects on the patient. In the rest of this segment the effects of waiting on people and previous research on this topic will be investigated. In addition, after looking at artificial neural network theory, works in health space using ANN and Keras will be discussed.

2.2 The Effects of Waiting in Healthcare

A patient waits for both their doctor visit and for the required tests after the examination. Waiting times are harmful to the patient in several different aspects and are not unique to our country. According to a study from Nigeria, their patients have to wait for between 90 and 180 minutes before an examination (Oche & Adamu (2013)). Another study from the US recorded the waiting times in Atlanta as 60 minutes and in Michigan 188 minutes (Dos Santos & Stewart, 1994). Tran et al (Tran, Nguyen, Minh Nong & Tran, 2017) examined additional studies on the topic of waiting. In some of these, it was determined that 50% of Irish patients waited for 60% (McCarthy, McGee & O'Boyle, 2000) and that Vietnamese hospitals had long waiting times due to overloading. The waiting times before examinations in Vietnam were 92 + 72 minutes in Vietnam.

The fact that waiting times are long around the world is notable. For these waits to have negative effects on the treatment process is unavoidable. As an example, for patients with heart problems, or patients waiting cataract surgeries, waiting times could mean death (Guttmann, Schull, Vermeulen, & Stukel, 2011). Even with diseases like cancer, the situation could be critical. Waiting times in different situations could have different consequences. Another study has investigated the mental and emotional negative effects of wait times on patients with orthopedic surgeries (Wright & Menaker, 2011). Another aspect of waiting times is that they affect not only the patient but also their relatives and friends (Day, 2013). Delays during waiting time and not knowing the reason for the delays cause the patient to be worried (Joseph, Hijal, Kildea, Hendren, & Herrera, 2017). Joseph A. has the compiled the articles above in his own thesis. Besides, phlebotomy patients could be fearful of the procedure and their stress levels rise as the wait time increases.

Jaworsk et al have conducted a survey in Massachusetts General Hospital and predicted waiting times using linear regression analysis (Jaworsky, Pianykh, & Oglevee, 2016). They have displayed the predicted waiting times on screens, in just two units of the hospital. The survey was made up of 5 questions. The first question was about whether the patients knew their waiting times. The question had possible 4 answers. "No", "Yes – Display", "Yes –Staff", "Not interested". The second question asked if their examination started on time. The possible answers were "Yes", "Early" and "Delayed". The third question asked if the patients noticed their waiting times on the screen. The possible answers were "No", "Yes – Useful" and "Yes – Not useful". Fourth question had the following sentence "The information on the wait time display looked . . ." and asked the respondent to fill the blank with one of the following options: Accurate/Inaccurate, Readable/Not Readable, Confusing/Straightforward. The fifth question asked if the patient thought that the displays should be kept, should added to all waiting rooms or should be removed. The results were satisfactory, 71% of patients thought the waiting time screens were beneficial and 82% of patients answered the fifth question with "screens should be kept" or "screens should be added to other rooms" (Jaworsky, Pianykh, & Oglevee, 2016).

2.3 Waiting Time Studies

Until now, all studies about waiting times have been based on polyclinics or a single unit of a hospital. Different units of hospitals have different waiting times due to different reasons. As a result, the important factors affecting the waiting time in a unit need to be considered. The algorithm to make the waiting time prediction needs to be selected carefully as well. Hemaya et al tried to make predictions with a linear regression using the patients' arrival times (Hemaya & Locker, 2012). Of course, it's difficult to make accurate predictions with a limited amount of parameters. Piankyh et al have looked at previous queues and created a prediction model using 25 parameters (Pianykh, & Rosenthal, 2015). Following the creation of their method, they made an application that displays predicted wait times and conducted a survey for 10 days to observe patient satisfaction. The dissatisfied patients were generally the ones who experienced delays. As a result, the authors aimed to make better predictions with a larger team and using Machine Learning (Curtis, Liu, Bollerman, & Pianykh, 2018).

There are many different machine learning algorithms including:

- neural network
- random forest
- support vector machine
- elastic net
- multivariate adaptive regression splines
- k-th nearest neighbor
- gradient boosting machine
- bagging
- classification and regression tree
- linear regression

Looking at the results, it was determined that elastic net was the most successful algorithm from the 10 considered (Sun, Teow, Heng, Ooi & Tay, 2012).

2.4 Artificial Neural Networks

2.4.1 The Explanation of Artificial Neural Networks

Artificial neural networks are a combination of neuronal complexes and, of course, artificial neurons that work much like biological neurons. Thus, it receives many inputs in different weights and produces an output depending on the input.

The relative position of the neural in the network (number and grouping, and their types of connections) is called the network architecture. In a network architecture, the connections between them and the number of layers are important. The first layer, where inputs are given to the network is called the “input layer”, where the system reveals its results is called the "output layer" and, if needed, layers between these two layers are called "hidden layers" (Figure 2.1).

Input layer: This layer receives the inputs and depending on the strength of relationship (connection) with the next layer, sends the input signal to the next layer. The strength of the relationship (connection) between each neuron and the other neuron is called the neuron's weight.

Hidden layer: The number of hidden layers and the number of neurons is arbitrary. These layers are the intermediate layers between the input and output layers and the place where the whole calculation is made.

Output layer: After the input data is processed in the Intermediate layer, the data is made available in the output layer.

The artificial neural network acts as a function that takes the same number of inputs as the number of input neurons and gives the same number of outputs as the number of output neural.

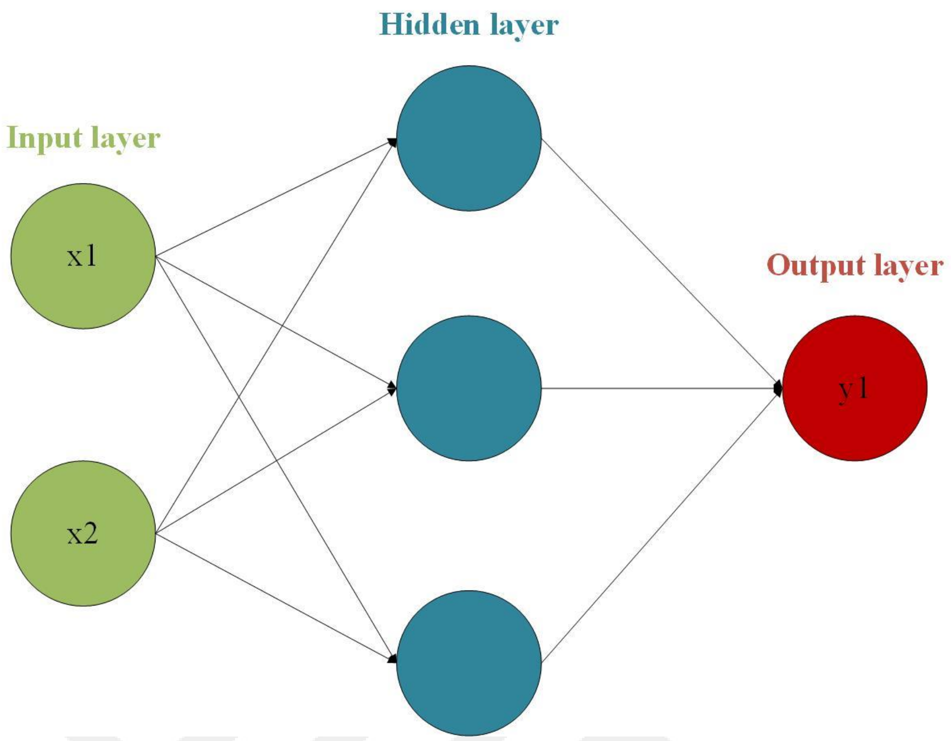

Figure 2.1 Structure of the neural network. The model consists of 2 inputs and 1 output. F(x1 ,x2) = y1

In many neural networks, neuronal connections are in such a way that the neurons of the Intermediate layers receive their input from all of their lower layer neurons (typically the layer of the input neurons). Thus, in a neural network, the signals gradually move from an input layer to higher layers, and eventually reach the final layer and the network output. Such a path is referred to with the technical term "feed forward" (Figure 2.2). Neuronal communication is very important in neural networks and in some way determines the strength of a neural network. The rule is that it divides neuronal communication into two groups. A type of neuronal communication is in additive, and in a way causes the signal to gather in the next neuron. The second type of communication between neurons causes the signal to be subtracted in the next neuron. Within the process, one group of strengthens communication to the next layer and the other group weakens it.

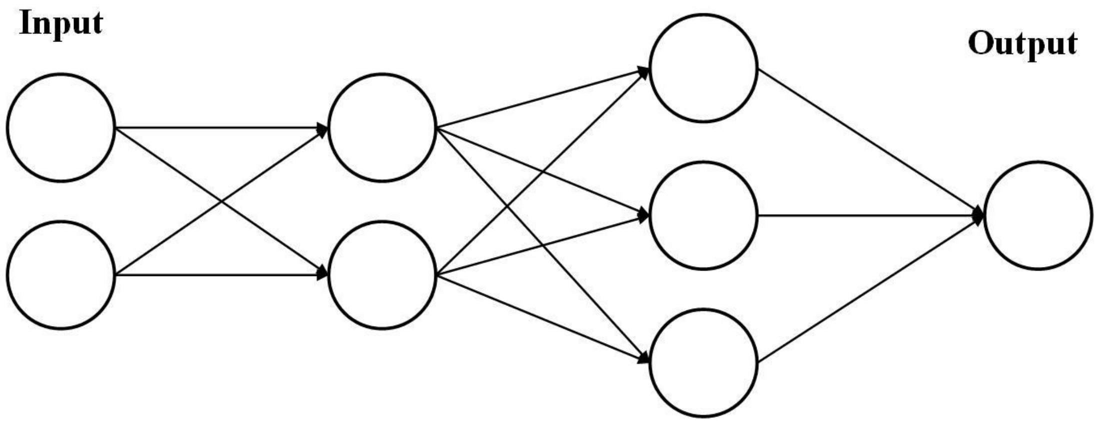

Figure 2.2 Feed-forward neural network

Another type of neuronal connection in neural networks is known as “feedback” (figure 2.3). In this type of communication, the output of a neuron layer is connected to the previous layer (or to a layer that is a few steps lower).

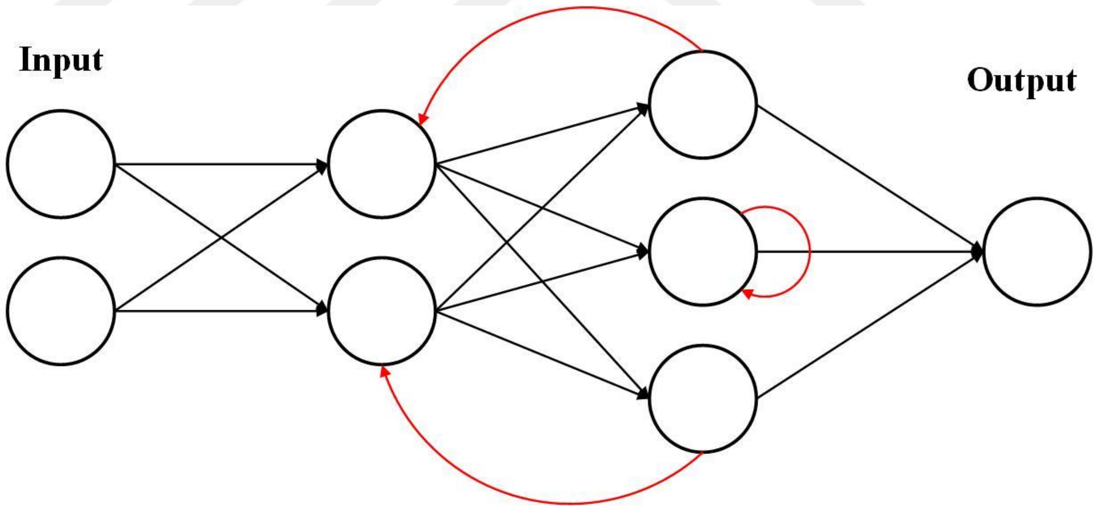

Figure 2.3 Feed-back neural network

Types of Artificial Neural Networks (Aggarwal & Yonghua 1998)

- Single-Layer Perceptron
- Multilayer Feedforward Network

- Hopfield neural networks
- Self-organising network
- radial basis function network

2.4.2 Learning in Artificial Neural Networks

A neural network, unlike digital computers, does not need clear instructions, just like a human, it is capable of learning through a number of specific examples. The learning of neural networks is in fact the regulation of the communication weights of these neurons to take different samples and to convert the output of the network to the desired output.

2.4.3 Types of Learning in Artificial Neural Networks

In terms of learning, neural networks are divided into several categories (Ang, Mirzal, Haron, & Haza 2016) (Figure 2.4):

- Supervised learning: In this type of learning, each neural network has two separate stages of learning and testing. For example, If we want a neural network using image processing to recognize the image of two people from different angles, first we need to teach the network a few different photos of each person, and then we can expect that the network to recognize new images with a few percentage of the error.
- Unsupervised learning: In this type of learning, neural networks do not have learning stages. For example, if we want from network to categorize two groups of images of bananas and apples without being clear that feature of two groups of photo, The Unsupervised network itself finds out the feature of two type of photographs and categorize them in two spaces.
- Semi-supervised learning: A supervised learning disadvantage is that the neural network may not be able to learn new positions not covered by the new

experimental data set without a teacher. A learning style of reinforcement resolves this limitation.

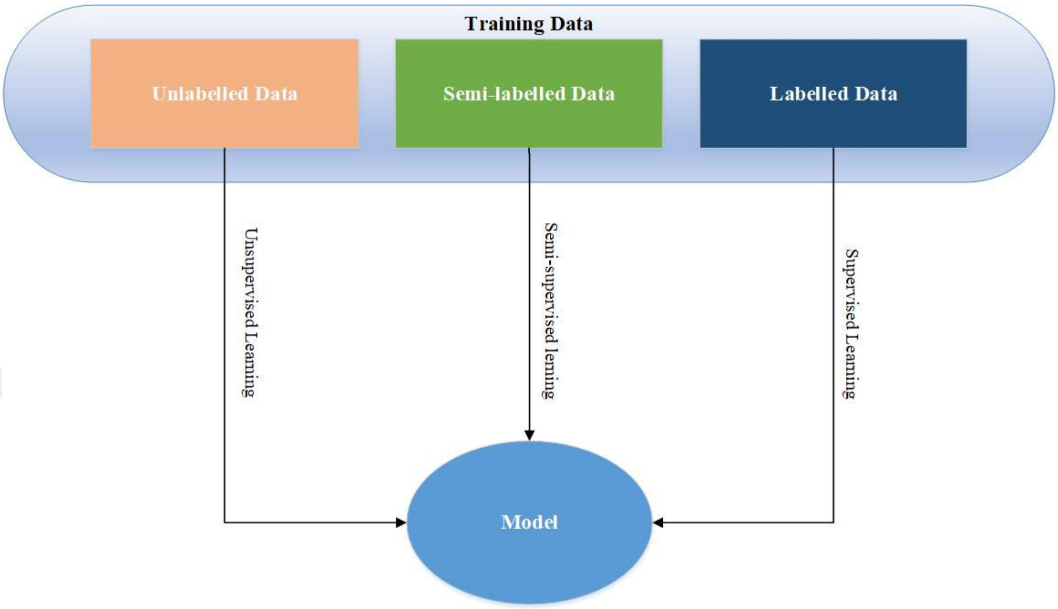

Figure 2.4 Type of machine learning

Learning algorithms in artificial neural networks:

- Back Propagation
- Delta Learning Rule
- Forward Propagation
- Hebb Learning Rule
- Simulated Annealing
- Genetic Algorithms

2.4.4 How in Artificial Neural Networks Learn

As a neural network learns, it develops a set relationship between inputs and outputs. This relationship is determined by calculating the relative importance of the inputs and outputs of the system. Based on trial and error, the system compares its results with the results given by the expert, until it reaches a certain level of accuracy.

With each test, the weights assigned to the inputs are changed to give the desired results.

Learning steps:

1. **Initialization**: Primary weights are randomly selected in the range.
2. **Activation**: Neurons or Perceptrons are activated by the use of optimal inputs and outputs. (Actualization) The actual output is calculated by repeating.
3. **Weight Training**: Neuronal or perceptron weights change. Weight correction is done using one of the training algorithms such as Delta Rule.
4. **Iteration**: Increasing the amount of repetition and returning to the second stage and repeating the process until convergence.

2.4.5 Advantages and Disadvantages of Artificial Neural Networks

Advantages:

The arrival of artificial neural networks into the world of information processing has allowed researchers to tackle many unresolved issues. Neural networks are included in soft computing because they do not follow a linear code like the normal algorithms of computation, such as the genetic algorithm. Hence, many consider them as artificial intelligence tools.

Disadvantages:

- Opaque internal functions, their inner workings are not clear even for their designers.

- They may require a lot of training time and may not converge to an acceptable point.
- Using randomized initial weights will result in different results.
- Due to repetitive calculations, in some cases the training and trial speed is low.
- Hardware simulation costs do not justify their application in many contexts

2.4.6 Neural network structure

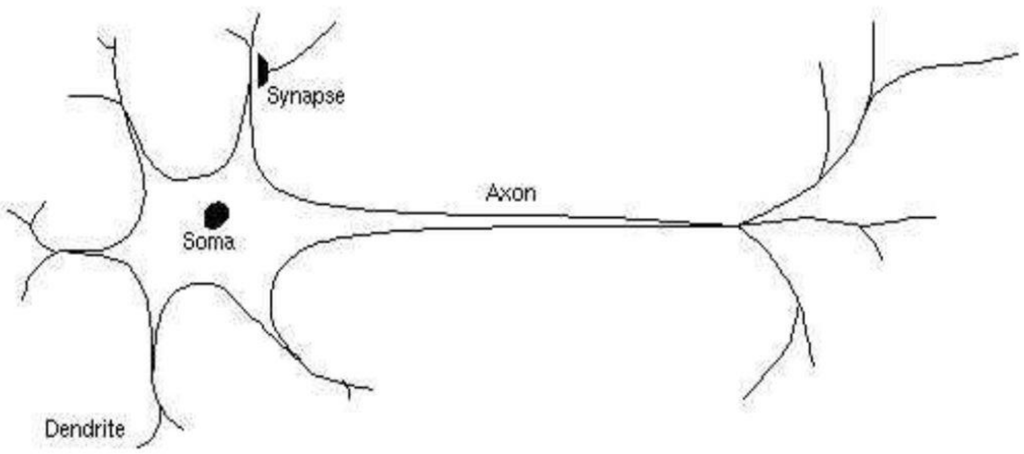

Figure 2.5 Biological nervous system. (adapted from Neural Networks - Neuron, n.d.)

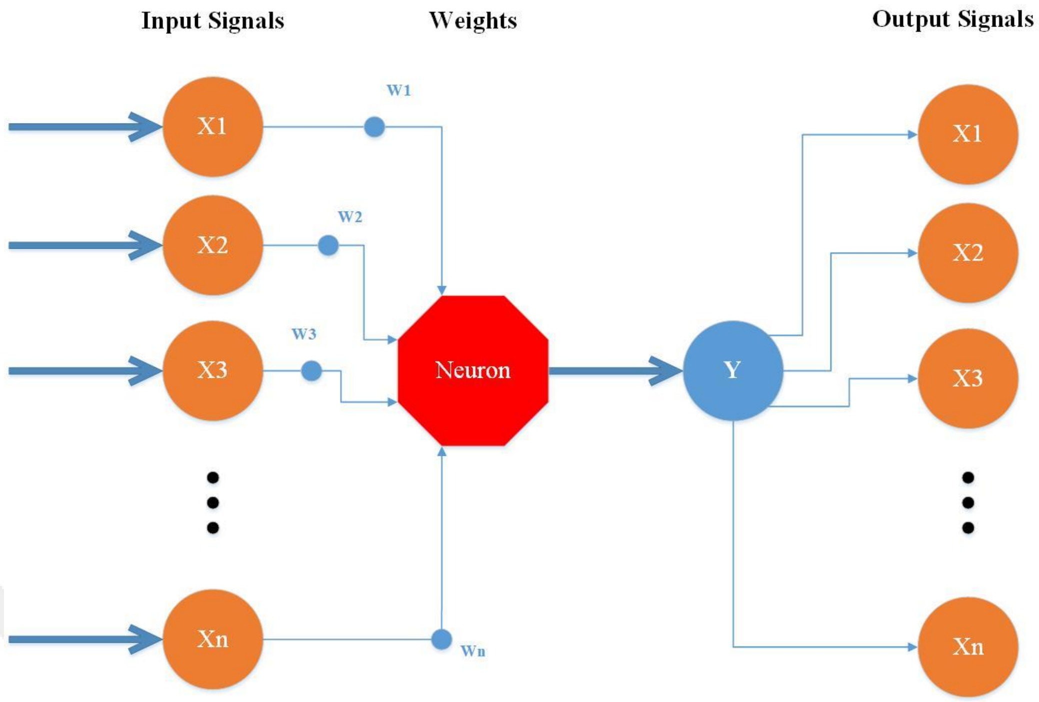

Figure 2.6 The basic ANN structure is similar to the biological nervous system

As you can see in the two pictures above (Figure 2.6) (Figure 2.5), the first image is a natural biological neuron. The information is entered into the neuron via the input, or the same dendritic, the same inputs are visible in the second image with the values (x1, ... , xm). In the artificial neural network model, we assign to each input a weight (w1, ... , .wm). These weights are in fact the importance of inputs for us, that's mean as the weight increases, the input is more important for network training. Then all the inputs are summed (Σ) and entered axon in a single layer. Next, apply the Activation Function to the data. Activation Function is actually defined in terms of the problem requirement and the type of neural network. This function contains a mathematical formula for updating weights in the network.

$$\sum_{i=1}^{m} bias + (w^i + x^i) \tag{2.1}$$

After doing the calculations, at this point, our information flows through the outgoing synapses to other neurons, and this phase continues until the network is trained.

In converting from figure to figure the soma becomes the neuron, dendrites turn into inputs, axon turns into an output and synapses turn into weights. ANN is called perceptron when ANN have single neuron.

2.5 Novelty of the Proposed Study

We have reviewed some studies about predicting waiting times in hospitals. Studies generally focus on regression techniques to predict waiting times. Besides this, collection of data is also a challenging task in such studies. Because there are unique factors for each department when it comes to predicting waiting times.

As far as we know there are no waiting time studies ran in a phlebotomy department. This also affects the types of features the neural network has. So the features utilized to make predictions in our case differ from other studies. We obtained our data from the Phlebotomy Unit of Izmir University of Health Sciences Tepecik Training And Research Hospital. Our study is also novel in the way it’s related to ML models. Observed waiting time studies have generally used regression models. We utilized Keras to implement Neural Network.

CHAPTER THREE
METHOD

3.1 System Overview

We developed a classification software tool that aims to predict patient waiting time at the hospital with machine learning. When we look at the overall flow of the system it is possible to explain it in six steps (Figure 3.1). As the data set needs to be prepared, it initially goes through some preprocessing. In the second step, we separated the dataset into a training set and test set. As third step, we created layers and set model of Artificial Neural Network. In the fourth step, we set training parameters as loss and optimization function. After doing all the above things, we train the train data set model in the fifth step. Finally, we used an evaluation to assess the performance of trained models on the test set.

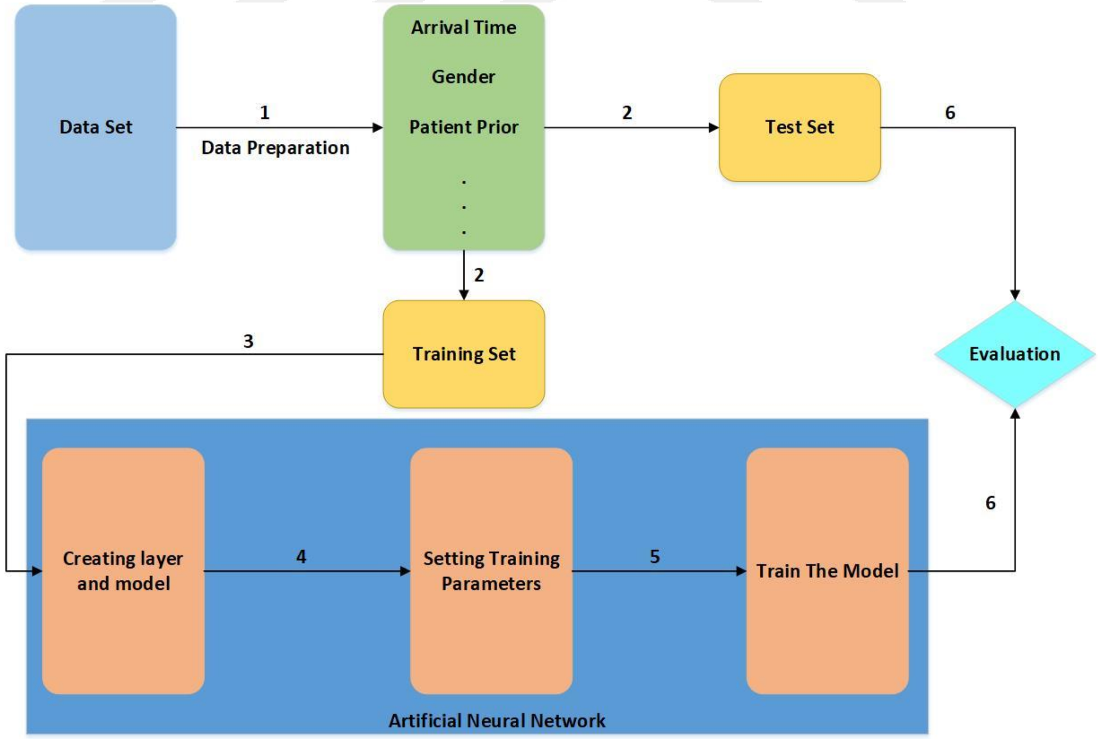

Figure 3.1 The flow of the proposed model

3.2 Utilized Programs

3.2.1 Python

Python is a high-level, object-oriented, and professional programming language that becomes more commonly used every day. It has a lot of wonderful features that have revolutionized the programming world from the development of web applications to the creation of computer games. Python first entered the programming world in 1991, and in recent years has attracted the attention of programmers and gaining new fans every day (Applications for Python, n.d.).

Over the past few years, the process of analyzing information-related data has become very complex, and the Python language has become a star in its field. Due to the popularity of the Python language, today we see that many libraries used in machine learning and data science provide an interface, or rather a Python language interface (Applications for Python, n.d.).

In this study we utilized python for the preparation of data and artificial neural network. We used Python version 3.7.

Applications of Python programming language

- Automation apps
- Scripting
- Data analysis
- Development of web applications

3.2.2 Library

We utilized multiple libraries for process on data in python. There are seven libraries for processing data: Keras, Json, DateTime, Matplotlib, Sicint-learn, Numpy and Pandas.

3.2.2.1 Keras

The Keras is one of the most important libraries for deep learning. In fact, Keras is not a deep learning library of its own. Keras employs Google Tensorflow, Microsoft CNTK, and Theano to offer a high-level API (application programming interface) that can use the deep learning libraries. In this way, it is possible to train your deep learning architecture using different packages (Home – Keras Documentation, n.d.).

We used version 2.2.4 of this library in this study.

3.2.2.2 Json

The Json library is used to express JavaScript Object Notation (JSON). Json is a lightweight data exchange format inspired by JavaScript object invariant syntax (json, n.d.).

We used version 3.7 of this library in this study.

3.2.2.3 Datetime

The datetime module is a class used to perform operations on the date and time. This library is presented for operation on simple and complex ways (Datetime — Basic date and time type, n.d.).

We used version 3.7.3 of this library in this study.

3.2.2.4 Matplotlib

The Matplotlib Library is a graphing library for the Python Programming Language. The library has a "Object-Oriented Application Programming Interface" for embedding charts into applications using the "General-Purpose GUI Toolkits" toolkit, such as Tkinter, wxPython, " Quotes (Qt) and Gitiki (GTK) (Matplotlib: Python plotting — Matplotlib 3.1.1 documentation, n.d.).

Matplotlib is originally written by John D. Hunter, a Neurobiologist, and has an active development team. This library has been distributed under a BSD-Style License.

Pyplot is a module from Matplotlib charting library that provides a Matlab-like interface. Matplotlib is designed to be as capable as Matlab (and even more so), and also benefits from working with Python, Open Source, and being free (Matplotlib: Python plotting — Matplotlib 3.1.1 documentation, n.d.).

We used version 3.0.2 of this library in this study.

3.2.2.5 Scinit-Learn

The Scinit-learn library, based on SciPy, is a powerful yet easy-to-use tool for classroom learning applications, including clustering, regression and classifications. Learning to use it is quick and easy, and its flexibility allows it to be used in a variety of issues. This module provides simple and effective tools for "Data Mining" and "Data Analysis". SKLearn is available and reusable for everyone in a variety of contexts.

We used version 0.21.2 of this library in this study.

3.2.2.6 Numpy

NumPy is one of the library in Python programming language (Python). Using this library, it is possible to use matrices and big multi-dimensional arrays. In order to perform mathematical and logical operations, they are much more efficient.

We used version 1.15.4 of this library in this study.

3.2.2.7 Pandas

When you are a user of Python programming language and intend to work on artificial intelligence or data science, Pandas is a valuable tool. The Pandas library is an open-source library with a BSD license that works superbly, and it's very easy to use to structure or analyze data.

We created DateFrame for parameters and removed missing value with using this library. We utilized the Pandas library to encode some parameters.

We used version 0.24.1 of this library in this study.

3.3 Data Preparation

The data was taken from the Phlebotomy Unit of Izmir University of Health Sciences Tepecik Training And Research Hospital. At first, we obtained the identification number of patient (szCdGndr), the identification number of patient (szPrtclNmbr), patient priority (szCdPrrty), time of phlebotomy (dCllct), identification number of the related phlebotomist (obCllct(szCdLSP)), patient arrival time (obArry), time for the phlebotomy tube to begin packaging (dPckStrt), when the packaging process ended (dPckDn), identification number of the assistant doing the packaging (obPck(szCdLSP)), call time to draw blood from the patient (arDsply) and the verification time of the patient (dVrfy)(Figure 3.2). We used all this data to determine the input parameters. This data is stored in a JSON file and the data from that JSON file is used for processing.

```
{
"patient":[
    {
        "szCdGndr" : "M",
        "szPrtclNmbr" : "29446330",
        "szCdPrrty" : "Routine",
        "arCntnr" : [
            {
                "obCllct" : {
                    "dCllct" : "2018-01-02T05:09:52.376Z",
                    "szCdLSP" : "1000100378"
                }
            }
        ],
        "obArrv" : {
            "dArrv" : "2018-01-02T04:31:42.263Z"
        },
        "obPck" : {
            "dPckStrt" : "2018-01-02T05:08:44.500Z",
            "dPckDn" : "2018-01-02T05:09:00.101Z",
            "szCdLSP" : "1000100204"
        },
        "arDsply" : [
            {
                "dDsply" : "2018-01-02T05:09:00.102Z"
            }
        ],
        "obVrfy" : {
            "dVrfy" : "2018-01-02T05:09:47.902Z",
            "szCdLSP" : "1000100378"
        }
    }
]
}
```


Figure 3.2 Derived JSON data for the patient

3.3.1 Arrival Time

Our first input data was arrival time and arrival time is determined by a delta T. Here we examined the delta in five minute intervals. In the other words, we divided the timeframe to five minutes intervals and then recorded the interval in which a patient arrived.

In the hospital we analyzed, the phlebotomy unit opened to the patients on 07:30 and closed at 15:00. Because of this we had ninety intervals of five minutes. For example, if a patient arrives at 10:32, and the arrival interval of this patient was calculated as 36, and for this patient, the arrival time parameter is 36.

$$\textit{(Patient Arrival Time - Time the Hospital Start) / 300} \qquad (3.1)$$

3.3.2 Patient Priority

The second input parameter we considered was the patient's priority. In the hospital we studied, there were eleven priority. These priorities in order from the highest priority to the lowest priority are STAT (Statim), Timing, Personnel, Disabled, Pregnant, Oncology, Organ-Transplantation, Elderly, Kid, Veteran and Routine.

3.3.3 Waiting Patient Count with Priority

One of the other input parameter is to determine how many patients waiting with equal priorities and higher priorities in the same T delta time interval for the patient's arrival time.

We calculated the waiting patients count with priority based on the arrival time of patients, the time of patient's call and the priority of the patient. For example, if a patient with STAT priority comes and in the waiting room there are 10 pending patients and only 1 patient has with Timing priority and 1 patient has STAT priority, the incoming patient will be the 3rd person during waiting.

3.3.4 Waiting Patient Count with the Same Priority

Another input parameter is the waiting patient count with the same priority. We obtained the waiting patients count with the same priority based on the arrival time, the time of patient's call and the priority of the patient. For example, if a patient with Oncology priority comes and in the waiting room there are 16 pending patients and only 3 patient has with Oncology, So here, there are 3 patients with the same priority.

3.3.5 Current Patient Duration

One of the other input parameters is the previously called patient's waiting time with the same priority for the new incoming patient. After trying different classifications of this parameter, we divided this parameter into 6 classifications. These classifications include 0 to 5 minutes, 5 to 10 minutes, 10 to 15 minutes, 15 to 20 minutes, more than 20 minutes and no previous patient with the same priority.

3.3.6 Phlebotomists Count

The phlebotomists count is one of the other input parameters. We calculated the phlebotomists count for each 5 minute interval. In the other words, we have calculated how many phlebotomists are working in every 5 minutes interval.

The phlebotomists count is one of the most important parameters because the phlebotomists count causes the queue to flow quickly or slowly. When there are a more nurses, the waiting time of the patient is shorter and vice versa, if the number of nurses is few, the waiting time of the patient is longer.

3.3.7 Incoming Patient Count

The number of patients incoming is one of the other input parameters. This parameter was used to determine the input rate. We obtained the incoming patient count for each 5 minute interval from a kiosk stationed at the unit.

3.3.8 Outgoing Patient Count

Another input parameters is the number of patients whose blood collection procedure ended within an interval. In this parameter, we calculated the rate outgoing patients. We used this parameter to detect duration of procedures inside.

3.3.9 Day of Week

One of the input parameters is the day of the week. We used this parameter because some days were more crowded than others. At first we categorized the day of the week, and after testing, we gave up classifying this parameter because it had better results if there was no classification.

3.3.10 Waiting Time

The output parameter is the patient's waiting time. We have categorized the waiting times in several different classifications. After testing all of these classifications, we classified the waiting time at 5 minute intervals. We identified the first class as less than 5 minutes, the second class as between 5 and 10, the third class as between 10 and 15 minutes, the fourth class as the waiting time between 15 and 20, and the last class as those waiting for more than 20 minutes.

3.4 One-Hot Vector

The "categorical" variable is a type of data that can only capture a limited number of values. For example, if there is a survey about car brands, brands would be the result of a class variable, because a brand that people could buy a car from is restricted to a few specific car manufacturing companies. Therefore, responses will be limited to a number of specific options. Or, if people are asked what color their cars are, their responses will be limited to a number of options (color names, such as white, blue, red, black, violent, etc.).

If these variables are used in most Machine Learning models, an error message will be issued. The reason for this is that many models do not have the ability to perform a direct process without encoding on the class data. One of the popular ways to encode class data is "One-Hot Vector" or "One-Hot Encoding."

The One-Hot encoding method is one of the most used approaches, and it functions quite well except for cases where the number of variables exceed a threshold .(Usually this method is not suitable for variables with more than 15 different values. In some cases, where the number of variables is lower, it may also not be appropriate). One-hot encoding creates new binary columns, each of which is one of the values that takes a variable.

For a better understanding of the subject, an example is given below (Figure 3.3). It is assumed that there is a variable called Color in the dataset whose possible values are Red, Yellow, and Green. Now, all values of this variable are converted to three columns separated by the headings Yellow, Red, and Green. As shown in the image, there are five "samples" in the dataset, where the value of the color variable for them is Red, Red, Yellow, Green and Yellow respectively. Now, for the first instance whose color is red, the red column is numbered 1 and the yellow and green columns are given 0. For the second instance, which is red, it works the same way. For a third instance whose color is yellow, the red and green columns are zero and in the yellow column is one. For the fourth variable whose color is Green, the value is zero in the red and yellow houses and zero in the green house. It works the same way for other samples.

	Red	Green	Yellow
Red	1	0	0
Red	1	0	0
Yelow	0	0	1
Green	0	1	0
Yellow	0	0	1

Figure 3.3 A simple example of the One-hot vector

We encoded four parameters of our dataset using One-Hot Vector. One of the features we encoded was the day of the week. This encoding is like the example shown above. For example if the day of week is Thursday, in the Thursday a 1 is written 1 and in the other days of week columns are given 0. The other parameters that got encoded were patient priority, current patient duration and patient wait time.

3.5 Layer and Model

In the artificial neural network there are three layers: input layer, hidden layer and output layer.

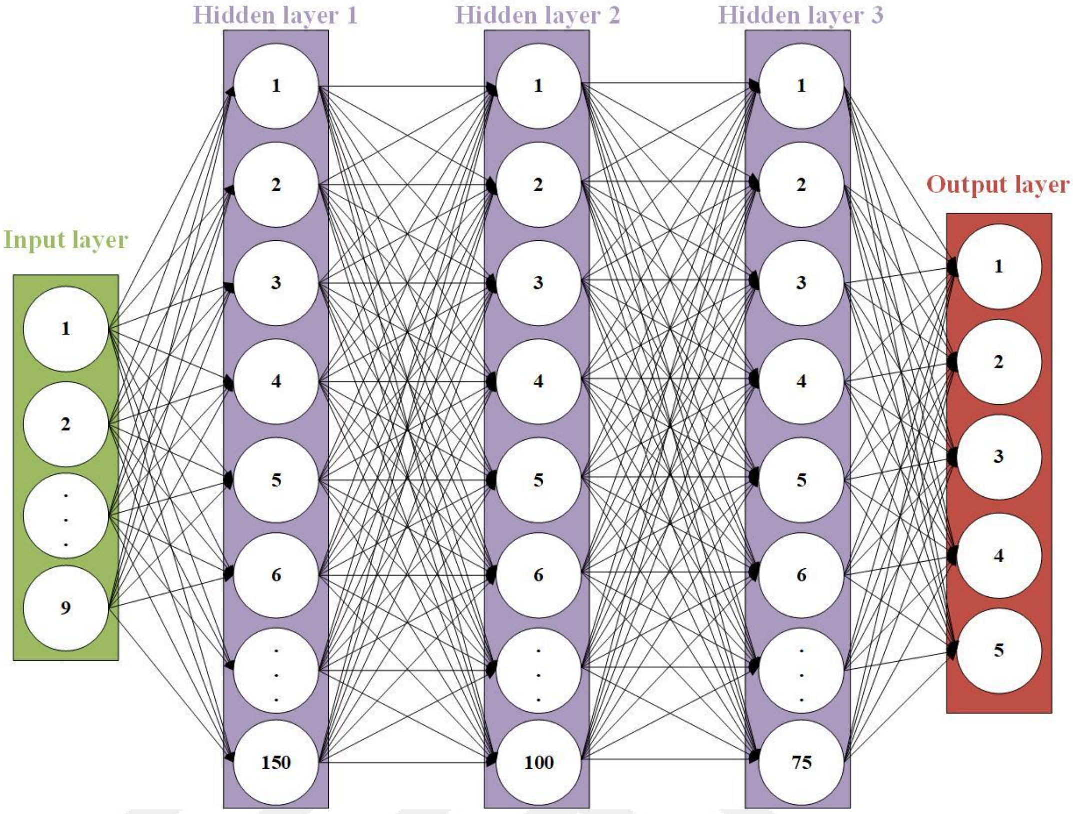

Figure 3.4 Structure of the proposed model of Neural Network

3.5.1 Input Layer

Input layer is the layer from which the inputs from the outside world come into the artificial neural network. This layer contains as many neurons as the number of entries from the outside world, the inputs are generally transmitted to the lower layers without any treatment.

In this layer, we have 9 neurons. These neuron include: Arrival Time, Day of Week, Phlebotomists Count, Current Patient Duration, Waiting Patient Count with priority, Waiting Patient Count with the Same Priority, Incoming Patient Count, Outgoing Patient Count and patient priority. These 9 neurons are our input parameters.

3.5.2 Hidden Layer

The data moves from the input layer to this layer. The number of hidden layers may vary from network to network. Some artificial neural networks do not have a hidden layer, and some artificial neural networks have more than one intermediate layer. The number of neurons in the hidden layers is independent of the number of inputs and outputs. In networks with more than one hidden layer, the number of neurons between the hidden layers may also be different. The number of hidden layers and the number of neurons in these layers increases the complexity and duration of the calculation, artificial neural network can be used to solve more complex problems.

In the hidden layer, we have three layers and each of layer has unique neurons. At first, the parameters we selected in the input layer are delivered to the first layer of the hidden layer. The task of this layer is processing the inputs obtained from the previous layer. Therefore, the layer is responsible for extracting the required features from the input data.

We used 150 neurons in the first layer of the hidden layer and the data comes from the input layer to this layer and after processing on the data, it goes to the second layer of the hidden layer. In the second layer of the hidden layer we have 100 neurons and the data comes from the first layer of the hidden layer to this layer and after processing on the data, it goes to the third layer of the hidden layer. In the last layer of the hidden layer we have 75 neuron.

The hidden layer is an intermediate layer between the input layer and the output layer. Activation function applied on this layer.

3.5.2.1 Activation Function

Activation functions are very important function in ANN and are usually non-linear or linear mathematical functions. Each activation function receives a number and then performs some mathematical operations on it. The output of activation function is a number that ranges from zero to one or between 1 and -1. Figure 3.5 shows a neuron with the activation function f.

As you can see, the activation function receives a number as an input, which is the same as the result of calculating the sigma function as well as the bias.

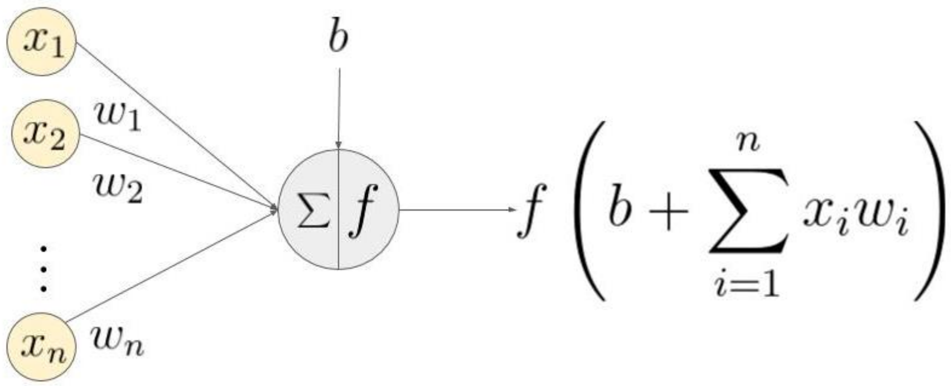

Figure 3.5 Neural Network with activation function

We utilized a sigmoid function in the activation function. (Sigmoid functions are also called logistic functions.) The curve is in the shape of S which can map any value to a value between 0 and 1 (Figure 3.6). If the curve approaches to the positive infinite, the prediction is mapped to 1 and if the curve approached the negative infinite, the prediction was mapped to 0.

$$F\ (t) = \frac{1}{1+e^{-t}} \tag{3.2}$$

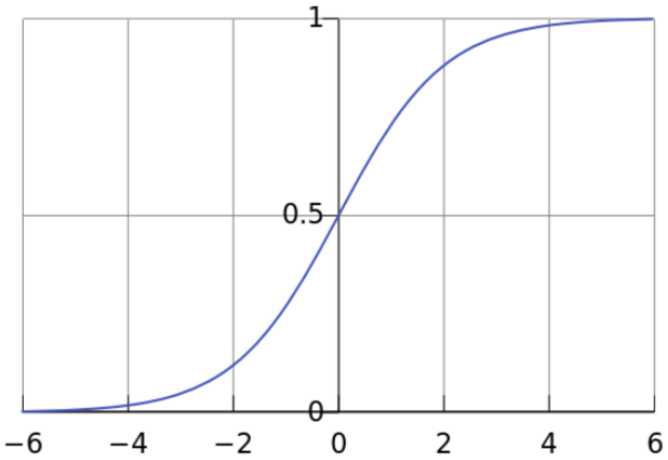

Figure 3.6 Sigmoid curve

If the output of the sigmoid function is greater than a numeric threshold of 0.5, then the result can be classified as 1 or YES, and if it is less than this numeric threshold of 0.5, it can be classified as 0 or NO. Using cancer prediction as an example, if the output is 0.75, then we can say it is 75% probable that the patient will suffer from cancer.

3.5.3 Output Layer

The output layer is the layer that processes the information from the intermediate layers and produces the outputs corresponding to the data from the input layer of the network. In feedback networks, the new weight values of the network are calculated using the output generated in this layer.

As shown in the figure below (Figure 3.7), each layer had different inputs and outputs. The input for each layer consists of the output of the previous layer. The output of the final layer is the prediction of the model.

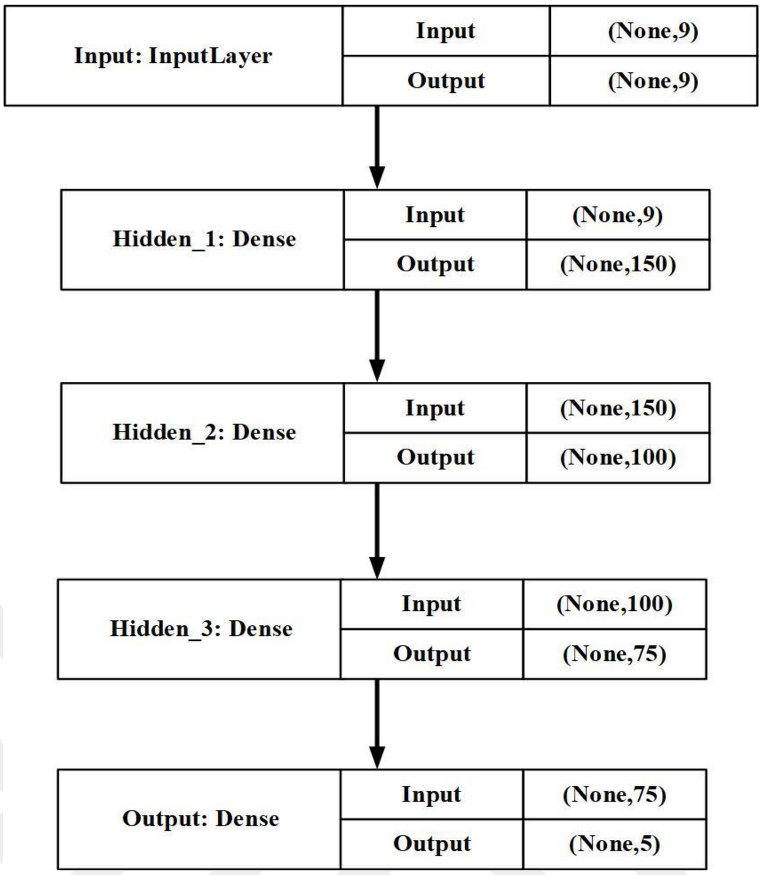

Figure 3.7 Graph of the proposed model of Neural Network

3.6 Compile

Before training a model, you must configure the learning process, which is done through the compile method.

Compile *(**optimizer**=None, **loss**=None, **metrics**=None)*

It takes three arguments:

1. **Optimizer**: An optimizer function is one of the two parameters needed to compile a model of Keras. Optimization is the process of quest for parameters that minimize or maximize our functions. We generally use indirect optimization when developing a machine learning model (Keras n.d.).

Available Optimizer functions:

- Adam
- Nadam
- Adamax
- Adadelta
- Adagard
- RMSprop
- SGD

2. **Loss**: A loss function is one of the two parameters needed to compile a model. You can pass one of the following two; the name of an actual loss function, or a TensorFlow/Theano symbolic function that reverts a scalar for one of two data-points and gets the following two arguments:

y_true**:** True labels. TensorFlow/Theano tensor.

y_pred**:** Predictions. TensorFlow/Theano tensor of the same shape as y_true.

The actual optimization goal is the average of the output sequence at all data points (Losses – Keras Documentation n.d.).

Loss functions include:

- mean_squared_error
- mean_absolute_error
- mean_absolute_percentage_error
- mean_squared_logarithmic_error
- squared_hinge
- hinge
- categorical_hinge
- logcosh
- categorical_crossentropy
- sparse_categorical_crossentropy

- binary_crossentropy
- kullback_leibler_divergence
- poisson
- cosine_proximity

3. Metrics: List of metrics to be evaluated by the model during training and testing. Usually using metrics = ['accuracy']. You can also pass a dictionary such as metrics = {'A__output': 'accuracy'} to specify different metrics for the different outputs of a multi-output model (Losses – Keras Documentation n.d.).

In this study, as the optimizer we utilized the Adam function. Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments (Kingma, Diederik, P. & Jimmy, B. A. 2015). We used categorical_crossentropy in loss function because the output was in categorical format.

3.7 Fit

Fit function trains the model for a dedicated number of epochs (repetitions on a dataset).

Fit(x*=None,* ***y****=None,* ***batch_size****=None,* ***epochs****=1,* ***validation_split****=0.0)*

Arguments of in the fit function:

- **X**: x is array of training data. The selected features are entered by X for training.
- **Y**: y is array of target data.

- **Batch_size**: the batch_size is an optional integer input. It denotes the number of instances per gradient update. If not specified, batch_size is 32 by default.
- **Epochs**: The epoch is argument is an integer. Number of epochs to train the model. An epoch is a repetition the whole x and y data provided. It's worth noting that in conjunction with **initial_epoch**, **epochs** is to be understood like "final epoch". The model was trained not only for several iterations through the epoch, but only until the epoch of the index epochs was reached.

- **Validation_split**: the validation_split is a float that ranges from 0 to 1. It denotes the proportion of data to be used as verification data. The model will separate this part of the training data and will use it for training, and will judge its loss and any model metrics related to this data at the end of each epoch. The verification data is selected from the final samples in the **x** and **y** data provided before mixing (Model (functional API) – Keras Documantation n.d.).

We have set Batch_size to 100, Epoch to 100 and 50 and Validation_split to 0.2 in fit function.

3.8 Evaluation

After obtaining the confusion matrix with True Positive (TP), False Negative (FN) results, we calculated accuracy information which is given in equation 3.3.

$$Accuracy = \frac{TP}{TP+FN} \qquad (3.3)$$

True Positive means if predicted the patient waiting time as in actual waiting time interval. Otherwise it is False Negative.

CHAPTER FOUR
RESULTS

In this study, we have done many different tests with different classifications. We aimed to test the model using different number of hidden layers and number of neurons in each layer with different number of epochs. We will mention the numbers in the following sections. We also aimed to compare waiting time results in different time intervals. Three different approaches were applied; dividing time intervals to eight (0-1, 1-3, 3-7, 7-11, 11-16, 16-22, 22-30 and 30-up minutes), six (0-3, 3-7, 7-12, 12-18, 18-25 and 25-up minutes), and five (0-5, 5-10, 10-15, 15-20 and 20-up minutes).

Table 4.1 Result of 8 classification Neural Network test

	Hidden_layer_1	Hidden_layer_2	Hidden_layer_3	Hidden_layer_4	Epochs	Loss	Val_Loss	Acc	Val_acc
Test1	250	100	50	-	50	73%	70%	69%	71%
Test2	250	100	50	-	100	67%	71%	74%	70%
Test3	250	150	75	40	50	71%	71%	70%	71%
Test4	250	150	75	40	100	66%	73%	72%	70%
Test5	400	200	100	50	50	68%	72%	71%	71%
Test6	400	200	100	50	100	62%	77%	74%	70%
Test7	100	50	25	-	50	74%	70%	68%	70%
Test8	100	50	25	-	100	73%	70%	69%	72%

Table 4.2 Result of 6 classification Neural Network test

	Hidden_layer_1	Hidden_layer_2	Hidden_layer_3	Hidden_layer_4	Epochs	Loss	Val_Loss	Acc	Val_acc
Test1	250	100	50	-	50	39%	41%	84%	83%
Test2	250	100	50	-	100	36%	43%	85%	84%
Test3	250	150	75	40	50	39%	40%	83%	84%
Test4	250	150	75	40	100	36%	43%	85%	84%
Test5	400	200	100	50	50	37%	42%	85%	84%
Test6	400	200	100	50	100	30%	46%	88%	82%
Test7	100	50	25	-	50	42%	39%	83%	84%
Test8	100	50	25	-	100	40%	39%	83%	84%

Table 4.3 Result of 5 classification Neural Network test

	Hidden_layer_1	Hidden_layer_2	Hidden_layer_3	Hidden_layer_4	Epochs	Loss	Val_Loss	Acc	Val_acc
Test1	250	100	50	-	50	29%	32%	88%	87%
Test2	250	100	50	-	100	25%	36%	89%	86%
Test3	250	150	75	40	50	29%	32%	88%	87%
Test4	250	150	75	40	100	24%	37%	90%	86%
Test5	400	200	100	50	50	26%	36%	89%	86%
Test6	400	200	100	50	100	21%	40%	91%	85%
Test7	100	50	25	-	50	31%	31%	87%	87%
Test8	100	50	25	-	100	30%	31%	87%	87%

For each structure we calculated accuracies as explained in section 3.8. The confusion matrices for the best result in each case can be seen in table 4.1, 4.2 and 4.3. Tests 2, 5 and 3 are the selected ones for each case respectively. As seen in Tables 4.4, 4.5 and 4.6, True predictions are colored yellow. When we calculate accuracy from these confusion matrices, it reaches %74, %85 and %88, respectively. When we reduced number of time intervals, it reached 88% accuracy.

Table 4.4 Confusion matrix result for 8 classifications

	0 – 1	1 – 3	3 – 7	7 – 11	11 – 16	16 – 22	22 – 30	30 - up
Class 1	19903	675	90	0	0	0	0	0
Class 2	3044	1298	795	11	1	1	0	0
Class 3	246	368	2508	413	45	3	0	3
Class 4	26	15	570	947	544	49	13	9
Class 5	1	1	56	315	395	362	95	0
Class 6	6	2	3	33	82	229	478	19
Class 7	2	0	2	10	6	25	271	423
Class 8	4	1	4	2	0	0	11	193

Table 4.5 Confusion matrix result for 6 classifications

	0 - 3	3-7	7-12	12-18	18- 25	25 - up
Class 1	25163	629	25	1	0	0
Class 2	830	2082	671	29	1	3
Class 3	67	399	1394	541	54	35
Class 4	7	16	262	544	268	215
Class 5	6	1	26	72	175	362
Class 6	6	4	5	1	20	724

Table 4.6: Confusion matrix result for 5 classifications

	0 - 5	5-10	10-15	15-20	20- up
Class 1	27139	691	7	2	0
Class 2	566	2132	573	61	63
Class 3	38	340	655	268	111
Class 4	8	24	113	239	506
Class 5	12	7	36	17	1020

4.1 Comparison of Neural Network Results with Different Number of Hidden Layers and Neurons

We applied 8 different tests as seen in tables 4.1, 4.2 and 4.2. We planned four different structures and two different numbers of epochs. First structure has 3 hidden layers with 250, 100 and 50 neurons, respectively. Another structure has 4 hidden layers using 250, 150, 75 and 40 neurons for each hidden layers, respectively. Third structure has 4 hidden layers as 400, 200, 100 and 50 are the numbers of neurons, respectively. The last structure has 3 hidden layers that has 100, 50 and 25 in each. For each structure, we used two different number of epochs which are 50 and 100.

Considering results in the Tables 4.1, 4.2 and 4.3, every structure can be the best for different cases. However, in general, the second structure has successful results in all cases. In table 4.1, the two structure has the best result with 100 epochs. Figure 4.1 shows the structure performed with 72% accuracy and 67% loss (Figure 4.1). However, the second structure is the best one in table 4.2 and 4.3, respectively. The second structure in Table 4.2 reached 85% accuracy and 30% loss (Figure 4.2) and in table 4.3 has the highest results with 88% accuracy and 29% loss (Figure 4.3). After all, the second structure with four hidden layers using 250, 150, 75 and 40 neurons seems to be the most successful structure.

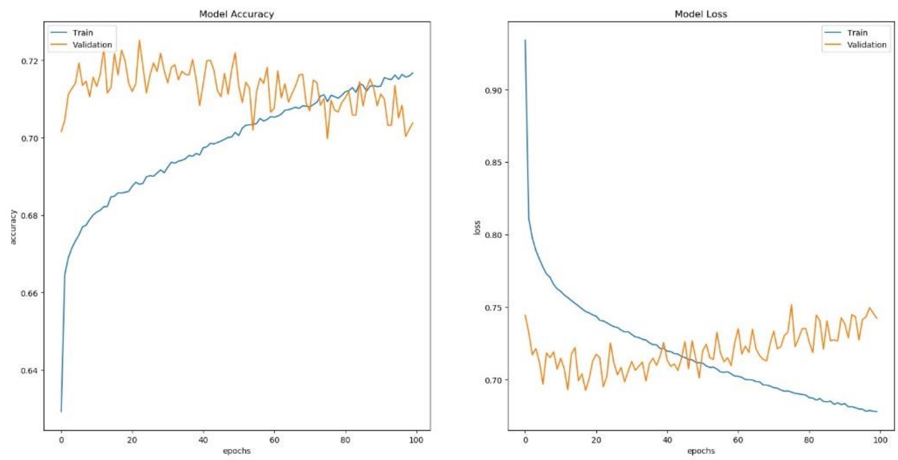

Figure 4.1 Line graph of resulting ANN model with 8 classifications (Test 2 in Table 4.1)

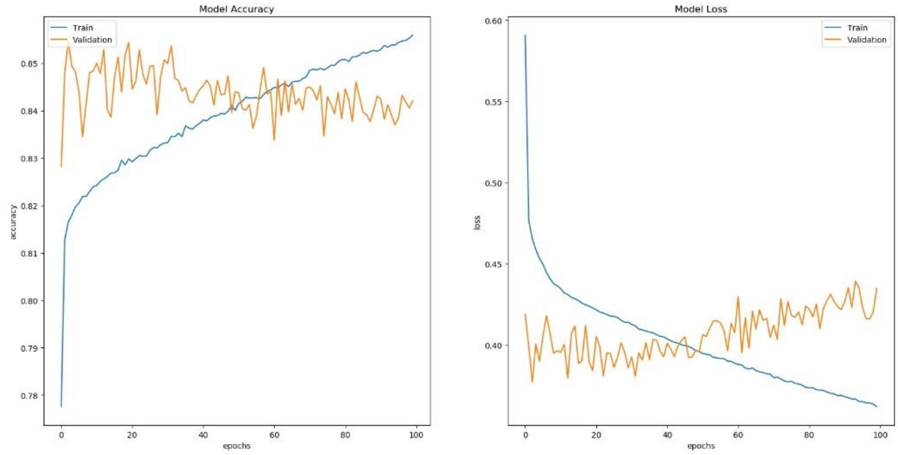

Figure 4.2 Line graph of resulting ANN model with 6 classifications (Test 5 in Table 4.2)

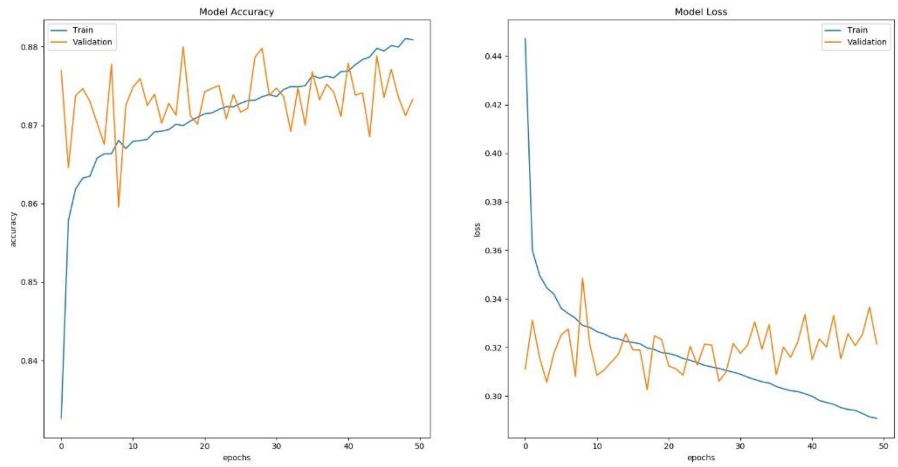

Figure 4.3 Line graph of resulting ANN model with 5 classifications (Test 3 in Table 4.3)

Overfit: A problem common to neural networks is "overfitting". This occurs when the network, in a manner of speaking memorizes the training set and doesn't have a lot of errors, but when put against new data, the network fails and number of errors increase substantially. Figure 4.4 illustrates how a model becomes overfit to its test set.

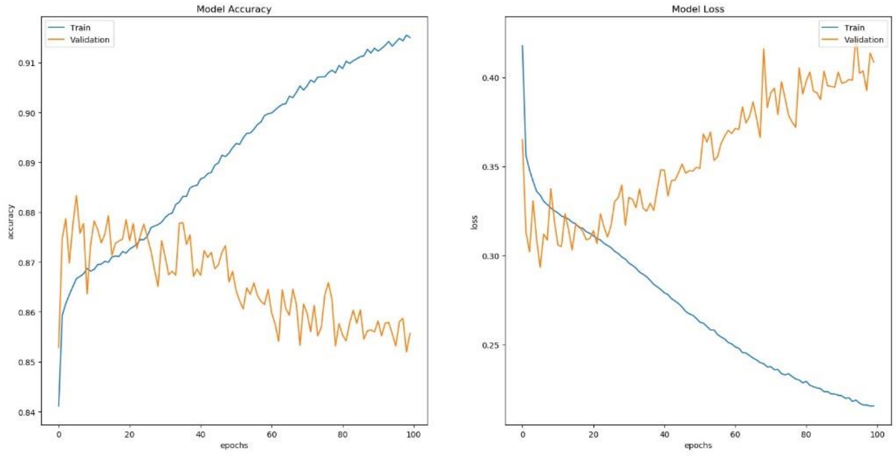

Figure 4.4 Overfitting of ANN model

4.2 Comparison of Different Number of Classes

In this part, we compared waiting time results in different time intervals. Firstly, there were 8 time intervals which are 0-1, 1-3, 3-7, 7-11, 11-16, 16-22, 22-30 and 30-up minutes. These time intervals are designed in minutes. For this part, output layers made up of 8 neurons which correspond to each time interval as the target class. The second time interval portioning includes 6 target classes which are 0-3, 3-7, 7-12, 12-18, 18-25 and 25-up minutes. For this part, 6 neurons are constructed as output layer. The last time interval portioning includes 5 target classes which are 0-5, 5-10, 10-15, 15-20 and 20-up minutes. For this part, 5 neurons are constructed as output layer.

When tables 4.1, 4.2, and 4.3 are compared, partitioning with five time interval seems to be the best when we compare each structure. Average validation accuracy of eight time partition intervals is %71 whereas %84 and %88 are the average validation accuracy for six and five partition intervals, respectively.

CHAPTER FIVE
CONCLUSION AND FUTURE WORK

Waiting time prediction is a challenging task due to many reasons explained above. By constructing different meaningful features and trying different time intervals, we achieved successful achievements. This study shows the importance of this challenging period.

We are planning to improve the success by considering to produce new different features and new Machine Learning methods.

REFERENCES

Aggarwal, R. & Yonghua S. (1998). Artificial neural networks in power systems. Part 2: Types of artificial neural networks. *Power Engineering Journal*, *12*(1), 41–47.

Ang, J. C., Mirzal, A., Haron, H., & Hamed, H. N. A. (2016). Supervised, Unsupervised, and Semi-Supervised Feature Selection: A Review on Gene Selection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *13*(5), 971–989

Applications for Python. (n.d.). Retrieved June 20, 2018, from https://www.python.org/about/apps/

Curtis, C., Liu, C., Bollerman, T., & Pianykh, O. (2018). Machine learning for predicting patient Wait Times and appointment delays. *Journal of the American College of Radiology*, *15*(9), 1310-1316.

Datetime — Basic date and time type. (n.d.). Retrieved January 10, 2019, from https://docs.python.org/3/library/datetime.html

Day, B. (2013). The consequences of waiting. In *Reducing Wait Times for Health Care: What Canada Can Learn from Theory and International Experience* (1rd ed.) (45-75). Canada: Fraser Institute.

Dos Santos, L. M. & Stewart, G. (1994). Rosenberg NM: pediatric emergency department walk-outs. *Pediatric Emergency Care, 10*(2), 76–78.

Guttmann, A., Schull, M. J., Vermeulen, M. J., & Stukel, T. A. (2011). Association between waiting times and short term mortality and hospital admission after departure from emergency department: population based cohort study from Ontario, Canada. *British Medical Journal*, *342*(7809), d2983.

Hemaya, S. A. & Locker, T.E. (2012). How accurate are predicted waiting times, determined upon a patient's arrival in the emergency department? *Emergency Medicine Journal, 29*(4), 316-318.

Hodge, W., Horsley, T., Albiani, D., Baryla, J., Belliveau, M., Buhrmann, R., O'Connor, M., Blair, J., & Lowcock, E. (2007). The consequences of waiting for cataract surgery: a systematic review. *Canadian Medical Association Journal*, 176(9), 1285-1290.

Home – Keras Documentation.(n.d.). Retrieved February 09, 2019, from https://keras.io/

Jaworsky, C., Pianykh, O., & Oglevee, C. (2016). Patient feedback on waiting time displays. *American Journal of Medical Quality*, *32*(1), 108-108.

Joseph, A., Hijal, T., Kildea, J., Hendren, L. & Herrera, D. (2017). Predicting waiting times in Radiation Oncology using machine Learning. *16th IEEE International Conference on Machine Learning and Applications*, *16*(1), 1024-1029.

Json. (n.d.). Retrieved January 16, 2019, from https://www.json.org

Kingma, Diederik, P. & Jimmy, B. A. (2015) Adam: A Method for Stochastic Optimization. *arXiv:1412.6980* [cs.LG], July 2015.

Kulkarni, G. S., Urbach, D. R., Austin, P. C., Fleshner, N. E., & Laupacis, A. (2009). Longer wait times increase overall mortality in patients with bladder cancer. *The Journal of Urology*, *182*(4), 1318-1324.

Losses – Keras Documentation (n.d). Retrieved February 10, 2019, from https://keras.io/losses/

Matplotlib: Python plotting — Matplotlib 3.1.1 documentation. (n.d.). Retrieved June 10, 2018, from https://matplotlib.org/

McCarthy, K., McGee, H. M. & O'Boyle, C. A. (2000). Outpatient clinic waiting times and nonattendance as indicators of quality. *Psychology, Health & Medicine*, 5(3), 287–293.

Model (functional API) – Keras Documantation. (n.d.). Retrieved February 10, 2019, from https://keras.io/models/model/

Neural Networks - Neuron. (n.d.). Retrieved January 10, 2019, from https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural_networks/Neuron/index.html

Oche, M. O. & Adamu, H. (2013). Determinants of patient waiting time in the general outpatient department of a tertiary health institution in north western Nigeria. *Annals of Medical and Health Sciences Research*, *3*(4), 588-592.

Pianykh, O.S. & Rosenthal, D. I. (2015). Can we predict patient wait time? *Journal of the American College of Radiology*, *12*(10), 1058-1066.

Sobolev, B. G., Fradet, G., Hayden, R., Kuramoto, L., Levy, A. R., & FitzGerald, M. J. (2008). Delay in admission for elective coronary-artery bypass grafting is associated with increased in-hospital mortality. *BMC health services research*, *8*(1), 185.

Sporns, O., Chialvo, D., Kaiser, M. & Hilgetag, C. (2004). Organization, development and function of complex brain networks. *Trends in Cognitive Sciences*, *8*(9), 418–425.

Sun, Y., Teow, K. L., Heng, B. H. , Ooi, CK. & Tay, S. Y. (2012). Real-time prediction of waiting time in the emergency department. Using quantile regression. *Annals of Emergency Medicine*, *60*(3), 299-308.

Tran, T. D., Nguyen, U. V., Minh Nong, V. & Tran, B. (2017). Patient waiting time in the outpatient clinic at a central surgical hospital of Vietnam: Implications for resource allocation. *F1000research*, *6*(3), 454.

Tu, J. V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, *49*(11), 1225–1231.

Wright, J. G. & Menaker, R. J. (2011). Waiting for children's surgery in Canada: the Canadian Paediatric surgical wait times project. *Canadian Medical Association Journal*, *183*(9), E559-E564.