T.C.
HASAN KALYONCU UNIVERSITY
GRADUATE SCHOOL OF
NATURAL AND APPLIED SCIENCES

A COMPARATIVE STUDY OF DEEP LEARNING METHODS ON FLEXURAL BUCKLING LOAD PREDICTION OF ALUMINUM ALLOY COLUMNS

M.Sc. THESIS
IN
ELECTRONICS AND COMPUTER ENGINEERING

Supervisor
Assist. Prof. Dr. Bülent HAZNEDAR

Zeliha Begüm KILINÇ

January 2020

A COMPARATIVE STUDY OF DEEP LEARNING METHODS ON FLEXURAL BUCKLING LOAD PREDICTION OF ALUMINUM ALLOY COLUMNS

M.Sc. Thesis
In
Electronics and Computer Engineering
Hasan Kalyoncu University

Supervisor
Asst. Prof. Dr. Bülent HAZNEDAR

Zeliha Begüm KILINÇ
January 2020

GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES INSTITUTE
MSc ACCEPTANCE AND APPROVAL FORM

Electronics-Computer Engineering M.Sc. (Master of Science) programme student **Zeliha Begüm KILINÇ** prepared and submitted the thesis titled **"A comparative study of deep learning methods on flexural buckling load prediction of aluminum alloy columns"** defended successfully on the date of 15/01/2020 and accepted by the jury as an M.Sc. thesis.

Position	Title, Name and Surename Department/University	Signature:
Jury Head	Prof. Dr. Abdülkadir ÇEVİK Civil Engineering Department Gaziantep University	
Jury Member	Assoc. Prof. Dr. M. Fatih HASOĞLU Computer Engineering Department Hasan Kalyoncu Universit	
M.Sc. Supervisor Jury Member	Assist. Prof. Dr. Bülent HAZNEDAR Computer Engineering Department Hasan Kalyoncu University	

This thesis is accepted by the jury members selected by the institute management board and approved by the institute management board.

Prof. Dr. Mehmet KARPUZCU

Document no: ENS.FR.32 Yayın Tarihi: 26.03.2018 Rev no/ Date: 00/--

ABSTRACT

A COMPARATIVE STUDY OF DEEP LEARNING METHODS ON FLEXURAL BUCKLING LOAD PREDICTION OF ALLIMINUM ALLOY COLUMNS

KILINÇ, Zeliha Begüm

M.Sc. in Electronic - Computer Engineering

Supervisor: Asst. Prof. Dr. Bülent HAZNEDAR

January 2020

In recent years, aluminum alloy columns have been widely used in construction fields. This is due to the light weight of aluminum alloys, high corrosion resistance, long life, low maintenance costs, the possibility of recovery, versatility of the metal and the possibility to obtain endless variety of profiles has many advantages. The calculation of the critical buckling loads of the columns is the most important issue. However, it is known that heat treated aluminum alloys have higher proof stress yield strength than non-heat treated aluminum alloys. In this study, buckling load estimation of heat treated aluminum alloy columns is made by using deep learning method and soft computing techniques and laboratory test results are compared. Sequential Model is used while using deep learning method. Adam, Adamax, Nadam, Adadelta, Adagrad, RMSProp and SGD and the optimizer functions of deep learning are evaluated separately. In addition, the results are evaluated using both MAE and MSE Loss functions for each optimizer. As a result of the study, it is understood that the optimizer and loss functions used together are more successful when estimating value for the dataset using deep learning model.

Keywords: Prediction of buckling load, Aluminum alloy columns, Soft computing, Deep learning, Artificial neural network

ÖZET

ALÜMİNYUM ALAŞIMLI KOLONLARIN EĞİLME BURKULMA YÜKÜ TAHMİNİ ÜZERİNE DERİN

ÖĞRENME YÖNTEMLERİNİN KARŞILAŞTIRMALI BİR ÇALIŞMASI

KILINÇ, Zeliha Begüm

Yüksek Lisans, Elektronik Bilgisayar Mühendisliği

Tez Danışmanı: Dr. Öğr. Üyesi Bülent HAZNEDAR

Ocak 2020

Son yıllarda, Alüminyum alaşımlı sütunlar inşaat alanlarında çok yaygın kullanılmaktadır. Bunun sebebi alüminyum alaşımlarının hafifliği, yüksek korozyon direnci, uzun ömürlülüğü, düşük bakım maliyetleri, geri kazanma imkanı, metalin çok yönlülüğü ve sonsuz değişik şekilde profil elde edebilme olanağı gibi pek çok avantaja sahip olmasıdır. Kolonların kritik burkulma yüklerinin hesaplanması ise buradaki en önemli konudur. Bu problem çeşitli hesaplama teknikleri ve bilinen yaklaşımlar kullanılarak giderilmeye çalışılsa da doğrusal olmayan sonuçları içermektedir. Bu da kritik burkulma yükü tahminini zorlaştırmaktadır. Buna rağmen, Isıl işlem görmüş alüminyum alaşımlarının ısıl işlem görmemiş alüminyum alaşımlarına göre daha yüksek prova gerilimi verim mukavemetine sahip olduğu bilinmektedir. Bu çalışmada, ısıl işlem görmüş alüminyum alaşımlı kolonların derin öğrenme yöntemi kullanılarak burkulma yükü tahmini yapılmış softcomputing teknikleri ve laboratuvar ortamından alınan test sonuçları ile karşılaştırılmıştır. Derin öğrenme yöntemi kullanılırken Sequential Modelinden yararlanılmıştır. Derin öğrenmenin optimizer fonksiyonlarından olan Adam, Adamax, Nadam, Adadelta, Adagrad, RMSProp ve SGD ve kullanılarak ayrı ayrı değerlendirilmiştir. Ayrıca, her bir optimizer için hem MAE hem de MSE Loss fonksiyonları kullanılarak sonuçlar değerlendirilmiştir. Çalışmanın sonucunda, derin öğrenme modelinin kullanıldığı veriseti için değer tahmini yaparken birlikte kullanılan hangi Optimizer ve Loss fonksiyonları ikilisinin daha başarılı olduğu anlaşılmıştır.

Anahtar Kelimeler: Burkulma tahmini, Alüminyum alaşımlı kolonlar, Yapay sinir ağları, Derin Öğrenme.

ACKNOWLEDGEMENT

I would like to thank my thesis advisor Bülent Haznedar for his support and helpfulness in sharing his knowledge and experience with me during the selection, writing and preparation process of my thesis.I would like to express my gratitude to Abulkadir Çevik for sharing his sincerity with me and the owner of the dataset used in my thesis.I would also like to thank my husband Çağan Kılınç, who encouraged me in this process, supported me financially and did not spare his help and labor, and all my family who raised my motivation and encouraged me when I gave up ...

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

ANN	Artificial Neural Network
ADAM	Adaptive Moment Estimation
ECCS	European Convention for Constructional Steelwork
GEP	Gene-expression programming
MAD	Mean Absolute Deviation
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
NN	Neural Networks
RMSE	Root Mean Squared Error
SC	Soft Computing
STD	Standart Deviation

CHAPTER 1
INTRODUCTION

1.1 General Overview

Aluminum alloys are non-linear metallic materials with round stress-strain curves not well represented by simplified elastic-excellent plastic material. Some aluminum alloys have tensile strength up to 600 MPa. In general, some aluminum alloys which are not very suitable for high temperatures can easily be used up to temperatures up to 300 ° C. Aluminum; It has high corrosion resistance in many chemical media including air, water, saline and petrochemicals. Maintaining its mechanical properties at low temperatures is another outstanding feature of aluminum. In general, the welding properties of high strength aluminum alloys are not good. For this reason, aircraft are produced using riveted joints. However, there is a high-strength aluminum alloy that can be welded, albeit in a small number.

Heat treatment for aluminum alloys is generally limited to processes applied to increase the strength and hardness of machined and cast alloys which may exhibit precipitation hardening. Heat treatment to increase the strength of aluminum alloys; solution; dissolution of the phases (formation of solid solution), quenching; formation of supersaturated structure, aging; precipitation of dissolved atoms at room temperature or higher (precipitation hardening).

27% of the aluminum alloys produced annually in the world are used in the construction sector. It is used in the Both of non-structural elements (doors, windows, facade cladding ...) and structural elements (bridge decks, wide span roofing, panels and shell structures). Aluminum alloys are also used as formwork for on-site concrete casting. By producing composite elements with fiber-reinforced polymer and carbon-reinforced polymer materials, the relatively low elasticity of aluminum can be removed. Nowadays, aluminum has become the most widely used metal after steel. However, the buckling estimates of the aluminum alloy are a major problem for aluminum alloy columns.

Recently, deep learning methods have been used in many fields and they are improving day by day. It is known that the amount of dataset is important when training the network in deep learning methods. The more training and test data the success of the model is increasing. However, even when there is not enough data, it is possible to achieve successful results even in small data sets by changing the variables such as epoch, mini-batch, layer number and optimizer method correctly. Since the dataset used in this study is not very large, it is utilized in the variables of hyper parameters.

The objective of this paper is to analyze the application of deep learning modelling for strenght prediction of aluminum alloy columns failing by flexural buckling. Sequential Model (SM) is presented as deep learning techniques used in the study. Adam, Adamax, Nadam, Adadelta, Adagrad, SGD and RMSProp optimizers which are the functions to SM is used. Also MAE,MSE are used as loss functions.The training and test sets for deep learning model are obtained from Cevik et al., (2009) related with the subject of 'Flexural buckling load prediction of aluminum alloy columns using soft computing techniques'. This paper also deals with comparative of the results of the soft computing techniques and deep learning modelling. These results are the statistical methods of *RMSE, MSE, MAD, MAE, MAPE, MEAN, STD* respectively.The proposed deep learning model is presented in explicit form to be used in practical applications.

The other chapters are explained below:

- Chapter 2, is a literature survey which is related on previous studies on aluminum alloy columns, deep learning and analyses according to GEP (Gene-Expression Programming) and NN (Neural Networks) accuracy.

- Chapter 3, general the background of the study about flexural buckling load prediction of the aluminum alloy columns datasets utilize in deep learning.

- Chapter 4, illustrates the broad scope of the necessary process and analyses that can be accomplished with deep learning.

- Chapter 5, discusses the results obtained from processing and analysis of the study.

- Chapter 6, related with the overall results.

CHAPTER 2
BACKGROUND & LITERATURE REVIEW

2.1 Aluminum Alloy Columns

Yuanqing et al. (2017), experimental and numerical and numerical investigations of the behavior of extruded aluminum alloy I section columns with fixed pin end conditions against compression buckling have been conducted. A total of 11 column tests, including two pieces of heat-treated aluminum 6061-T6 and 6063-T5, are performed to obtain the results of compressed buckling strengths. Before the column tests are performed, the material properties of the two aluminum alloys are decided and determined from tensile coupon tests, while the first local and global geometric defects are measured differently by experimental techniques to reach different values. Using the ABAQUS software system in a comfortable way, finite element (FE) models have been developed to help explain the linearity of the material and the defects found in the initial shape, ie geometry. FE models also provide reliable simulation of the tightened buckling behavior with the fixed connection end conditions of the columns verified, verified and tested according to the test results given as a result of the entered data. Based on its test and numerical results, all calculation methods in current design standards, including European, Chinese, American and Australian / New Zealand specifications, are evaluated in detail. It has been shown that the design provisions of all four standards provide relatively conservative strength estimates, especially for aluminum alloys with a tensile hardening capacity that makes them more specific.

Studies have been made to develop a finite element model to investigate the buckling movements of some aluminum columns exposed to fire. The results of this model were verified and validated in the results of the experiments. Considering a parametric study with the help of the finite element model, it was concluded in EN 1999-1-2 that as a result of the bending movements of the aluminum columns exposed to fire, the simple calculation model could not give accurate estimation of the resistance against buckling in fire. This paper proposes a different way of suggesting an alternative design model by emphasizing the stress-strain relationship of aluminum alloys at high temperatures. Estimates of this model give approximate values to the results found in the finite element model (Maljhaars, 2009).

Wang et al. (2017) showed that the failure behavior of the aluminum alloy elements which are faced with the axial compression load and failed in the general buckling mode is moved and focused on the finite element simulation. The column elements play a major role in extrusion production using approximately a new heat-treated aluminum alloy 6082-T6. Thus, it is not difficult to say that it is produced. We have tried to develop an accurate finite element model for the simulation of aluminum alloy columns and have been quite successful. The developed finite element model shows that it can predict the damaged shapes and final loads of the columns tested according to the experimental results, but some notes have been entered. A parametric study has been carried out on the stability of aluminum alloy column elements which fail with general alloys. A suitable column curve is recommended for the aluminum alloy design; it is slightly more conservative than the current Chinese curve in GB50429.

Rasmussen et al. al (2000) forms a column curve formulation that can produce accurate strength for extruded aluminum alloys that fail to bend bending. The formulation uses a simple extension of the Perry curve and checked its validity for all alloys used in practice. It is believed that the material properties are typically expressed in terms of Ramberg-Osgood parameters obtained from a stud column test of the finished product. First, the formulation has been shown to be able to produce ECCS a, b- and c-column curves closely for aluminum alloys. Secondly, a better agreement is reached by accepting ECCS's recommendations based on the column curve selection based on the alloy type (heat treated or not heat treated). In addition, the tests can be obtained using the column curve formulation proposed on paper compared to the column curves of ISO.

2.2 Deep Learning

Deep learning is a machine learning type that try to teaches computers to do what comes naturally to humans; learn by using example. Deep learning is the clef technology behind driverless cars, unmanned aerial vehicles, enabling them to get to know a stop sign or to seperate a pedestrian from a lamppost. It is the clef to voice control in user devices like phones, tablets, wireless headset, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It is achieving results that are not possible before.

Deep learning has attracted a lot of attention in recent years and for good reasons. Results not previously possible. Recent advances in deep learning have improved to the point where deep learning performs better than humans in some tasks, such as classifying objects in images. While deep learning is first theoretical in the 1980s, there are two main reasons why it has been useful recently; Deep learning must have a large amount of tagged data in the system. For example, the development of vehicles without drivers requires millions of photos, images, pictures and thousands of hours of video. Deep learning requires significant computing power; High-performance GPUs have a parallel architecture that is efficient for deep learning. When combined with clusters or cloud computing, this allows development teams to reduce training time from weeks to hours or less for a deep learning network (Abdulkader, 2019).

Shalev-Shwartz and Ben-David (2014) explained that a machine can as well memorize e-mails which are labelled as spam and non-spam in order to establish relations and differentiate the types from each other in future encounters, which in the end makes a perfect example for “Machine Learning”. The term “Machine Learning” became more popular in the past decades and a conceptually well-known approach among many individuals.

The methodologies introduced and developed in the style of machine learning enable a large number of automation capabilities to be used to solve many problems in different research areas. Again, different types of algorithms are used to find different solutions to problems related to machine learning and give different perspectives to research or problem area. (İnce, 2019).

2.3 Overview Studies

Çevik et al. (2009) tried to apply soft calculation techniques for the calculation of strength, ie strength, of heat-treated rolled aluminum alloy columns that failed with bending buckling. Neural networks (NN) and genetic programming (GP) are used as computational techniques on experimental data obtained from the literature. It has been found that the proposed NN and GEP models are more accurate than the existing models and related codes (EC9 and ISO), which are preferred and proposed by Rasmussen from the previous analytical terms and codes applied and used.

According to Üner (2019), the study is investigate whether more information can be extracted from this remaining set of experiments with a deep learning-based approach. They experiment with 6 different deep learning architectures that use gene expression data from the LINCS L1000 project, chemical structure fingerprints of drugs, SMILES string representation of drug structure, and the atomic structure of the drug molecules. The multilayer perceptron (MLP) based model which uses chemical structures and gene expression features achieve 88% microAUC and 79% macro-AUC, thus offering better performance in comparison to the state-of-the-art studies on side effect prediction. They observe that the chemical structure is more predictive than the gene expression profiles despite the fact that the features are extracted with different deep learning models. Finally, the convolutional neural network-based model that uses only SMILES strings of the drugs provides 82% macro-AUC, and 88%micro-AUC improvements, better performing than the models that use gene expression and chemical structure features simultaneously.

Abdulkader (2019), the main purpose of this study is to define and create new VO2max prediction models with the help of deep learning (DL). Data set was tried to be separated into training and test data by using 70-30%, 80-20% split rate and 10 times cross validation. When the comparison was applied, the aim of the study was to develop the VO2max estimation models based on Multi-Layer Sensor (MLP), Support Vector Machine (SVM) and Single Decision Tree (SDT). The performance of the estimation models was evaluated using the Standard Estimation Error (SEE) and Multiple Correlation Coefficient (R). As a result, DL VO2max can be used easily and safely in the desired branch.

Yetiş (2019), studied within the scope of this thesis, 2D floor plans and facade drawings collected and / or produced from scratch are used as data sets. These data are semantically divided into three different Convolutionary Neural Networks to obtain relevant architectural information, because with the widespread use of Deep Learning, it shows promising success in solving a wide range of problems. Semantically segmented drawings are then converted to 3D models using Digital Geometry Processing methods. Finally, a web application was defined on the system in order to allow any user to choose, buy and use 3D models comfortably and easily. In 2D, the results of two different case studies are evaluated for the results given by semantic segmentation and in 3D, and different comparisons are made with different measurement methods to represent the accuracy of the process to the system. As a result, this work has been done and made possible by recommending and applying the most advanced methods, recommending an automated process for the reconstruction of 3D models and making this system easily and easily available even for a person without technical knowledge.

Paker (2019) delineated nowadays with the increase in biological knowledge, the use of deep learning in bioinformatics and computational biology has increased. Deep learning is widely used to classify and analyze biological sequences. In recent years, deep neural network architectures such as Convolutional and Recurrent Neural Networks have been developed to achieve more successful results compared to classical machine learning algorithms. The problem discussed in this thesis is a bioinformatics problem. Therefore, it is discussed whether the given microRNA molecule binds to the mRNA molecule. MicroRNAs (miRNAs) are ~ 21-23 base length non-coding RNA molecules that play an important role in gene expression. After transcription, they target mRNAs and cause mRNA degradation or translation inhibition. Rapid and efficient determination of the binding sites of miRNAs is a major problem in bioinformatics. In this thesis, a deep learning approach based on Long Short Term Memory (LSTM) has been developed with the help of an existing duplex sequence model. The study provides a comparative approach based on different data sets and configurations. In addition, a web interface has been developed to efficiently and quickly identify human miRNA target regions and provide a visual interface to the end user. Compared to the six classical machine learning methods, the proposed LSTM model gives better results in terms of some evaluation criteria.

Pala (2019) has sought to systematically compare systematic videos using challenging conditions and conditions using some of the deeper learning-based methods recently developed to assist the facial recognition system and make it work better and more comfortably. Openface, VGGFace2 and Arcface tried to make evaluations in 3 different deep learning models and succeeded. The results of the UvA-NEMO video dataset for Pala make a difference in their work by showing that even the most successful deep learning-based face recognition methods provide poor performance in difficult distortions such as noise, blur and contrast.

Tanimu (2019) examined the identification of cracks in infrastructure such as roads and bridges that may cause future problems. The methods of detecting cracks are examined in two groups as destructive and non-destructive. This study aims to analyze the cracks in the materials using ground radar analysis due to its many advantages compared to other methods. In this study, a laboratory environment is created and ground radar and thermal image measurements of various blocks with and without fractures are made in different shapes and materials. After visual and thermal analysis, GPR raw data are analyzed by wavelet transform and entropy methods. Finally, the continuous wavelet transform coefficients are classified, separated and divided into different forms and branches with the help of the possibilities provided by the deep learning methods, especially with convolutional neural network. In addition, they have tried and used a case study of a bridge used for the aforementioned previous research studies to test the methods on a larger scale. The results show that the proposed Wavelet-CNN crack detection method is more useful and better than direct detection of cracks from raw data or b-scan signals. In this way, this system is preferred.

Munir (2019) examined to classify a single handwriting character so that the handwriting text can be translated into a digital form. To classify a complete word or text, the first step is to correctly recognize the lines of text. A text line detection system has been developed that can detect all text lines according to the curvature angle of the text lines by dividing the original image of an A4 scanned document. Each letter image in the text lines is identified and given as an introduction to the deep learning network for later recognition. A set of data from our own handwriting, which includes 2200 images of each letter, is also used, together with a public data set for the educational disruption of the deep learning network. A total of 26 x 7800 = 202, 800 pictures are used for the training of the artificial neural network. A GUI system has been designed that can retrieve a scanned document in image science using MATLAB and provide text line detection, letter detection and letter recognition results.

Süberk (2019) conducted point estimation using the help of deep learning methods of building density in some images of remote sensing optics. The aim of the project is to minimize the mean square error of the estimated intensity by introducing and applying architectural changes to the deep learning network and using and opting for increased educational data. The results obtained or reached show that it is possible to easily and quickly predict the density of the building using vanilla convolutional neural networks (CNN) aids. Adding the sigmoid layer to the system, reducing the network to a small data set and data amplification for easy amplification significantly increases its accuracy in regression. Data amplification is the most accurate or preferred method to reduce the mean square root error in this thesis.

Soydaş (2019) studied the subject of aircraft detection from satellite imagery is discussed, and different neural network architectures based on deep learning technique with traditional methods have been trained and tested. For the test, labeling is performed manually on the images containing the airport regions. An algorithm has been developed for rapid detection and high performance in large scale images, and the use of different architectures and the effects of training methods on success have been examined. In the study, firstly the literature is searched and different approaches are examined and then information about machine learning basics is shared. Deep learning, which is examined as a subtitle of machine learning, is also studied and mentioned. Convolutional neural networks, which constitute the most important point of the backbone of the study, have been introduced into the system and introduced to the system and the basic concepts have been emphasized. Many of the object detection models that work with the help of deep learning techniques use convective neural networks as feature extractors. Therefore, in the methodology section of the study, the sections where CNN and object detection architectures intersect are examined and discussed from different angles and the latest technology detection architectures are examined. The sliding window method and maximum printing algorithms are used safely and easily for the fast and high performance detection of large scale satellite images.

Rasheed (2019) examined the deep learning from different frameworks. Our study then focuses on object detection with a deep learning technique and compares which methods are suitable for the detection of one or more dogs. For object detection, firstly, classification and placement of objects in the image are provided. Object detection is a difficult issue involving computer vision and machine learning. In this study, the most impressive results are achieved by applying various deep learning techniques with the help of machine learning and computer vision, because deep learning has become one of the most popular and widely used methods in research communities. The convolutional neural network is a subtype that easily overcomes some competitions in its views on the computer system in the deep neural network, because good results are easily achieved and are well suited for use in object detection tasks. Based on the latest research of the discoveries in recent research, the technique appropriate for the studies to be made or to be conducted is considered to be the Faster R-CNN (Faster R-CNN) preferred for classifying images from multiple dog species. R-CNN, which is faster for object detection and classification, passes several different and distant paths, especially for detection. The R-CNN method, which is faster especially in terms of accuracy and speed, is generally preferred. This provides a faster R-CNN trained to classify and detect dog images.

According to Çakır (2019), study analyzed the effectiveness of supervised and unattended deep learning algorithms in detecting zero-day attacks. We conducted analyzes on different deep learning models and compared the performance of these algorithms with each other. The models tested are deep learning algorithms, forward neural networks, recursive neural networks and convolutional neural networks. Test results showed that unguided deep learning methods are successful in detecting attacks with an accuracy score of 95.3% and a f1-score of 97%. In addition to the test sets produced in the same environment as the training set, test sets created in different environments from the training set are used to test the deep learning methods. These tests showed that although deep learning methods cannot detect every attack in sets created in different environments, some attacks can detect low false positive rates.

Özgenel (2019) explained data design term as introduced to describe end to end process of problem solving with deep learning algorithms which is suitable for broad range of applications including problems in architecture. Data design defined as a holistic approach embracing the process from problem (re)formulation to evaluation of the results considering the interrelations of decisions made throughout the process. In this context, data design in architecture is exemplified with the task of crack detection in buildings in order to minimize subjectivity in the course of evaluating the results. For this purpose, the relation between data and deep learning framework, case specific evaluation requirements and strategies for enhancing the performance are inspected through image classification and semantic segmentation applications for crack detection. Concordantly, this study contributes to the literature not only with the introduction and framing of data design but also with the proposal of crack detection specific evaluation metrics for both image classification and segmentation applications and a novel method is proposed employing quad tree and deep learning algorithms in conjunction for semantic segmentation of objects with limited visual features.

As a result, data design and respective consequences are discussed in depth and demonstrated regarding the case dependency, decisions taken in the course of implementation and their influences to both process and the results.

Tienin (2019) examined the main purpose of this study is to compare traditional image processing methods with deep learning techniques. First, a two-class classification study is conducted. The aim of this study is to differentiate between cloud images and non-cloud images. Convolutional Neural Networks and SoftmaxWithLoss function are used for estimation and classification. In the training phase, 92% accuracy is achieved after fine tuning of the model. In the next step, detection and segmentation is performed on cloud images using image processing techniques. At this stage, edge detection methods and watershed algorithm are applied. In the third stage, Full Convolutional Networks (THA, FCN) and U-NET deep learning methods are used for segmentation. Segmentation of cloud images is provided by two different deep learning methods. U-NET is found to be 87% as Dice coefficient and 45% as loss during the training phase. FCN network, stochastic slope drop, Adam's momentum and Nesterov momentum techniques have been applied to different training methods and the highest success is achieved with Adam's momentum technique is 63.12%.

Sorkun (2018), focused on statistical time series estimation methods. The aim of this study is to investigate the suitability and competitiveness of time series estimation methods with deep learning on solar radiation data. In this context, time series estimation on solar irradiation is made by using the regenerative neural network variation Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models. By optimizing the parameters, values are found to form the model that best represents the solar radiation data. The seasonal performances of established RNN models and other machine learning methods are compared and a hybrid model is proposed according to experimental results. Finally, the effect of additional meteorological parameters on solar radiation estimation is investigated. The results show that LSTM and GRU models are suitable and competitive for estimating time series over 1 hour horizon on solar radiation data. Experiments have shown that the hybrid approach and additional meteorological parameters improve the performance of the model.

Anwer (2017) defined deep learning which is used in medical filed, to propose deep learning based clinical support system for diagnosis breast cancer. Deep learning is a subfield of machine learning to solve problems of artificial intelligence. This study aims to investigate whether the deep learning approaches are also capable of producing successful results in the medical area. Many of deep learning algorithms have been used in the diagnosis of breast cancer, such as fully connected neural networks and recurrent neural network, the new order is the first time that the new architecture of convolution neural networks is used to diagnosis breast cancer. Wisconsin Breast Cancer datasets from the UCI Machine Learning repository is used to test the ability of deep learning different techniques. The experiments with deep learning approaches show a promising direction towards supporting medical decision making.

Eriş (2018) aimed to use the deep learning methods which have been successful in many use cases in judicial evidence examinations. For this purpose, a method of extracting evidence to assist in forensic examinations is proposed using deep learning object detection algorithms. The results show that the classification of images obtained from digital evidences according to their contents can be performed 93% faster using deep learning. It has also been shown that human errors will be reduced as a result of classification. It is aimed to integrate the proposed method into the forensic evidence analysis software tools by applying it to audio, text and video data as well as image data in further studies.

Kutlu (2019) delineated in recent years, artificial intelligence algorithms are one of the most researched and developed applications. Machine learning, artificial neural networks, classification, clustering algorithms are used in artificial intelligence applications. One of these methods is deep learning. Deep learning is an advanced machine learning class. Using the Deep Learning method, video analysis, image classification, speech recognition and natural language processing are very successful. The data and experiences to be provided by the projects that will cover the areas of Deep Learning and Unmanned Aerial Vehicles will contribute to the development of high value-added products in these technologies by increasing the number of qualified studies on these issues. In this study, a control software that evaluates image data from unmanned aerial vehicles and makes various inferences (classification, positioning, marking) is created. By using the method of retraining the last layers of the pre-trained artificial neural network models with our data set, it is tried to reduce the training time and increase the success. In these studies, 2 pre-educated models are used and as a result of training of these models, as a result of 190 thousand step training, 25.39 and 27.87 mAP values are reached.

Kantepe (2019) producted recommendation system is designed using the deep learning method AutoEncoder. This designed recommendation system is implemented using the Python language on the TensorFlow platform. MovieLens 1M dataset is used which consists of the movie ratings ranging from 1 to 5 and collected by the GroupLens researchers from the users of MovieLens website. The study is written in four sections. Furthermore, it is tried to find the best optimization algorithm to improve the success of the system and the effect of increasing the amount of data that is analyzed. The results of the algorithms of Gradient Descent, Gradient Descent with Momentum, RmsProp and Adam are depicted in both tables and graphs. In the third section, findings are evaluated and as a result, the Adam is shown to be the best algorithm with 1,363 points of test error. It is observed that the more data a training set has, the more successful the recommendation system is. In the fourth section, all the results from the processes mentioned above are summarized and the studies which can be conducted in order to implement a better recommendation system as future plans are stated.

Koyun and Afşin (2017), used 2160000 characters to train the convolutional neural network. The authors state that the method developed by themselves is better when compared to the OCR tool embedded in the Matlab environment.

Uçar and Bingöl (2018), have briefly introduced the layers of DKSA (Deep Convolutional Neural Networks), which is a kind of deep learning and is widely used in image processing applications, and gave information about their architecture. Using the Caffe program to implement DKSAs, the authors built the applications on two computers with embedded Nvidia Jetson TK1 / TX1 development cards and Nvidia GTX550 / GTX960 graphics cards. On the cards and the computer, they use the LeNet network to recognize handwritten numbers, and in their assessment of performance, speed and accuracy, they stated that GPUs are faster than CPUs, although they are close to accuracy.

In the study conducted by Büber and Şahingöz in 2017, it is aimed to perform image processing by using deep learning approach and to recognize numerical characters. In the application, 50000 data from MNIST data set consisting of handwriting pictures taken from 250 different people are used for training and 10000 data are used for testing. In addition, the effects of hyperparameters on the performance of the algorithm should be determined before using the deep learning method. The authors stated that the success rate of the numerical hyper parameters determined to be used in the test step for the best case is 94.75%.

In 2016, Kaynar et al. presented a deep learning-based approach to automatically detect spam mail. Authors using 2 different data sets used 75% of the data in Turkish and English data sets for educational purposes and 25% for testing purposes. The proposed model is a classifier with 2 deep auto encoders connected back to back and a softmax layer at the outputs. This classifier has a 98% success rate.

In the studies conducted by Işık and Artuner in 2016, it is aimed to identify the radio signals received through Software Based Radio with deep learning neural networks. The authors made a convolutional neural network study with the images obtained from the Signal Identification Wiki and used 82 of 137 images for training, 22 for verification and the remaining 33 for testing. It is stated that 82% performance is achieved at the end of the operation of the Caffe platform-trained network.

Light and Artuner (2016) worked on recognition of radio signals with deep learning Neural Networks. In their work, the authors state that they aim to correctly identify the properties of fabrics with a distinctive texture rather than detecting fabric defects. They used 1000 sample data, 153 of which are incorrect during the training phase, and 100 of which 60 are error-free during the testing phase. The authors concluded that the proposed method yielded 88% success.

In 2017, the study conducted by Razavi and Yalçın aimed to identify the plant type from images collected from smart agricultural stations. The authors proposed a deep learning method that can automatically extract attributes from two-dimensional plant images. In the construction of CNN architecture, 4800 images of 16 plant classes in TARBIL data set are used. The authors state that 3 different deep learning methods are used, the CNN-based approach works on 16 kinds of plants with an accuracy of approximately 97.47% and the classification accuracy is better than other methods. Yalçın examined to automatically identify human activities in RGB-D video images. Yalçın wanted to classify the activities of people from skeletal movements and deep learning through 3D skeletal joint data. The raw data from human images are pre-processed with operations such as translation, rotation and scaling. Using the Human3.6M activity data set for the training of the proposed method, the author used a model with 5 hidden layers as a deep understanding network, 47 neurons per layer, a non-instructional learning rate of 0.001 and an instructional learning rate of 0.12.

Arık et al. 2017, delineated a system that classifies skin lesions for the detection of melanoma by deep learning methods is proposed. In this study, approximately 65% of 1279 healthy and diseased images from ISIC archive are used for education and 35% for testing. The authors stated that because of the data limitation, the system they recommended is 70% successful and expert dermatologists predicted the disease at an average rate of 75%.

Yetkin and Hamamcı (2016), worked on a deep learning-based method for estimating the relative exposure between the patient's planar image and the previously taken head MR images is presented. In practice, the MR volume obtained from a healthy subject using the 3D gradient-echo sequence is used. After this volume is rendered in 3D, 324 images are obtained with 18 different light angles, and 260 images are used for training and 64 for testing in the convolutional neural network.

(Shahbazi et al. 2014) studied the elastic buckling of smart lightweight column structures integrated with a pair of surface piezoelectric layers using artificial intelligence. The finite element modeling of Smart lightweight columns is found using ANSYS® software. Then, the first buckling load of the structure is calculated using eigenvalue buckling analysis. To determine the accuracy of the present finite element analysis, a compression study is carried out with literature. Later, parametric studies for length variations, width, and thickness of the elastic core and of the piezoelectric outer layers are performed and the associated buckling load data sets for artificial intelligence are gathered. Finally, the application of soft computing-based methods including artificial neural network (ANN), fuzzy inference system (FIS), and adaptive neuro fuzzy inference system (ANFIS) are carried out. The comparison of the results reveal that, the ANFIS model with Gaussian membership function provides high accuracy on the prediction of the buckling load in smart lightweight columns, providing better predictions compared to other methods. However, the results obtained from the ANN model using the feed-forward algorithm are also accurate and reliable.

Yuan and Bao (2018) assessed a guided wave (GW)-CNN based fatigue crack diagnosis method is proposed. GW features extracted from GW signals of different excitation-acquiring paths with different excitation frequencies are employed as the input of a trained CNN for fatigue crack diagnosis. An experiment verification is performed on a kind of attachment lug specimen, which is an important connecting component on aircraft structures. The results show that this proposed method is promising.

Shon et al. developed a deep conditional generative model for structured output prediction using Gaussian latent variables. The model is trained efficiently in the framework of stochastic gradient variational Bayes, and allows for fast prediction using stochastic feed-forward inference. In experiments, we demonstrate the effectiveness of our proposed algorithm in comparison to the deterministic deep neural network counterparts in generating diverse but realistic structured output predictions using stochastic inference. Furthermore, the proposed training methods are complimentary, which leads to strong pixel-level object segmentation and semantic labeling performance on Caltech-UCSD Birds 200 and the subset of Labeled Faces in the Wild dataset.

Merwe, analyzed the most effective column design method is to be found by comparing the different column design criteria and methods of design which are in use for different materials. The materials under consideration are carbon and low-alloy steels (structural steels), stainless steels and an aluminum alloy that is regarded as suitable for structural design.

Durmuş et. al (2006) Effects of ageing conditions at various temperatures, load, sliding speed, abrasive grit diameter in 6351 aluminum alloy have been investigated by using artificial neural networks. The experimental results are trained in an ANNs program and the results are compared with experimental values. It is observed that the experimental results coincided with ANNs results.

Hassan et. al (2009) studied the potential of using feed forward backpropagation neural network in prediction of some physical properties and hardness of aluminum–copper/silicon carbide composites synthesized by compocasting method. Two input vectors are used in the construction of proposed network; namely weight percentage of the copper and volume fraction of the reinforced particles. Density, porosity and hardness are the three outputs developed fromthe proposed network. Effects of addition of copper as alloying element and silicon carbide as reinforcement particles to Al–4 wt.% Mg metal matrix have been investigated by using artificial neural networks. The maximum absolute relative error for predicted values does not exceed 5.99%. Therefore, by using ANN outputs, satisfactory results can be estimated rather than measured and hence reduce testing time and cost.

Kechagias et. al assesed Al7075 alloy surface quality, achieved in slot milling, constitutes the subject of the current research study. Twenty seven slots are cut using different cutting conditions by a KC633M drill-slot end mill cutter. The independent variables considered are the depth of cut (ap, mm), cutting speed (Vc, m/min), and feed rate (f,mm/rev). Process performance is estimated using the statistical surface texture parameters Ra, RSm, and Rt; all measured in microns. To predict the surface roughness within the limits of the parameters involved, an artificial feed forward back propagation neural network model is designed for the data obtained.

CHAPTER 3
MATERIALS & METHODS

3.1 Aluminum Alloy Columns

Aluminum tubular elements are used in curtain walls, space structures and other structural applications. Aluminum tubular elements are normally made of aluminum for heat treatment because heat treatment alloys have higher yield stress than important properties of non-heat treated alloys. Aluminum structural elements are specially based on ultimate strength of compact sections. Because of this, it is necessary to investigate the behavior and design of aluminum columns, plates, beams and beam-columns of slender sections. Although aluminum alloys are well suited for a variety of applications in structures, their unique material properties make structural reactions different from steel. In the literature, important experimental researches about Aluminum column test have been made. Although aluminum alloys are well suited for some applications in marine structures, their unique material characteristics make the structural response different than steel. When structural elements are subjected to compressive loads, the buckling and collapse capacity is one of the most crucial factors governing the design. Ultimate strength of longitudinally stiffened panels is very important because it governs the structural capacity. Such panels are subjected to longitudinal compression, transverse compression, shear, and local bending. As the mechanical properties of aluminum alloy typically vary more significantly between the parent metal, weld metal and HAZ (Heat Affected Zone), as compared to those of steel, it would be anticipated that the existing formulations for steel structures may not be accurate when applied to aluminum alloy panels. The scope of research on the extensive experimental data and numerical data made in the European Construction Steels Convention (ECCS) in the 1960s and 1970s. Based on the results of these tests, column curves of welds referred to as ECCS a-, b- and c-curves (ECCS, 1978) are recommended for aluminum alloys, where a- and b-curves are accepted by ECCS. Heat treated and non-heat treated alloys, respectively. The major difference in the sequence of different curves for all heat-treated and non-heat-treated alloys is that the greater difference in softening of non-heat-treated alloys is greater than for heat-treated alloys. ECCS column curves cannot be used in accordance with the design as they are in tabular form.

Liu et. al. 2015 presented experimental and quantitative research on the buckling behavior of shape alloyed aluminum columns under axial compression. The structural component having the cross section examined is generally used as columns in an aluminum alloy framed structure as shown in Figure 3.1 (a). The Fiber Reinforced Plastic (FRP) wall can be easily fixed to the channel of the section as shown in Figure 3.1 (b).

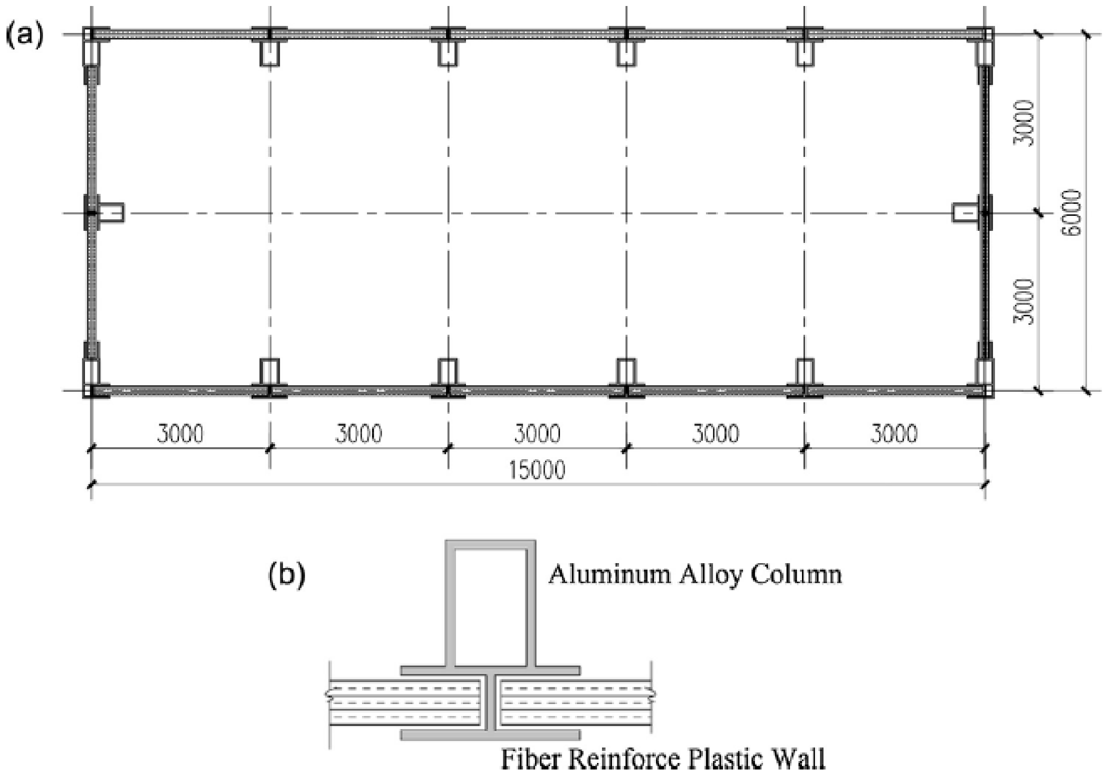

Figure 3.1. Application of aluminum alloy column with section.

Advantages of using aluminum alloys as a structural material are high strength / weight ratio, formability, electrical-thermal conductivity, recycle, light weight, corrosion resistance and production purpose. There are the disadvantages of using aluminum alloys for structural applications, such as the aluminum modulus, which is roughly one third of the steel and causes the aluminum element to fail easily by buckling. Previous research of aluminum structural elements focused mainly on the ultimate strength of compact (non-thin) sections. However, the use of aluminum thin-walled sections has increased in recent years. In aluminum structures, welds are divided into two types: (1) transverse welds, (2) longitudinal welds, to share their effects on element strength. While transverse welds are often used in connections, longitudinal welds are used for the production of built-up elements. However, there is little research on the behavior of aluminum columns containing transverse welds.

Therefore, it is necessary to investigate the behavior and design of aluminum columns, beams and beam columns of thin sections.

3.2 Artificial Intelligence (AI)

From the beginning of time to the end, it went hand in hand with technological advances to improve the lives of people. Since the first tools and the discovery of the wheel, technology, the concrete offspring of human intelligence, has made humanity longevity, comfort, and provided means for understanding the cosmos that surrounds it (Makaritou, 2019). The disrupting processes are named Revolutions denoting that those events are abrupt and radical in nature. The progress of a human kind and civilizations has been driven by the progress of technology. Though, perception of a word technology of a modern person might be associated more with a digital technology, such as computers, internet and smartphones, for the ancient people, their technology are invention of the wheel and utilization of simple tools (Rahimov, 2019). In fact, since human beings survive in a digital age, it is very important to examine and investigate related issues further and make logical predictions to come to a conclusion and eventually prepare for developments in future differences in industries. Artificial intelligence can be broadly defined as the intelligence formed by software programs in machines, which can be seen as similar to the natural intelligence shown in humans and other animals. In practice, intelligence is attributed to artificial entities when those entities simulate human-like functions such as learning and problem solving features (Özmen, 2019). Artificial intelligence applications are used in many areas to make everyday life more convenient and efficient. They are used to clean houses, take care of old or sick people, to work at dangerous jobs replacing human beings, to give medical advisory and healthcare services, to prevent fraudulent situations in finance, etc. AI is just one of the areas of computer science that aims to create intelligent machines that work and respond, such as human preparation, solutions to problems, speaking, recognizing, introducing, teaching and learning. Also, Artificial intelligence is a branch of computer science that aims to create intelligent machines. It has become an necessary part of the technology production. Research related with artificial intelligence is highly technical and specialized. The core problems of artificial intelligence contain programming computers for certain traits such as: Information, sensivity, becomes skilled, program solving, ability to manipulate and move objects (Okyay, 2018).

Also it can be defined as, ANN is a programming technique that uses a large number of cells that mimic the functions of the human brain and functional connections between these cells. The functions on the created network (Figure 3.2) are optimized by changing the variables on the neurons that symbolize each cell. Using the input and output data obtained, ANN is trained and it is tried to reduce the error between the outputs of the ANN and the outputs that should have. When this error level goes down to the minimum level, ANN training is accepted as complete (Kara, 2019). Artificial neural networks (ANNs) or connective systems can be used in conjunction with the so-called unlimited inspiration and intelligent computer systems using biological neural networks. Such systems or programs "learn" to successfully consider tasks, taking into account examples, without programming a particular system with specific rules. In recognizing images in the system, they can define the desired contents by analyzing the sample images entered in the system in a healthy way. For example, they can learn to identify images that contain cats. Using "cat" or "no cat" in the entered data according to the desired result and using the results to identify cats in other images. They do this without prior knowledge of cats, such as feathers, sharp ears, tail, whiskers and cat-like faces. Instead, they automatically generate descriptive properties from the samples they process. An ANN is based on a collection of connected units or nodes called artificial neurons that loosely model neurons in a brain in a biological system. In this collection, each neuron has connection tasks. Each link, ie neurons, can transmit a signal to other neurons, ie other links, such as synapses in a biological brain. An artificial neuron that receives a signal can then process it, understand it, and point to the neurons connected to it. In this way, they can work in a systematic way depending on each other. This enables communication between neurons. In embodiments of the ANN system, the signal in a connection is a real number, and the output of information from each neuron is calculated by a nonlinear function of the sum of its inputs. These functions are calculated differently. These connections are called edges. Knowledgeable neurons and so-called connecting edges typically have a weight adjusted as learning progresses. Weight increases or decreases the strength of the signal in a link.

Neurons may have a threshold value at which a signal is sent only when the sum of the signals delivered exceeds this threshold. That is to say, the situation depends on the sum of the inputs. Typically, neurons exist in layers and are processed into the system in a healthy manner by means of edges. Different layers can perform different transformations in their input. Sometimes these conversions can be achieved. The signals probably cross the layers more than once, and once this transition is made, the signals go from the first layer (the input layer) to the last layer (the output layer) as expected, and the system is completed. The major objective of the ANN approach is to solve the problems systematically, that is, not artificially, but as if they are natural within a human brain. However, over time, attention led to the fulfillment of specific tasks and deviations from biology. Over time, these problems led to different functions. ANNs have been used in a variety of tasks, including computer vision, communication, speech, processing, machine translation, social networks filtering for people, video games, medical diagnostics and even traditionally considered activities reserved for people. Over time, these tasks have made people's lives easier.

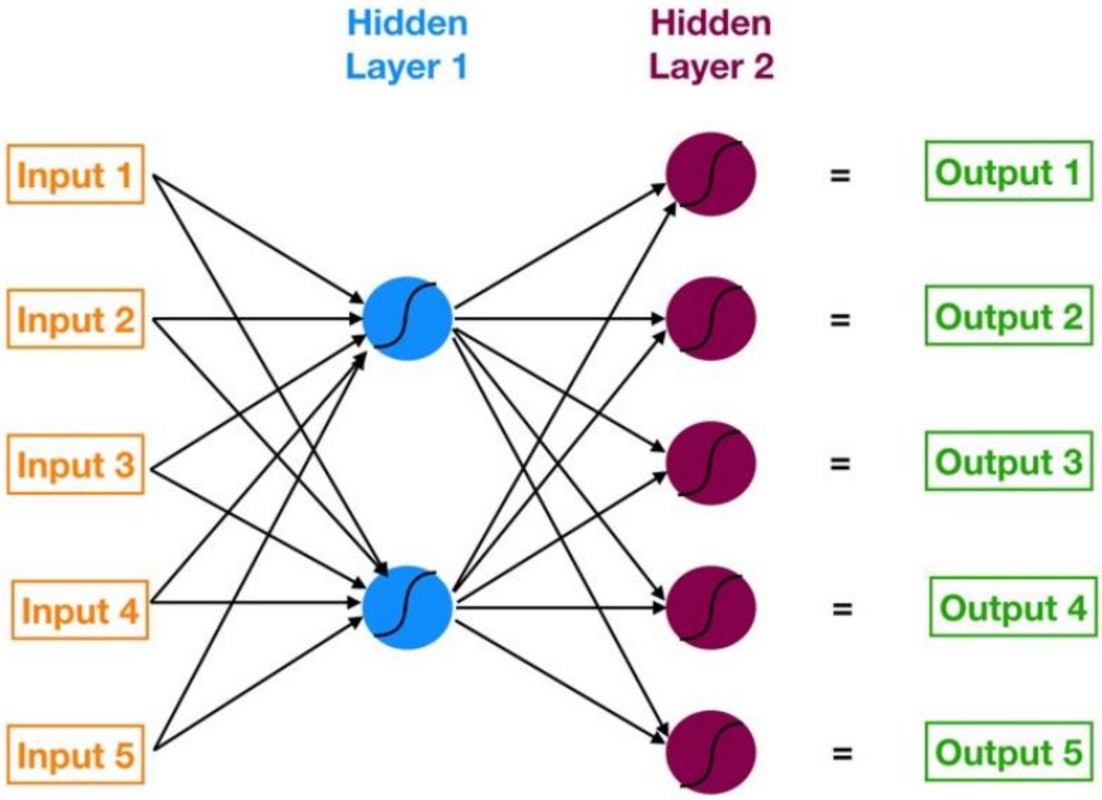

Figure 3.2. General view of an ANN.

Artificial neural networks are another data mining technique that should be used for estimation and classification. Neural networks are signal processing systems that attempt to mimic the biological nervous system by presenting a numerical data model of multiple neuronal compositions connected to a network (Haykin & Lippmann, 1994). Neurons founded in the input and output layers and hidden layers or layers, if any. When a neuron significantly entered the system, the weights associated with that neuron can change. This means that the neuron will be more effective than other neurons of the same level as the neuron itself. Artificial neural networks learn by adjusting the weights between neurons. (Mumbuçoğlu, 2019).

3.3 Deep Learning

Deep Learning history can be traced back to 1943 when a computer model is developed by Walter Pitts and Warren McCulloch based on neural networks, with imitation of human brain. They used a combination of algorithms and mathematics they called “threshold logic” to mimic the thought process. Furthermore, in the 90s Yann LeCun & Yoshua Bengio (1995) take a further step from natural Multilayer Perceptron (MLP) with the sole purpose to reduce both high computational tasks and availability of high number of dimensions. Subsequent to that, a paper is published that presented a better pattern recognition system by canceling unrelated variables (Kamachy, 2019).

With the importance and popularity of analysis task, lots of algorithms have been tried for having better scores for this. Various natural language processing and machine learning techniques are applied for accuracy classification. Recently, deep learning algorithms like CNN and LSTM networks have shown state of the good scores in accuracy analysis. These scores are high in the literature but they are not satisfactory for the implementation needs in real life. Therefore, it is an important interest area for researchers. People have been trying different combinations and ensembles of these deep learning methods for increasing accuracy (Kamış, 2019).

Structurally, deep learning can be thought as a more complex form of artificial neural networks (ANN). There is a feed-forward structure in traditional neural networks. Each input neuron has a hidden layer to which each neuron is attached, and each neuron in the hidden layer is connected to each neuron in the output layer. For classification problems, the number of units in the output layer is equal 3 to the number of classes. Typically, a single linear output neuron is used to predict a continuous output. All connections in the network are lead from the input layer to the output layer, and it is possible to create deep networks by adding more hidden layers to which each neuron binds to each neuron in the following layer. Deep learning provides a very strong framework for the supervised learning process. By adding more units to a layer, a deep network can represent increased complexity functions. It is easy to match an input vector to an output vector, with the help of human easily. But, that process can be accomplished through deep learning model with modeled large data sets and exemplary training samples. To put it more simply, classical (simple) neural networks have only one hidden layer, whereas deep neural networks have more than one hidden layer (Figure 3.3) (Paker, 2019).

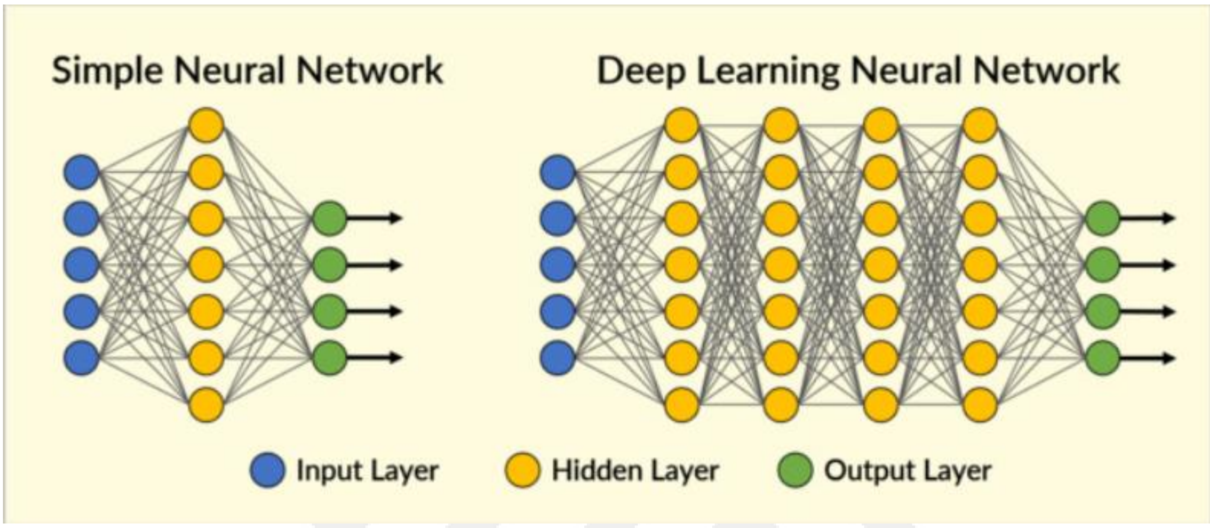

Figure 3.3. Difference between a simple neural network and a deep learning neural network.

The main definition of DL is that it is a Neural Network with many hidden layers, so "deep" here refers to the depth of layers consisting more than 2 hidden layers. It provides automatic feature extraction by determining the properties of the input data which can be used as a pointer to label the input data accurately. Each layer extracts features from the output of previous layer. This is a revolution in computer vision tasks. In contrast, shallow networks required manual designs for feature extraction and a great amount of experience in the image processing field. On the other hand, transformations from input images to vectors led to losses of much interesting information.

There are 7 main applications that are mainly practiced in DL (Hordri et. al 2016):

- Automatic Speech Recognition (ASR)
- Image Recognition
- Natural Language Processing
- Drug Discovery and Toxicology
- Customer Relationship Management

 Recommendation Systems

 Bioinformatics

 Deep Learning Frameworks

 Tensor Flow

 Theano

 Keras

 Torch

 Caff

On the other hand, Deep learning is part of a family of artificial neural networks based on a broader and more useful machine and learning methods. In this way, healthy learning systems are established by working with artificial neural networks together with deep learning. Learning may not always be controlled. Learning in different forms can be controlled, semi-controlled or not. Deep learning architectures such as deep neural networks, deep belief networks, repetitive neural networks and convoluted neural networks have been applied to fields such as computer vision, speech, communication, introduction, recognition, natural language processing, voice recognition, machine translation, bioinformatics, drug design. Deep learning has made human life easier as can be predicted after entering human life. Medical image analysis, material examination, and board game programs are comparable to human experts here, and in some cases have produced superior results. Usage area is increasing day by day.

Deep Learning is a machine learning method. It allows us to train the data set and the artificial intelligence to predict the output after a given or desired information is input. In this way, it provides convenient classifications and to reach the result after the required details are entered in the required information as soon as possible. Both supervised and unsupervised learning can be used to train artificial intelligence. The more data input, the better artificial intelligence features will be revealed. Because as classifications enter, the desired data will be reached more easily, but on the other hand, as the data increases, things will become more complex, and as they become complex, undesirable shifts from artificial intelligence to machine learning will occur. As it becomes more complex, the shifts from machine learning to deep learning will begin and user will see how useful learning is. In the machine learning, the experiences of human beings to date are introduced to the machine by means of parameters. Because how much data, parameters, experience is entered in technology, results will increase in direct proportion. After entering the information such as pear and apple shape, color, size, stalk, the result will be easily reached. However, deep learning can learn this different on its own. Only by showing the apple and pear to the deep learning system itself creates its own rules, to reveal the differences in color and shape are the main distinguishing features itself. Thus, it can perform its operations by creating its own discriminatory abilities without the need of basic human abilities.

3.3.1 Deep Learning Library

The so-called deep learning library is actually a library of software systems. There are multiple systems, soft wares. These are ; TensorFlow(Python), Caffe(Python), Theano(Python), Keras(Python), Torch(C++), Deeplearning4j(Java), Covnetjs(Java), Mxnet(Python), PyLearn2(Python), Deep Learn Toolbox-Matlab, Accord.NET(C#), Sci-Kit Learn(Python), Accord.MachineLearning. These software systems work in different ways from each other. The common points are software, many of which are used for deep learning.

3.4 Keras

Keras is a python library for deep learning. It also works with Theano or Tensorflow libraries, which are also based on symbolic operations and used for deep learning. GPU or CPU work on these basic libraries and helps to reach a conclusion. Since it is a higher level library, user can develop applications more easily than Theano or Tensorflow. It is quite common use. It is a library that can be easily used by many developers around the world. Also many examples can even find about the problem in the Kaggle competitions. There are two main API structures in Keras. The models to be installed can be designed by installing one of these two different main structures. Sequential models have to be designed in layers. But it is more simple and understandable, comfortable and easy to use. Functional structure is designed as functions. It makes it possible to design more flexible and advanced models. Since the input layer data will be used in the structure call the model, the input size of the data must be specified. It does not need to be specified for, because the other layers receive data from the preceding layer. To get the results of the output layer, the output size must be determined. The output size is generally the output size of the layer before the last activation layer.

3.4.1 Keras Models

There are two ways to create Keras models: sequential and functional.The sequential API gives user opportunities to create multiple layer models for most problems.It is limited that it does not allow models that share layers or have multiple inputs or outputs.Alternatively, the functional API allows to create models with much more flexibility because they can easily identify models where layers are connected to more layers than only previous and later layers.In fact, user can connect layers to another layer (literally). As a result, siamese networks and now it becomes possible to create complex networks such as networks. Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error. This makes Keras easy to learn and easy to use. As a Keras user, user are more productive, allowing user to try more ideas than userr competition, faster which in turn helps user win machine learning competitions. This ease of use does not come at the cost of reduced flexibility: because Keras integrates with lower-level deep learning languages (in particular TensorFlow), it enables to implement anything user could have built in the base language. In particular, as keras, the Keras API integrates seamlessly with userr Tensor Flow workflows (Figure 3.4).

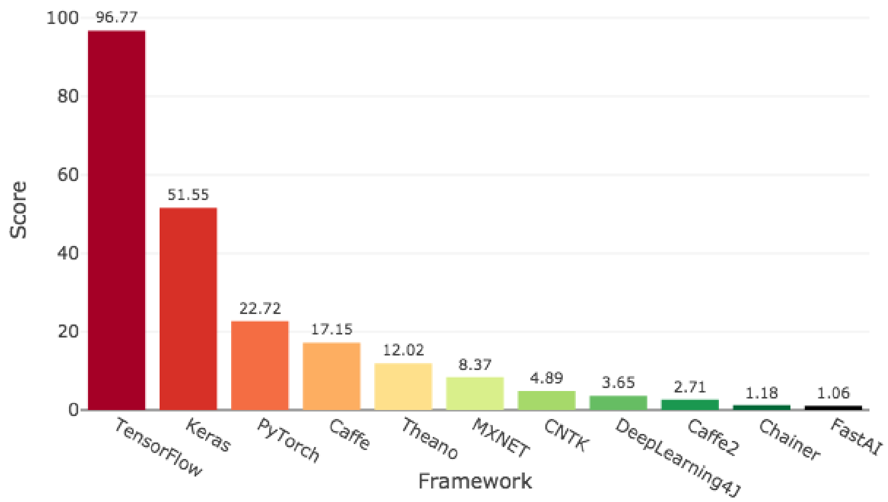

Figure 3.4. Deep learning frameworks ranking.

With over 250,000 individual users as of mid-2018, Keras has stronger adoption in both the industry and the research community than any other deep learning framework except TensorFlow itself (and the Keras API is the official frontend of TensorFlow, via the tf.keras module).User are already constantly interacting with features built with Keras -- it is in use at Netflix, Uber, Yelp, Instacart, Zocdoc, Square, and many others. It is especially popular among startups that place deep learning at the core of their products. Keras is also a favorite among deep learning researchers, coming in terms of mentions in scientific papers uploaded to the preprint some server. Keras has also been adopted by researchers at large scientific organizations, in particular CERN and NASA.

3.4.1.1 Functional API

The Keras functional API is the way to go for defining complex models, such as multi-output models, directed acyclic graphs, or models with shared layers. This guide already familiar with the Sequential model. A layer instance is callable (on a tensor), and it returns a tensor. Input tensor(s) and output tensor(s) can then be used to define, such a model can be trained just like Keras Sequential models. With the functional API, it is easy to reuse trained models: Any model can treat as if it are a layer, by calling it on a tensor. Note that by calling a model just reusing the architecture of the model, user are also reusing its weights.

3.4.1.2 Methods of Sequential

Choosing the right file editing system for database records or digital files affects how much data can be done and how useful and efficient user system is. The more organized the files are, the more efficient the system works. Sequential file organization means that computers store their data, information, or files in a specific file, not in a location, but in a specific order based on the data or file type. This sequence is processed regularly and makes the system useful. Sequential file organization is transparent to the user, and the methods of organizing sequential files work with a variety of data and different operating system environments. It helps the system to be preferred according to the needs and to install files to suit the preferred system, to obtain effective data and to convert it into different ways. The simplest sequential file organization is to save the files in the order in which user create them. The first file is saved first, the next is second and so on. This method is straight forward and it lets user to delete files without disturbing the sequence. While the method lets user place newly created files at the end of the sequence, there is no indication of the location of information. The low complexity of the organizational method means finding information takes a long time since a search has to look through every file, starting with the first one. Indexed sequential files solve this problem but at a cost of organizational simplicity. The system lists files or data in a pre-determined order, such as alphabetically. When user creates a file, the system either has to insert the new file in the proper sequence or re-index the whole file list. It takes longer to create new files and place them in the index, but it takes less time to find information because user know where in the indexed system to start looking. When designing a sequential file system, user have to look at how often user need to search for data and how often user plan to add or change files. Archival-type systems constantly grow as user adds new files but searches are comparatively rare. A system that files the information in sequence, as user archives it, is appropriate. A call-center database is relatively static but subject to constant searches. It makes sense to use an alphabetical or other index that lets user find specific records quickly. The two critical parameters for sequential file organization methods are speed and storage space. When user store files sequentially, there is no room to later make a file bigger. The system either has to store the additional information elsewhere or it has to make room and resave all the files. Both methods take a lot of additional time. Sequential file systems solve this problem by reserving extra space for each file when the system creates it. This avoids the delay but uses up a lot more space. When setting up the system, user has to decide whether time saved or space used is more valuable and use the corresponding method to organize the files.

The method of sequential analysis is first attributed to Abraham Wald with Jacob Wolfowitz, W. Allen Wallis, and Milton Friedman at Columbia University's Statistical Research Group. At the same time, George Barnard led a group working on optional stopping in Great Britain. Another early contribution to the method is made by K.J. Arrow with D. Blackwell and M.A. Girshick. A similar approach is independently developed from first principles at about the same time by Alan Turing, as part of the Banburismus technique used at Bletchley Park, to test hypotheses about whether different messages coded by German Enigma machines should be connected and analysed together. This work remained secret until the early 1980s.Peter Armitage introduced the use of sequential analysis in medical research, especially in the area of clinical trials. Sequential methods became increasingly popular in medicine following Stuart Pocock's work that provided clear recommendations on how to control Type 1 error rates in sequential designs.

3.5 Optimizers of Sequential

Adaptive Moment Estimation (Adam) is another method that computes adaptive learning rates for each parameter. In addition to storing an exponentially decaying average of past squared gradients Vt like Adadelta and RMSprop, Adam also keeps an exponentially decaying average of past gradients mt, similar to momentum. Norms for large pp values generally become numerically unstable, which is why ℓ1ℓ1 and ℓ2ℓ2 norms are most common in practice. However, ℓ∞ℓ∞ also generally exhibits stable behavior. For this reason, the authors propose AdaMax (Kingma and Ba, 2015) and show that vtvt with ℓ∞ℓ∞ converges to the following more stable value. As we have seen before, Adam can be viewed as a combination of RMSprop and momentum: RMSprop contributes the exponentially decaying average of past squared gradients vtvt, while momentum accounts for the exponentially decaying average of past gradients mtmt. We have also seen that Nesterov accelerated gradient (NAG) is superior to vanilla momentum.

Given the ubiquity of large-scale data solutions and the availability of low-commodity clusters, distributing SGD to speed it up further is an obvious choice. SGD by itself is inherently sequential: Step-by-step, we progress further towards the minimum. Running it provides good convergence but can be slow particularly on large datasets. In contrast, running SGD asynchronously is faster, but suboptimal communication between workers can lead to poor convergence. Additionally, we can also parallelize SGD on one machine without the need for a large computing cluster. The following are algorithms and architectures that have been proposed to optimize parallelized and distributed SGD.

Whereas momentum can be seen as a ball running down a slope, Adam behaves like a heavy ball with friction, which thus prefers flat minima in the error surface Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions, is introduced based on the thoughts and predictions made of adjustable simple moments. The procedures that need to be performed are easy to implement and pour into tools, are also very efficient in terms of calculations, require little memory, do not change in diagonal rescaling of gradients and are well suited for large problems in terms of data and / or parameters. This method is also suitable for non-stationary purposes and problems with very noisy and / or sparse gradients. Hyper parameters have intuitive interpretations and typically require minimal adjustments and adjustments. Empirical results show that Adam works well in practice and is compared to other stochastic optimization methods. Finally, AdaMax, a variant of Adam based on the eternity norm (Kingma, 2014).

Reddi, Kale and Kumar is working More recently, stochastic optimization methods, which have been successfully and successfully used in the training of deep networks such as RMSProp, Adam, Adadelta, Nadam, have been imposed on using gradient updates scaled with square roots of exponential moving averages of square past gradients. In many types of applications, e.g. learning with large output spaces, empirically it has been observed that these algorithms cannot approach an optimal result and solution (or a critical point in non-convex environments). One reason for such errors is the exponential moving average used in algorithms. Here is a clear example of a simple convex optimization setting where Adam cannot merge with the optimal solution, and user can see the exact problems in the previous analysis of the Adam algorithm. Conducted analyzes indicate that convergence problems can be corrected by equipping such algorithms with long term memory of past gradients, and that the Adam algorithm not only corrects convergence problems, but can also suggest new variants that often result in improved empirical performance.

3.6 Evaluation Methods

In many sciences, as mentioned, the use of qualitative or quantitative (concrete or abstract) methods has become a matter of debate, with particular schools of thought in each discipline favoring one method and disdains another. Qualitative methods argue that quantitative methods are meant to conceal the reality of the social events studied because they underestimate, ignore, ignore or neglect the most immeasurable factors. The modern tendency (and indeed the majority tendency throughout the history of social science) is to use eclectic approaches: quantitative methods can be used with a global qualitative framework, and qualitative methods can be used to understand the meaning of numbers generated by quantitative methods.

But even this practice has raised a debate as to whether quantitative and qualitative research methods are complementary: some investigators argue that combining these two different approaches to a research method is useful and that the social world helps create a more complete situation. They believe and trust that the epistemologies that support each approach are so divergent that they cannot be reconciled even in a research project (Gadd, 2006).

3.6.1 MAD (Mean Absolute Deviation)

The mean absolute deviation (or mean absolute deviation (MAD)) about a given point of a data set (or only "mean absolute deviation") is the average of the absolute deviations of the data given or the positive difference (Usually central values). It is a summary statistics of statistical distribution or variability. In the general form, the central point may be the mean, median, mode, or any central trend measure, or the result of any random data point associated with the given data set. The absolute values of the difference between the data points and their central tendencies are summed and divided by the number of data points.

3.6.2 MAPE

Many organizations focus primarily on the MAPE when assessing forecast accuracy. Most people are comfortable thinking in percentage terms, making the MAPE easy to interpret. It can also convey information when user doesn’t know the item’s demand volume. For example, telling user manager, "we are off by less than 4%" is more meaningful than saying "we are off by 3,000 cases," if user manager doesn’t know an item’s typical demand volume.

The MAPE is scale sensitive and should not be used when working with low-volume data. Notice that because "Actual" is in the denominator of the equation, the MAPE is undefined when Actual demand is zero. Furthermore, when the Actual value is not zero, but quite small, the MAPE will often take on extreme values. This scale sensitivity renders the MAPE close to worthless as an error measure for low-volume data.

3.6.3 MEAN (Actual / Predicted)

In statistics, a forecast error is the difference between the actual or real and the predicted or forecast value of a time series or any other phenomenon of interest. Since the forecast error is derived from the same scale of data, comparisons between the forecast errors of different series can only be made when the series are on the same scale.

In simple cases, a forecast is compared with an outcome at a single time-point and a summary of forecast errors is constructed over a collection of such time-points. Here the forecast may be assessed using the difference or using a proportional error. By convention, the error is defined using the value of the outcome minus the value of the forecast.

In other cases, a forecast may consist of predicted values over a number of lead-times; in this case an assessment of forecast error may need to consider more general ways of assessing the match between the time-profiles of the forecast and the outcome. If a main application of the forecast is to predict when certain thresholds will be crossed, one possible way of assessing the forecast is to use the timing-error—the difference in time between when the outcome crosses the threshold and when the forecast does so. When there is interest in the maximum value being reached, assessment of forecasts can be done using any of, the difference of times of the peaks the difference in the peak values in the forecast and outcome, the difference between the peak value of the outcome and the value forecast for that time point. Forecast error can be a calendar forecast error or a cross-sectional forecast error, when we want to summarize the forecast error over a group of units. If we observe the average forecast error for a time-series of forecasts for the same product or phenomenon, then we call this a calendar forecast error or time-series forecast error. If we observe this for multiple products for the same period, then this is a cross- sectional performance error (Grafe et al., 2010).

3.6.4 Standart Deviation

The standard deviation is a system used for a measure of the amount of variation or distribution of a set of values. A low standard deviation indicates that the values tend to be close to the average of the set, while a high standard deviation indicates that the values are spread over a wider range. In this way, the distance and proximity to the intended or targeted results are found. The standard deviation of a random variable, statistical population, data set or probability distribution is the square root of its variance. In practice it is algebraically simpler, although less robust than the average absolute deviation. A useful feature of standard deviation is that, in contrast to variance, it is expressed in the same units as the data. In addition to expressing the variability of a population, standard deviation is commonly used to measure confidence in statistical results. For example, the margin of error in the query data is determined by calculating the expected standard deviation in the results if the same survey is performed more than once. This derivation of the standard deviation is often referred to as the mean while the estimate is called "standard error" or "standard error of average". If an infinite number of samples are drawn and an average is calculated for each sample, it is calculated as the standard deviation of all instruments to be calculated from this population. The standard deviation of a population and the standard mean of a statistic derived from this population are quite different, but related. The reported error margin of a survey is calculated from the standard error of the mean. In science, many investigators report the standard deviation of experimental data, results, and, by convention, more than just two standard deviations from an empty expectation are considered statistically significant. Normal random error or change in measurements is thus distinguished from the possible original. Standard deviation is also important in finance where the standard deviation in the return on investment is a measure of the volatility of the investment.

3.6.5 Correlation

Correlation or dependence is any statistical relationship, whether causal or not, between two random variables or bivariate data. In the broadest sense correlation is any statistical association, though it commonly refers to the degree to which a pair of variables are linearly related. Familiar examples of dependent phenomena include the correlation between the physical statures of parents and their offspring, and the correlation between the demand for a limited supply product and its price.[citation needed]

Correlations are useful because they can indicate a predictive relationship that can be exploited in practice. For example, an electrical utility may produce less power on a mild day based on the correlation between electricity demand and weather. In this example, there is a causal relationship, because extreme weather causes people to use more electricity for heating or cooling. However, in general, the presence of a correlation is not sufficient to infer the presence of a causal relationship.

CHAPTER 4
RESULTS & DISCUSSIONS

4.1 Overall Results

In this section, the predictive results of the deep learning model using the data obtained from the study named ‘Flexural buckling load prediction of aluminum alloy columns using soft computing techniques are compared with the results obtained from soft computing techniques in the same study. Sequential model of the learning method of the optimizers Adam, Adamax, RMSprops, Nadam, Adagrad, Adadelta are used and the model compiled with MAE and MSE loss is run 10 times; optimum results are used for evaluation. Dataset is divided into training set and testing set. The training set is 70% and the test set is 30%. Experimental values for 104 tests in Table 4.2 given with related material parameters.

The description of material parameters are shown in Table 4.1 taken from Cevik et al., (2009).An analytical statement was proposed (Frey and Rondal, 1978; Rondal and Maquoi, 1979) and was accepted by the ECCS. The calculation of the non-dimensional column strength is shown as:

$$X = \frac{1}{\theta + \sqrt{\varphi^2 - \lambda^2}},$$

$$\varphi = \frac{1}{2}(1 + \eta + \lambda^2),$$

The constants of the material parameters are $\alpha, \beta, \lambda_0, \lambda_1$. These are shown as:

$$\alpha(\mathrm{n,e}) = \frac{1.5}{(\mathrm{e}^{0.6}+0.03)(\mathrm{n}\left(\frac{0.0048}{\mathrm{e}^{0.55}}\right)^{1.4}+1.3)} + \frac{0.02}{\mathrm{e}^{0.6}},$$

$$\beta(\mathrm{n,e}) = \frac{0.36\exp(-\mathrm{n})}{\mathrm{e}^{0.45}+0.007} + \tanh\left(\frac{\mathrm{n}}{180} + \frac{6\mathrm{x}10^{-6}}{\mathrm{e}^{1.4}} + 0.04\right),$$

$$\lambda_0(\mathrm{n,e}) = 0.82\left(\frac{\mathrm{e}}{\mathrm{e}+0.0004} - 0.01\mathrm{n}\right) \geq 0.2,$$

$$\lambda_1(\mathrm{n,e}) = 0.8\,\frac{e}{e+0.0018}\left(1 - \left[\left(\frac{n-55}{n+\frac{6e-0.0054}{e+0.0015}}\right)^2\right]^{0.6}\right).$$

where $\sigma_{0.2}$, L and r are the ultimate stress, effective length and Radius of gyration respectively.

$$\chi = \frac{\sigma_u}{\sigma_{0.2}}, \quad \lambda = \sqrt{\frac{\sigma_u}{\sigma_{E_0}}}, \quad \sigma_{E_0} = \frac{\pi^2 E_0}{(\mathrm{L/r})^2}$$

Table 4. 1. Description Table of Dataset

Reference	**Ref. no**	**Type (axis)**	**Production**	**Alloy**	$\mathbf{E_0}$	$\boldsymbol{\sigma_{0.2}}$	**n**
Djalaly and Sfintesco	1	I (minor)	France	2017	72,600	310	7.15
	2	I (minor)	France	7020	70,630	320	18.12
	3	I (minor)	France	6081	68,670	288	16.16
Bernard,Frey,Janss,andMassonnet(1973)	4	I (minor)	Switzerland	7020	75,880	335	24.15
	5	I (minor)	Switzerland	7020	78,264	325	26.56
	6	CHS	Belgium	7020	72,170	340	35.78
	7	CHS	Switzerland	6082	67,300	299	29.45
	8	CHS	Norway	6082	74,650	245	19.94
Kloppel and Barsch (1973)	9	major	Germany	7020	72,100	330	33.6
	10	major	Germany	6082	72,100	293	29.9
	11	CHS	Germany	7020	72,100	330	33.6
	12	CHS	Germany	6082	72,100	293	29.9
Arnault (1967)	13	I (minor)	France	2017	73,575	312	11.9
	14	I (minor)	France	6081	68,670	288	66.3
	15	I (minor)	France	6082	68,670	315	22.5
	16	I (minor)	France	7020	71,120	322	37.3

Table 4.2. Dataset of Study.

E (Mpa)	$\boldsymbol{\sigma_{0.2}}$	**n**	**L/r**	**λ**	**TEST (Mpa)**	**GEP (Mpa)**	**NN (Mpa)**
72600	310	7,15	10	0,211	312,9	312,3	297,5
72600	310	7,15	50	1,055	225,4	218,5	235,3
72600	310	7,15	70	1,477	143,8	139,3	147
72600	310	7,15	85	1,793	97,6	95,74	102,2
72600	310	7,15	100	2,109	74,4	66,16	72,7
72600	310	7,15	120	2,531	51,4	41,72	49,73
72600	310	7,15	150	3,164	33,8	22,7	34,24
72600	310	7,15	10	0,214	321,5	322,6	347,1
70630	320	18,12	50	1,072	237,4	219,2	231,4

70630	320	18,12	70	1,5	137,8	136,6	143,7
70630	320	18,12	85	1,822	97,2	92,66	100
70630	320	18,12	100	2,143	71	63,49	71,38
70630	320	18,12	120	2,572	47,7	39,76	49,05
70630	320	18,12	150	3,215	30,7	21,52	34
70630	320	16,16	10	0,203	288,2	290,1	286
68670	288	16,16	50	1,016	213,6	208,8	213,8
68670	288	16,16	70	1,423	144	136,2	132,7
68670	288	16,16	85	1,728	93,7	94,79	92,44
68670	288	16,16	100	2,033	65,6	66,05	66,41
68670	288	16,16	120	2,44	46,7	41,95	46,32
68670	288	16,16	150	3,049	29,3	22,97	32,9
68670	288	16,16	48,3	1,022	226	241,2	248,5
75880	335	24,15	48,3	1,022	255,7	241,2	248,5
75880	335	24,15	48,3	1,022	263,3	241,2	248,5
75880	335	24,15	48,3	1,002	278,1	241,2	248,5
78264	325	26,56	64,5	1,311	178,9	175,4	183,2
78264	325	26,56	64,5	1,311	197,6	175,4	183,2
78264	325	26,56	64,5	1,311	224,9	175,4	183,2
78264	325	26,56	64,5	1,311	178,9	175,4	183,2
78264	325	26,56	64,5	1,311	215	175,4	183,2
78264	325	26,56	64,5	1,311	161,6	175,4	183,2
78264	325	26,56	64,5	1,311	169,6	175,4	183,2
78264	325	26,56	64,5	1,311	166,5	175,4	183,2
78264	325	26,56	48,3	0,982	251,4	243,3	258,6
78264	325	26,56	48,3	0,982	286,9	243,3	258,6
78264	325	26,56	48,3	0,982	263,2	243,3	258,6
78264	325	26,56	48,3	0,982	242	243,3	258,6
78264	325	26,56	48,3	0,982	262	243,3	258,6
78264	325	26,56	48,3	0,982	263,2	243,3	258,6

78264	325	26,56	48,3	0,982	257	243,3	258,6
78264	325	26,56	48,3	0,982	281,9	243,3	258,6
78264	325	26,56	32,4	0,659	290	300,2	295,1
78264	325	26,56	32,4	0,659	288,7	300,2	295,1
78264	325	26,56	32,4	0,659	298,7	300,2	295,1
78264	325	26,56	32,4	0,659	310,1	300,2	295,1
78264	325	26,56	32,4	0,659	282,9	300,2	295,1
78264	325	26,56	32,4	0,659	283,7	300,2	295,1
78264	325	26,56	32,4	0,659	301,6	300,2	295,1
78264	325	26,56	32,4	0,659	295	300,2	295,1
78264	325	26,56	41,2	0,9	271,4	269,5	275,5
72170	340	35,78	41,2	0,9	265,6	269,5	275,5
72170	340	35,78	41,2	0,9	274,9	269,5	275,5
72170	340	35,78	41,2	0,9	264	269,5	275,5
72170	340	35,78	41,2	0,9	275,7	269,5	275,5
72170	340	35,78	51,2	1,086	201,7	202,5	203,6
67300	299	29,45	51,2	1,086	197,6	202,5	203,6
67300	299	29,45	51,2	1,086	206,5	202,5	203,6
67300	299	29,45	51,2	1,086	219,9	202,5	203,6
67300	299	29,45	51,2	1,086	203,9	202,5	203,6
67300	299	29,45	51,2	1,086	194,2	202,5	203,6
67300	299	29,45	51,2	1,086	202,8	202,5	203,6
67300	299	29,45	51,2	1,086	203,2	202,5	203,6
67300	299	29,45	28,9	0,527	208,6	237,6	202
74650	245	19,94	28,9	0,527	200,3	237,6	202
74650	245	19,94	28,9	0,527	197,9	237,6	202
74650	245	19,94	28,9	0,527	192,7	237,6	202
74650	245	19,94	51,9	0,947	163,6	193,4	164,3
74650	245	19,94	51,9	0,947	155,1	193,4	164,3
74650	245	19,94	51,9	0,947	191,4	193,4	164,3

74650	245	19,94	51,9	0,947	158,8	193,4	164,3
74650	245	19,94	51,9	0,947	153,3	193,4	164,3
74650	245	19,94	51,9	0,947	163,6	193,4	164,3
74650	245	19,94	51,9	0,947	153,2	193,4	164,3
74650	245	19,94	51,9	0,947	157,1	193,4	164,3
74650	245	19,94	90	1,937	82,2	82,91	85,81
72100	330	33,6	90	1,827	81,5	87,49	82,83
72100	293	29,9	90	1,827	79	87,49	82,83
72100	293	29,9	60	1,291	180,5	179,2	176,6
72100	330	33,6	60	1,291	169,7	179,2	176,6
72100	330	33,6	82	1,765	98,1	101,9	103,3
72100	330	33,6	82	1,765	98,1	101,9	103,3
72100	330	33,6	60	1,218	162,8	176,3	170,4
72100	293	29,9	82	1,665	101	106	99,71
72100	293	29,9	50	1,036	225,6	222,8	230,2
73575	312	11,9	70	1,451	144,2	143,6	143,7
73575	312	11,9	85	1,762	97,6	99,21	99,74
73575	312	11,9	100	2,073	74,5	68,8	71,13
73575	312	11,9	120	2,487	51,4	43,56	48,91
73575	312	11,9	150	3,109	33,8	23,76	34,24
73575	312	11,9	50	0,996	213,6	211,2	217,9
68670	288	66,3	70	1,394	143,9	139,3	133,3
68670	288	66,3	85	1,693	93,7	97,47	92,38
68670	288	66,3	100	1,992	65,6	68,18	66,51
68670	288	66,3	120	2,39	46,7	43,48	46,71
68670	288	66,3	150	2,998	29,2	23,85	33,38
68670	288	66,3	50	1,078	241,1	214,1	225,6
68670	315	22,5	70	1,509	149	132,6	140
68670	315	22,5	85	1,832	98,5	89,69	97,53
68670	315	22,5	50	1,071	237,4	220,7	218

71120	322	37,3	70	1,499	137,8	137,6	134,1
71120	322	37,3	85	1,82	97,2	93,43	93,53
71120	322	37,3	100	2,141	70,9	64,02	67,46
71120	322	37,3	120	2,569	47,7	40,12	47,26
71120	322	37,3	150	3,212	30,8	21,71	33,47

The flowchart of the application and execution of deep learning model used in the thesis and is shown in Figure 4.1. According to this chart, Firstly, The data set is scaled to give more correct results of Model. After that, the parameters of the model is defined and the dataset is trained and evaluated. If the execution count is smaller than 10; the application is executed until obtain the better results than soft computing techniques. If the execution count is equal 10; the last results are taken for comparing.

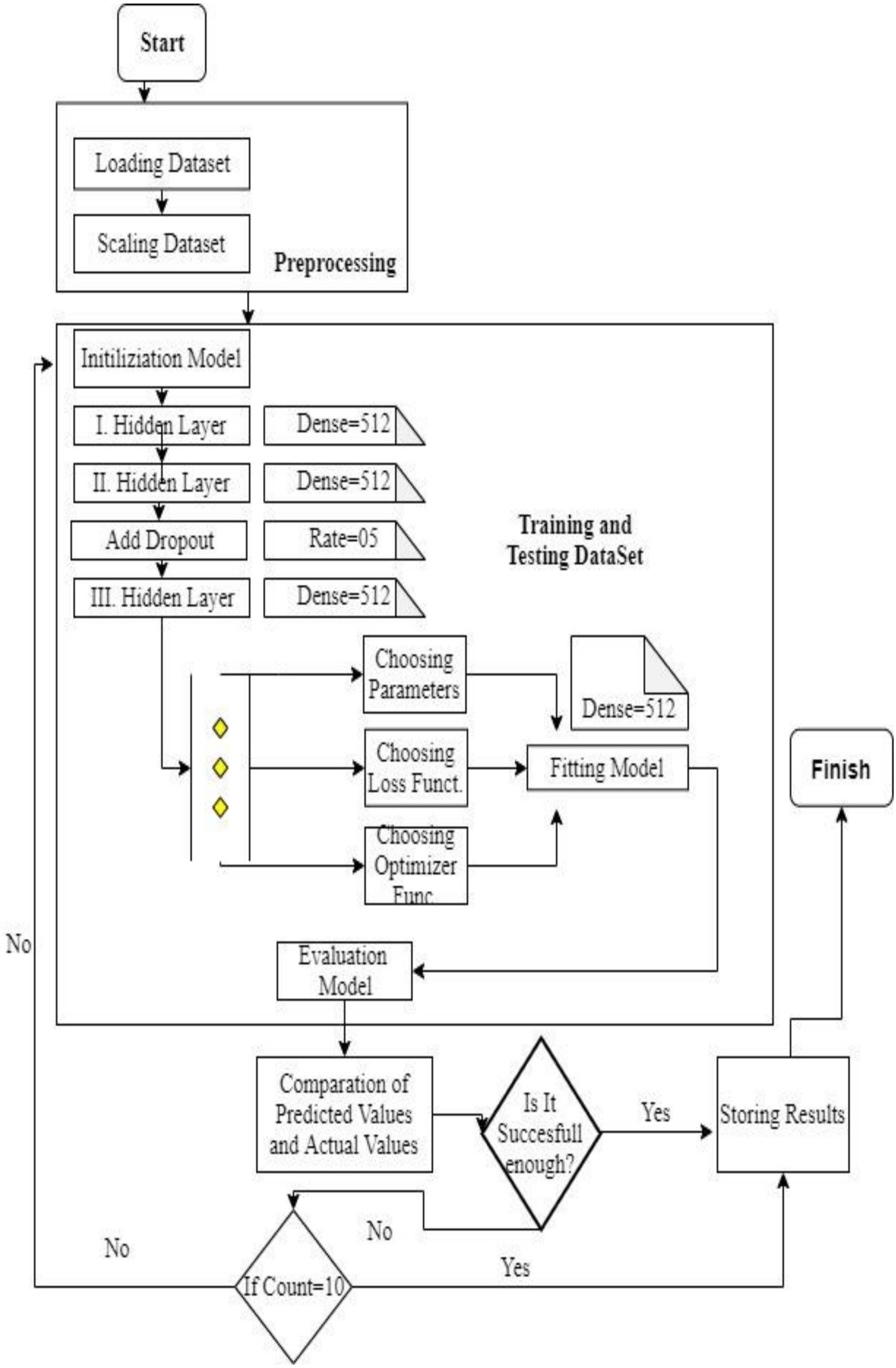

Figure 4.1. The Flowchart of the Application of Model

4.2 Comparison of Nadam Optimizer's Results and Laboratuar Test Results

When Nadam is used as an optimizer and MAE is used as loss, the model's train and test accuracy are shown in Figure 4.2 Accuracy of Test dataset is 9.38 and Accuracy of Train Dataset is 82.99. Mean of Test Dataset 51.56 and mean of train dataset is 23.51. The results obtained by using both loss parameters with the Nadam optimizer of the deep learning method with GEP and NN are compared with the previous methods, GEP and NN is shown in Table 4.3. For *MSE, MAD, RMSE, MEAN and Correlation*, the best results are seen in Nadam Optimizer and MAE Loss. For *MAE, MAPE*, the best results are seen in Nadam Optimizer and MSE Loss. In this case, it can be said that MAE loss function is more successful for Nadam optimizer in deep learning model.

Table 4.3. Statistical Results of MAE and MSE loss for Nadam

	MAD	MSE	RMSE	MAE	MAPE	MEAN	STD	Correlation
GEP	11.9448	300.4576	17.3337	0.0833	8.3350	1.0322	0.1308	0.9782
NN	7.2895	117.0553	10.8192	0.0419	4.1852	0.9988	**0.0586**	0.9915
NADAM & MAE	**6.4505**	**109.0971**	**10.4450**	0.0384	3.8352	**0.9992**	0.0622	**0.9920**
NADAM & MSE	6.4625	123.1020	11.0951	**0.0378**	**3.7775**	1.0097	0.0611	0.9912

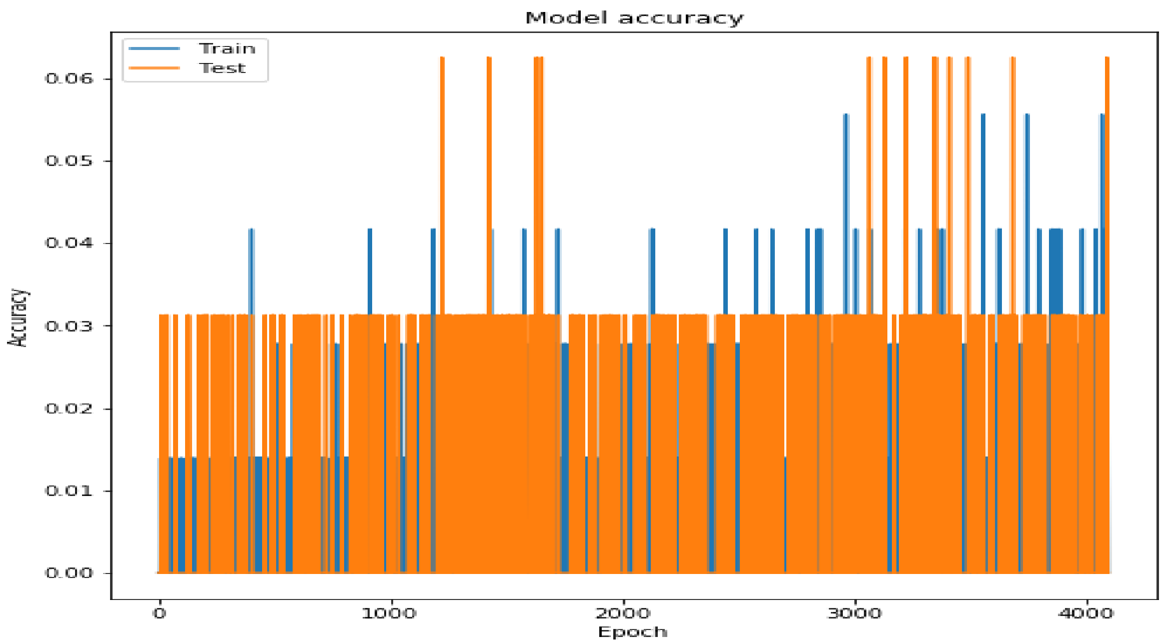

Figure 4.2. Comparison testing and training accuracy for Nadam Optimizer and MAE loss

In the graph in Figure 4.3, test results and model results are compared in terms of value. According to this graph, it can be said that the range with the highest deviations is between 150-250 and the R^2 value is 0.9841.

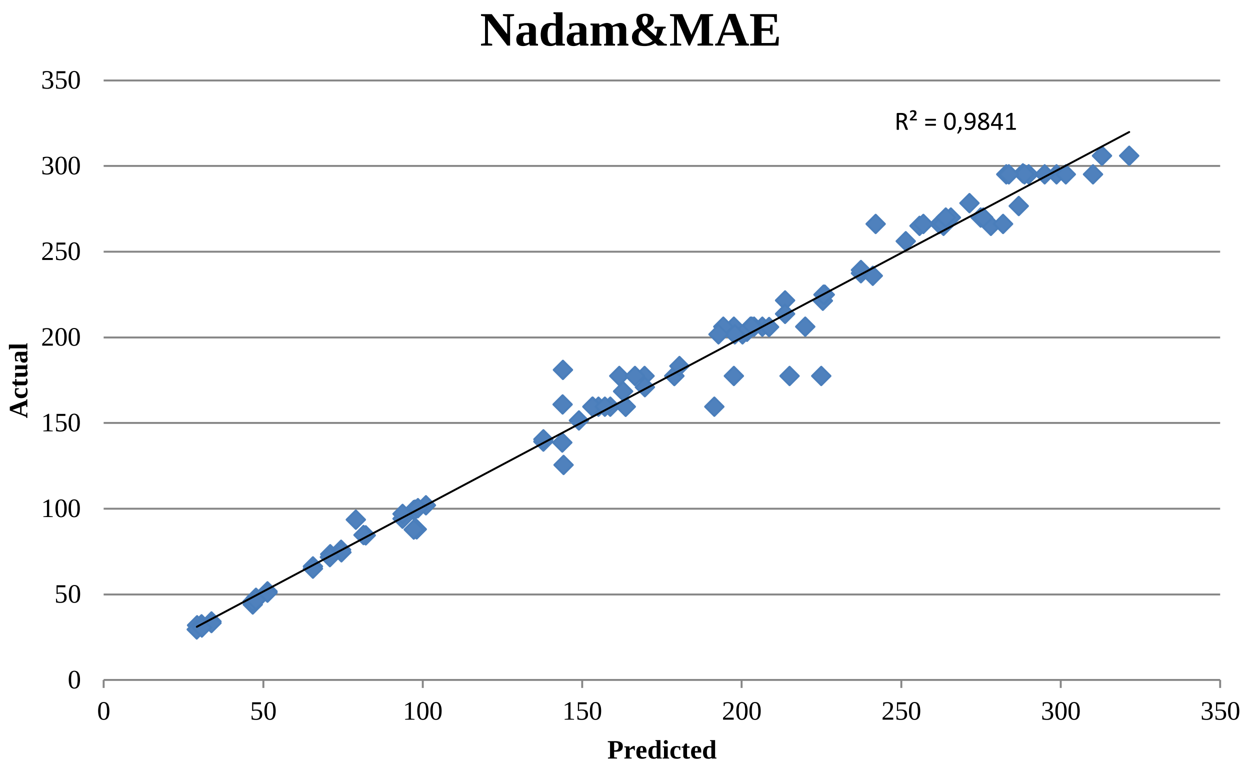

Figure 4.3. Comparison of Test Results and Predicted Results of Nadam Optimizer and MAE loss

105 rows in the dataset is an example for the deep learning results for Nadam & MAE as seen in Figure 4.4. When the closeness to the correct result is examined, it is observed that in 64 of 105 samples, deep learning model gives more realistic results than NN method.

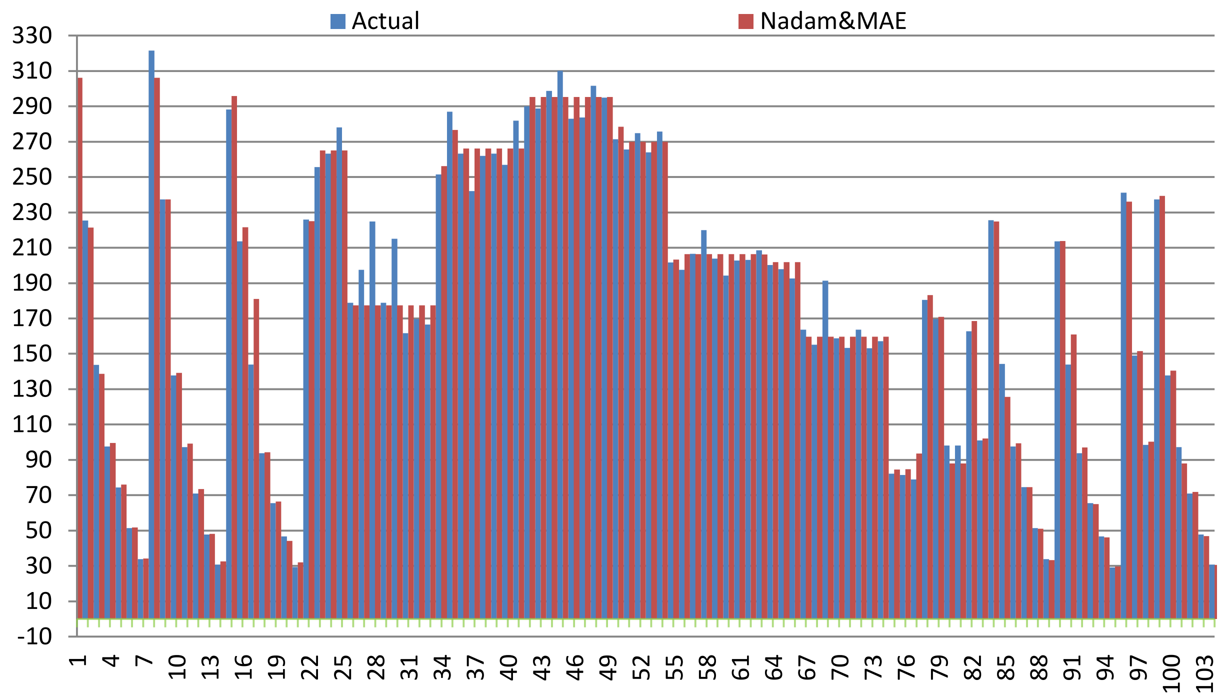

Figure 4.4. Actual datas and predicted results for Nadam Optimizer&MAE loss

The results of each line are compared with GEP and NN, which are the soft computing techniques, and the MAEs of the results obtained from the deep learning model are compared in Figure 4.5. As can be seen in the figure, the highest difference between the results belongs to GEP; it is observed that the points where the errors are minimum belong to the deep learning results.

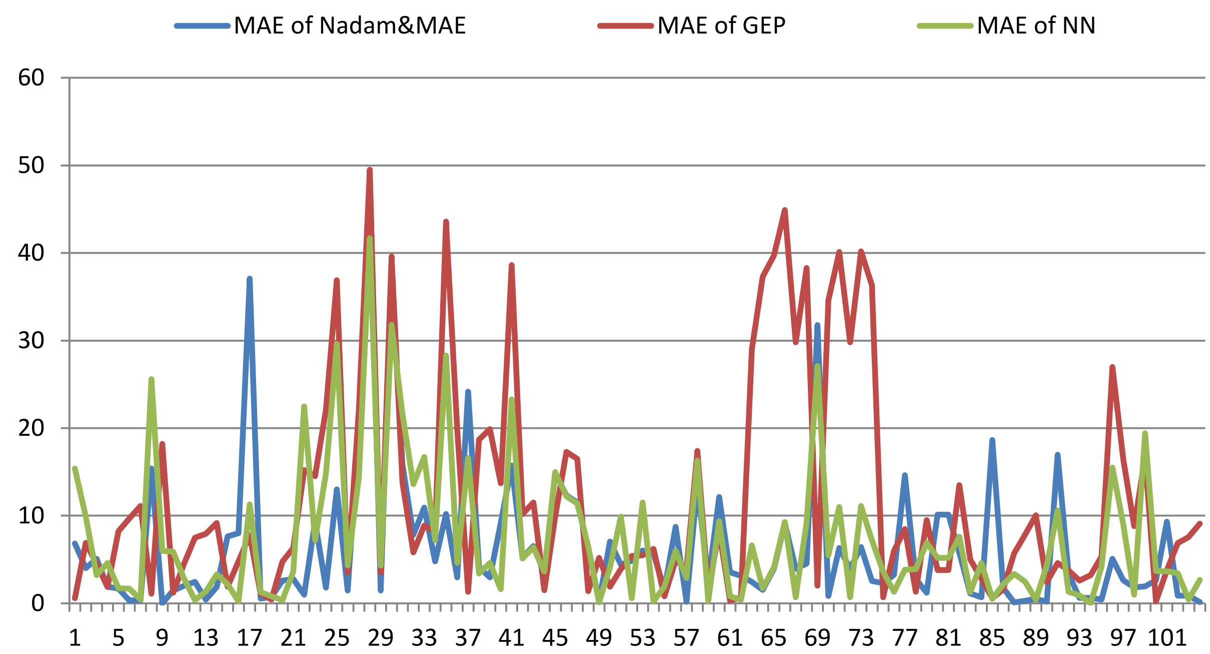

Figure 4.5. MAE of Each Results for Nadam Optimizer and MAE loss

Train and test accuracy of the model when Nadam is used as optimizer and MSE is used as loss in Figure 4.6 seen. The comparison of the test results of each row in the dataset with the estimation results from the model is as seen in Figure 4.7. Mean of Test Dataset 172.10 and mean of train dataset is 34.35. Std of Test Dataset is 165.86 and std of train dataset is 31.67.

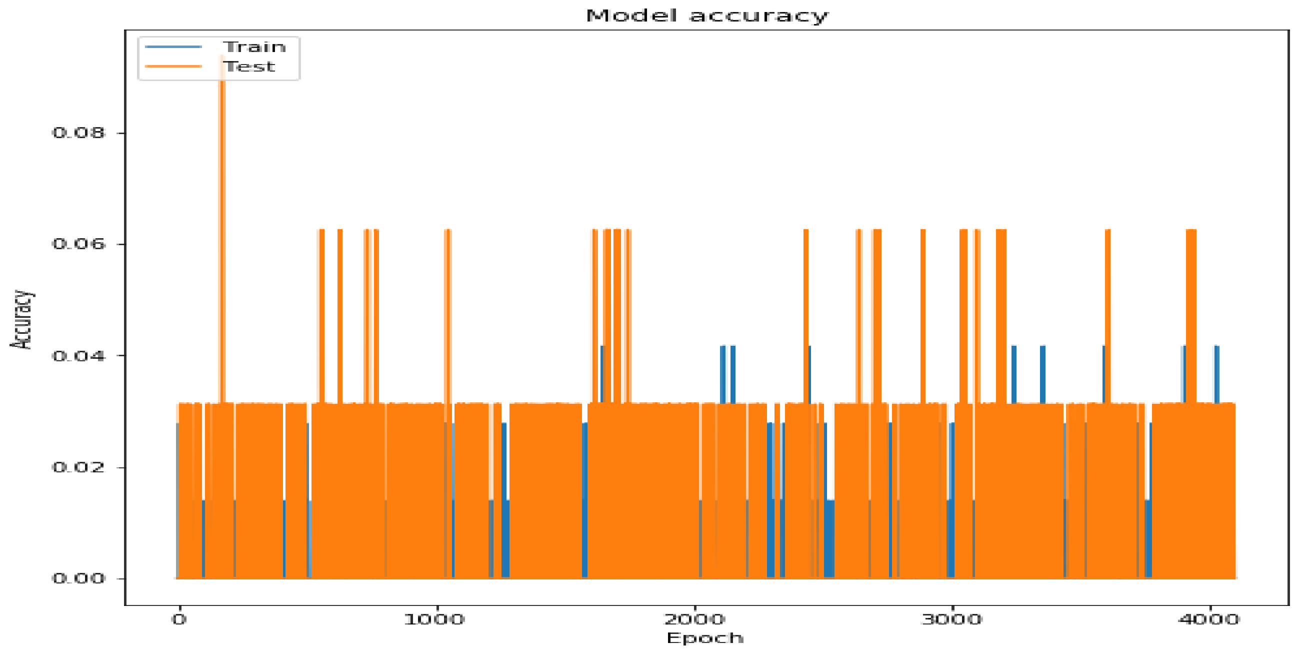

Figure 4.6. Comparison testing and training accuracy for Nadam Optimizer and MSE loss

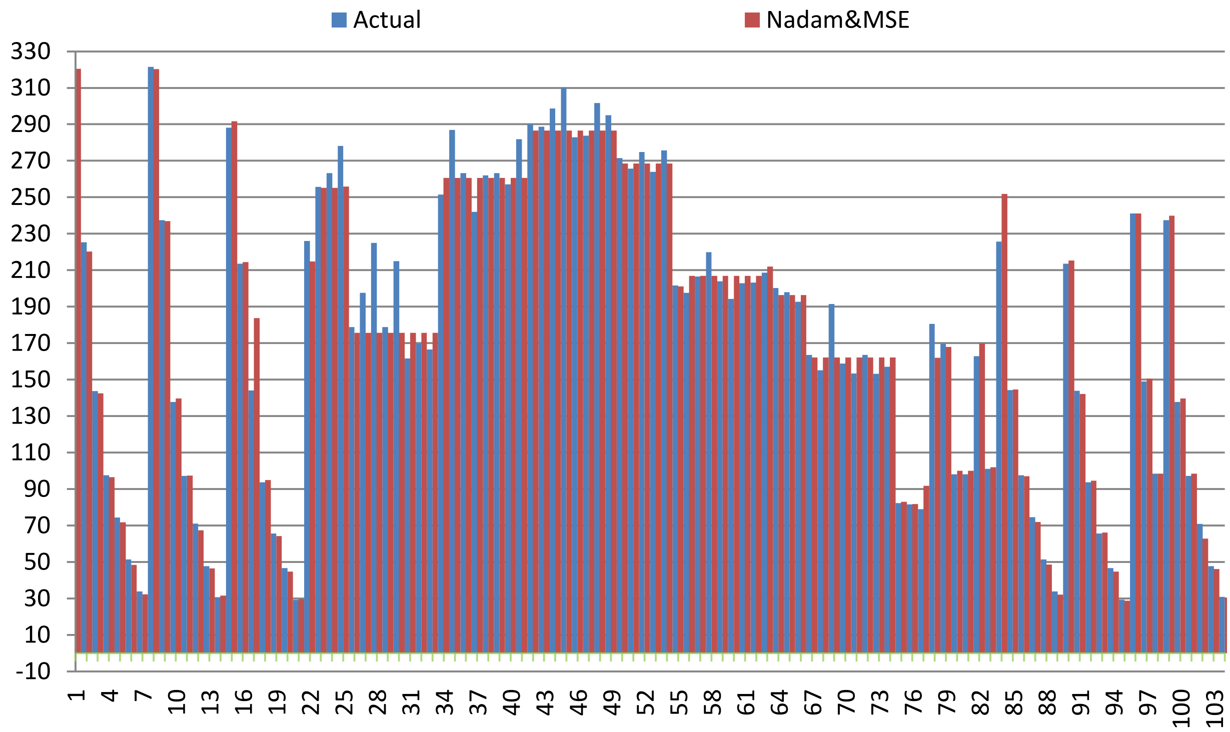

Figure 4.7. Actual datas and predicted results for Nadam Optimizer&MSE loss

The comparison of the test results in the dataset with the estimation results from the model is shown in Figure 4.7. When the closeness to the correct result is examined, it is observed that the deep learning model gives more realistic results than the NN method in 67 of 105 samples.

In the Figure 4.8, the test results and the model results are compared in terms of value. According to this graph, it can be said that the range with the most deviations is 150-200 and R^2 value is 0.9824.

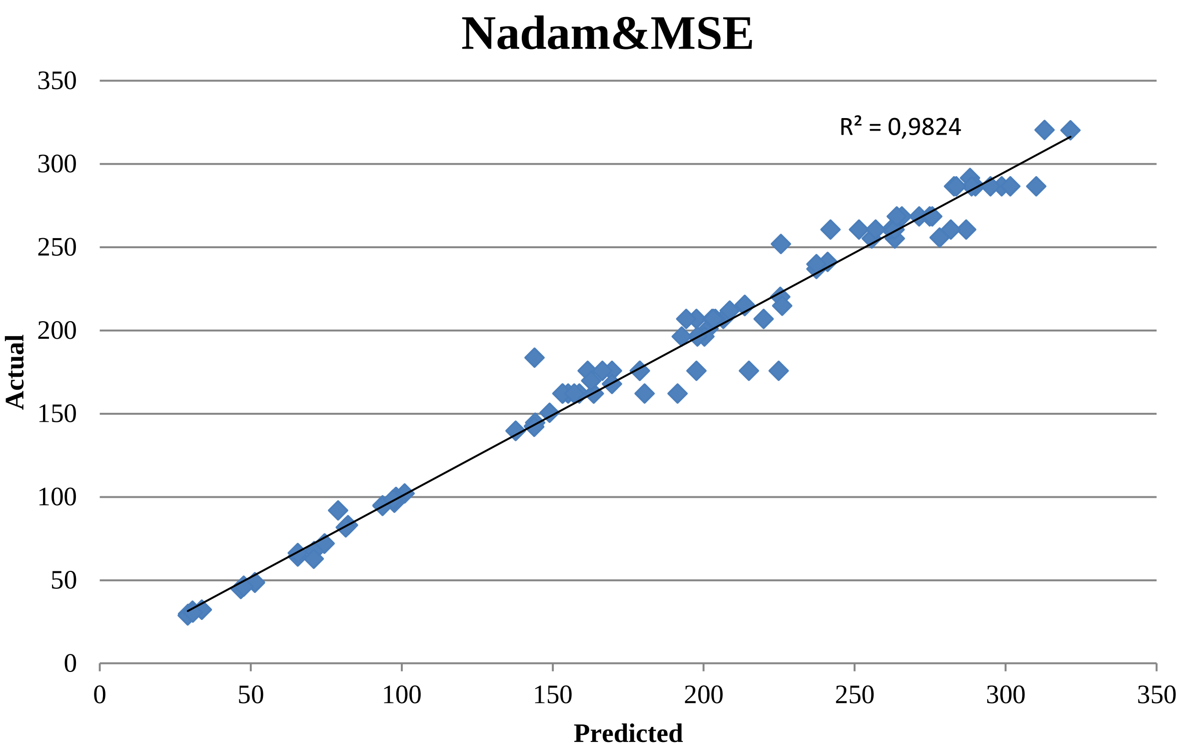

Figure 4.8. Comparison of Test Results and Predicted Results of Nadam Optimizer and MSE loss.

The results of each line are compared with GEP and NN, which are Soft Computing techniques, and the MAEs of the results obtained from the deep learning model are compared in Figure 4.9. As shown in the figure, the value at the point where the differences between the results are highest belongs to GEP. Except for 6 points, deep learning results are found to be successful.

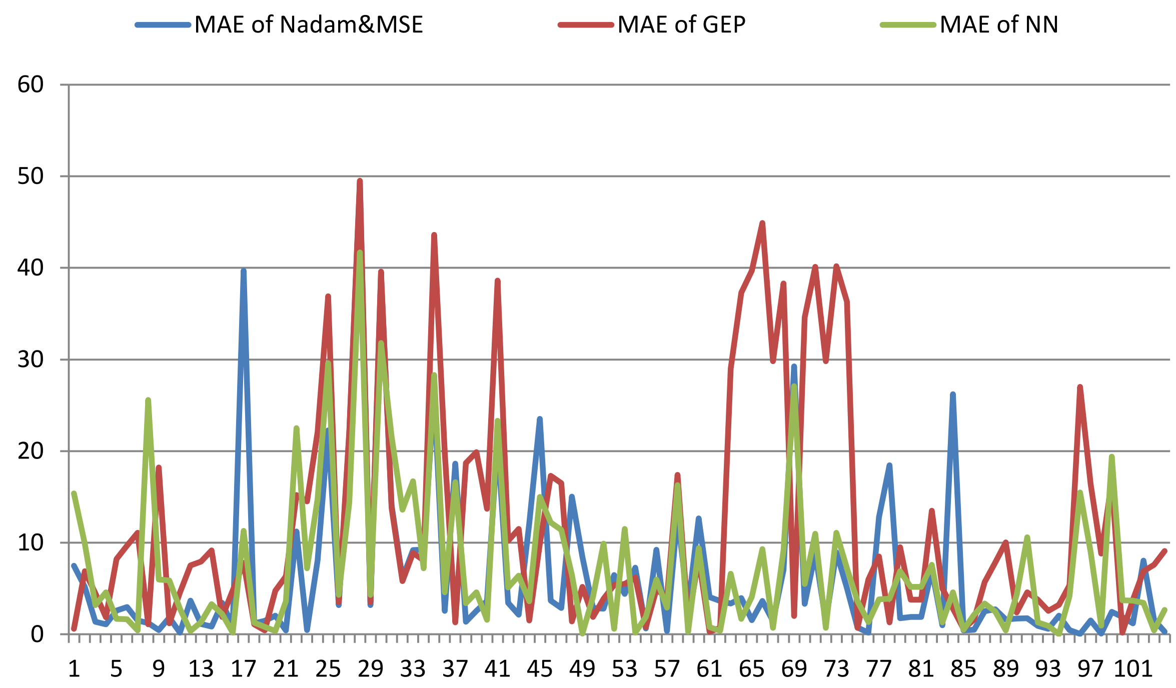

Figure 4.9. Mean Absolute Error of Each Results for Nadam Optimizer and MSE loss.

4.3 Comparison of Adamax Optimizer's Results and Laboratuar Test Results

In Table 4.4, the results of GEP and NN and Adamax optimizer of the deep learning method are compared using the two loss parameters. According to this table, the best results for *MAD, MAE, and MAPE* are found in Adamax Optimizer and MAE Loss. ; In *Correlation*, *MSE*, *RMSE*, Optimizer and MSE Loss, the last one is for standard deviation and mean for NN method. In this case, MSE and MAE Loss functions for Adamax optimizer can be interpreted as having almost equal performance.

Table 4.4. Statistical Results of Adamax Optimizer.

	MAD	MSE	RMSE	MAE	MAPE	MEAN	Std	Correlation
GEP	11.9448	300.4576	17.3337	0.0833	8.3350	1.0322	0.1308	0.9782
NN	7.2895	117.0553	10.8192	0.0419	4.1852	**0.9988**	**0.0586**	0.9915
ADAMAX & MAE	**5.8172**	115.3379	10.7395	**0.0337**	**3.3702**	1.0169	0.0596	0.9922
ADAMAX & MSE	6.4842	**110.5480**	**10.5142**	0.0463	4.6345	1.0309	0.0657	**0.9927**

Train and test accuracy of the model using Adamax as the optimizer and MAE as the loss as seen in Figure 4.10. The comparison of the test results of each row with the prediction results from the model is as presented in Figure 4.11. Accuracy of Test dataset is 9.38 and Accuracy of Train Dataset is 82.99.Mean of Test Dataset 59.002 and mean of train dataset is 23.730. Std of Test dataset is 91.266 and std of train dataset is 35.574.

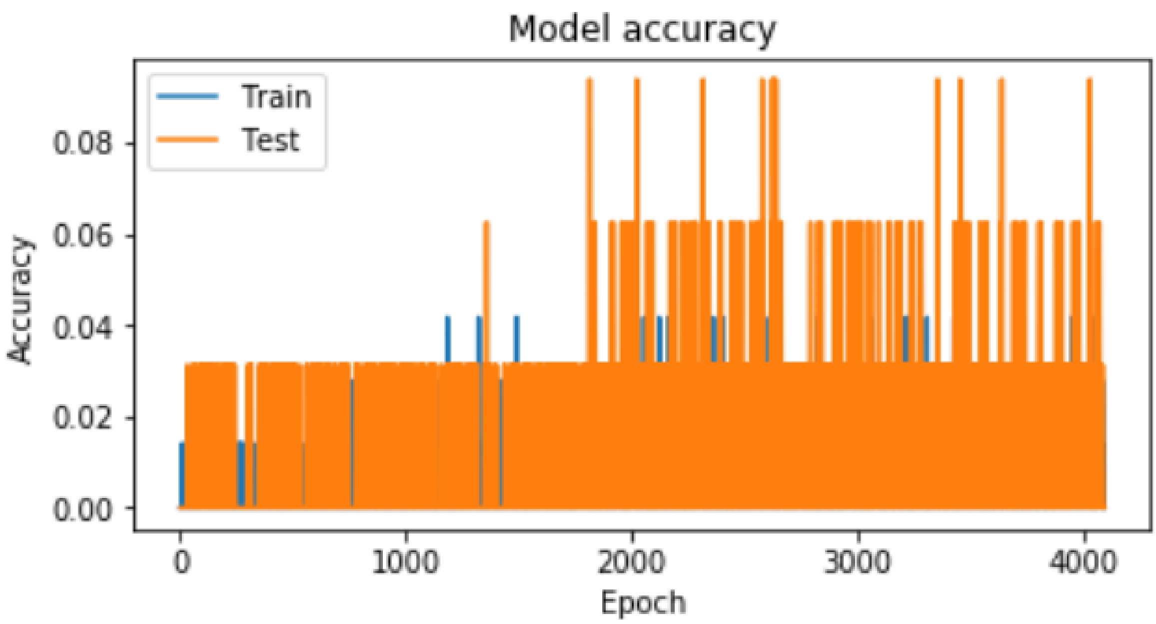

Figure 4.10. Comparison testing and training accuracy for Adamax Optimizer and MAE loss.

Comparison of the test results in the dataset with the estimation results from the model as presented in Figure 4.11. When the closeness to the correct result is examined, it is observed that in 68 of 105 samples, deep learning model gives more realistic results than NN method.

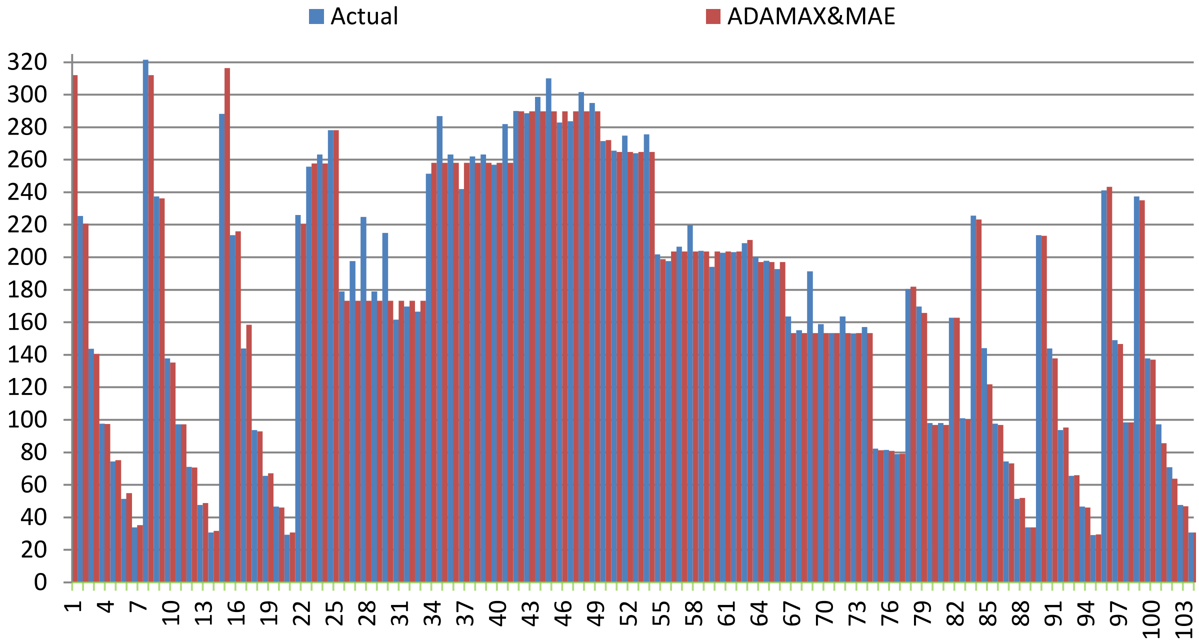

Figure 4.11. Actual datas and predicted results for Adamax Optimizer and MAE loss.

In the graph in Figure 4.12, test results and model results are compared in terms of value. According to this graph, it can be said that the maximum deviations are between 160-180 and R^2 value is 0.9845.

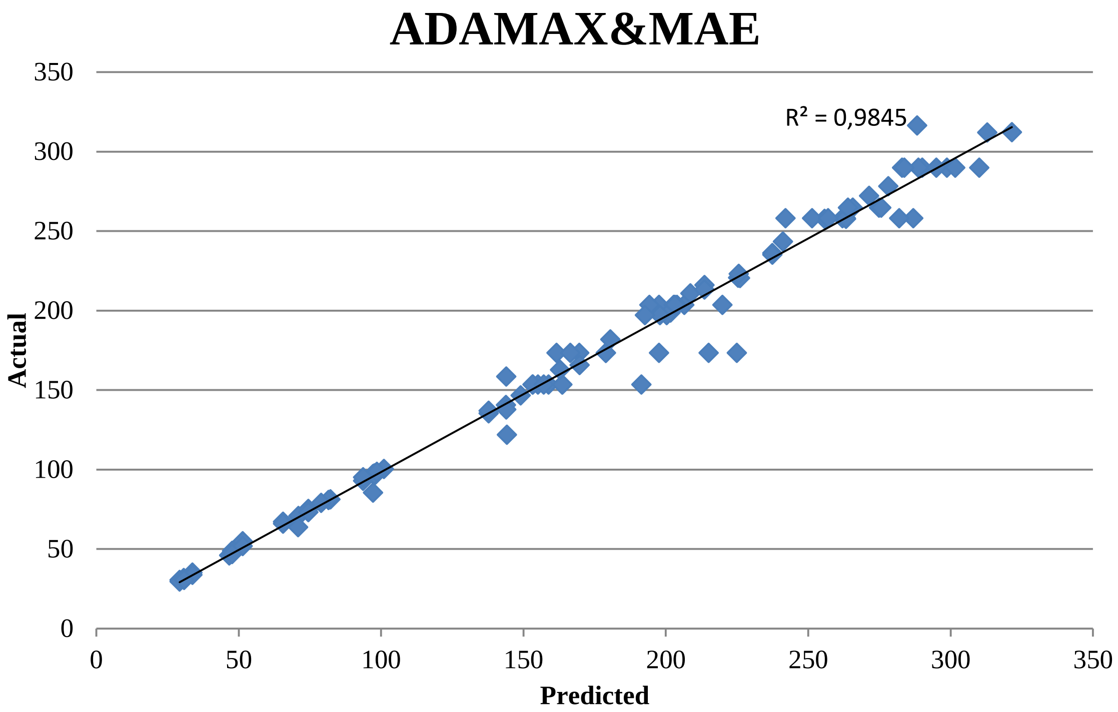

Figure 4.12. Comparison of Test Results and Predicted Results of Adamax Optimizer and MAE loss.

The results of each line are compared with GEP and NN, which are the soft computing techniques, and the MAEs of the results obtained from the deep learning model are compared in Figure 4.13. As can be seen in the figure, it is observed that the points where the errors are minimum belong to the deep learning results.

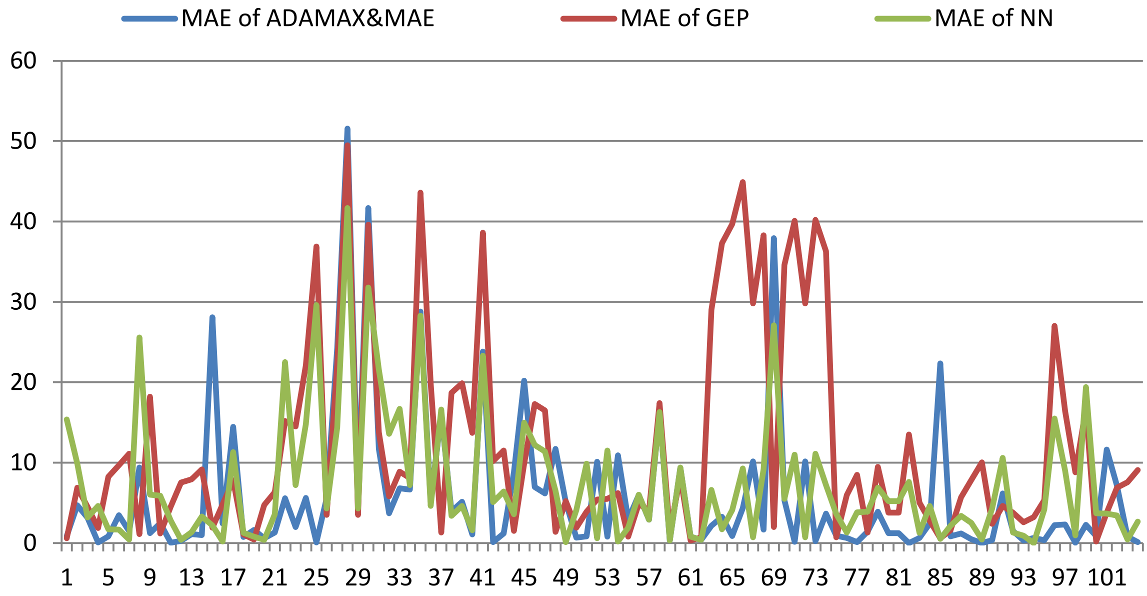

Figure 4.13. MAE of Each Results for Adamax Optimizer and MAE.

Train and test accuracy of the model using Adamax as the optimizer and MSE as the loss as seen in Figure 4.14. The comparison of the test results of each row with the prediction results from the model is as presented in Figure 4.15. Accuracy of Test dataset is 9.93 and Accuracy of Train Dataset is 78.40.Mean of Test Dataset 99.978 and mean of train dataset is 40.498. Std of Test dataset is 95.05 and std of train dataset is 37.953.

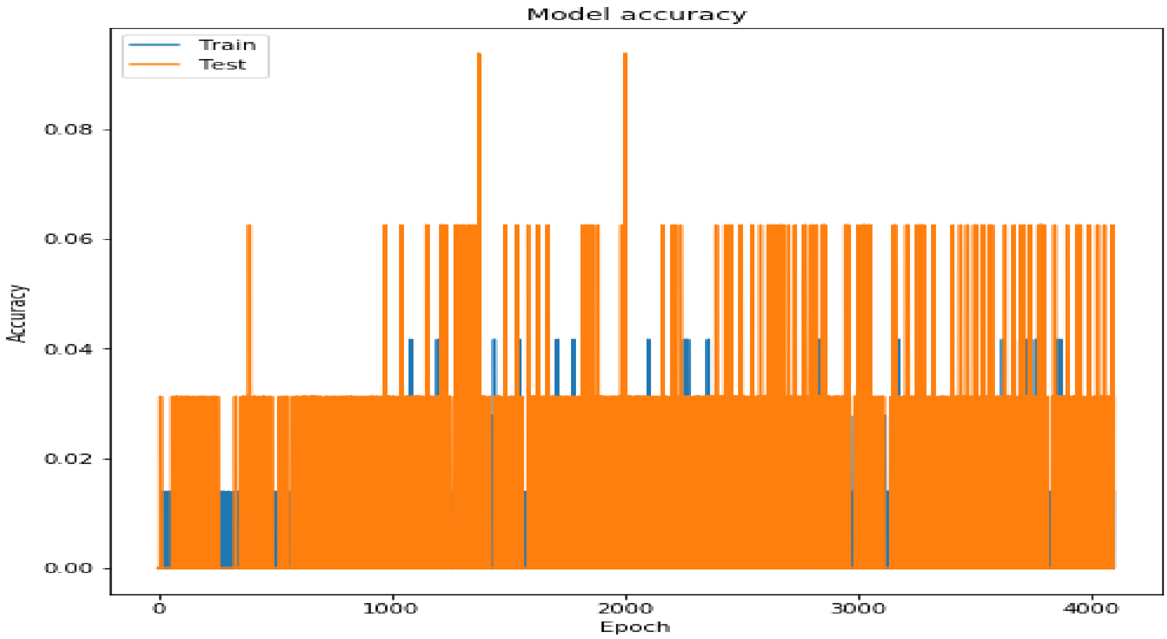

Figure 4.14. Comparison testing and training accuracy for Adamax Optimizer and MSE loss.

Comparison of the test results in the dataset with the estimation results from the model figure 4.15 as in. When the closeness to the correct result is examined, it is observed that in 65 of 105 samples, deep learning model gives more realistic results than NN method.

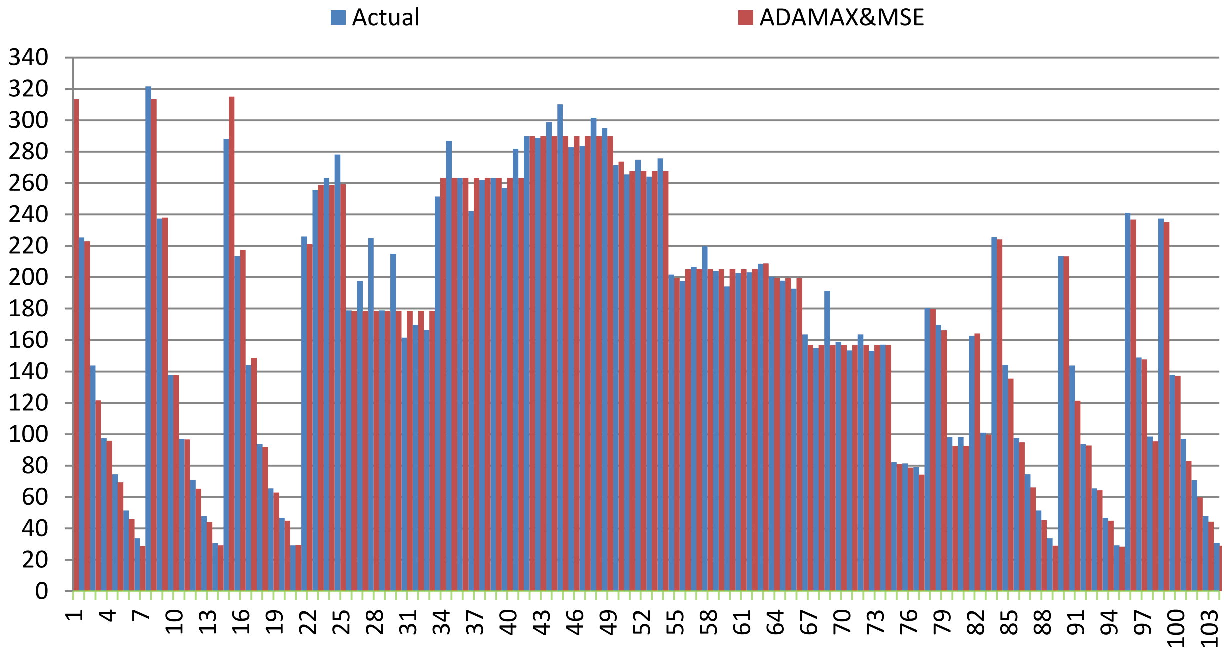

Figure 4.15. Actual datas and predicted results for Adamax Optimizer and MSE loss.

In the graph in Figure 4.16, test results and model results are compared in terms of value. According to this graph, it can be said that the maximum deviations are between 200-250 and R^2 value is 0.9855.

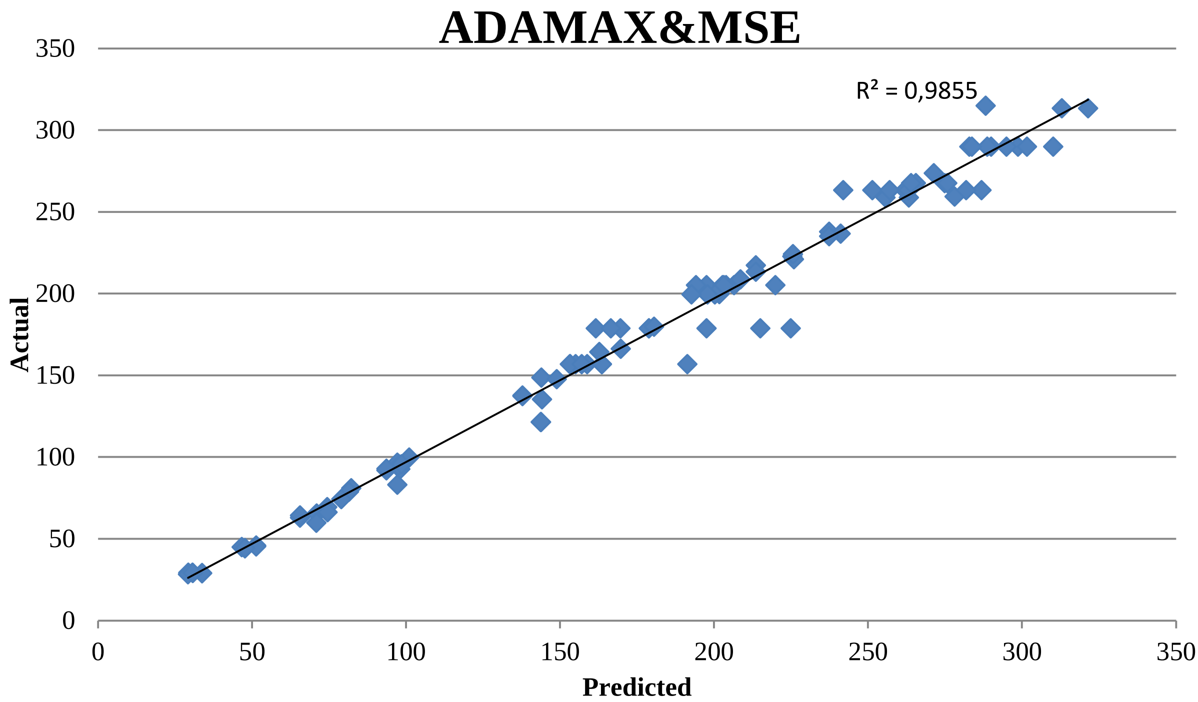

Figure 4.16. Comparison of Test Results and Predicted Results of Adamax Optimizer and MSE Loss

The results of each line are compared with GEP and NN, which are the soft computing techniques, and the MAEs of the results obtained from the deep learning model are compared in Figure 4.17. As can be seen in the figure, the highest difference between the results belongs to GEP; it is observed that the points where the errors are minimum belong to the deep learning results.

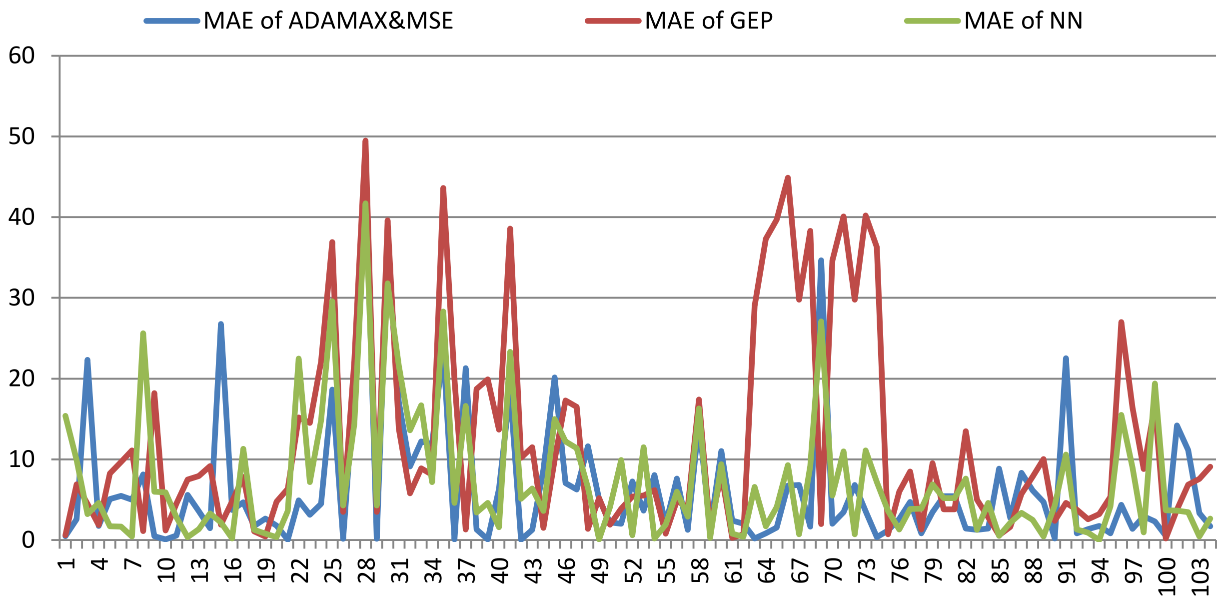

Figure 4.17. MAE of Each Results for Adamax Optimizer and MSE.

4.4 Comparison of Adam Optimizer's Results and Laboratuar Test Results

Table 4.5 compared the results obtained by using both loss parameters with the Adam optimizer of the deep learning method with GEP and NN. According to this table, the best results for *MAD*, *MSE*, *RMSE*, *MEAN*, *correlation* are found in Adam Optimizer and MAE Loss; *MAE*, *MAPE*, *MEAN for* the NN method. According to this situation, MAE Loss function is better for Adamax optimizer in deep learning model.

Table 4.5. Statistical Results of Adam Optimizer.

	MAD	MSE	RMSE	MAE	MAPE	MEAN	Std	Correlation
GEP	11.9448	300.4576	17.3337	0.0833	8.3350	1.0322	0.1308	0.9782
NN	7.2895	117.0553	10.8192	**0.0419**	**4.1852**	**0.9988**	**0.0586**	0.9915
ADAM & MAE	**6.5120**	**100.0853**	**10.0043**	0.0441	4.4052	1.0138	0.0767	**0.9928**
ADAM & MSE	6.9639	112.9846	10.6294	0.0461	4.6051	1.0174	0.0628	0.9920

Train and test accuracy of the model using Adam as the optimizer and MAE as the loss according to Figure 4.18. The comparison of the test results of each row with the prediction results from the model is showed in Figure 4.19. Mean of Test Dataset 54,600 and mean of train dataset is 85,735. Std of Test dataset is 5.347 and std of train dataset is 35.642 for Adam optimizer and MAE loss.

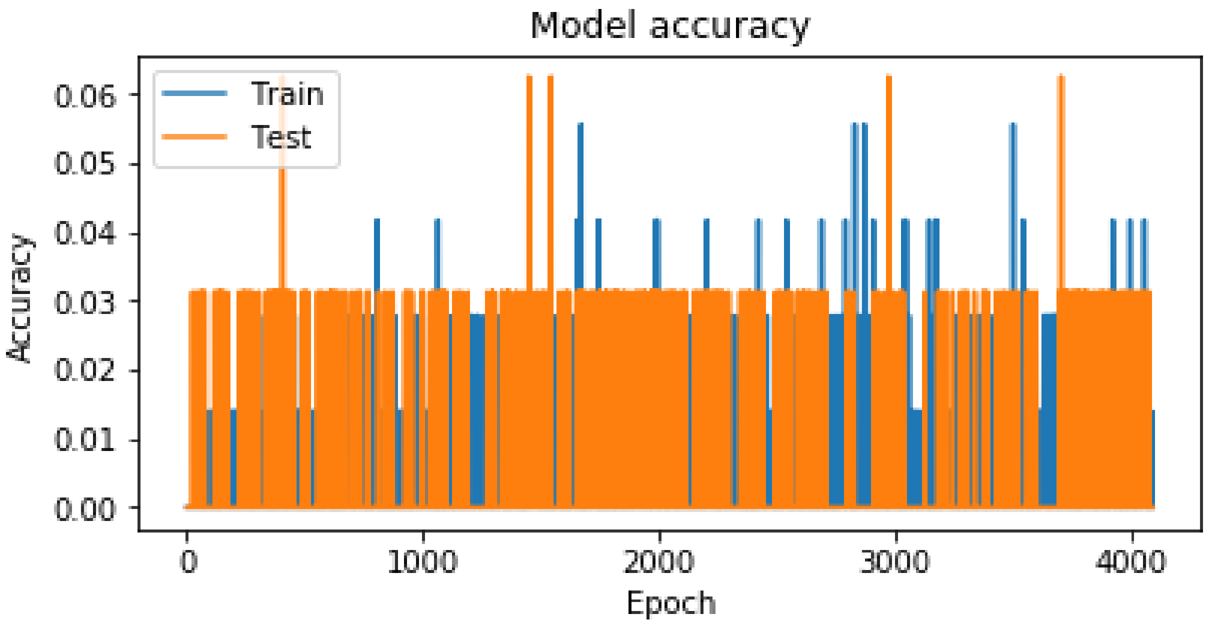

Figure 4.18. Comparison testing and training accuracy for Adam Optimizer and MAE loss.

Comparison of the test results in the dataset with the prediction results from the model showed at Figure 4.19. When the closeness to the correct result is examined, it is observed that in 67 of 105 samples, deep learning model gives more realistic results than NN method.

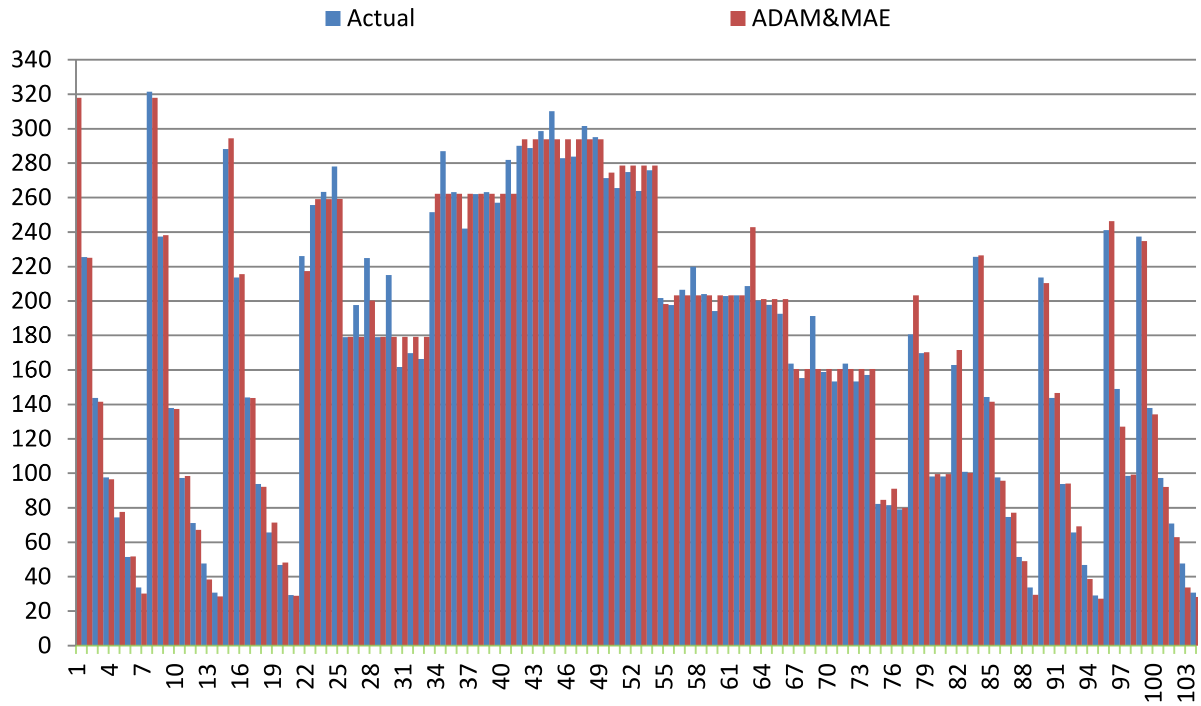

Figure 4.19. Actual datas and predicted results for Adam Optimizer and MAE loss.

In the graph in Figure.4.20, test results and model results are compared in terms of value. According to this graph, the range of 160-230 is the range with the highest deviation and R^2 value is 0.9857.

.

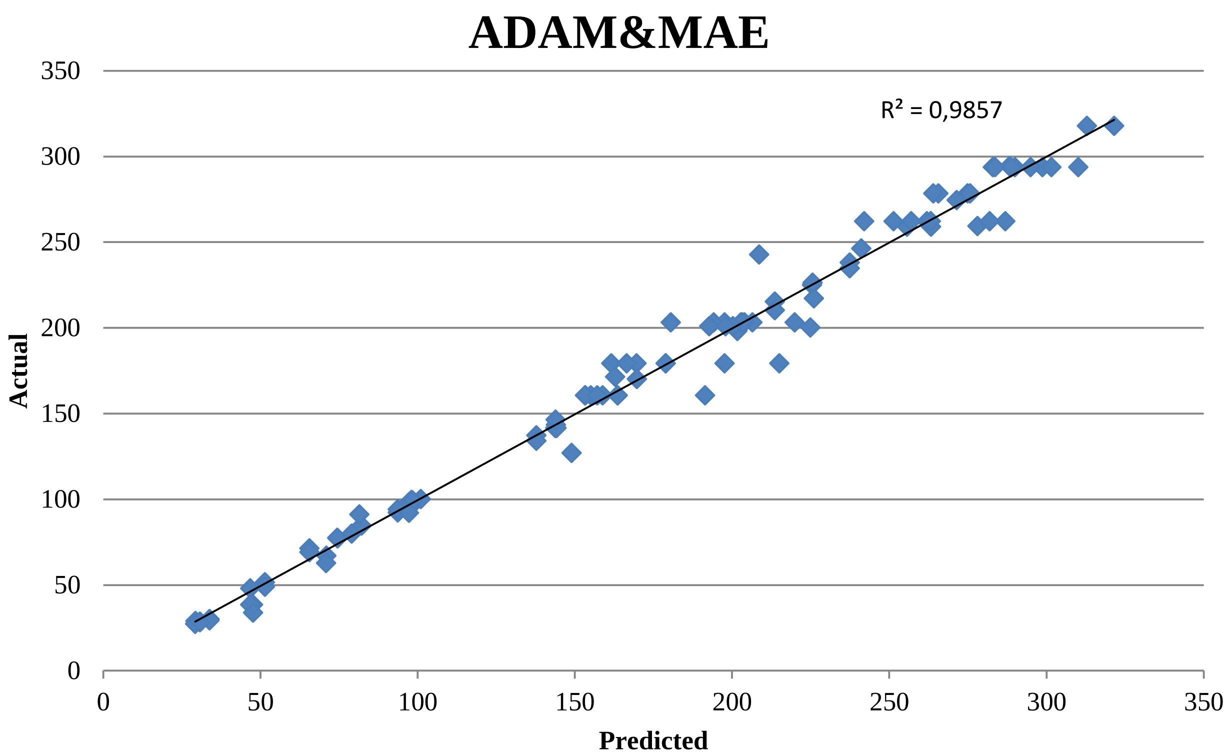

Figure 4.20. Comparison of Test Results and Predicted Results of Adam Optimizer and MAE loss.

When the deep learning model of each row is run with GEP and NN using Adam Optimizer and MAE loss functions, the MAEs of the results are compared in Figure 4.21. As can be seen in the figure, the difference between the results at the point where the highest value belongs to GEP; the closest points to real values belong to deep learning.

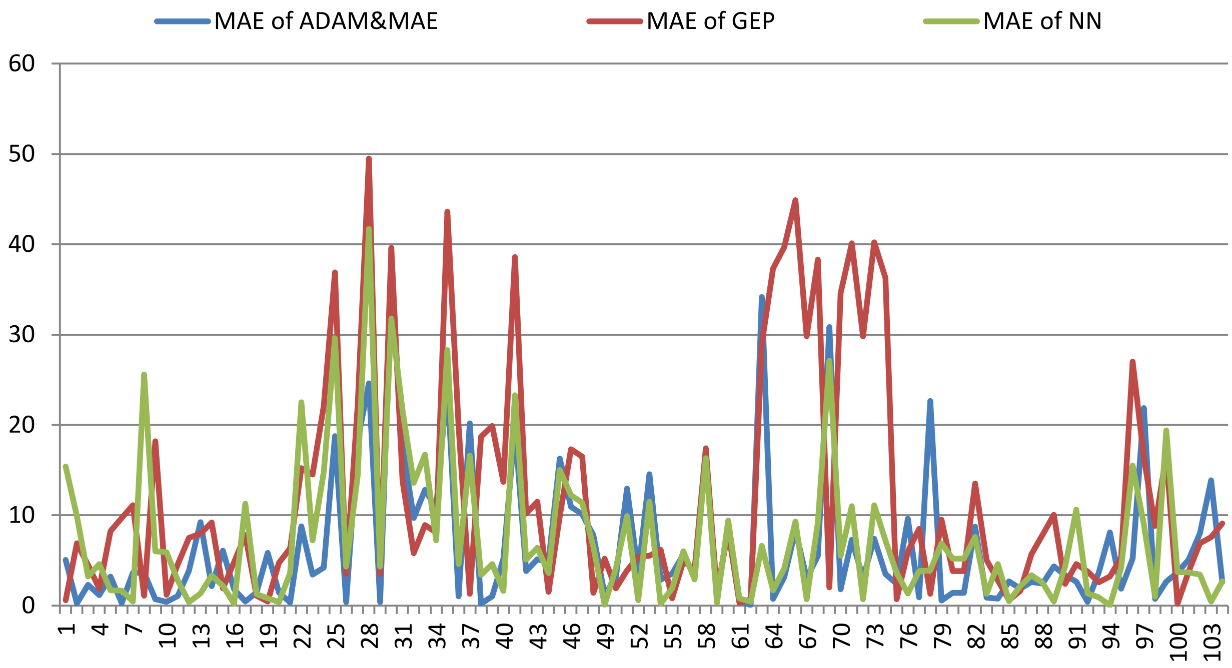

Figure 4.21. MAE of Each Results for Adam Optimizer and MAE.

Train and test accuracy of the model using Adam as the optimizer and MAE as the loss Figure 4.22. The comparison of the test results of each row with the prediction results from the model is indicated in Figure4.23. Mean of Test Dataset 111.816 and mean of train dataset is 40.457. Std of Test Dataset is 106.199 and std of traindataset is 37.961 for Adam optimizer and MSE loss.

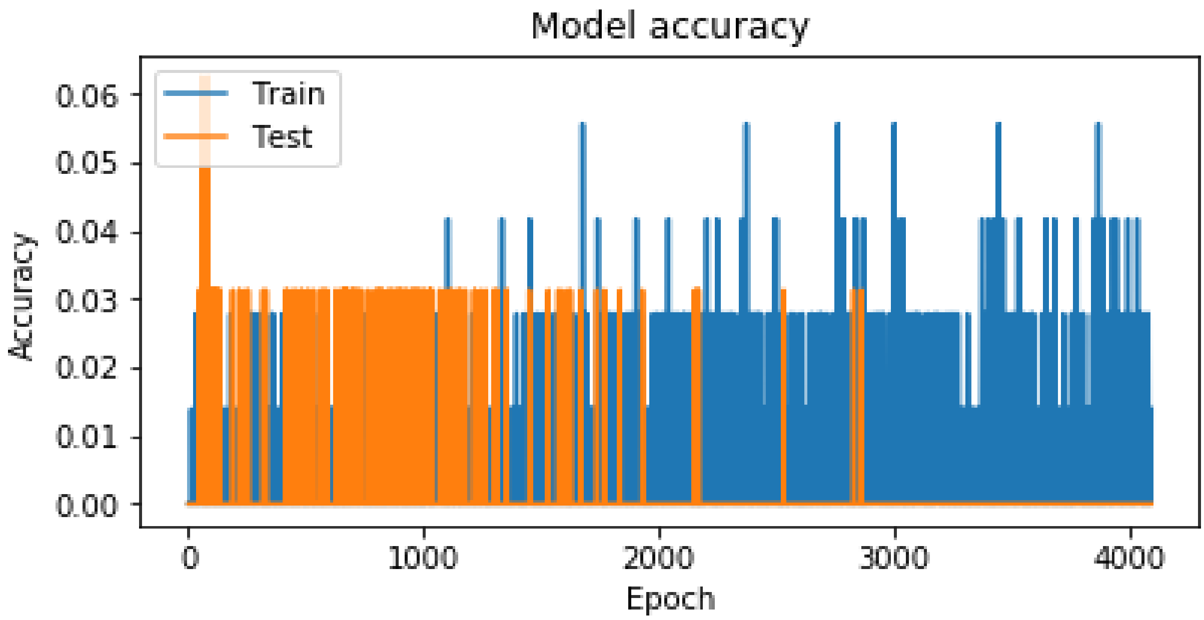

Figure 4.22. Comparison testing and training accuracy for Adam Optimizer and MSE loss.

The comparison of the test results in the dataset and the estimation results from the model is as in figure 4.23. When the closeness to the correct result is examined, it is observed that in 52 of 105 samples, deep learning model gives more realistic results than NN method.

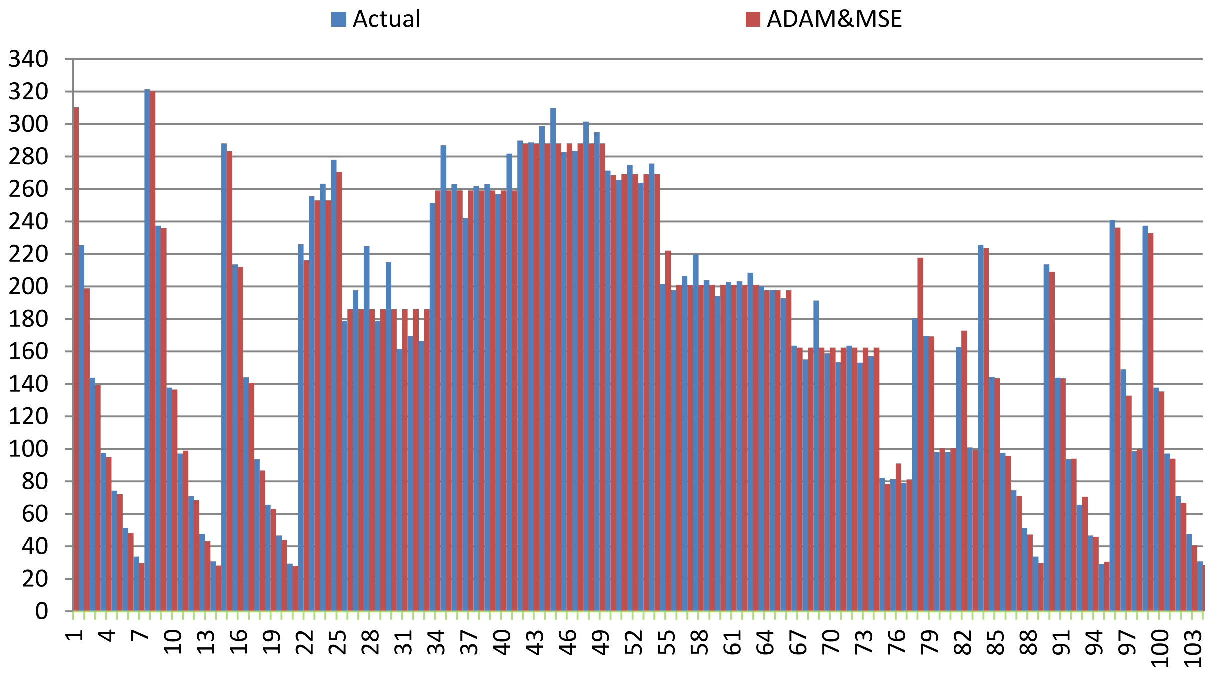

Figure 4.23. Actual datas and predicted results for Adam Optimizer and MSE loss.

In the graph in Figure 4.24, test results and model results are compared in terms of value. According to this graph, deviations between 160-180 and 270-300 are maximum and R^2 value is 0.9841

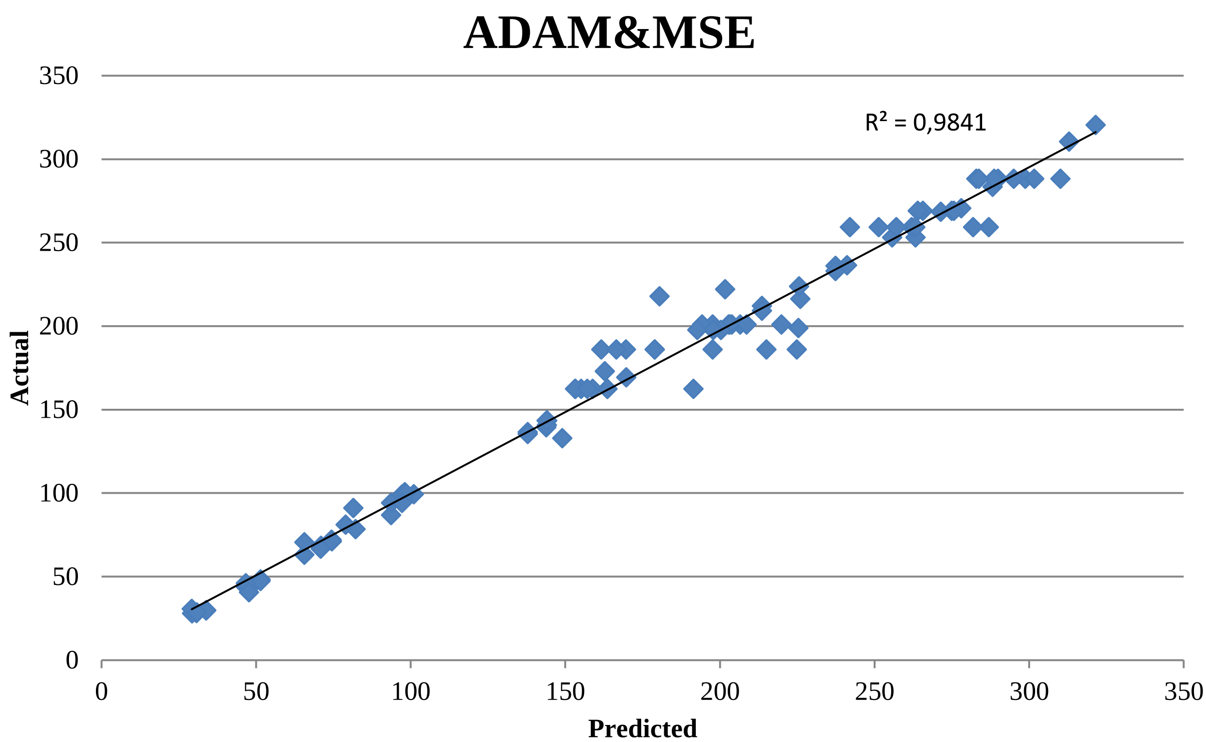

Figure 4.24. Comparison of Test Results and Predicted Results of Adam Optimizer and MSE loss.

When the deep learning model of each row is run with GEP and NN using Adam Optimizer and MSE loss functions, the MAEs of the results are compared in Figure 4.25. As can be seen in the figure, the difference between the results at the point where the highest value belongs to GEP; When the closeness to real values is considered, it is seen that NN is ahead in some points and deep learning results are closer in some points.

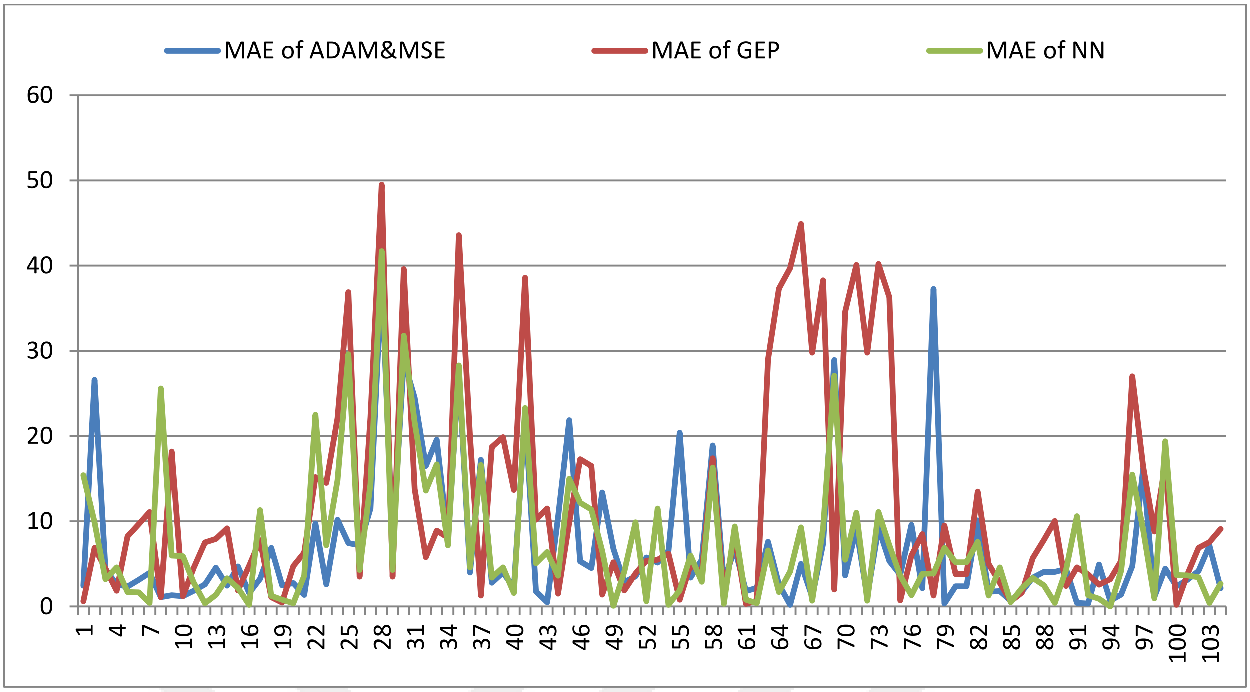

Figure 4.25. MAE of Each Results for Adam Optimizer and MSE

4.5 Comparison of Adadelta Optimizer's Results and Laboratuar Test Results

In Table 4.6, the results obtained by using both the loss parameters and the Adadelta optimizer of the deep learning method with the previous methods GEP and NN are compared. According to this table, the best results are found for *MAD, MSE, RMSE, MAE, MAPE, MEAN and correlation.* The standard deviation for Adadelta Optimizer and for MAE Loss is much better than NN method.

Table 4.6. Statistical Results of Adadelta Optimizer.

	MAD	MSE	RMSE	MAE	MAPE	MEAN	Std	Correlation
GEP	11.9448	300.4576	17.3337	0.0833	8.3350	1.0322	0.1308	0.9782
NN	7.2895	117.0553	10.8192	0.0419	4.1852	**0.9988**	**0.0586**	0.9915
ADADELTA & MAE	**6.3152**	**114.0623**	**10.6800**	**0.0393**	**3.9315**	1.0178	0.0649	**0.9919**
ADADELTA & MSE	7.2662	120.5526	10.9796	0.0460	4.5992	1.0031	0.0666	0.9915

When using Adadelta as optimizer and MAE as loss, the model's train and test accuracy is shown in Figure 4.26. The comparison of the test results of each row with the prediction results from the model likes in Figure 4.27. Mean of Test Dataset 50.836 and mean of train dataset is 23.599. Std of Test Dataset is 76.909 and std of train dataset is 35.286 for Adam optimizer and MAE loss.

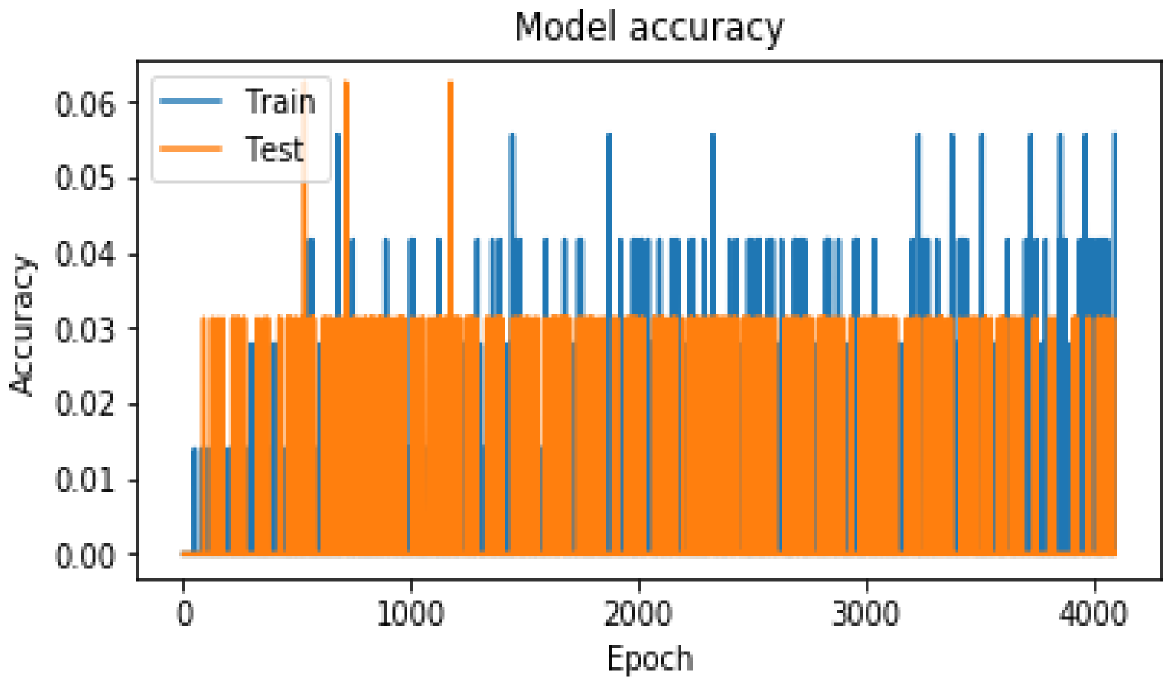

Figure 4.26. Comparison testing and training accuracy for Adadelta Optimizer and MAE loss.

Comparison of the test results in the dataset with the estimation results from the model illustrated in Figure 4.27. When the closeness to the correct result is examined, it is observed that in 63 of 105 samples, the deep learning model gave more realistic results than the NN method.

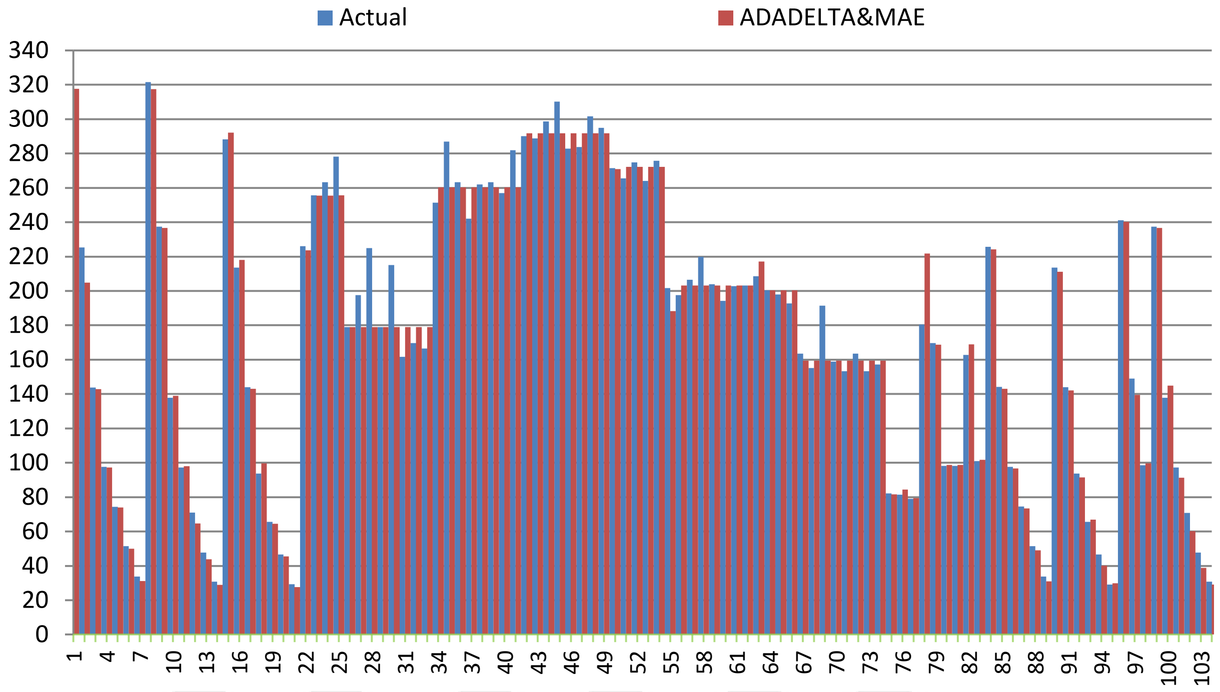

Figure 4.27. Actual datas and predicted results for Adadelta Optimizer and MAE loss.

In the graph in Figure.4.28, the test results and the model results are compared in terms of value. According to this graph, the range of 230-280 is the range with the highest deviations and R^2 value is 0.9838.

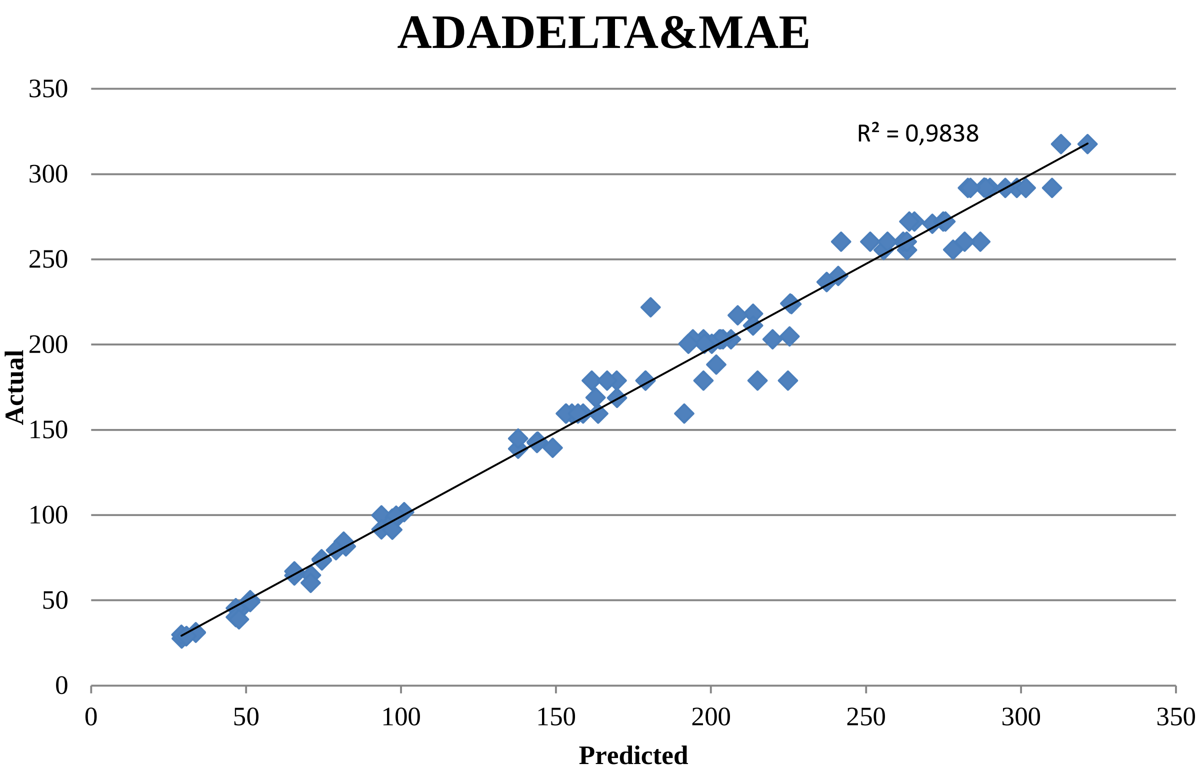

Figure 4.28. Comparison of Test Results and Predicted Results of Adadelta Optimizer and MAE loss.

When the deep learning model of GEP and NN of each row is run using Adadelta Optimizer and MAE loss functions, the MAEs of the results are compared in Figure 4.29. As can be seen in the figure, the difference between the results at the point where the highest value belongs to GEP; the closest points to real values belong to deep learning.

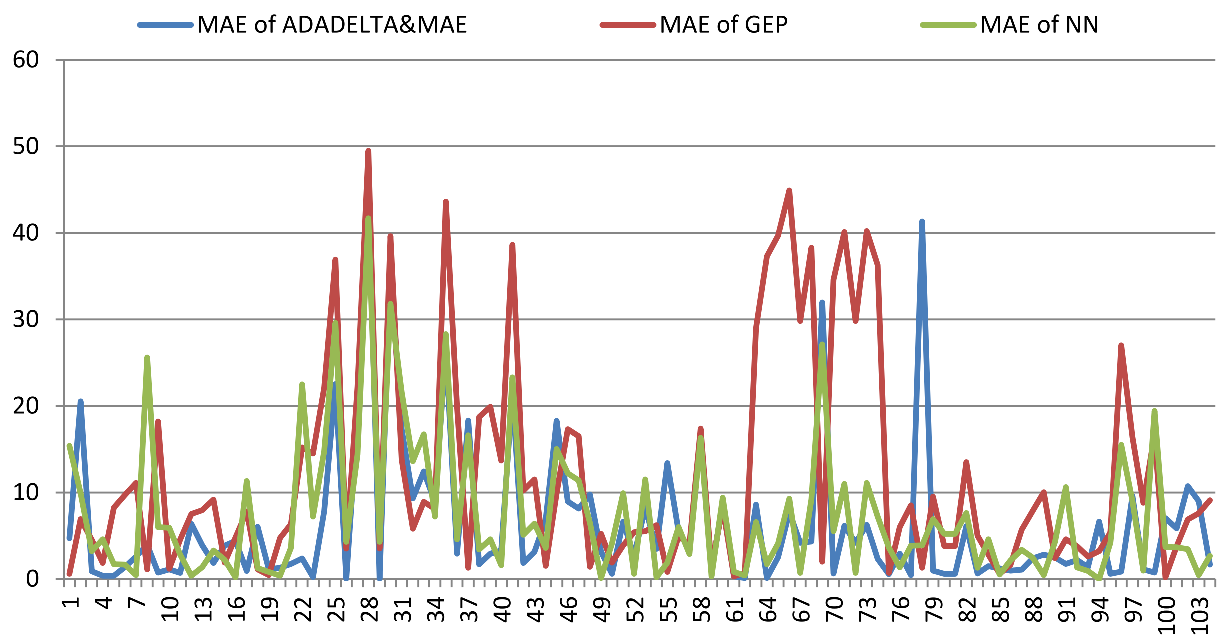

Figure 4.29. MAE of Each Results for Adadelta Optimizer and MAE.

Train and test accuracy of the model when using Adadelta as optimizer and MAE as loss also seen in Figure 4.30. The comparison of the test results of each row in the dataset with the estimation results from the model is as in Figure 4.31.Mean of Test Dataset 111.792 and mean of train dataset is 46.638. Std of Test dataset is 106.252 and std of train dataset is 43.672 for Adam optimizer and MAE loss.

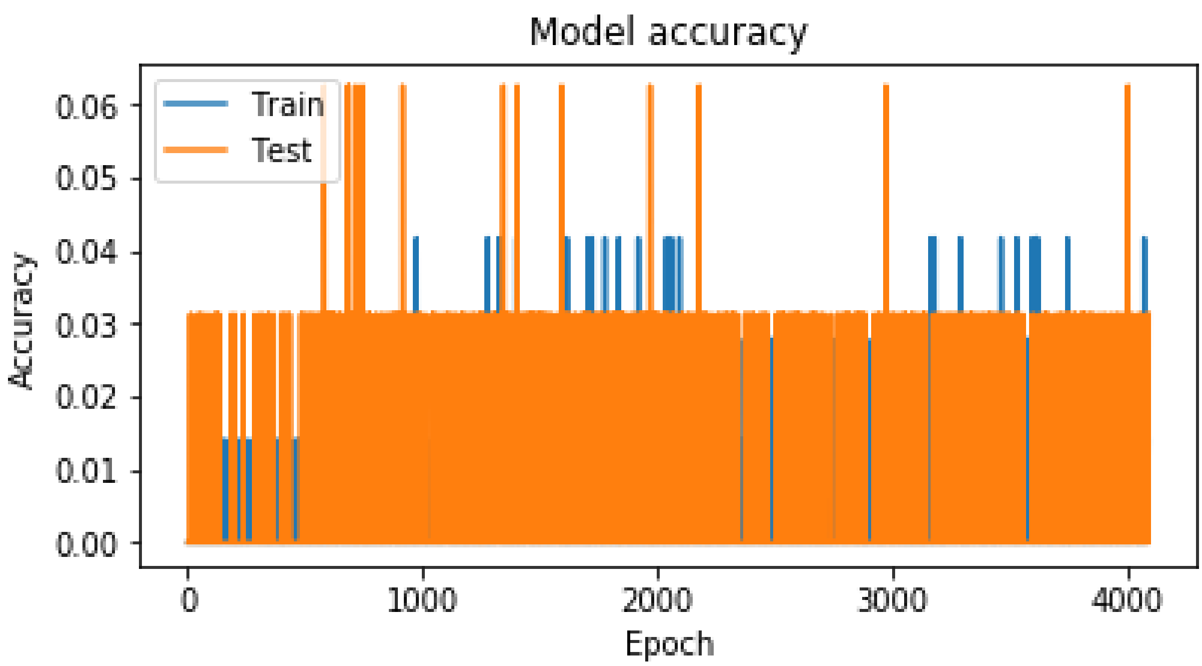

Figure 4.30. Comparison testing and training accuracy for Adadelta Optimizer and MSE

loss.

Comparison of the test results in the dataset with the prediction results from the model is illustrated in Figure 4.31. When the closeness to the correct result is examined, it is observed that deep learning model gives closer to the real results than the NN method in 56 of 105 samples.

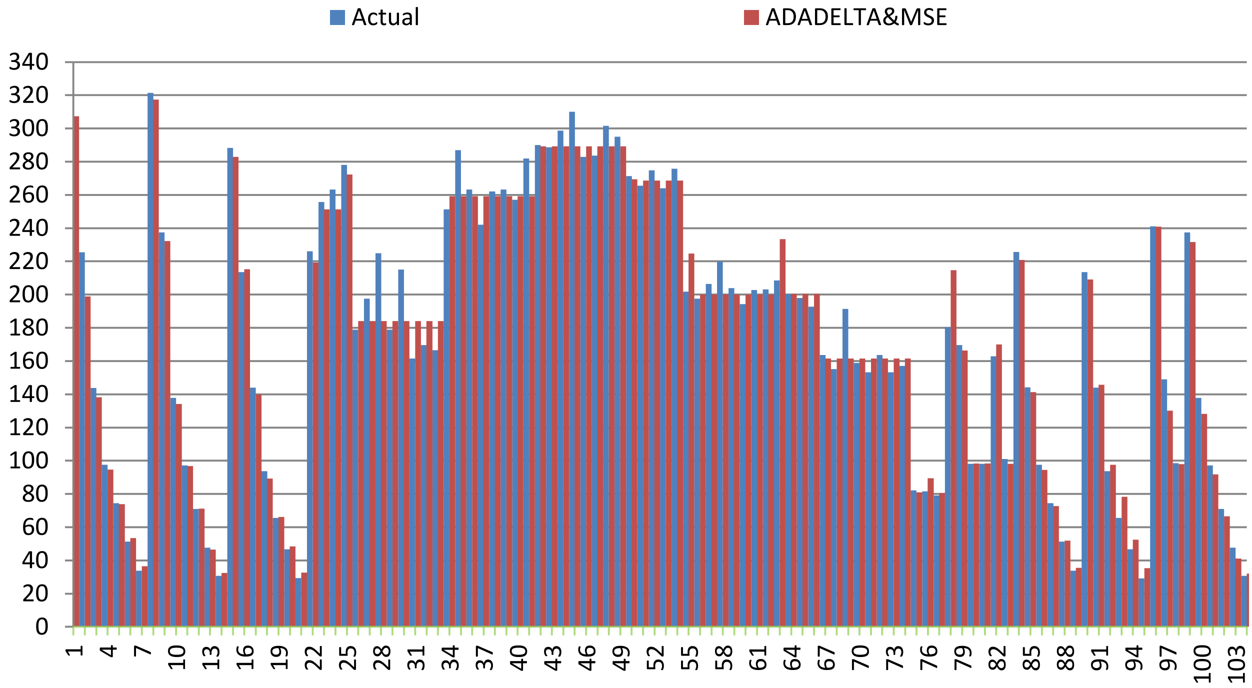

Figure 4.31. Actual datas and predicted results for Adadelta Optimizer and MSE loss.

In the graph in Figure.4.32, test results and model results are compared in terms of value. According to this graph, the range of 270-290 and 180-210 are the ranges with the highest deviations and R^2 is 0.983.

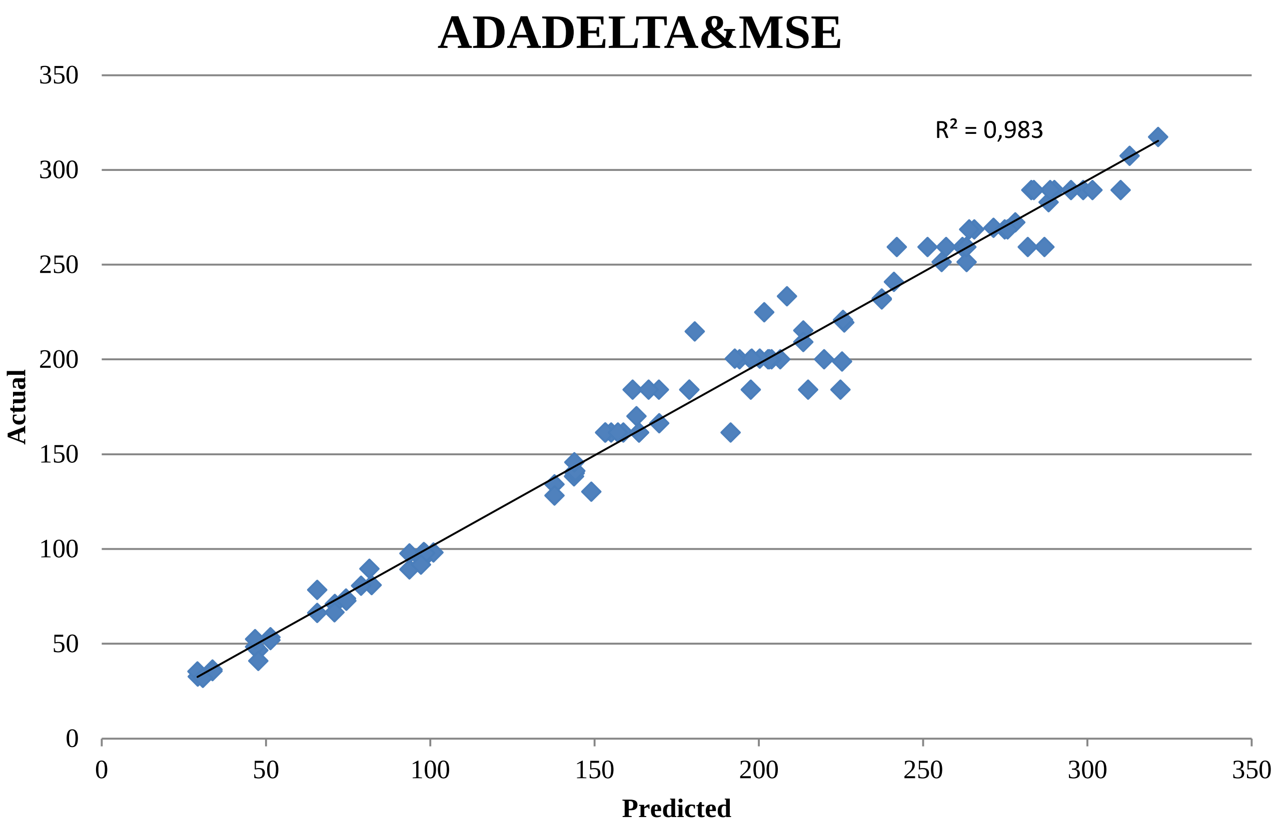

Figure 4.32. Comparison of Test Results and Predicted Results of Adadelta Optimizer and MSE loss.

When the deep learning model of each row with GEP and NN is run using Adadelta Optimizer and MSE loss functions, the MAEs of the results are compared in Figure 4.33.

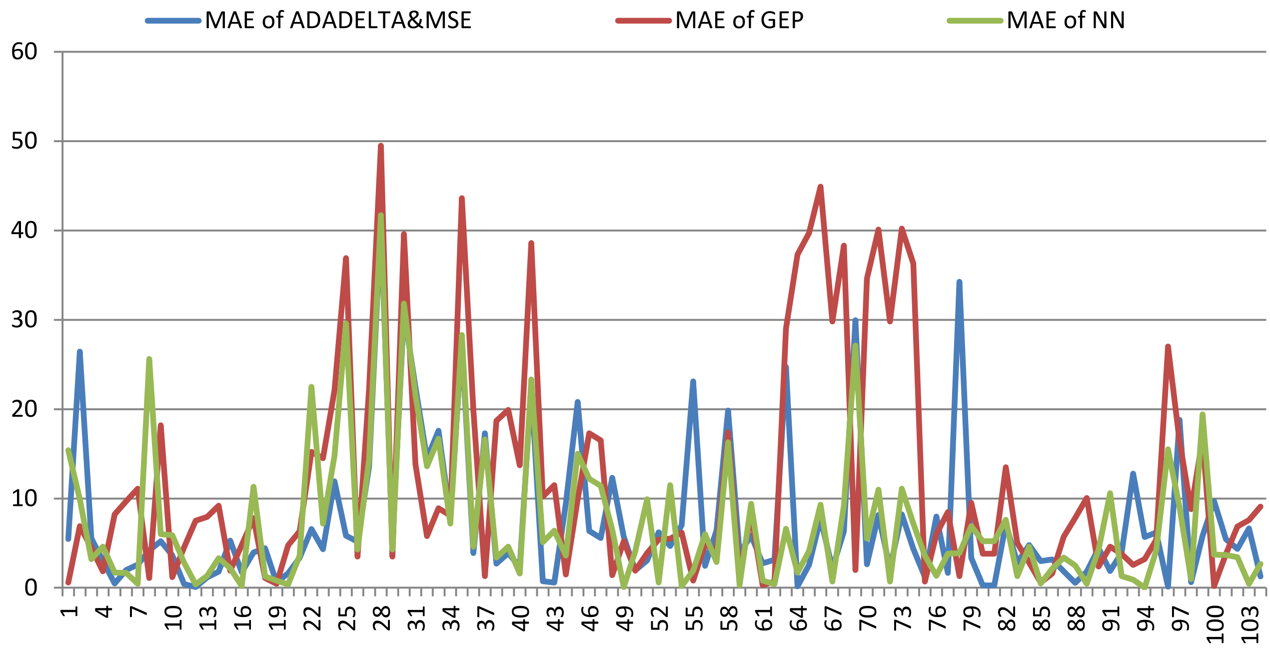

Figure 4.33. MAE of Each Results for Adadelta Optimizer and MSE.

4.6 Comparison of Adagrad Optimizer's Results and Laboratuar Test Results

The model results that used Adagrad optimizer are shown in Table 4.7.

Table 4.7. Statistical Results of Adagrad Optimizer.

	MAD	**MSE**	**RMSE**	**MAE**	**MAPE**	**MEAN**	**Std**	**Correlation**
GEP	11.9448	300.4576	17.3337	0.0833	8.3350	1.0322	0.1308	0.9782
NN	7.2895	**117.0553**	**10.8192**	**0.0419**	**4.1852**	**0.9988**	**0.0586**	**0.9915**
ADAGRAD & MAE	**6.9978**	136.4832	11.6826	0.0452	4.5225	1.0068	0.0754	0.9902
ADAGRAD & MSE	7.0842	134.2247	11.5855	0.0459	4.5881	1.0077	0.0712	0.9903

When using Adagrad as an optimizer and MAE as a loss, the model's train and test accuracy are also seen in Figure 4.34. The comparison of the test results for each row in the dataset with the predicted results from the model is demonstrated in Figure 4.35. Mean of Test

Dataset 71.730 and mean of train dataset is, 22.988. Std of Test Dataset is 110.752 and std of train dataset is 34.118 for Adagrad optimizer and MAE loss.

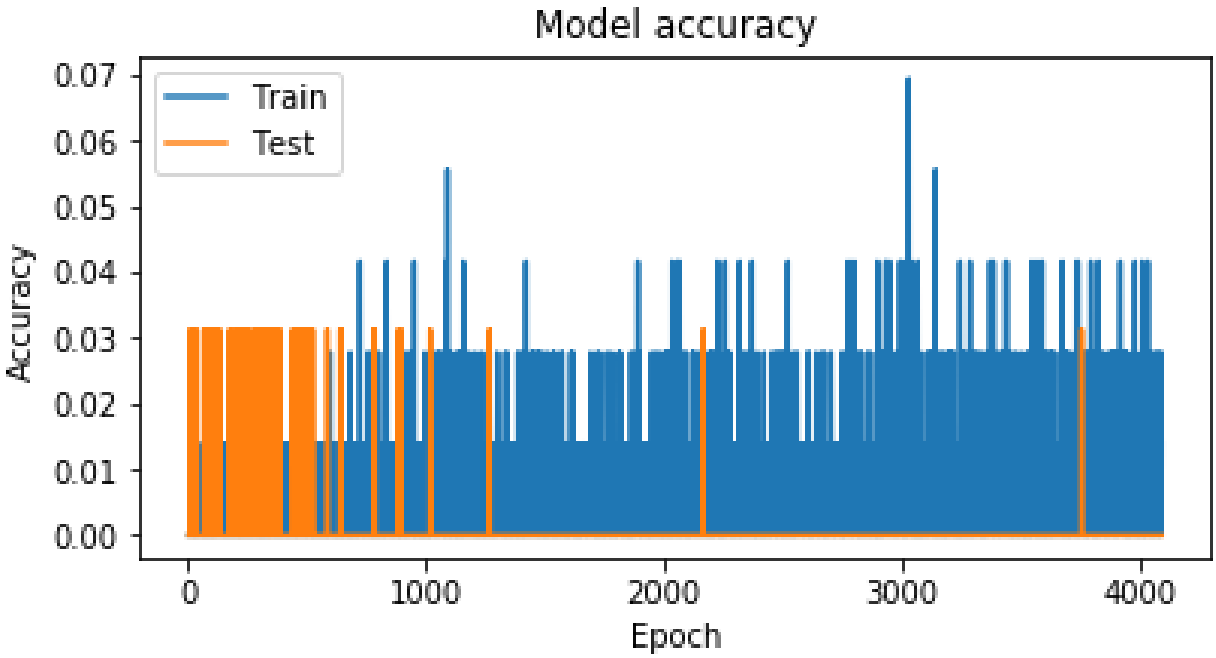

Figure 4.34. Comparison testing and training accuracy for Adagrad Optimizer and MAE loss.

Comparison of the test results in the dataset with the estimation results from the model as in Figure 4.35..When the closeness to the correct result is examined, it has been observed that deep learning model gives closer results to the reality than the NN method in 64 of 105 samples.

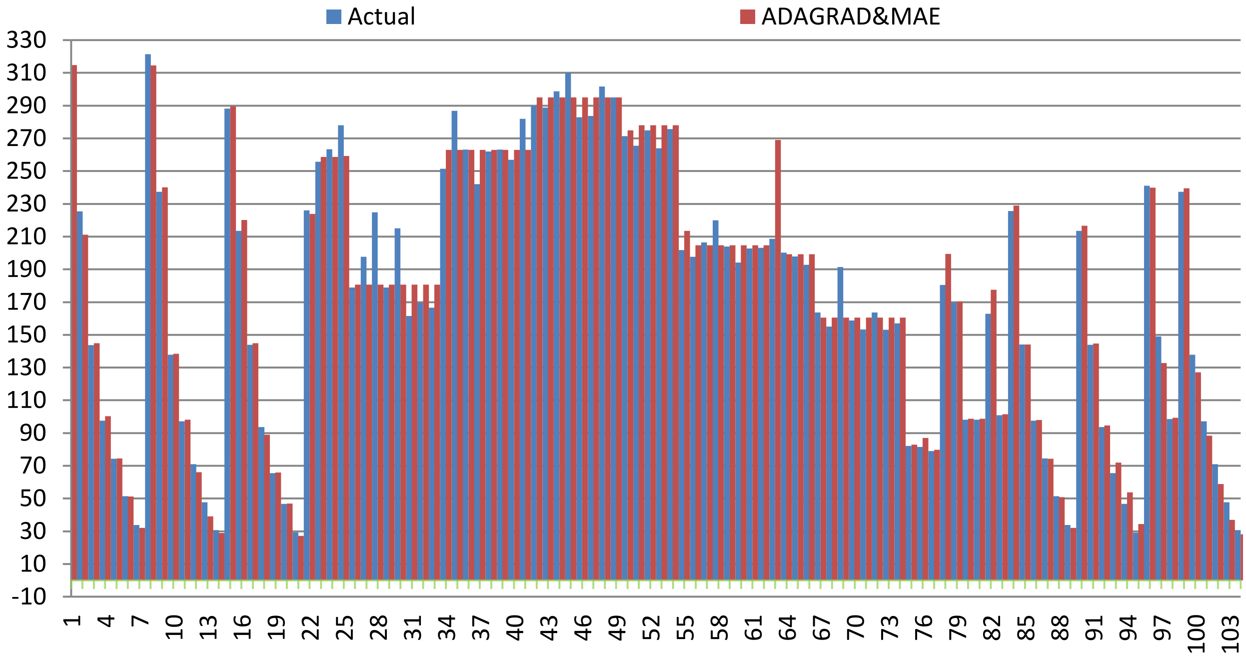

Figure 4.35. Actual datas and predicted results for Adagrad Optimizer and MAE loss.

In the graph in Figure 4.36, test results and model results are compared in terms of value. According to this graph, the range of 220-270 is the range with the highest deviations and R^2 value is 0.9805

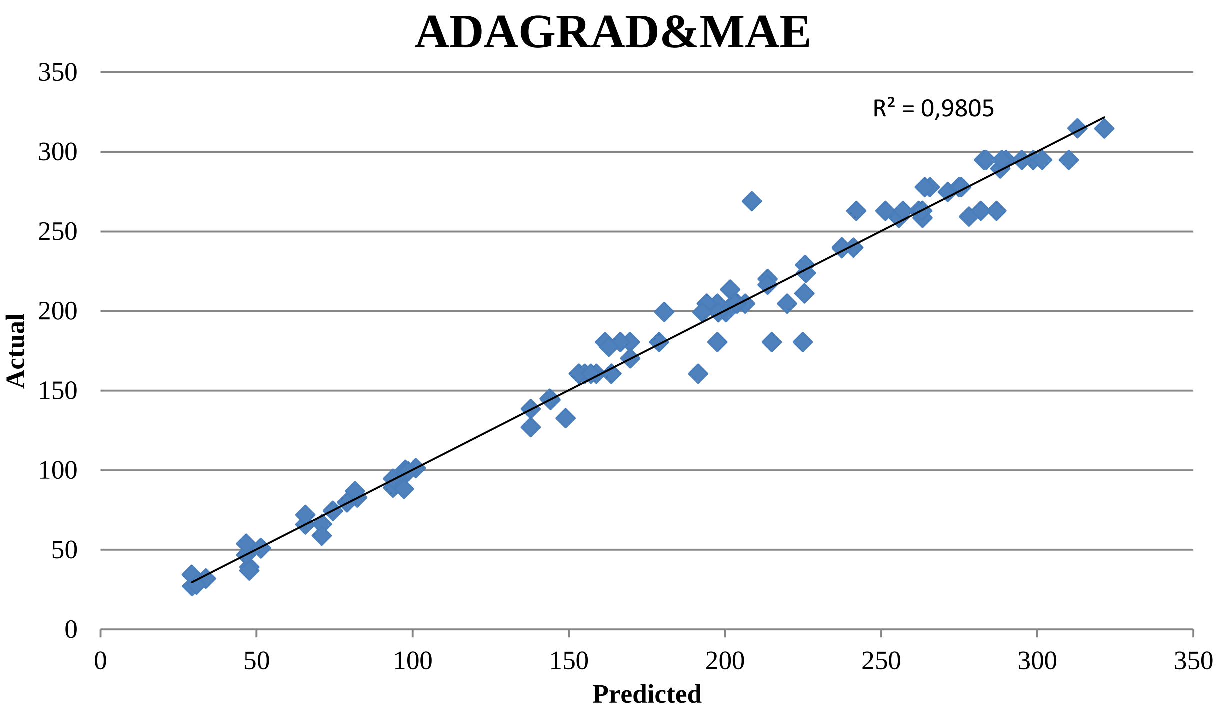

Figure 4.36. Comparison of Test Results and Predicted Results of Adagrad Optimizer and MAE loss.

When the deep learning model of GEP and NN of each line is run using Adagrad Optimizer and MAE loss functions, the differences between the results and the actual results are compared in Figure 4.37. As shown in the figure, the value at the point where the differences between the results are highest belongs to deep learning.

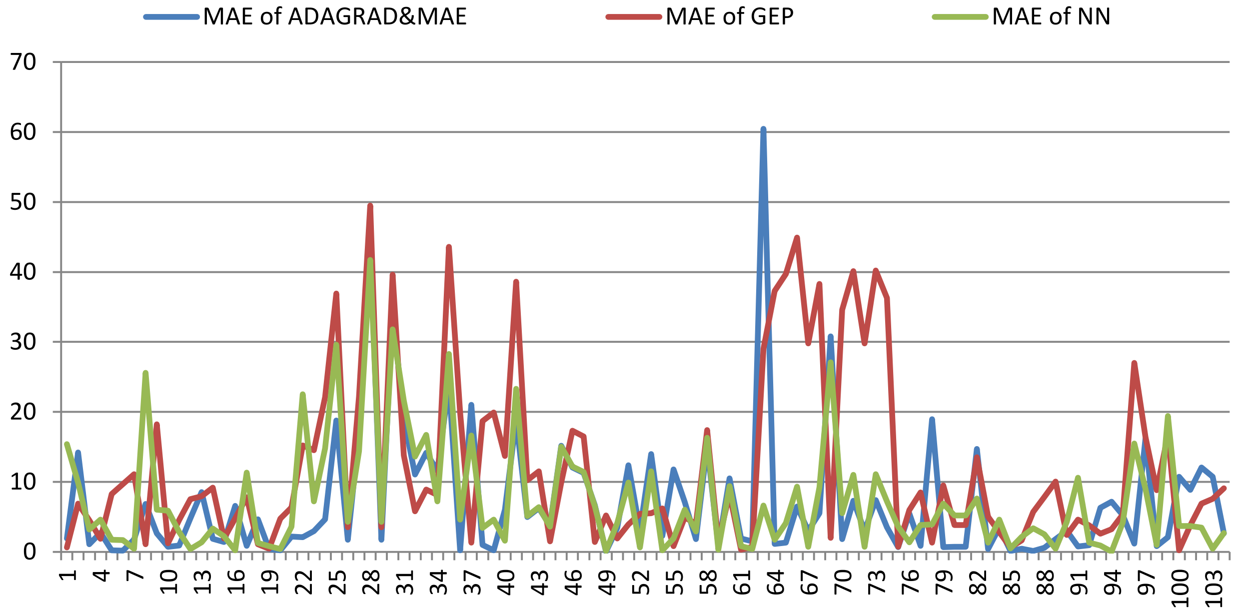

Figure 4.37. MAE of Each Results for Adagrad Optimizer and MAE.

When using Adagrad as optimizer and MSE as loss, the model's train and test accuracy are indicated in Figure 4.38. The comparison of the test results for each row in the dataset and the predicted results from the model is as in Figure 4.39. Mean of Test Dataset 130.897 and mean of train dataset is 42.285. Std of Test dataset is 125.127 and std of train dataset is 39.741 for Adagrad optimizer and MAE loss.

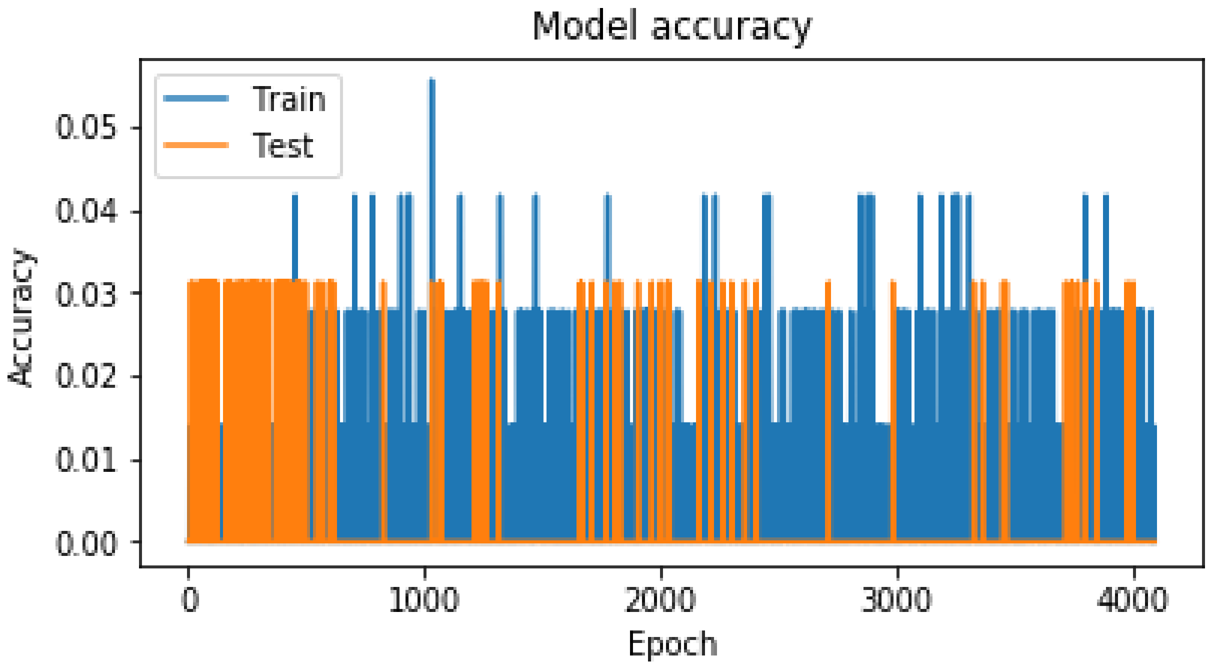

Figure 4.38. Comparison testing and training accuracy for Adagrad Optimizer and MSE loss.

Comparison of the test results in the dataset with the estimation results from the model seen in Figure 4.39. When the closeness to the correct result is examined, it is observed that deep learning model gave closer results to the reality than the NN method in 57 of 105 samples.

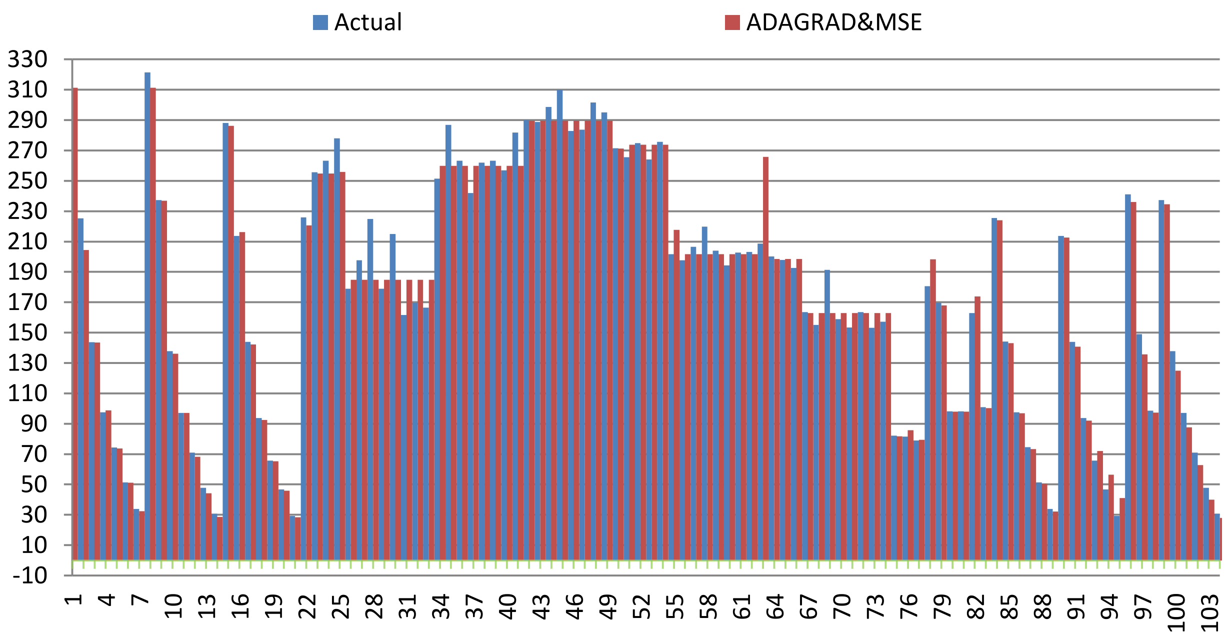

Figure 4.39. Actual datas and predicted results for Adagrad Optimizer and MSE loss.

Figure 4.40 in the graph of the test results are compared in terms of the value of the model results. According to this graph, the range of 160-180 and 240-820 is the range with the highest deviations and R^2 value is 0.9806.

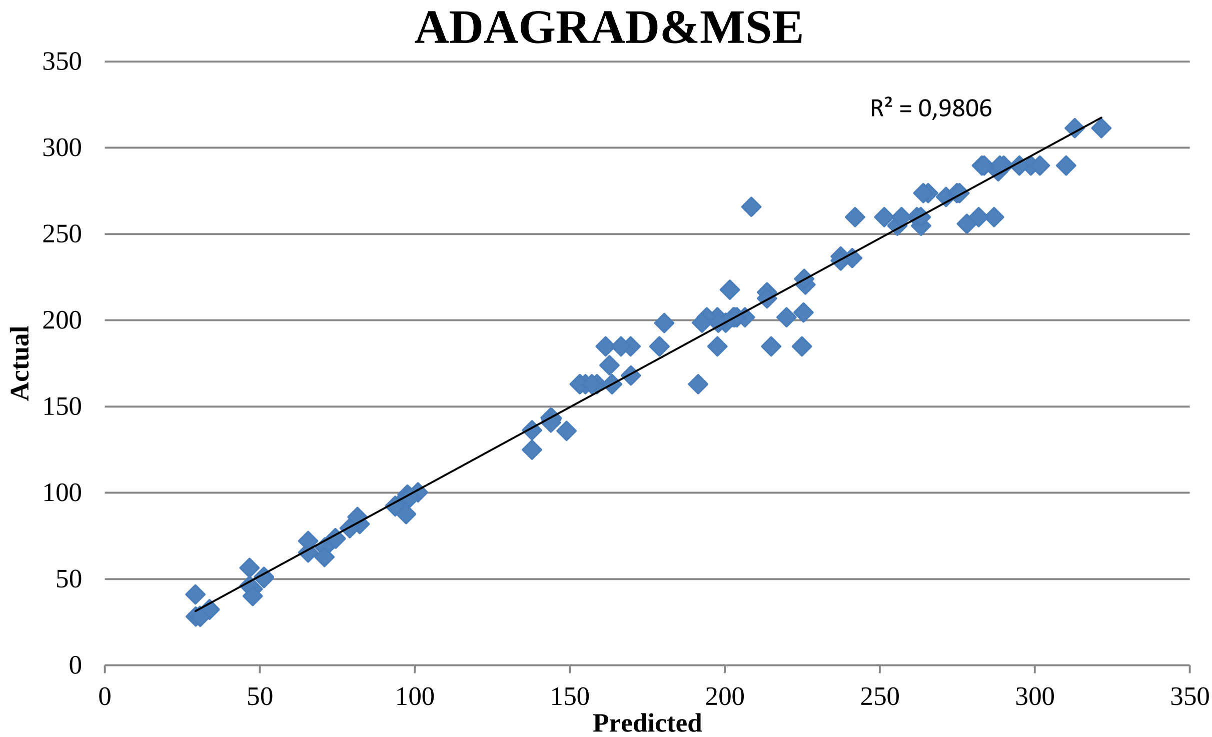

Figure 4.40. Comparison of Test Results and Predicted Results of Adagrad Optimizer and MSE loss.

When the deep learning model of GEP and NN of each row is run using Adagrad Optimizer and MSE loss functions, the differences between the results and the actual results are compared in Figure 4.41. As can be seen in the figure, the difference between the results at the highest point, although the value of deep learning is close to the point of 0 is also remarkable.

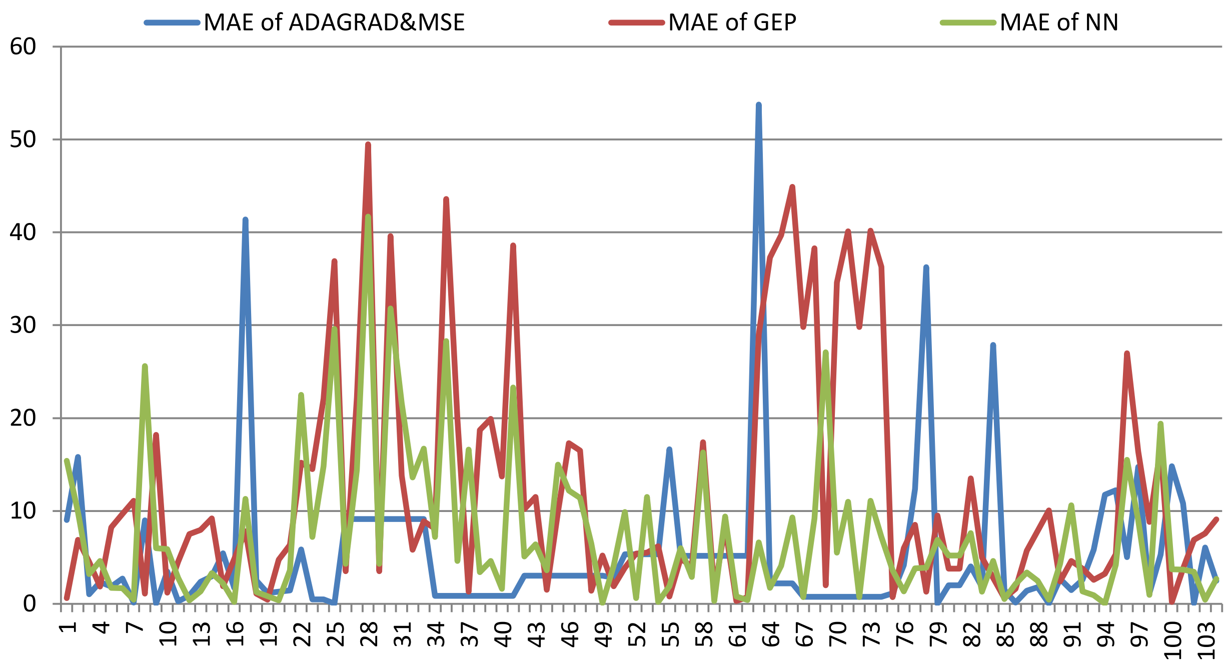

Figure 4.41. MAE of Each Results for Adagrad Optimizer and MSE.

4.7 Comparison of RMSProp Optimizer's Results and Laboratuar Test Results

In Table 4.8, the results obtained by using both loss parameters are compared with the previous methods GEP and NN and RMSProp optimizer of deep learning method. According to this table, the best results for *MAD, MAE, MAPE, RMSE, MSE, correlation are* seen in RMSProp Optimizer and MAE Loss. In this case, we can say that the MAE loss function is more successful in the deep learning model for RMSPROP Optimizer.

Also, according to Standart Deviation evaluation results, NN method of soft computing techniques is better than deep Learning methods that used RMSProp Optmizer.

Table 4.8. Statistical Results of RMSPROP Optimizer.

	MAD	**MSE**	**RMSE**	**MAE**	**MAPE**	**MEAN**	**STD**	**Correlation**
GEP	11.9448	300.4576	17.3337	0.0833	8.3350	1.0322	0.1308	0.9782
NN	7.2895	117.0553	10.8192	0.0419	4.1852	0.9988	**0.0586**	0.9915

RMSPROP & MAE	**6.5172**	**116.2108**	**10.7801**	**0.0388**	**3.8776**	**1.0213**	0.0720	**0.9925**
RMSPROP & MSE	6.9357	117.9624	10.8611	0.0438	4.3766	1.0230	0.0628	0.9922

When RMSPROP is used as an optimizer and MAE is used as a loss, the model's train and test accuracy are shown in Figure 4.42.

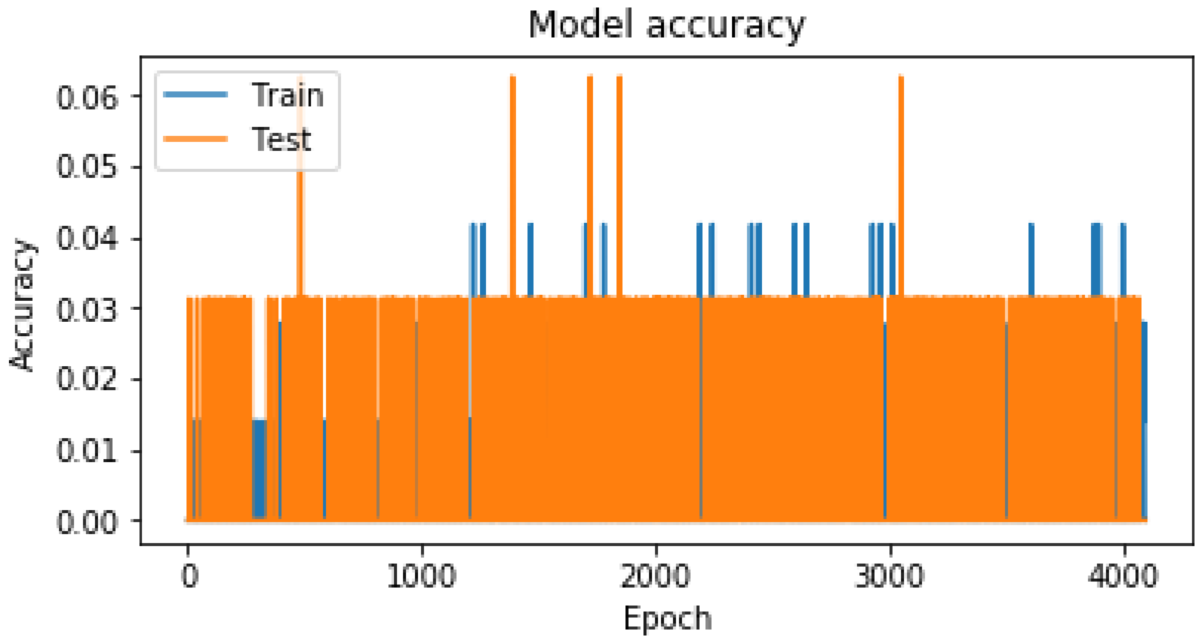

Figure 4.42. Comparison testing and training accuracy for RMSPROP Optimizer and MAE loss.

The comparison of the test results for each row in the dataset and the predicted results from the model is as shown in Figure.4.43. Mean of Test Dataset 91.529 and mean of train dataset is 45.341.Std of Test Dataset is 142.794 and std of train dataset is 66.679 for RMSPROP optimizer and MAE loss.

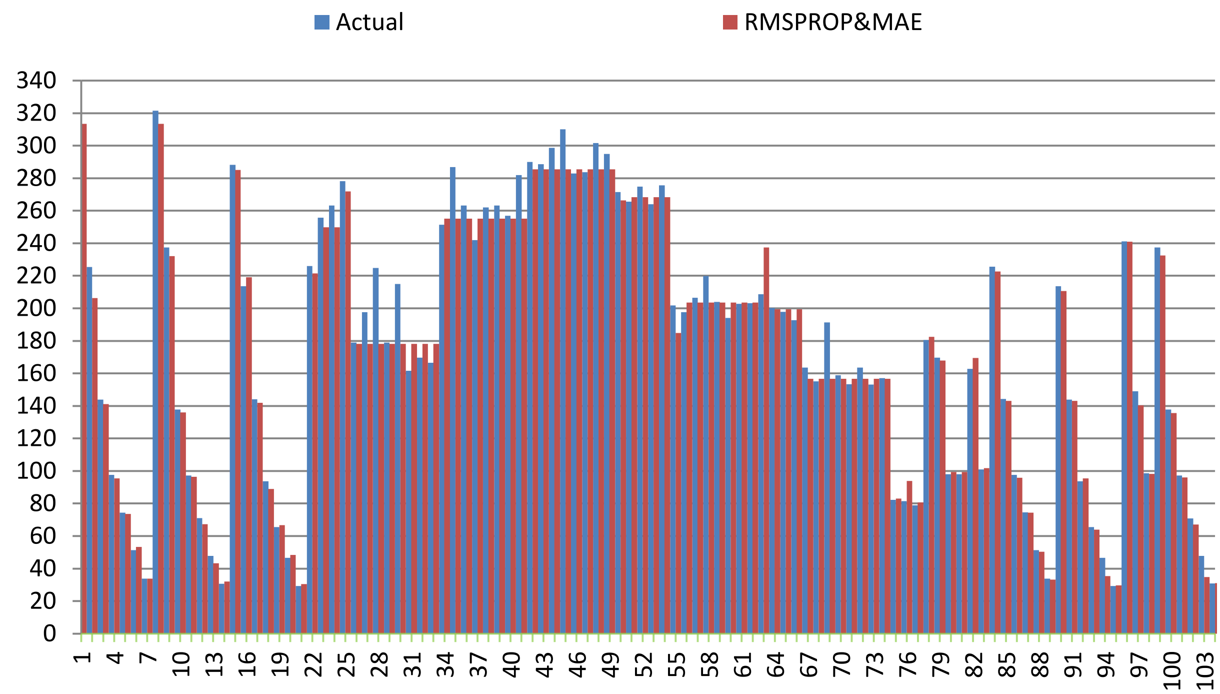

Figure 4.43. Actual datas and predicted results for RMSPROP Optimizer and MAE loss.

In the Figure 4.44, test results and model results are compared in terms of value. According to this graph, the range of 270-300 is the range with the highest deviations and R^2 is 0.985.

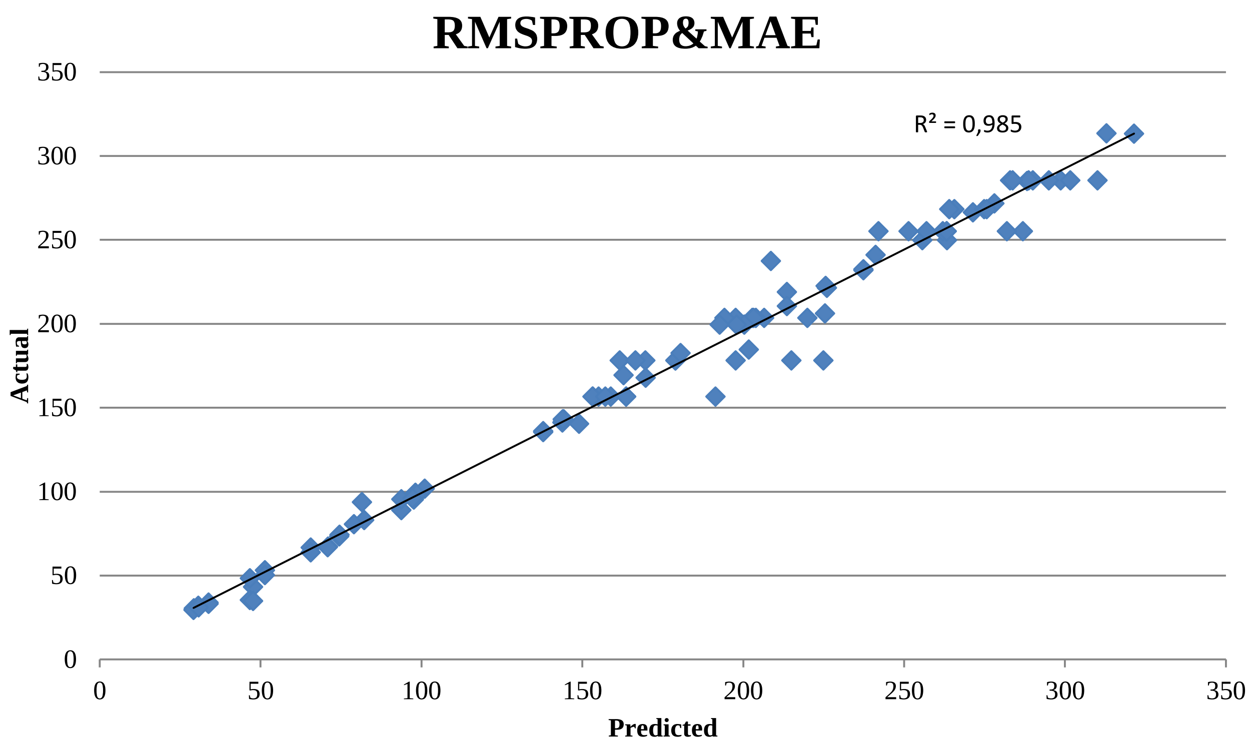

Figure 4.44. Comparison of Test Results and Predicted Results of RMSPROP Optimizer and MAE loss.

When the deep learning model of GEP and NN of each row is run using RMSPROP Optimizer and MAE loss functions, the differences between the results and the actual results are compared in Figure 4.45. As shown in the figure, the value at the point where the differences between the results are highest belongs to deep learning.

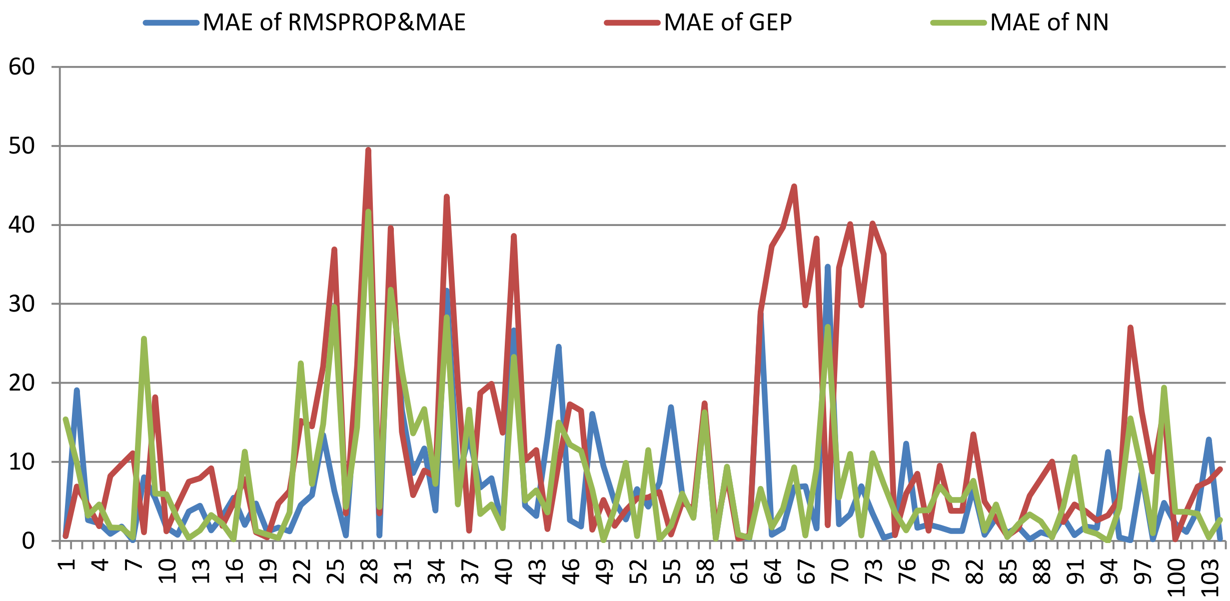

Figure 4.45. MAE of Each Results for RMSPROP Optimizer and MAE.

Train and test accuracy of the model when using RMSPROP as optimizer and MSE as loss in Figure 4.46 also seen. The comparison of the test results for each row in the dataset and the predicted results from the model is as in Figure.4.47. Mean of Test Dataset 106.457 and mean of train dataset is 49.272. Std of Test Dataset is 101.364 and std of train dataset is 46.305 for RMSPROP optimizer and MSE loss.

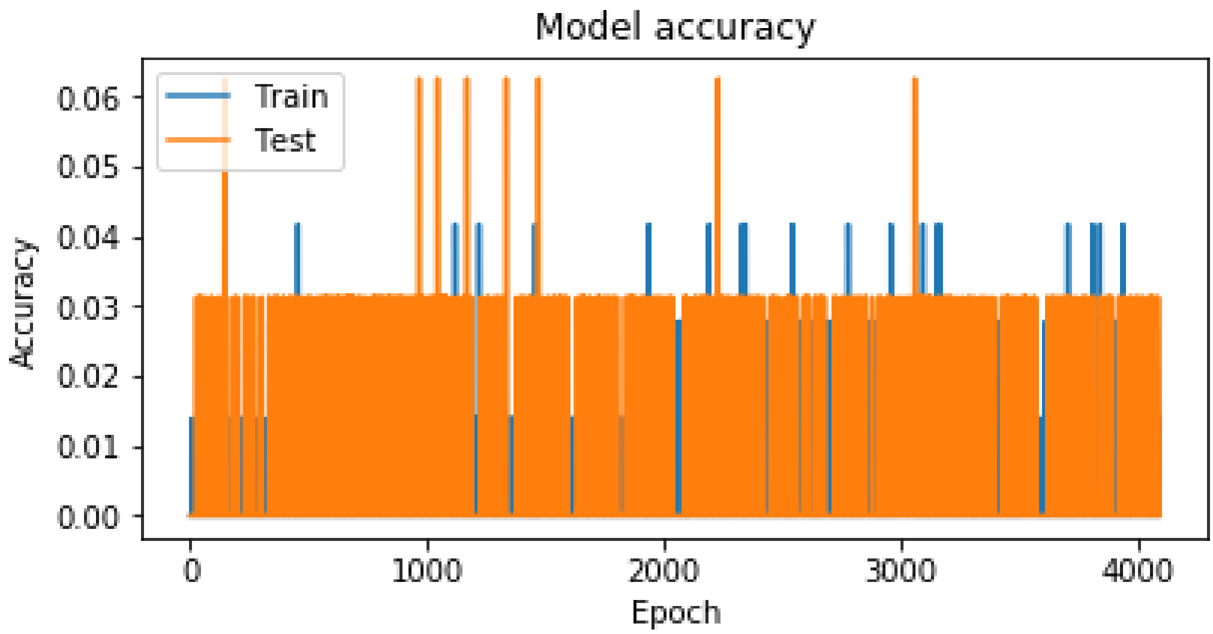

Figure 4.46. Comparison testing and training accuracy for RMSPROP Optimizer and MSE loss.

Comparison of the test results in the dataset with the prediction results from the model is illustrated in Figure 4.47. When the closeness to the correct result is examined, it is observed that deep learning model gave closer results to the reality than the NN method in 60 of 105 samples.

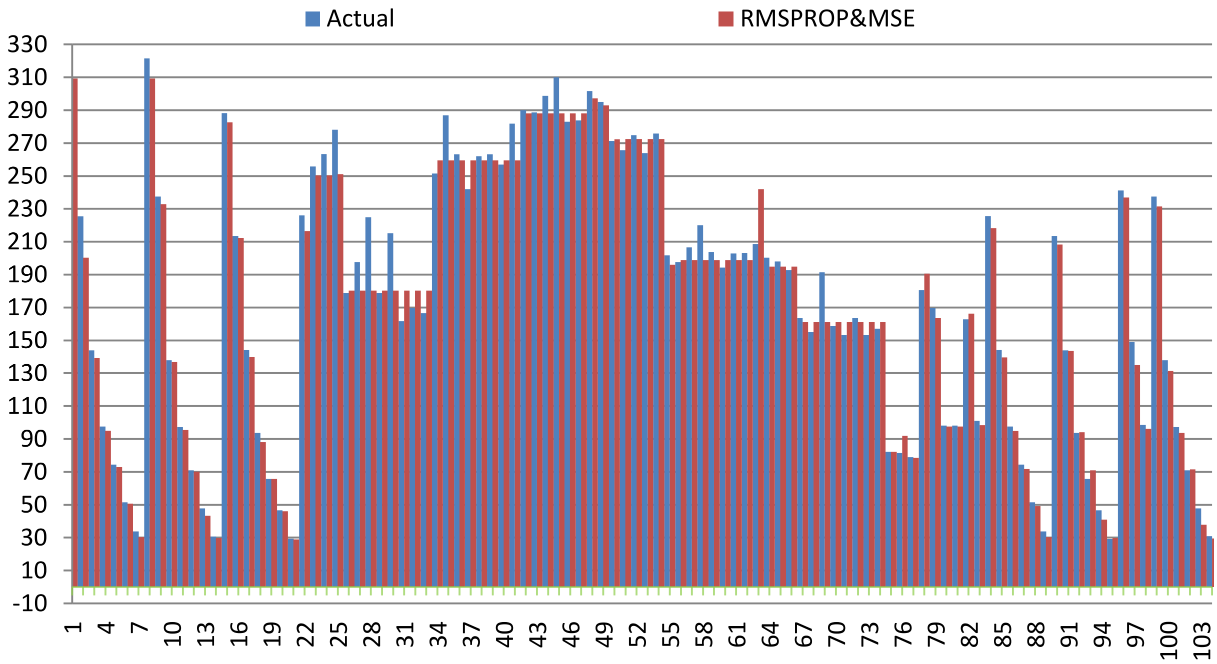

Figure 4.47. Actual datas and predicted results for RMSPROP Optimizer and MSE loss.

In the Figure 4.48 graph is compared with the test results in terms of model results. According to this graph, the range of 240-300 is the range with the highest deviations and R^2 is 0.9844

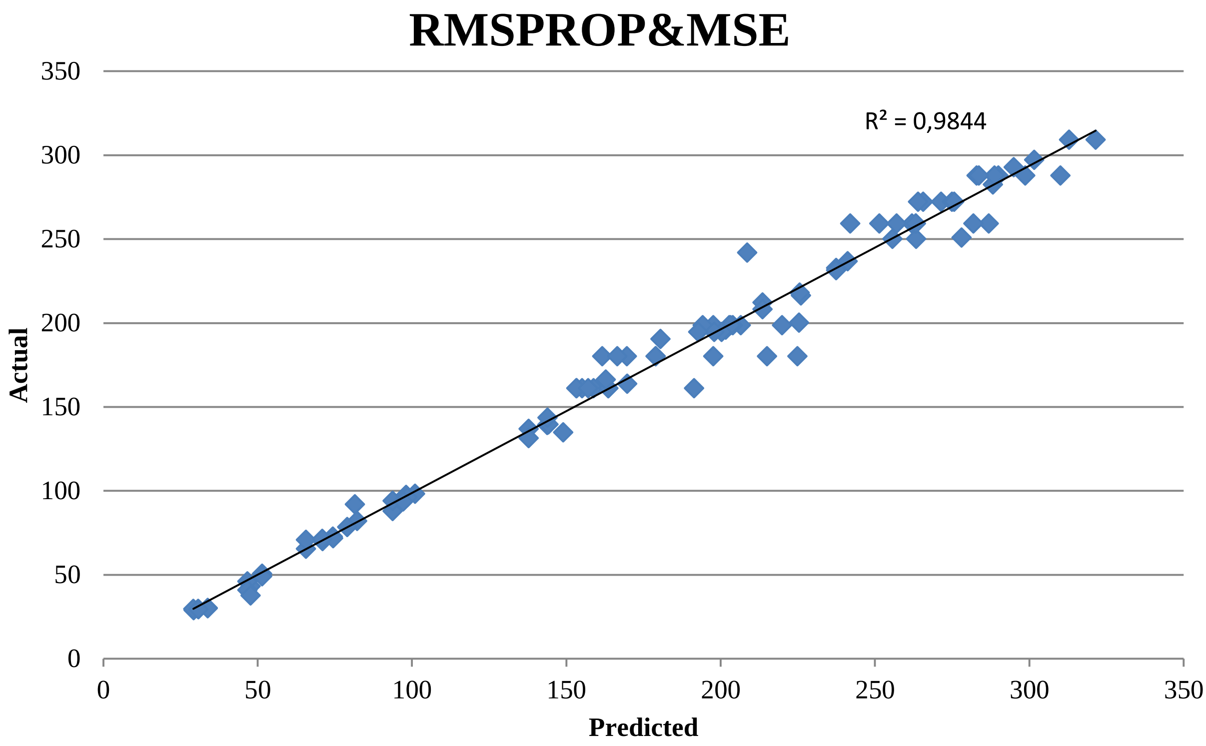

Figure 4.48. Comparison of Test Results and Predicted Results of RMSPROP Optimizer and MSE loss.

When the deep learning model of GEP and NN of each row is run using RMSProp Optimizer and MSE loss functions, the differences between the results and the actual results are compared in Figure 4.49. As can be seen in the figure, the difference between the results is the highest point of the GEP and 2-3 points except RMSProp optimizer and MSE loss using the deep learning model generally achieves the best results.

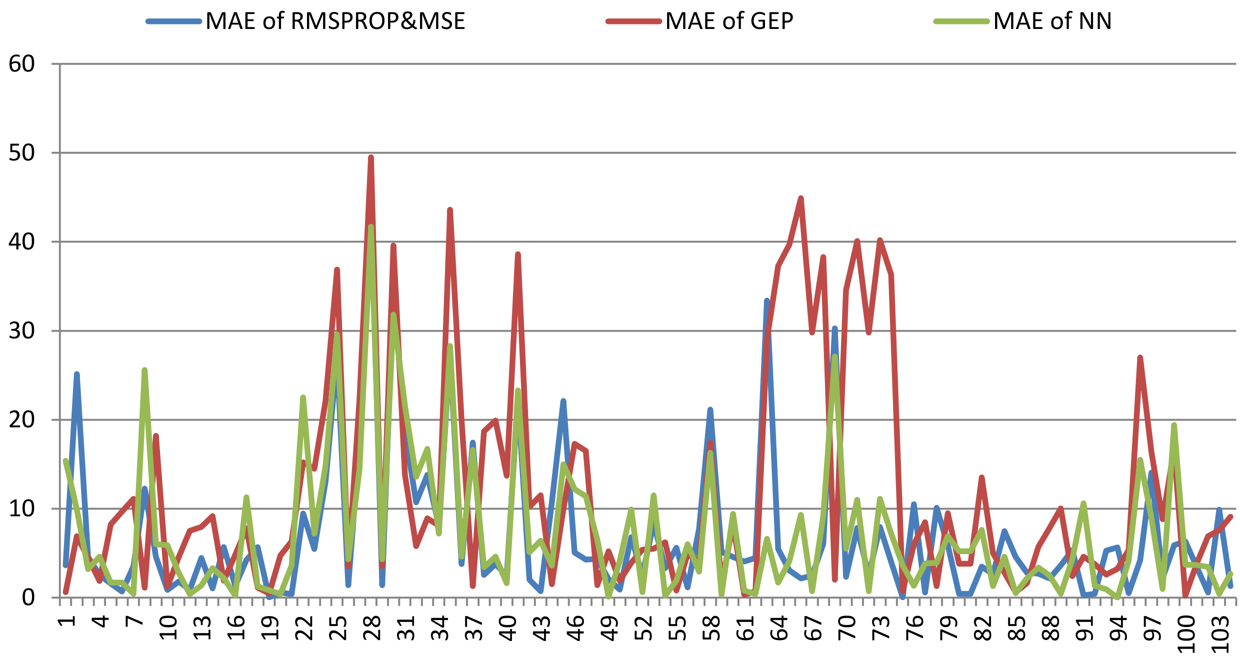

Figure 4.49. MAE of Each Results for RMSPROP Optimizer and MSE.

When using RMSPROP as optimizer and MSE as loss, the train and test accuracy of the model Figure 4.50. The comparison of the test results for each row in the data set and the estimation results from the model is as in Figure.4.51. Std of Test Dataset is 101.364 and std of train dataset is 46.305 for RMSPROP optimizer and MSE loss.

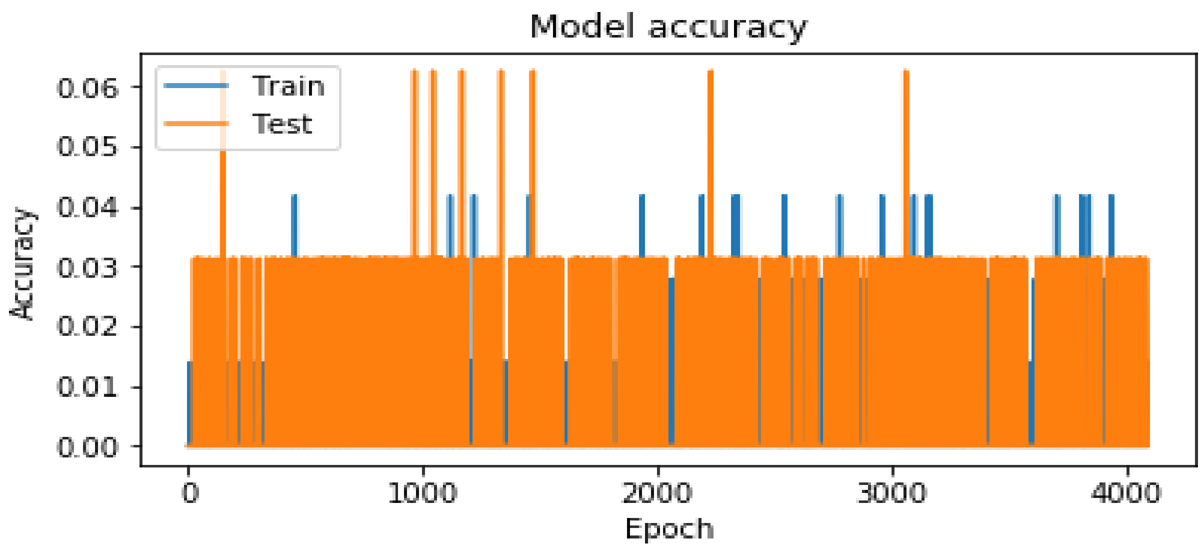

Figure 4.50. Comparison testing and training accuracy for RMSPROP Optimizer and MSE loss.

Comparison of the test results in the dataset with the estimated results from the model that seen in Figure 4.51. When the closeness to the correct result is examined, it is observed that in

60 of 105 samples, deep learning model gives more realistic results than NN method.

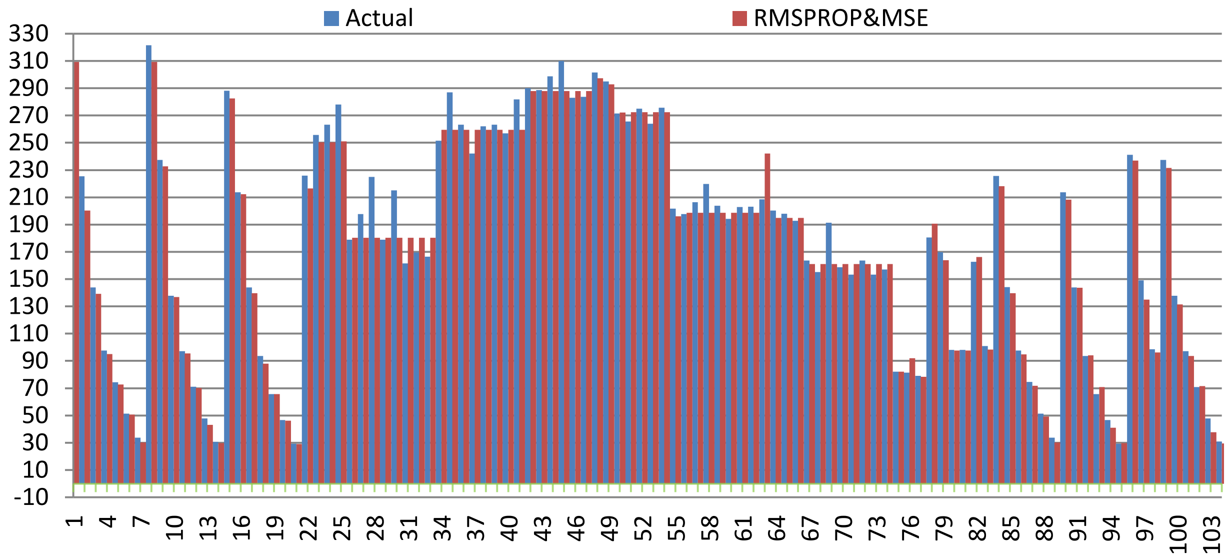

Figure 4.51. Actual datas and predicted results for RMSPROP Optimizer and MSE loss.

In the graph, the test results and the model results are compared in terms of value in Figure 4.52. According to this graph, the range of 240-300 is the range with the highest deviations and R^2 value is 0.9844.

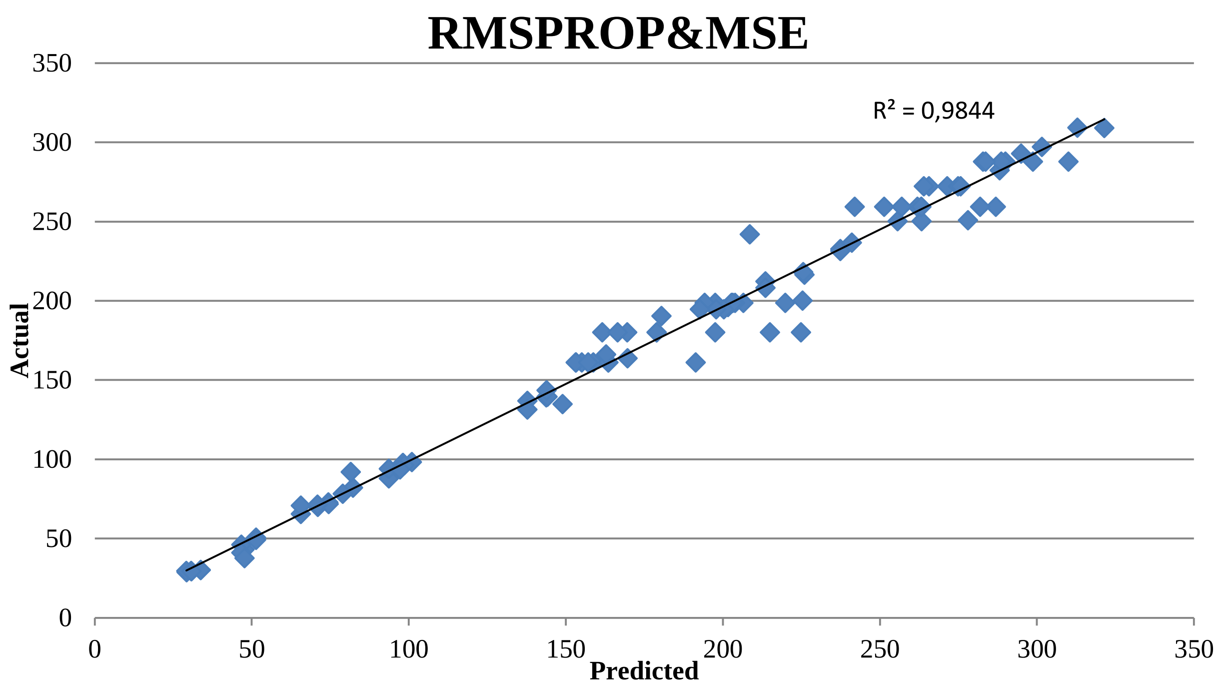

Figure 4.52. Comparison of Test Results and Predicted Results of RMSPROP Optimizer and MSE loss.

When the deep learning model is run using the Adagrad Optimizer and MSE loss functions, the differences between the results and the actual results of each row are compared with GEP and NN, in Figure 4.53.As can be seen in the figure, the highest difference between the results is GEP's result and 2-3 points except the RMSPROP optimizer and MSE loss using the deep learning model generally achieves the best results.

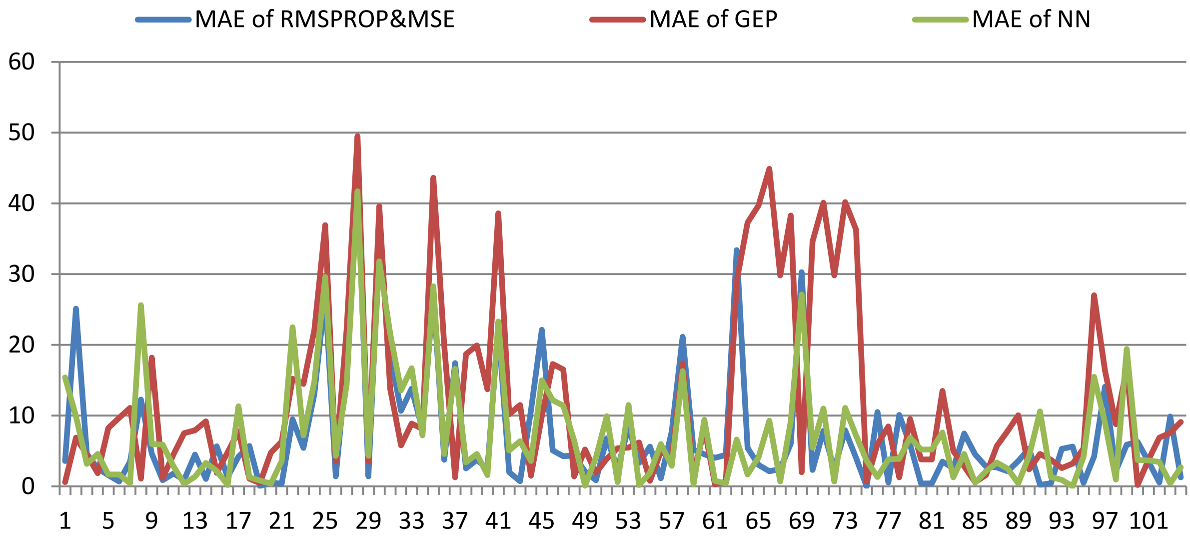

Figure 4.53. MAE of Each Results for RMSPROP Optimizer and MSE Loss.

4.8 Comparison of All Optimizer for MAE loss in Statistical Results

When all the optimizers are compared for MAE loss according to the Table 4.9, it is seen that the best result for *RMSE*, *MSE* is taken from the deep learning model using Adam optimizer. For *MAD*, *MAPE*, *MAE*, *Standard deviation* and *correlation*, the best result appears to be the ADAMAX optimizer.

Table 4.9. Statistical Results of All Optizimer for MAE loss.

	MAD	MSE	RMSE	MAE	MAPE	MEAN	STD	Correl ation
GEP	11.9448	300.4576	17.3337	0.0834	8.3350	1.0322	0.1308	0.9782
NN	7.2895	117.0553	10.8192	0.0419	4.1852	0.9988	**0.0586**	0.9915

ADAM & MAE	6.5120	**100.0853**	**10.0043**	0.0469	4.6943	1.0138	0.0767	0.9928
ADADELTA& MAE	6.3152	114.0623	10.6800	0.0393	3.9315	1.0178	0.0649	0.9919
ADAGRAD & MAE	6.9978	136.4832	11.6826	0.0452	4.5225	1.0068	0.0754	0.9902
RMSPROP & MAE	6.5172	116.2108	10.7801	0.0388	3.8776	1.0213	0.0720	**0.9925**
SGD & MAE	9.0132	166.9209	12.9198	0.0616	6.1643	1.0483	0.0867	0.9897
NADAM & MAE	6.4505	109.0971	10.4450	0.0384	3.8352	**0.9992**	0.0622	0.9920
ADAMAX & MAE	**5.8172**	115.3379	10.7395	**0.0337**	**3.3702**	1.0169	0.0596	0.9922

Also, The Comparison of All Optimizer's results are shown in Figure 4.54 for MAE Loss.

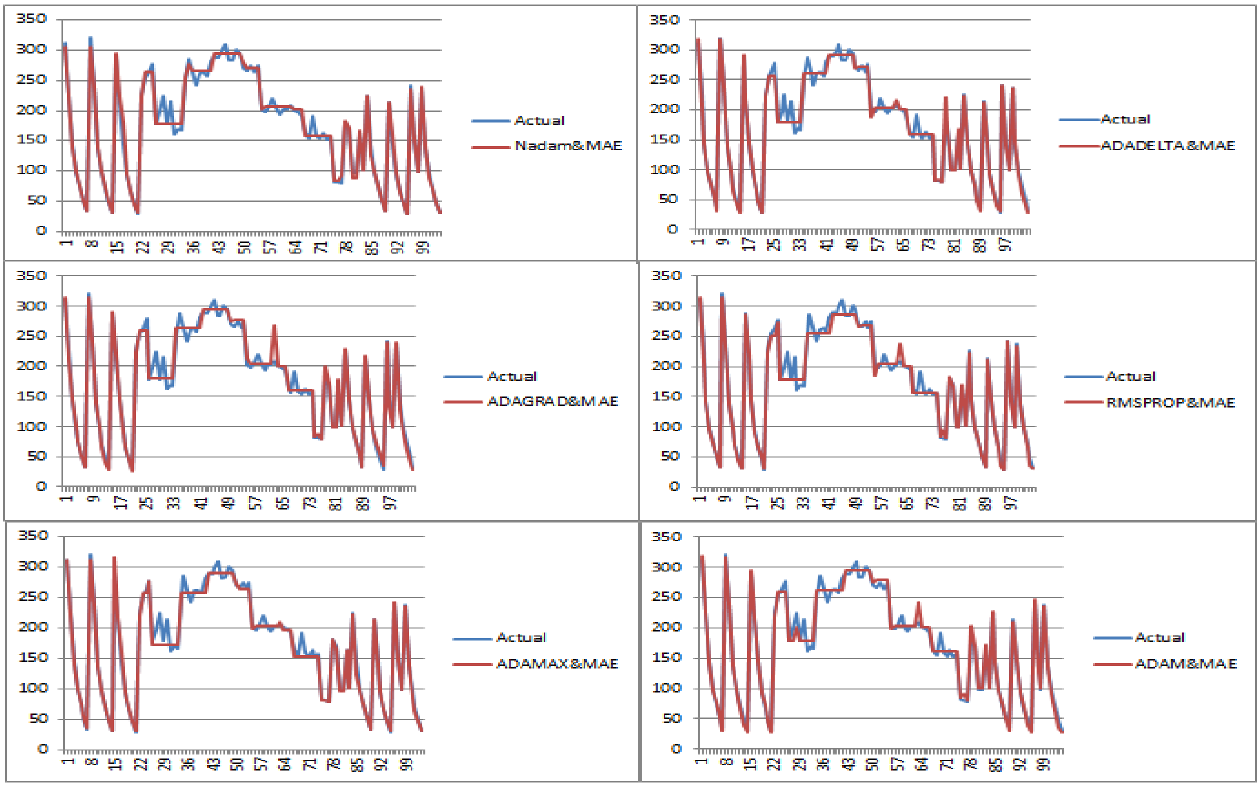

Figure 4.54. Actual values and Predicted values of all Optimizer and MAE Loss.

4.9 Comparison of All Optizimer for MSE loss in Statistical Results

When all the optimizers are compared for MSE loss according to the Table 4.10, it is seen that the best result for MAD, MAE, MAPE is taken from the deep learning model using Nadam optimizer. The best results for *RMSE*, *MSE and correlation* are seen in ADAMAX optimizer, and in Adadeltada for *Mean*, *standard deviation*. In addition, it is observed that the NN model for standard deviation is closer to zero with a slight difference from deep learning algorithms.

Table 4.10 Statistical Results of All Optizimer for MSE Loss.

	MAD	**MSE**	**RMSE**	**MAE**	**MAPE**	**MEAN**	**STD**	**Correlation**
GEP	11.9448	300.4576	17.3337	0.0834	8.3350	1.0322	0.1308	0.9782
NN	7.2895	117.0553	10.8192	0.0419	4.1852	**0.9988**	**0.0586**	0.9915
ADAM & MSE	6.9639	112.9846	10.6294	0.0461	4.6051	1.0174	0.0628	0.9920

ADADELTA & MSE	7.2662	120.5526	10.9796	0.0460	4.5992	1.0031	0.0666	0.9915
ADAGRAD & MSE	7.0842	134.2247	11.5855	0.0459	4.5881	1.0077	0.0712	0.9903
RMSPROP & MSE	6.9357	117.9624	10.8611	0.0438	4.3766	1.0230	0.0628	0.9922
SGD & MSE	6.9947	121.6747	11.0306	0.0439	4.3891	1.0213	0.0638	0.9917
NADAM & MSE	**6.4625**	123.1020	11.0951	**0.0378**	**3.7775**	1.0097	0.0611	0.9912
ADAMAX & MSE	6.4842	**110.5480**	**10.5142**	0.0463	4.6345	1.0309	0.0657	**0.9927**

The Comparison of All Optimizer's Results are shown in Figure 4.55 for MSE Loss.

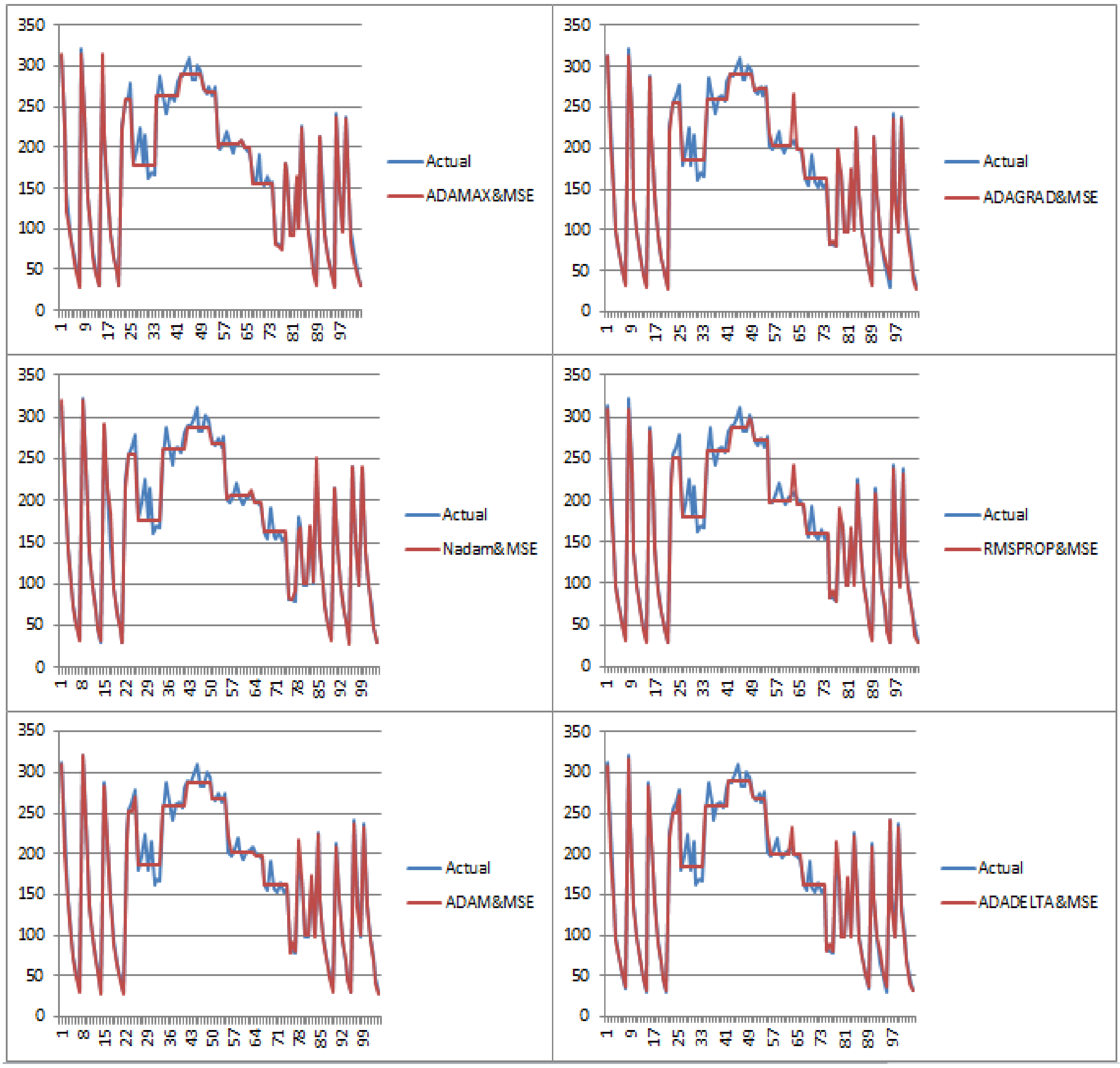

Figure 4.55. Actual values and Predicted values of all Optimizer and MSE Loss.

4.10 Comparison of All Optimizer's Results in Statistical Evaluations

When evaluated in terms of MAD, as shown in Figure 4.56, it is seen that Deep Learning is generally more successful than soft computing. According to this graph, it can be said that SGD and MAE functions which are used as optimizer for Sequential model have the most unsuccessful results. The best results are obtained from the model using ADAMAX optimizer and MAE loss.

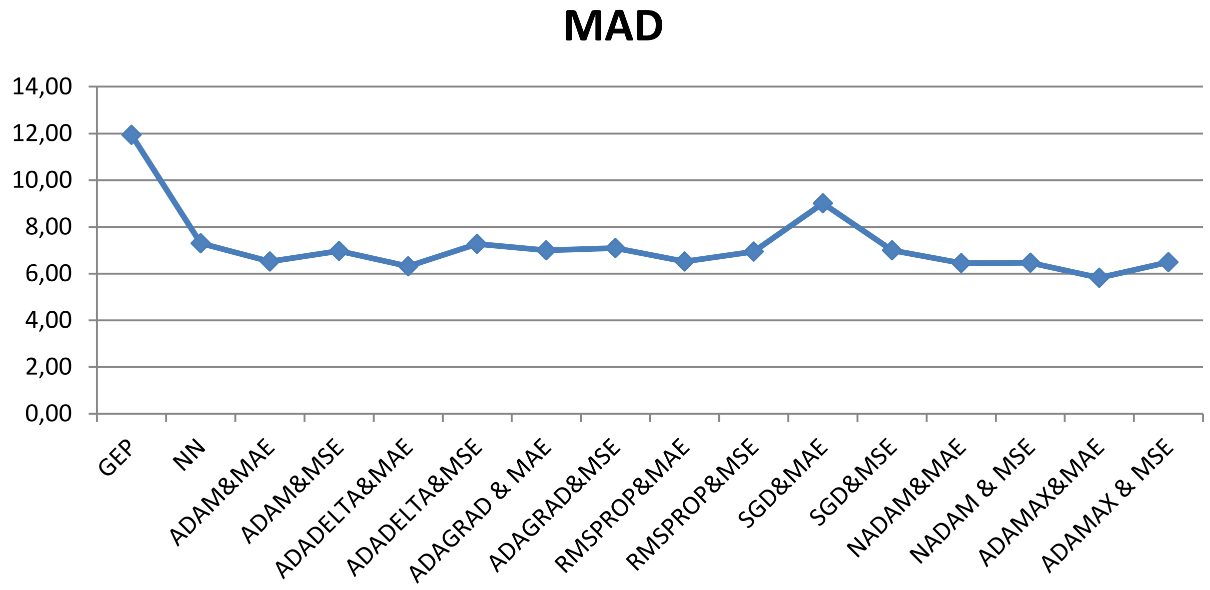

Figure 4.56. Mean Absolute Deviations of All Optimizers and Losses.

When evaluated in terms of MSE, as shown in Figure 4.57, it is seen that Deep Learning is generally more successful than soft computing. According to this graph, we can say that the SGD and MAE functions used as the Optimizer for Sequential model have the most unsuccessful results. However, it is seen that it is between the GEP model and the NN model when compared to soft computing techniques. In other words, when the weakest deep learning method, SGD optimizer and MAE loss functions are used, the model gives better results compared to GEP and is slightly weaker than NN.

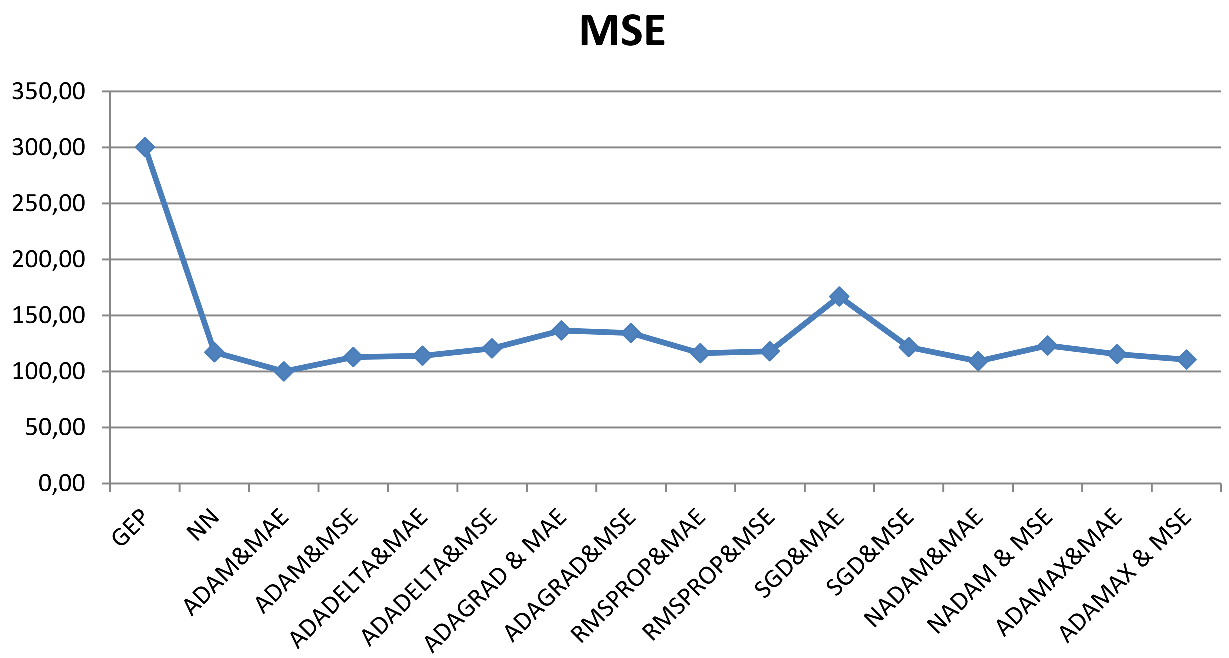

Figure 4.57. Mean Square Errors of All Optimizers and Losses.

When evaluated in terms of Standard Deviation, as shown in (Figure 4.58), it is seen that Deep Learning technique is slightly weaker than the soft computing techniques, although GEP is more successful than NN, although approximate values are obtained. According to this graph, we can say that the SGD and MAE functions used as the Optimizer for Sequential model have the most unsuccessful results. When comparing the deep learning methods, it can be said that SGD optimizer and MAE loss functions have the weakest performance, ADAMAX optimizer and MAE Loss functions have the best performance.

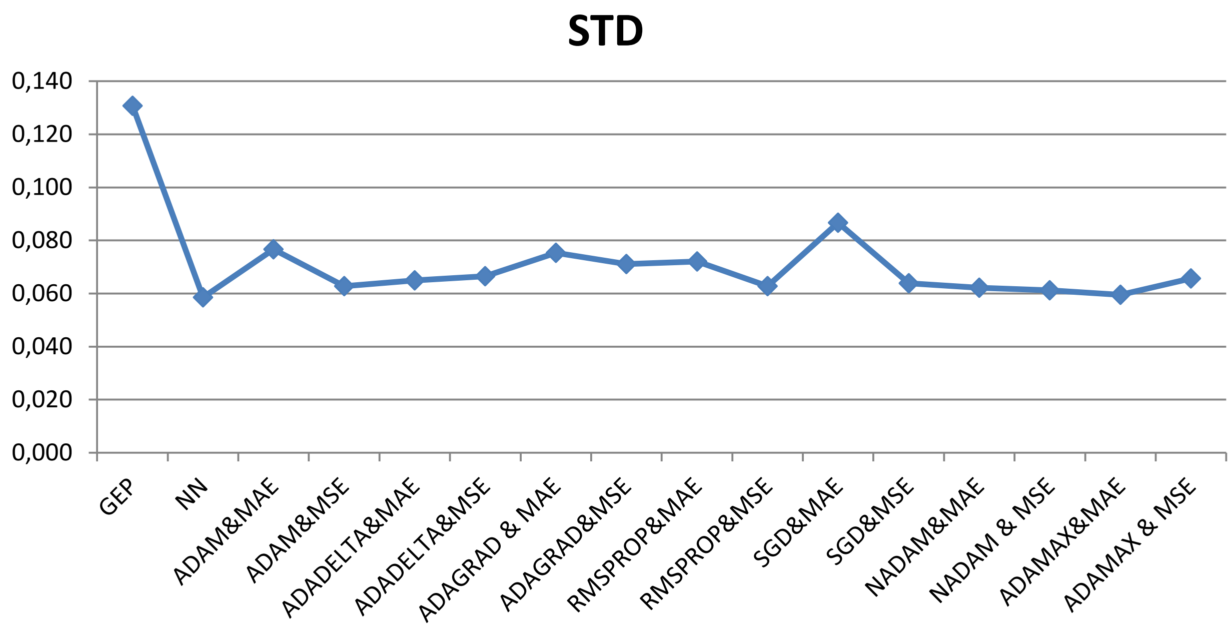

Figure 4.58. Standart Deviations of All Optimizers and Losses.

When evaluated in terms of Correlation, as shown in Figure 4.59, Deep Learning is generally more successful than soft computing. According to this graph, it can be said that the SGD and MAE functions used as the Optimizer for Sequential model have the most unsuccessful results. However, it is seen that it is between the GEP model and the NN model when compared to soft computing techniques. In other words, when the weakest deep learning method, SGD optimizer and MAE loss functions are used, the model gives better results compared to GEP and is slightly weaker than NN. The best results are obtained from the model using ADAM Optimizer and MAE Loss.

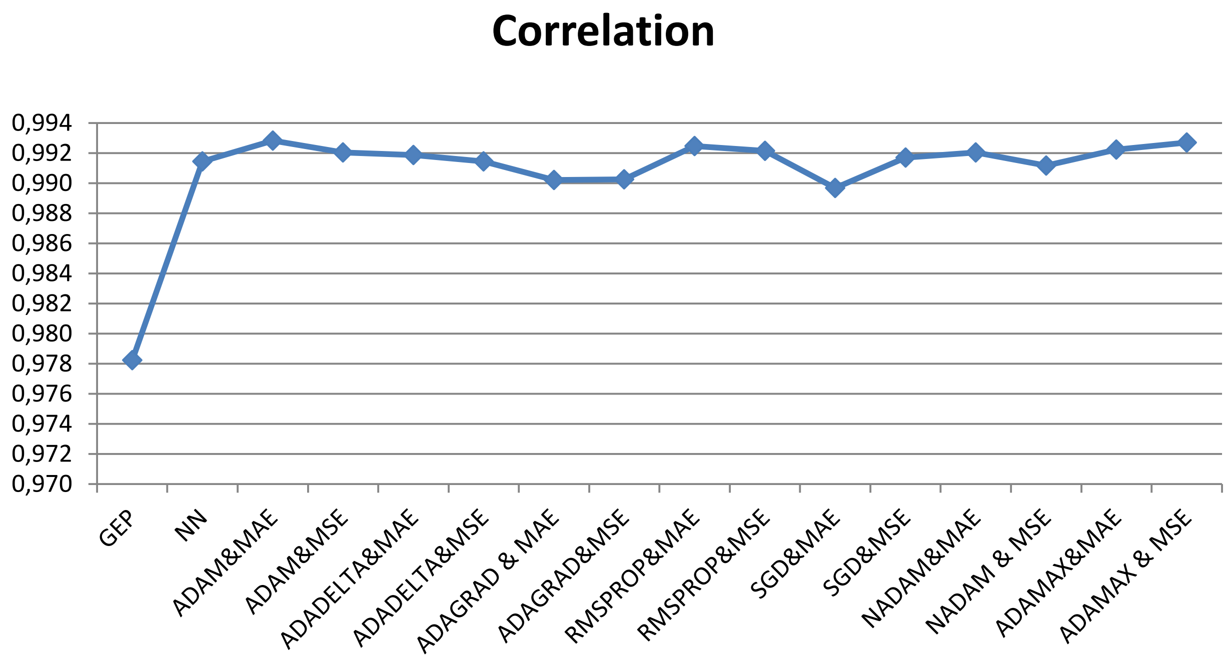

Figure 4.59. Correlations(R) of All Optimizers and Losses.

When evaluated in terms of *RMSE*, as shown in Figure 4.60, the results of the Deep Learning technique and the NN method of soft computing techniques are very similar. Despite these close values, deep learning generally resulted in fewer errors. According to this graph, we have the most unsuccessful results of the SGD and MAE functions used as Loss in the Sequential model for the dataset we have here. However, when compared to soft computing techniques, it is seen that it is better than both of GEP model and the NN model. In other words, when the weakest deep learning method, SGD optimizer and MAE loss functions are used, the model gives better results compared to GEP and is slightly stronger than NN. The best results are obtained from the model using NADAM Optimizer and MAE Loss.

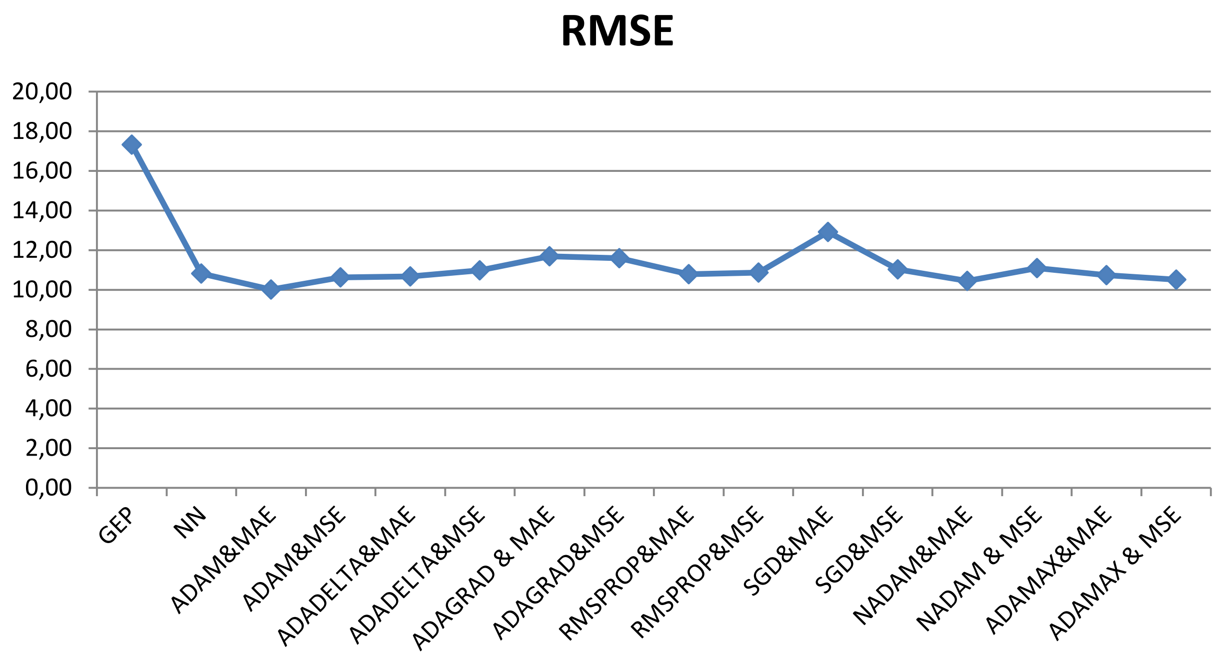

Figure 4.60. Root Mean Square Errors(RMSE) of All Optimizers and Losses.

When evaluated in terms of MAE, as shown in Figure 4.61, the results of the Deep Learning technique and the NN method from soft computing techniques are very similar. Despite these close values, deep learning generally resulted in fewer errors. According to this graph, the most unsuccessful results are the SGD and MAE functions used as Loss in the Sequential model for the dataset. However, it is seen that it is between the GEP model and the NN model when compared to soft computing techniques. In other words, when the weakest deep learning method, SGD optimizer and MAE loss functions are used, the model gives better results compared to GEP and is slightly better than NN. The best result is from the model using the ADAMAX Optimizer and MAE Loss.

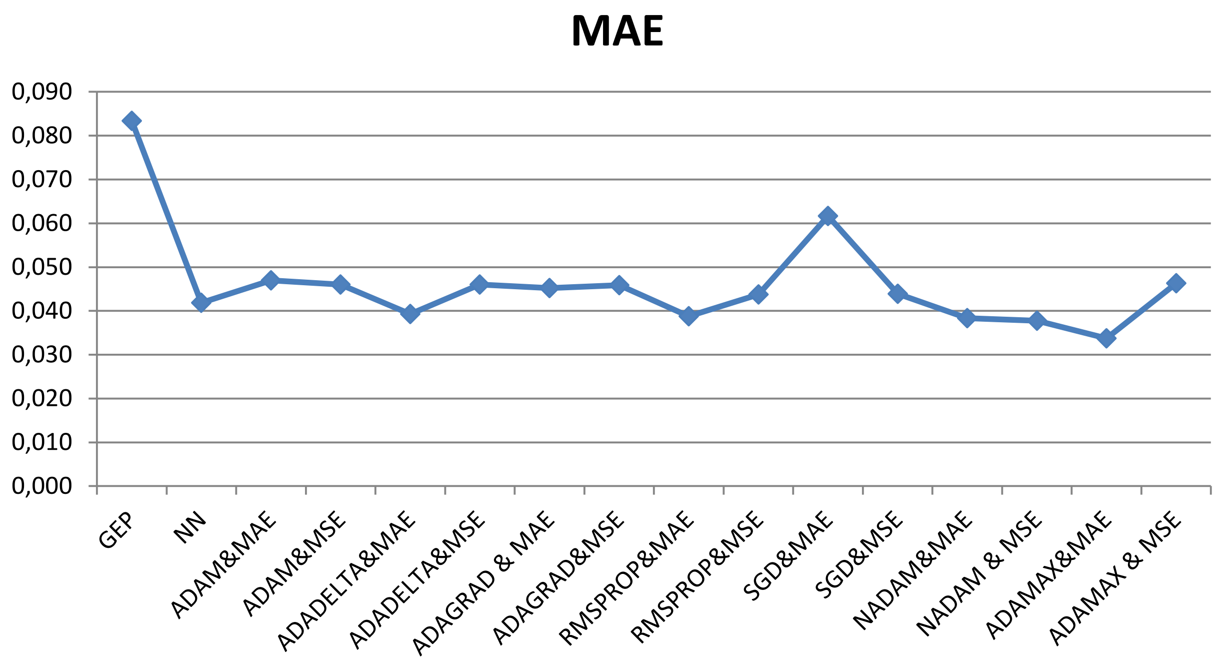

Figure 4.61. Mean Absolute Errors(MAE) of All Optimizers and Losses.

When evaluated in terms of *MAPE*, as shown in Figure 4.62, the results of the Deep Learning technique and the NN method of soft computing techniques are very similar. In general, the comparison results of the values are the same as those of the MAE.

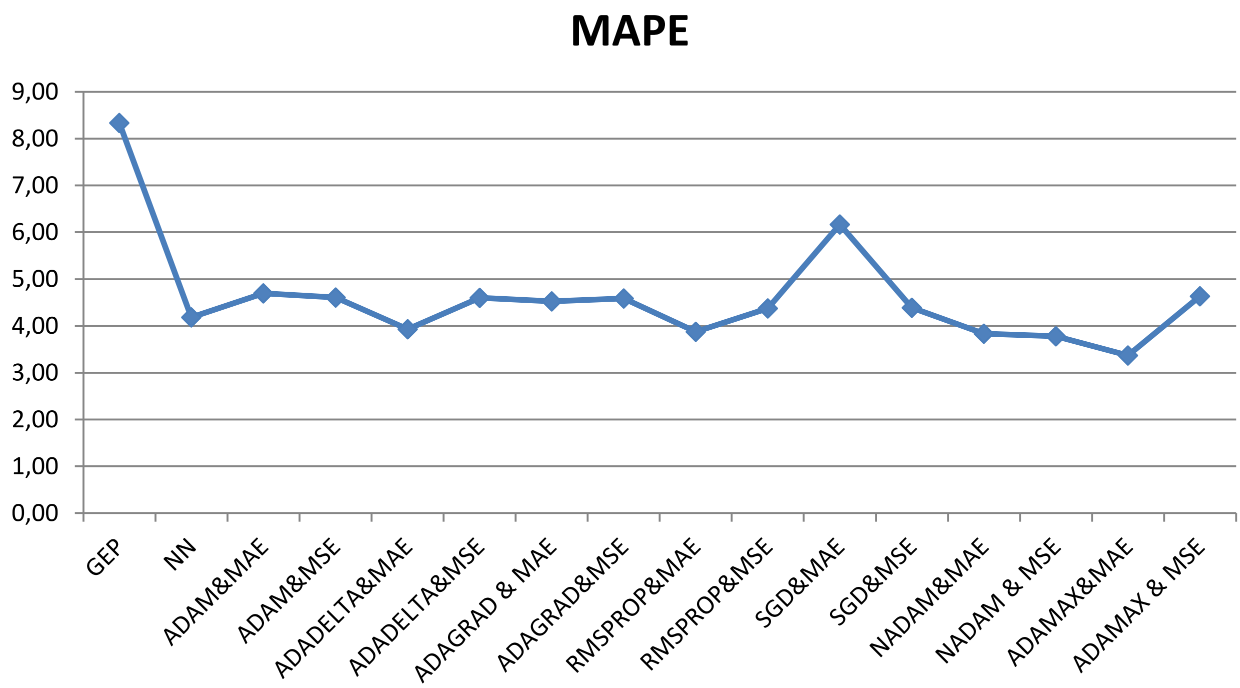

Figure 4.62. Mean Absolute Percentage Errors(MAPE) of All Optimizers and Losses.

The actual values and the estimated values are evaluated in terms of MEAN obtained from the ratio shown in Figure 4.63. According to this graph, we have the most unsuccessful results of the SGD and MAE functions used as Loss in the Sequential model for the dataset we have here. In the case of the weakest deep learning method SGD optimizer and MAE loss functions according to the data available, the model has remained weak from both soft computing techniques and other deep learning methods. Of all the best results are obtained from the model using the ADAM Optimizer and MAE Loss.

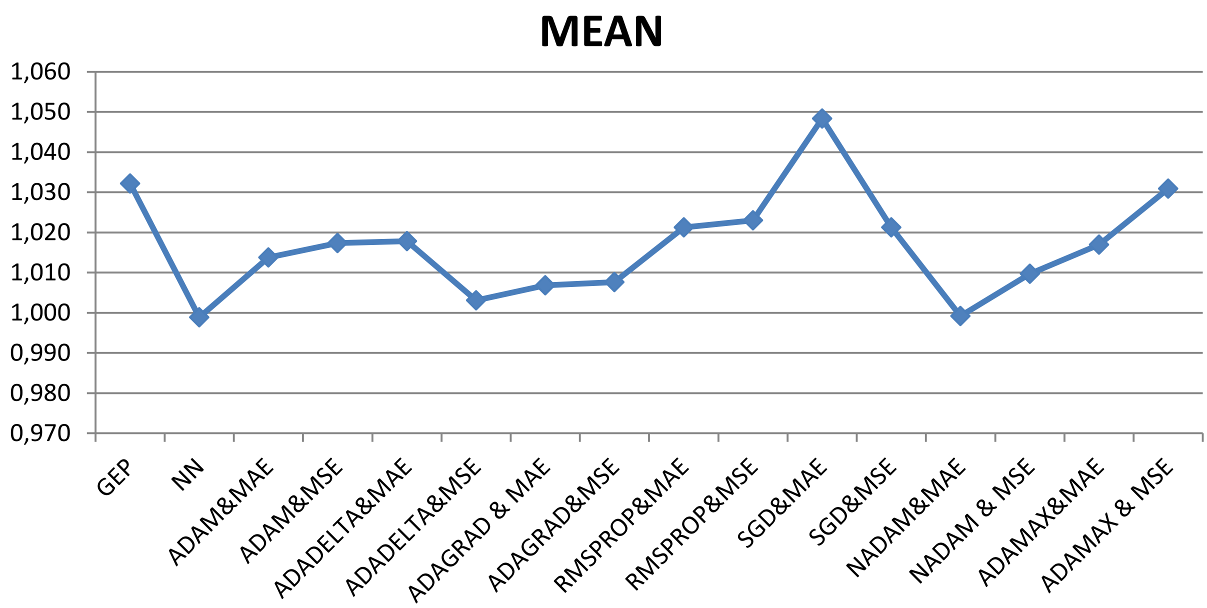

Figure 4.63. Mean of All Optimizers and Losses for Average Rate of Results.

The ratios of the each actual results and predicted results are shown in the Table 4.11.

Table 4. 11. The table of Mean(Actual/Predicted) of all Optimizers and Losses

30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**TEST**
21.5200	39.7600	63.4900	92.6600	136.6000	219.2000	322.6000	22.7000	41.7200	66.1600	95.7400	139.3000	218.5000	312.3000	**GEP**
1.4266	1.1997	1.1183	1.0490	1.0088	1.0830	0.9966	1.4890	1.2320	1.1245	1.0194	1.0323	1.0316	1.0019	**GEP/TEST**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
34.0000	49.0500	71.3800	100.0000	143.7000	231.4000	347.1000	34.2400	49.7300	72.7000	102.2000	147.0000	235.3000	297.5000	**NN**
0.9029	0.9725	0.9947	0.9720	0.9589	1.0259	0.9262	0.9871	1.0336	1.0234	0.9550	0.9782	0.9579	1.0518	**NN / TEST**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
28.5597	38.4529	67.1401	98.2641	137.3973	238.0961	317.8589	30.1635	51.6808	77.6277	96.4429	141.5111	225.1544	317.9392	**ADAM & MAE**
1.0749	1.2405	1.0575	0.9892	1.0029	0.9971	1.0115	1.1206	0.9946	0.9584	1.0120	1.0162	1.0011	0.9842	**Actual / Predicted**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
28.2584	43.1429	68.4049	99.0425	136.5735	236.0872	320.3874	29.8346	48.2754	72.0705	94.9985	139.4325	198.7884	310.4535	**ADAM & MSE**
1.0864	1.1056	1.0379	0.9814	1.0090	1.0056	1.0035	1.1329	1.0647	1.0323	1.0274	1.0313	1.1339	1.0079	**ADAM & MSE / TEST**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
28.8438	43.8207	64.6368	97.9065	138.9176	236.6602	317.5379	31.2220	50.0304	74.0299	97.2218	142.9055	204.8545	317.6058	**ADADELTA & MAE**
1.0644	1.0885	1.0984	0.9928	0.9920	1.0031	1.0125	1.0826	1.0274	1.0050	1.0039	1.0063	1.1003	0.9852	**ADADELTA & MAE / TEST**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
32.5021	46.5324	71.0743	96.7895	134.1539	232.1720	317.3362	36.3939	53.4052	73.9290	94.5760	138.2569	198.9798	307.4277	**ADADELTA & MSE**
0.9446	1.0251	0.9990	1.0042	1.0272	1.0225	1.0131	0.9287	0.9625	1.0064	1.0320	1.0401	1.1328	1.0178	**ADADELTA & MSE / TEST**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
28.8587	39.1573	66.2007	98.0816	138.5146	240.0514	314.6771	32.0462	51.2155	74.5886	100.3319	144.8950	211.1847	314.8179	**ADAGRAD & MAE**
1.0638	1.2182	1.0725	0.9910	0.9948	0.9890	1.0217	1.0547	1.0036	0.9975	0.9728	0.9924	1.0673	0.9939	**ADAGRAD & MAE / TEST**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
28.5843	44.1920	68.2117	97.1077	136.1547	236.9385	311.2856	32.4346	51.1402	73.6294	98.8269	143.4680	204.4010	311.3629	**ADAGRAD & MSE**
1.0740	1.0794	1.0409	1.0010	1.0121	1.0019	1.0328	1.0421	1.0051	1.0105	0.9876	1.0023	1.1027	1.0049	**ADAGRAD & MSE / TEST**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
32.0339	43.2498	67.2779	96.4174	136.1044	232.0196	313.4233	33.8946	53.2414	73.5067	95.3622	141.1737	206.3118	313.5049	**RMSPROP & MAE**
0.9584	1.1029	1.0553	1.0081	1.0125	1.0232	1.0258	0.9972	0.9654	1.0122	1.0235	1.0186	1.0925	0.9981	**RMSPROP & MAE / TEST**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
29.6609	43.2316	70.0921	95.3934	136.9389	232.7943	309.2395	30.2929	50.7124	72.8114	95.0368	139.2408	200.2680	309.2925	**RMSPROP & MSE**
1.0350	1.1034	1.0130	1.0189	1.0063	1.0198	1.0396	1.1158	1.0136	1.0218	1.0270	1.0327	1.1255	1.0117	**RMSPROP & MSE / TEST**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
31.6888	48.8179	70.7380	97.1738	135.3408	236.1871	312.0942	35.2548	54.8532	75.2236	97.5335	140.6541	220.7023	312.0407	**ADAMAX & MAE**
0.9688	0.9771	1.0037	1.0003	1.0182	1.0051	1.0301	0.9587	0.9370	0.9891	1.0007	1.0224	1.0213	1.0028	**ADAMAX & MAE / TEST**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
29.2339	44.1304	65.4019	96.6357	137.7304	237.8976	313.3866	28.8106	45.9135	69.3333	95.8472	121.4907	222.8074	313.3816	**ADAMAX & MSE**
1.0502	1.0809	1.0856	1.0058	1.0005	0.9979	1.0259	1.1732	1.1195	1.0731	1.0183	1.1836	1.0116	0.9985	**ADAMAX & MSE / TEST**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
29.6609	43.2316	70.0921	95.3934	136.9389	232.7943	319.2395	30.2929	50.7124	72.8114	95.0368	139.2408	220.2680	309.2925	**SGD & MSE**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
1.0350	1.1034	1.0130	1.0189	1.0063	1.0198	1.0071	1.1158	1.0136	1.0218	1.0270	1.0327	1.0233	1.0117	**SGD & MSE / TEST**
28.4622	51.9466	78.9505	102.0858	142.3948	240.0171	328.1571	33.0945	56.8609	79.4010	101.2854	148.6743	222.2754	313.1813	**SGD & MAE**
1.0786	0.9183	0.8993	0.9521	0.9677	0.9891	0.9797	1.0213	0.9040	0.9370	0.9636	0.9672	1.0141	0.9991	**SGD & MAE / TEST**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
32.5501	48.1457	73.4388	99.1934	139.2753	237.3905	306.1177	34.1803	51.7188	76.0908	99.4824	138.7031	221.3883	306.0709	**NADAM & MAE**
0.9432	0.9907	0.9668	0.9799	0.9894	1.0000	1.0502	0.9889	0.9938	0.9778	0.9811	1.0367	1.0181	1.0223	**NADAM & MAE / TEST**
30.7000	47.7000	71.0000	97.2000	137.8000	237.4000	321.5000	33.8000	51.4000	74.4000	97.6000	143.8000	225.4000	312.9000	**Actual**
31.5470	46.5659	67.3348	97.3834	139.6196	236.9210	320.2730	32.2964	48.4320	71.7889	96.5002	142.4381	220.2315	320.3855	**NADAM & MSE**
0.9731	1.0244	1.0544	0.9981	0.9870	1.0020	1.0038	1.0466	1.0613	1.0364	1.0114	1.0096	1.0235	0.9766	**NADAM & MSE / TEST**

161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
175.4000	175.4000	175.4000	175.4000	175.4000	175.4000	241.2000	241.2000	241.2000	241.2000	22.9700	41.9500	66.0500	94.7900	136.2000	208.8000	290.1000
0.9213	1.2258	1.0200	1.2822	1.1266	1.0200	1.1530	1.0916	1.0601	0.9370	1.2756	1.1132	0.9932	0.9885	1.0573	1.0230	0.9935
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
183.2000	183.2000	183.2000	183.2000	183.2000	183.2000	248.5000	248.5000	248.5000	248.5000	32.9000	46.3200	66.4100	92.4400	132.7000	213.8000	286.0000
0.8821	1.1736	0.9765	1.2276	1.0786	0.9765	1.1191	1.0596	1.0290	0.9095	0.8906	1.0082	0.9878	1.0136	1.0852	0.9991	1.0077
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
179.2912	179.2912	179.2912	200.2912	179.2912	179.2912	259.3592	259.1068	259.1068	217.2464	28.9708	48.1383	71.4407	92.2686	143.5519	215.3797	294.2734
0.9013	1.1992	0.9978	1.1229	1.1021	0.9978	1.0723	1.0162	0.9869	1.0403	1.0114	0.9701	0.9182	1.0155	1.0031	0.9917	0.9794
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
186.0886	186.0886	186.0886	186.0886	186.0886	186.0886	270.6652	253.0947	253.0947	216.2476	27.9384	43.9411	63.1078	86.8131	140.7526	212.0264	283.4883
0.8684	1.1554	0.9614	1.2086	1.0619	0.9614	1.0275	1.0403	1.0103	1.0451	1.0487	1.0628	1.0395	1.0793	1.0231	1.0074	1.0166
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
178.9167	178.9167	178.9167	178.9167	178.9167	178.9167	255.5600	255.4624	255.4624	223.6411	27.5783	45.4210	64.5003	99.7321	143.0608	218.0136	292.0424
0.9032	1.2017	0.9999	1.2570	1.1044	0.9999	1.0882	1.0307	1.0009	1.0105	1.0624	1.0282	1.0170	0.9395	1.0066	0.9798	0.9868
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
184.0639	184.0639	184.0639	184.0639	184.0639	184.0639	272.2893	251.3987	251.3987	219.4086	32.6634	48.3372	66.2049	89.2792	140.0350	215.3395	282.8964
0.8780	1.1681	0.9719	1.2219	1.0735	0.9719	1.0213	1.0473	1.0171	1.0300	0.8970	0.9661	0.9909	1.0495	1.0283	0.9919	1.0187
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
180.6325	180.6325	180.6325	180.6325	180.6325	180.6325	259.3236	258.6586	258.6586	223.9315	27.1397	46.9299	65.9950	89.0339	144.8568	220.1748	289.5772
0.8946	1.1903	0.9904	1.2451	1.0939	0.9904	1.0724	1.0179	0.9886	1.0092	1.0796	0.9951	0.9940	1.0524	0.9941	0.9701	0.9952
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
184.8612	184.8612	184.8612	184.8612	184.8612	184.8612	255.8244	254.7498	254.7498	220.5913	28.2894	45.9647	65.2765	92.3916	142.3026	216.2309	286.1995
0.8742	1.1630	0.9678	1.2166	1.0689	0.9678	1.0871	1.0336	1.0037	1.0245	1.0357	1.0160	1.0050	1.0142	1.0119	0.9878	1.0070
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
178.2033	178.2033	178.2033	178.2033	178.2033	178.2033	271.7908	249.8946	249.8946	221.4481	30.5142	48.3911	66.7587	88.9555	141.9625	219.0594	285.0725
0.9068	1.2065	1.0039	1.2620	1.1088	1.0039	1.0232	1.0536	1.0232	1.0206	0.9602	0.9651	0.9826	1.0533	1.0144	0.9751	1.0110
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
180.3040	180.3040	180.3040	180.3040	180.3040	180.3040	251.0436	250.2519	250.2519	216.5321	28.9096	46.1327	65.6360	88.0081	139.7696	212.2851	282.5416
0.8963	1.1924	0.9922	1.2473	1.0959	0.9922	1.1078	1.0521	1.0218	1.0437	1.0135	1.0123	0.9995	1.0647	1.0303	1.0062	1.0200
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
173.3206	173.3206	173.3206	173.3206	173.3206	173.3206	278.1826	257.6875	257.6875	220.4388	30.6191	46.0915	67.2191	92.9279	158.4556	216.0183	316.3096
0.9324	1.2405	1.0322	1.2976	1.1401	1.0322	0.9997	1.0218	0.9923	1.0252	0.9569	1.0132	0.9759	1.0083	0.9088	0.9888	0.9111
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
178.7355	178.7355	178.7355	178.7355	178.7355	178.7355	259.4624	258.8353	258.8353	221.0664	29.3538	44.9288	62.9595	91.9645	148.6814	217.3279	314.9626
0.9041	1.2029	1.0009	1.2583	1.1055	1.0009	1.0718	1.0172	0.9879	1.0223	0.9982	1.0394	1.0419	1.0189	0.9685	0.9828	0.9150
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
180.3040	180.3040	180.3040	180.3040	180.3040	180.3040	251.0436	250.2519	250.2519	216.5321	28.9096	46.1327	65.6360	88.0081	139.7696	212.2851	282.5416
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
0.8963	1.1924	0.9922	1.2473	1.0959	0.9922	1.1078	1.0521	1.0218	1.0437	1.0135	1.0123	0.9995	1.0647	1.0303	1.0062	1.0200
186.6862	186.6862	186.6862	186.6862	186.6862	186.6862	270.5714	270.4817	270.4817	227.0977	27.2974	49.9770	67.9567	106.4333	145.7286	227.2152	302.9268
0.8656	1.1517	0.9583	1.2047	1.0585	0.9583	1.0278	0.9734	0.9454	0.9952	1.0734	0.9344	0.9653	0.8804	0.9881	0.9401	0.9514
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
177.4376	177.4376	177.4376	177.4376	177.4376	177.4376	265.0812	265.0709	265.0709	225.0263	32.0664	44.0951	66.3407	94.2491	181.0604	221.6208	295.8348
0.9107	1.2117	1.0082	1.2675	1.1136	1.0082	1.0491	0.9933	0.9646	1.0043	0.9137	1.0591	0.9888	0.9942	0.7953	0.9638	0.9742
161.6000	215.0000	178.9000	224.9000	197.6000	178.9000	278.1000	263.3000	255.7000	226.0000	29.3000	46.7000	65.6000	93.7000	144.0000	213.6000	288.2000
175.7184	175.7184	175.7184	175.7184	175.7184	175.7184	255.8408	255.2207	255.2207	214.7573	29.7239	44.6492	64.1239	94.9000	183.6905	214.4993	291.6239
0.9197	1.2235	1.0181	1.2799	1.1245	1.0181	1.0870	1.0317	1.0019	1.0524	0.9857	1.0459	1.0230	0.9874	0.7839	0.9958	0.9883

301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
300.2000	300.2000	300.2000	300.2000	300.2000	300.2000	300.2000	243.3000	243.3000	243.3000	243.3000	243.3000	243.3000	243.3000	243.3000	175.4000	175.4000
1.0047	0.9450	0.9424	1.0330	0.9950	0.9617	0.9660	1.1587	1.0563	1.0818	1.0769	0.9947	1.0818	1.1792	1.0333	0.9493	0.9669
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
295.1000	295.1000	295.1000	295.1000	295.1000	295.1000	295.1000	258.6000	258.6000	258.6000	258.6000	258.6000	258.6000	258.6000	258.6000	183.2000	183.2000
1.0220	0.9614	0.9587	1.0508	1.0122	0.9783	0.9827	1.0901	0.9938	1.0178	1.0131	0.9358	1.0178	1.1094	0.9722	0.9088	0.9258
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
293.8148	293.8148	293.8148	293.8148	293.8148	293.8148	293.8148	262.1867	262.1867	262.1867	262.1867	262.1867	262.1867	262.1867	262.1867	179.2912	179.2912
1.0265	0.9656	0.9629	1.0554	1.0166	0.9826	0.9870	1.0752	0.9802	1.0039	0.9993	0.9230	1.0039	1.0943	0.9589	0.9287	0.9459
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
288.2112	288.2112	288.2112	288.2112	288.2112	288.2112	288.2112	259.1996	259.1996	259.1996	259.1996	259.1996	259.1996	259.1996	259.1996	186.0886	186.0886
1.0465	0.9843	0.9816	1.0759	1.0364	1.0017	1.0062	1.0876	0.9915	1.0154	1.0108	0.9336	1.0154	1.1069	0.9699	0.8947	0.9114
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
291.8408	291.8408	291.8408	291.8408	291.8408	291.8408	291.8408	260.2903	260.2903	260.2903	260.2903	260.2903	260.2903	260.2903	260.2903	178.9167	178.9167
1.0334	0.9721	0.9694	1.0626	1.0235	0.9892	0.9937	1.0830	0.9874	1.0112	1.0066	0.9297	1.0112	1.1022	0.9658	0.9306	0.9479
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
289.2788	289.2788	289.2788	289.2788	289.2788	289.2788	289.2788	259.2948	259.2948	259.2948	259.2948	259.2948	259.2948	259.2948	259.2948	184.0639	184.0639
1.0426	0.9807	0.9779	1.0720	1.0326	0.9980	1.0025	1.0872	0.9911	1.0151	1.0104	0.9333	1.0151	1.1065	0.9696	0.9046	0.9214
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
294.9634	294.9634	294.9634	294.9634	294.9634	294.9634	294.9634	262.9990	262.9990	262.9990	262.9990	262.9990	262.9990	262.9990	262.9990	180.6325	180.6325
1.0225	0.9618	0.9591	1.0513	1.0127	0.9788	0.9832	1.0719	0.9772	1.0008	0.9962	0.9202	1.0008	1.0909	0.9559	0.9218	0.9389
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
289.5777	289.5777	289.5777	289.5777	289.5777	289.5777	289.5777	259.7662	259.7662	259.7662	259.7662	259.7662	259.7662	259.7662	259.7662	184.8612	184.8612
1.0415	0.9797	0.9769	1.0709	1.0315	0.9970	1.0015	1.0852	0.9894	1.0132	1.0086	0.9316	1.0132	1.1045	0.9678	0.9007	0.9174
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
285.5289	285.5289	285.5289	285.5289	285.5289	285.5289	285.5289	255.2415	255.2415	255.2415	255.2415	255.2415	255.2415	255.2415	255.2415	178.2033	178.2033
1.0563	0.9936	0.9908	1.0861	1.0461	1.0111	1.0157	1.1044	1.0069	1.0312	1.0265	0.9481	1.0312	1.1240	0.9849	0.9343	0.9517
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
297.2431	287.9784	287.9784	287.9784	287.9784	287.9784	287.9784	259.4483	259.4483	259.4483	259.4483	259.4483	259.4483	259.4483	259.4483	180.3040	180.3040
1.0147	0.9851	0.9824	1.0768	1.0372	1.0025	1.0070	1.0865	0.9906	1.0145	1.0098	0.9327	1.0145	1.1058	0.9690	0.9234	0.9406
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
289.8833	289.8833	289.8833	289.8833	289.8833	289.8833	289.8833	258.0717	258.0717	258.0717	258.0717	258.0717	258.0717	258.0717	258.0717	173.3206	173.3206
1.0404	0.9787	0.9759	1.0697	1.0304	0.9959	1.0004	1.0923	0.9958	1.0199	1.0152	0.9377	1.0199	1.1117	0.9741	0.9606	0.9785
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
289.9661	289.9661	289.9661	289.9661	289.9661	289.9661	289.9661	263.2947	263.2947	263.2947	263.2947	263.2947	263.2947	263.2947	263.2947	178.7355	178.7355
1.0401	0.9784	0.9756	1.0694	1.0301	0.9956	1.0001	1.0707	0.9761	0.9996	0.9951	0.9191	0.9996	1.0897	0.9548	0.9315	0.9489
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
287.9784	287.9784	287.9784	287.9784	287.9784	287.9784	287.9784	259.4483	259.4483	259.4483	259.4483	259.4483	259.4483	259.4483	259.4483	180.3040	180.3040
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
1.0473	0.9851	0.9824	1.0768	1.0372	1.0025	1.0070	1.0865	0.9906	1.0145	1.0098	0.9327	1.0145	1.1058	0.9690	0.9234	0.9406
305.5955	305.5955	305.5955	305.5955	305.5955	305.5955	305.5955	277.1498	277.1498	277.1498	277.1498	277.1498	277.1498	277.1498	277.1498	186.6862	186.6862
0.9869	0.9284	0.9257	1.0147	0.9774	0.9447	0.9490	1.0171	0.9273	0.9497	0.9453	0.8732	0.9497	1.0352	0.9071	0.8919	0.9085
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
295.2260	295.2260	295.2260	295.2260	295.2260	295.2260	295.2260	266.1706	266.1706	266.1706	266.1706	266.1706	266.1706	276.7124	256.1845	177.4376	177.4376
1.0216	0.9610	0.9582	1.0504	1.0118	0.9779	0.9823	1.0591	0.9655	0.9888	0.9843	0.9092	0.9888	1.0368	0.9813	0.9384	0.9558
301.6000	283.7000	282.9000	310.1000	298.7000	288.7000	290.0000	281.9000	257.0000	263.2000	262.0000	242.0000	263.2000	286.9000	251.4000	166.5000	169.6000
286.5650	286.5650	286.5650	286.5650	286.5650	286.5650	286.5650	260.6307	260.6307	260.6307	260.6307	260.6307	260.6307	260.6307	260.6307	175.7184	175.7184
1.0525	0.9900	0.9872	1.0821	1.0423	1.0075	1.0120	1.0816	0.9861	1.0099	1.0053	0.9285	1.0099	1.1008	0.9646	0.9475	0.9652

197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
237.6000	237.6000	237.6000	202.5000	202.5000	202.5000	202.5000	202.5000	202.5000	202.5000	202.5000	269.5000	269.5000	269.5000	269.5000	269.5000	300.2000
0.8329	0.8430	0.8779	1.0035	1.0015	0.9590	1.0069	1.0859	1.0198	0.9758	0.9960	1.0230	0.9796	1.0200	0.9855	1.0071	0.9827
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
202.0000	202.0000	202.0000	203.6000	203.6000	203.6000	203.6000	203.6000	203.6000	203.6000	203.6000	275.5000	275.5000	275.5000	275.5000	275.5000	295.1000
0.9797	0.9916	1.0327	0.9980	0.9961	0.9538	1.0015	1.0801	1.0142	0.9705	0.9907	1.0007	0.9583	0.9978	0.9641	0.9851	0.9997
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
201.0236	201.0236	242.7558	203.2478	203.2478	203.2478	203.2478	203.2478	203.2478	203.2478	198.1576	278.5450	278.5450	278.5450	278.5450	274.4814	293.8148
0.9845	0.9964	0.8593	0.9998	0.9978	0.9555	1.0032	1.0819	1.0160	0.9722	1.0179	0.9898	0.9478	0.9869	0.9535	0.9888	1.0040
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
197.7094	197.7094	201.0011	201.0011	201.0011	201.0011	201.0011	201.0011	201.0011	201.0011	222.1208	269.1488	269.1488	269.1488	269.1488	268.5485	288.2112
1.0010	1.0131	1.0378	1.0109	1.0089	0.9662	1.0144	1.0940	1.0274	0.9831	0.9081	1.0243	0.9809	1.0214	0.9868	1.0106	1.0236
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
200.4200	200.4200	217.1958	203.1065	203.1065	203.1065	203.1065	203.1065	203.1065	203.1065	188.2963	272.2302	272.2302	272.2302	272.2302	270.8222	291.8408
0.9874	0.9994	0.9604	1.0005	0.9985	0.9561	1.0039	1.0827	1.0167	0.9729	1.0712	1.0127	0.9698	1.0098	0.9756	1.0021	1.0108
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
200.4771	200.4771	233.3222	200.0543	200.0543	200.0543	200.0543	200.0543	200.0543	200.0543	224.8109	268.6769	268.6769	268.6769	268.6769	269.4384	289.2788
0.9871	0.9991	0.8940	1.0157	1.0137	0.9707	1.0192	1.0992	1.0322	0.9877	0.8972	1.0261	0.9826	1.0232	0.9885	1.0073	1.0198
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
199.1887	199.1887	269.0527	204.7073	204.7073	204.7073	204.7073	204.7073	204.7073	204.7073	213.4652	277.9537	277.9537	277.9537	277.9537	274.8126	294.9634
0.9935	1.0056	0.7753	0.9926	0.9907	0.9487	0.9961	1.0742	1.0088	0.9653	0.9449	0.9919	0.9498	0.9890	0.9556	0.9876	1.0001
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
198.5349	198.5349	265.7561	201.6951	201.6951	201.6951	201.6951	201.6951	201.6951	201.6951	217.7123	273.7524	273.7524	273.7524	273.7524	271.3581	289.5777
0.9968	1.0089	0.7849	1.0075	1.0055	0.9628	1.0109	1.0903	1.0238	0.9797	0.9265	1.0071	0.9644	1.0042	0.9702	1.0002	1.0187
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
199.5149	199.5149	237.4563	203.5168	203.5168	203.5168	203.5168	203.5168	203.5168	203.5168	184.7576	268.3239	268.3239	268.3239	268.3239	266.4472	285.5289
0.9919	1.0039	0.8785	0.9984	0.9965	0.9542	1.0019	1.0805	1.0147	0.9709	1.0917	1.0275	0.9839	1.0245	0.9898	1.0186	1.0332
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
194.8485	194.8485	242.0129	198.7524	198.7524	198.7524	198.7524	198.7524	198.7524	198.7524	196.0873	272.4029	272.4029	272.4029	272.4029	272.2941	292.9143
1.0157	1.0280	0.8619	1.0224	1.0204	0.9771	1.0259	1.1064	1.0390	0.9942	1.0286	1.0121	0.9692	1.0092	0.9750	0.9967	1.0071
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
197.0442	197.0442	210.7366	203.5790	203.5790	203.5790	203.5790	203.5790	203.5790	203.5790	198.7407	264.7829	264.7829	264.7829	264.7829	272.0548	289.8833
1.0043	1.0165	0.9899	0.9981	0.9962	0.9539	1.0016	1.0802	1.0143	0.9706	1.0149	1.0412	0.9970	1.0382	1.0031	0.9976	1.0177
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
199.4527	199.4527	208.7884	205.2157	205.2157	205.2157	205.2157	205.2157	205.2157	205.2157	199.7548	267.6340	267.6340	267.6340	267.6340	273.5788	289.9661
0.9922	1.0042	0.9991	0.9902	0.9882	0.9463	0.9936	1.0716	1.0063	0.9629	1.0097	1.0301	0.9864	1.0271	0.9924	0.9920	1.0174
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
194.8485	194.8485	242.0129	198.7524	198.7524	198.7524	198.7524	198.7524	198.7524	198.7524	196.0873	272.4029	272.4029	272.4029	272.4029	272.2941	287.9784
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
1.0157	1.0280	0.8619	1.0224	1.0204	0.9771	1.0259	1.1064	1.0390	0.9942	1.0286	1.0121	0.9692	1.0092	0.9750	0.9967	1.0244
202.6262	202.6262	213.6086	212.4827	212.4827	212.4827	212.4827	212.4827	212.4827	212.4827	186.5036	287.4524	287.4524	287.4524	287.4524	281.2361	305.5955
0.9767	0.9885	0.9766	0.9563	0.9544	0.9140	0.9596	1.0349	0.9718	0.9300	1.0815	0.9591	0.9184	0.9563	0.9240	0.9650	0.9653
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
201.8349	201.8349	206.1508	206.3374	206.3374	206.3374	206.3374	206.3374	206.3374	206.3374	203.2993	270.0374	270.0374	270.0374	270.0374	278.4670	295.2260
0.9805	0.9924	1.0119	0.9848	0.9829	0.9412	0.9882	1.0657	1.0008	0.9577	0.9921	1.0210	0.9776	1.0180	0.9836	0.9746	0.9992
197.9000	200.3000	208.6000	203.2000	202.8000	194.2000	203.9000	219.9000	206.5000	197.6000	201.7000	275.7000	264.0000	274.9000	265.6000	271.4000	295.0000
196.3347	196.3347	211.9780	206.8498	206.8498	206.8498	206.8498	206.8498	206.8498	206.8498	201.0724	268.4249	268.4249	268.4249	268.4249	268.5313	286.5650
1.0080	1.0202	0.9841	0.9824	0.9804	0.9388	0.9857	1.0631	0.9983	0.9553	1.0031	1.0271	0.9835	1.0241	0.9895	1.0107	1.0294

162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
176.3000	101.9000	101.9000	179.2000	179.2000	87.4900	87.4900	82.9100	193.4000	193.4000	193.4000	193.4000	193.4000	193.4000	193.4000	193.4000	237.6000
0.9234	0.9627	0.9627	0.9470	1.0073	0.9030	0.9315	0.9914	0.8123	0.7921	0.8459	0.7927	0.8211	0.9897	0.8020	0.8459	0.8110
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
170.4000	103.3000	103.3000	176.6000	176.6000	82.8300	82.8300	85.8100	164.3000	164.3000	164.3000	164.3000	164.3000	164.3000	164.3000	164.3000	202.0000
0.9554	0.9497	0.9497	0.9609	1.0221	0.9538	0.9839	0.9579	0.9562	0.9324	0.9957	0.9330	0.9665	1.1649	0.9440	0.9957	0.9540
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
171.5516	99.5077	99.5077	170.2530	203.1490	79.9051	91.1030	84.5915	160.5835	160.5835	160.5835	160.5835	160.5835	160.5835	160.5835	160.5835	201.0236
0.9490	0.9859	0.9859	0.9968	0.8885	0.9887	0.8946	0.9717	0.9783	0.9540	1.0188	0.9546	0.9889	1.1919	0.9659	1.0188	0.9586
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
172.9290	100.4880	100.4880	169.3539	217.7828	81.1461	91.1029	78.4782	162.4608	162.4608	162.4608	162.4608	162.4608	162.4608	162.4608	162.4608	197.7094
0.9414	0.9762	0.9762	1.0020	0.8288	0.9736	0.8946	1.0474	0.9670	0.9430	1.0070	0.9436	0.9775	1.1781	0.9547	1.0070	0.9747
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
168.9006	98.7012	98.7012	168.7313	221.8060	79.4311	84.4033	81.6652	159.4332	159.4332	159.4332	159.4332	159.4332	159.4332	159.4332	159.4332	200.4200
0.9639	0.9939	0.9939	1.0057	0.8138	0.9946	0.9656	1.0065	0.9854	0.9609	1.0261	0.9615	0.9960	1.2005	0.9728	1.0261	0.9615
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
169.9662	98.3658	98.3658	166.3429	214.7385	80.6633	89.4756	80.9543	161.4408	161.4408	161.4408	161.4408	161.4408	161.4408	161.4408	161.4408	200.4771
0.9578	0.9973	0.9973	1.0202	0.8406	0.9794	0.9109	1.0154	0.9731	0.9490	1.0134	0.9496	0.9836	1.1856	0.9607	1.0134	0.9612
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
177.5190	98.8063	98.8063	170.3630	199.4571	79.8570	86.9545	82.8580	160.6072	160.6072	160.6072	160.6072	160.6072	160.6072	160.6072	160.6072	199.1887
0.9171	0.9929	0.9929	0.9961	0.9050	0.9893	0.9373	0.9921	0.9782	0.9539	1.0186	0.9545	0.9887	1.1917	0.9657	1.0186	0.9674
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
173.8657	98.0008	98.0008	167.9153	198.3138	79.4357	85.8347	81.7975	162.8799	162.8799	162.8799	162.8799	162.8799	162.8799	162.8799	162.8799	198.5349
0.9364	1.0010	1.0010	1.0106	0.9102	0.9945	0.9495	1.0049	0.9645	0.9406	1.0044	0.9412	0.9750	1.1751	0.9522	1.0044	0.9706
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
169.4630	99.3664	99.3664	168.0119	182.5452	80.6735	93.8170	83.0462	156.6812	156.6812	156.6812	156.6812	156.6812	156.6812	156.6812	156.6812	199.5149
0.9607	0.9873	0.9873	1.0100	0.9888	0.9793	0.8687	0.9898	1.0027	0.9778	1.0442	0.9784	1.0135	1.2216	0.9899	1.0442	0.9658
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
166.2868	97.6692	97.6692	163.8120	190.6082	78.4434	91.9995	82.2114	161.1371	161.1371	161.1371	161.1371	161.1371	161.1371	161.1371	161.1371	194.8485
0.9790	1.0044	1.0044	1.0359	0.9470	1.0071	0.8859	0.9999	0.9749	0.9507	1.0153	0.9514	0.9855	1.1878	0.9625	1.0153	0.9890
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
162.8073	96.8674	96.8674	165.7864	181.9173	79.1326	80.8577	81.3012	153.4435	153.4435	153.4435	153.4435	153.4435	153.4435	153.4435	153.4435	197.0442
1.0000	1.0127	1.0127	1.0236	0.9922	0.9983	1.0079	1.0111	1.0238	0.9984	1.0662	0.9991	1.0349	1.2474	1.0108	1.0662	0.9780
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
164.2208	92.6504	92.6504	166.2290	179.6750	74.2911	78.8623	80.9874	156.7569	156.7569	156.7569	156.7569	156.7569	156.7569	156.7569	156.7569	199.4527
0.9913	1.0588	1.0588	1.0209	1.0046	1.0634	1.0334	1.0150	1.0022	0.9773	1.0437	0.9779	1.0130	1.2210	0.9894	1.0437	0.9661
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
166.2868	97.6692	97.6692	163.8120	212.6082	78.4434	91.9995	82.2114	161.1371	161.1371	161.1371	161.1371	161.1371	161.1371	161.1371	161.1371	194.8485
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
0.9790	1.0044	1.0044	1.0359	0.8490	1.0071	0.8859	0.9999	0.9749	0.9507	1.0153	0.9514	0.9855	1.1878	0.9625	1.0153	0.9890
179.1404	104.6568	104.6568	177.2732	185.3857	82.7429	79.6768	84.8922	162.3173	162.3173	162.3173	162.3173	162.3173	183.1236	152.3173	165.3001	192.6262
0.9088	0.9373	0.9373	0.9573	0.9736	0.9548	1.0229	0.9683	0.9679	0.9438	1.0079	0.9444	0.9783	1.0452	1.0183	0.9897	1.0004
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
168.5784	87.9600	87.9600	170.9027	183.2766	93.6217	84.7476	84.4907	159.6410	159.6410	159.6410	159.6410	159.6410	159.6410	159.6410	159.6410	201.8349
0.9657	1.1153	1.1153	0.9930	0.9849	0.8438	0.9617	0.9729	0.9841	0.9597	1.0248	0.9603	0.9947	1.1989	0.9716	1.0248	0.9547
162.8000	98.1000	98.1000	169.7000	180.5000	79.0000	81.5000	82.2000	157.1000	153.2000	163.6000	153.3000	158.8000	191.4000	155.1000	163.6000	192.7000
169.8492	99.9906	99.9906	167.9310	162.0479	91.7932	81.6887	82.9406	162.1416	162.1416	162.1416	162.1416	162.1416	162.1416	162.1416	162.1416	196.3347
0.9585	0.9811	0.9811	1.0105	1.1139	0.8606	0.9977	0.9911	0.9689	0.9449	1.0090	0.9455	0.9794	1.1804	0.9566	1.0090	0.9815

237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
220.7000	89.6900	132.6000	214.1000	23.8500	43.4800	68.1800	97.4700	139.3000	211.2000	23.7600	43.5600	68.8000	99.2100	143.6000	222.8000	106.0000
1.0757	1.0982	1.1237	1.1261	1.2243	1.0741	0.9622	0.9613	1.0330	1.0114	1.4226	1.1800	1.0828	0.9838	1.0042	1.0126	0.9528
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
218.0000	97.5300	140.0000	225.6000	33.3800	46.7100	66.5100	92.3800	133.3000	217.9000	34.2400	48.9100	71.1300	99.7400	143.7000	230.2000	99.7100
1.0890	1.0099	1.0643	1.0687	0.8748	0.9998	0.9863	1.0143	1.0795	0.9803	0.9871	1.0509	1.0474	0.9785	1.0035	0.9800	1.0129
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
234.7537	99.2293	127.1129	246.3070	27.3414	38.5916	69.2292	94.1166	146.5489	210.2473	29.4636	48.9674	77.1248	95.7441	141.5304	226.3673	100.1310
1.0113	0.9927	1.1722	0.9789	1.0680	1.2101	0.9476	0.9956	0.9819	1.0159	1.1472	1.0497	0.9660	1.0194	1.0189	0.9966	1.0087
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
232.9618	99.6497	132.7963	236.3666	30.6014	46.0252	70.5231	94.0510	143.4592	209.1819	29.7351	47.3406	71.0943	95.8750	143.4927	223.7847	99.2613
1.0191	0.9885	1.1220	1.0200	0.9542	1.0147	0.9302	0.9963	1.0031	1.0211	1.1367	1.0857	1.0479	1.0180	1.0049	1.0081	1.0175
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
236.6606	99.5987	139.4109	240.2612	29.8038	40.0655	66.9705	91.5151	142.1746	211.1335	30.9846	49.0190	73.4436	96.6445	142.9845	224.1142	101.6181
1.0031	0.9890	1.0688	1.0035	0.9797	1.1656	0.9795	1.0239	1.0121	1.0117	1.0909	1.0486	1.0144	1.0099	1.0085	1.0066	0.9939
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
231.5644	97.8769	130.2123	240.9703	35.3946	52.3829	78.3615	97.6284	145.7903	209.1882	35.6090	51.9507	72.6495	94.4244	141.2200	220.8177	98.1761
1.0252	1.0064	1.1443	1.0005	0.8250	0.8915	0.8371	0.9598	0.9870	1.0211	0.9492	0.9894	1.0255	1.0336	1.0211	1.0217	1.0288
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
239.4946	99.2886	132.7819	239.9528	34.4478	53.8371	71.9067	94.6315	144.6758	216.5759	32.0118	50.8111	74.4034	98.0532	144.0697	229.0572	101.4012
0.9913	0.9921	1.1221	1.0048	0.8477	0.8674	0.9123	0.9902	0.9946	0.9863	1.0559	1.0116	1.0013	0.9954	1.0009	0.9849	0.9960
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
234.5599	97.2931	135.7597	236.0316	40.9781	56.4127	72.0038	92.0414	140.6551	212.6534	32.1916	50.4561	73.3679	96.9730	143.1655	223.9674	100.1806
1.0121	1.0124	1.0975	1.0215	0.7126	0.8278	0.9111	1.0180	1.0231	1.0045	1.0500	1.0187	1.0154	1.0065	1.0072	1.0073	1.0082
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
232.5604	98.2941	140.4165	241.0371	29.6406	35.4476	63.9628	95.5334	143.1614	210.6026	33.1707	50.3187	74.2849	95.8736	143.1868	222.6723	101.7612
1.0208	1.0021	1.0611	1.0003	0.9851	1.3174	1.0256	0.9808	1.0052	1.0142	1.0190	1.0215	1.0029	1.0180	1.0071	1.0131	0.9925
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
231.4953	96.3141	134.9305	236.9026	29.7098	41.0706	70.8825	94.1270	143.6509	208.2529	30.1917	49.2414	71.8102	94.8008	139.6432	218.1318	98.3971
1.0255	1.0227	1.1043	1.0177	0.9828	1.1371	0.9255	0.9955	1.0017	1.0257	1.1195	1.0438	1.0375	1.0295	1.0326	1.0342	1.0265
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
235.1194	98.5185	146.7070	243.3249	29.5576	46.0846	65.9219	95.2055	137.7448	213.2330	33.8078	51.8838	73.3009	96.7646	121.8431	223.1638	100.3787
1.0097	0.9998	1.0156	0.9909	0.9879	1.0134	0.9951	0.9842	1.0447	1.0017	0.9998	0.9907	1.0164	1.0086	1.1835	1.0109	1.0062
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
235.1152	95.5790	147.5918	236.7507	28.3812	44.9453	64.2370	92.8456	121.3873	213.4072	29.0800	45.2667	66.1688	94.8462	135.3523	224.1520	99.7184
1.0097	1.0306	1.0095	1.0184	1.0288	1.0390	1.0212	1.0092	1.1855	1.0009	1.1623	1.1355	1.1259	1.0290	1.0654	1.0065	1.0129
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
231.4953	96.3141	134.9305	236.9026	29.7098	41.0706	70.8825	94.1270	143.6509	208.2529	30.1917	49.2414	71.8102	94.8008	139.6432	218.1318	98.3971
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
1.0255	1.0227	1.1043	1.0177	0.9828	1.1371	0.9255	0.9955	1.0017	1.0257	1.1195	1.0438	1.0375	1.0295	1.0326	1.0342	1.0265
239.9789	103.4233	140.9155	230.4370	25.1531	44.2444	76.1723	97.7421	139.3574	226.7010	32.9878	56.8639	79.3645	102.2051	148.2354	233.8132	105.0700
0.9893	0.9524	1.0574	1.0463	1.1609	1.0555	0.8612	0.9586	1.0326	0.9422	1.0246	0.9039	0.9387	0.9549	0.9728	0.9649	0.9613
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
239.3339	100.3293	151.6189	236.0163	29.6282	46.0979	64.9898	97.0609	160.8652	213.7846	33.3061	51.1009	74.6033	99.3479	125.5618	224.9164	102.1295
0.9919	0.9818	0.9827	1.0215	0.9855	1.0131	1.0094	0.9654	0.8945	0.9991	1.0148	1.0059	0.9986	0.9824	1.1484	1.0030	0.9889
237.4000	98.5000	149.0000	241.1000	29.2000	46.7000	65.6000	93.7000	143.9000	213.6000	33.8000	51.4000	74.5000	97.6000	144.2000	225.6000	101.0000
239.8544	98.4260	150.5120	241.0564	28.7380	44.6561	66.1938	94.6325	142.1343	215.3229	32.1797	48.6613	71.9787	97.0909	144.6293	251.8297	101.9990
0.9898	1.0008	0.9900	1.0002	1.0161	1.0458	0.9910	0.9901	1.0124	0.9920	1.0504	1.0563	1.0350	1.0052	0.9970	0.8958	0.9902

30.8000	47.7000	70.9000	97.2000	137.8000
21.7100	40.1200	64.0200	93.4300	137.6000
1.4187	1.1889	1.1075	1.0404	1.0015
30.8000	47.7000	70.9000	97.2000	137.8000
33.4700	47.2600	67.4600	93.5300	134.1000
0.9202	1.0093	1.0510	1.0392	1.0276
30.8000	47.7000	70.9000	97.2000	137.8000
28.1936	33.8223	62.8972	92.1087	134.1122
1.0924	1.4103	1.1272	1.0553	1.0275
30.8000	47.7000	70.9000	97.2000	137.8000
28.6339	40.5657	66.7292	94.0599	135.4382
1.0756	1.1759	1.0625	1.0334	1.0174
30.8000	47.7000	70.9000	97.2000	137.8000
29.1225	38.7167	60.1726	91.3527	144.8150
1.0576	1.2320	1.1783	1.0640	0.9516
30.8000	47.7000	70.9000	97.2000	137.8000
32.0642	41.0405	66.5563	91.7830	128.1210
0.9606	1.1623	1.0653	1.0590	1.0755
30.8000	47.7000	70.9000	97.2000	137.8000
28.1590	36.9766	58.8501	88.3363	127.0542
1.0938	1.2900	1.2048	1.1003	1.0846
30.8000	47.7000	70.9000	97.2000	137.8000
28.0548	40.0281	62.7817	87.5884	124.8984
1.0978	1.1917	1.1293	1.1097	1.1033
30.8000	47.7000	70.9000	97.2000	137.8000
31.0763	34.8532	67.0006	96.0668	135.6222
0.9911	1.3686	1.0582	1.0118	1.0161
30.8000	47.7000	70.9000	97.2000	137.8000
29.5113	37.8138	71.4676	93.6957	131.4894
1.0437	1.2614	0.9921	1.0374	1.0480
30.8000	47.7000	70.9000	97.2000	137.8000
30.6724	46.9156	63.7825	85.5730	136.9219
1.0042	1.0167	1.1116	1.1359	1.0064
30.8000	47.7000	70.9000	97.2000	137.8000
29.0823	44.3722	59.7575	82.9879	137.2847
1.0591	1.0750	1.1865	1.1713	1.0038
30.8000	47.7000	70.9000	97.2000	137.8000
29.5113	37.8138	71.4676	93.6957	131.4894
30.8000	47.7000	70.9000	97.2000	137.8000
1.0437	1.2614	0.9921	1.0374	1.0480
26.7025	43.7464	65.4216	103.0273	130.9200
1.1535	1.0904	1.0837	0.9434	1.0526
30.8000	47.7000	70.9000	97.2000	137.8000
30.6489	46.8406	71.7721	87.8921	140.4430
1.0049	1.0183	0.9878	1.1059	0.9812
30.8000	47.7000	70.9000	97.2000	137.8000
30.5161	46.0997	62.8363	98.3892	139.7042
1.0093	1.0347	1.1283	0.9879	0.9864

CHAPTER 5
CONCLUSION

The aim of this study is to determine whether deep learning methods can be used to help the buckling load estimation of heat treated aluminum alloy columns and to analyze which method yields more successful results. Sequential is preferred as the model when applying deep learning method. Adam, Adamax, Adadelta, AdaGrad, Nadam, SGD, RMSProp optimizers are used. MAE and MSE Losses are used for each optimizer. Epoch number is kept constant as 4096 and results of all variations are considered and these results are evaluated in many respects. These evaluation methods are *RMSE, MSE, MAE, MSE, MAD, STD, MEAN, and Correlation*.

When evaluation methods *MAD, MAE, MAPE, and STD* are considered; the best results are obtained from the model using ADAMAX optimizer and MAE loss functions. Evaluation results are obtained as *MAD*: 5.8172, *MAE*: 0.0337, *MAPE*: 3.3701, STD: 0.0595 respectively.

When analyzed according to *MSE, RMSE, correlation* methods; the best results are obtained from ADAM optimizer and MAE Loss functions. Evaluation results are obtained as *MSE*: 100.0853, *RMSE*: 10.0042, *correlation*: 0.9928 respectively. When the *MEAN* results are analyzed, it is seen that the result of NN method (result: 0.9988) which is one of the soft computing techniques is closer to 1. The most successful MEAN result of the deep learning methods is ADADELTA optimizer and MSE Loss functions (result: 1.0030).

When the dataset is analyzed; it is seen that the places where the errors are most intense, are the lines where different results are obtained although the same inputs are used in the test results. In these rows, Deep learning Model makes a classification; consequently, the results of the same inputs are the same. Estimates are inaccurate because they differ in laboratory results.

In general, deep learning methods have been successful in prediction of the buckling load of aluminum alloy columns. This method is expected to be more successful if larger dataset is used to guide future studies. According to the results of the study, it is recommended to use MAE as a loss function and ADAM or ADAMAX as an optimizer in estimation processes. The SGD optimizer has the weakest results so that, it is not recommended.

REFERENCES

Abdulkader (2019), *Predicting The Maximal Oxygen Uptake Using Deep Learning*, **1**, Turkey

Anwer (2017), *Breast Cancer Dıagnosıs Wıth Deep Learnıng Methods,* **10**, Turkey

Arik, A., Gölcük, M., & Karslıgil, E. M. (2017). *Deep learning based skin cancer diagnosis. In Signal Processing and Communications Applications Conference,* Turkey

Büber, E., Sahingoz, O. K. (2017). *Image Processing and Adjustment of Optimal Parameters with the Machine-Care System* IDAP

J, Maljaar., L,Twilt., F, Soeten. (2009). *Flexural buckling of fire exposed aluminum columns,* Fire Safety Journal **44**(5), pp 711-717.

Cevik, A., Ekmekyapar, T., Atmaca, N., Guzelbey, İ. (2009). *Flexural buckling load prediction of aluminum alloy columns using soft computing techniques, Expert Systems with Applications*, **36**, pp 6332-6342.

Çakır (2019), *Zero-Day Attack Detectıon Wıth Deep Learnıng,* **5**, Turkey

Eriş (2018), *Evaluatıon Of Evıdence In Forensıc Informatıon Studıes Usıng Deep Learnıng Methods,* **4**, Turkey

Işık, G., Artuner, H. (2016). *Recognition of radio signals with deep learning Neural Networks. In Signal Processing and Communication Application Conference*

Kantepe (2019), *Deep Learnıng Methods Usıng Product Proposal System Desıgn,* **5,** Turkey

Kaynar, O., Görmez, Y., Işık, Y. E. (2016). *Spam Detection with Auto Coder Based Deep Learning Machines,*

Koyun, A., Afşin, E. (2017). *Two Dimensional Optical Character Recognition with Deep Learning*. Journal of Turkey Informatics Foundation of Computer Science and Engineering,

Kutlu (2019), *Obtaıned Wıth Unmanned Aırcraft Deep Of Imaged Images Analysıs Wıth Learnıng Methods,* **4**, Turkey

Munir (2019), *Handwrıtten Text Recognıtıon Usıng Deep Learnıng,* **7**, Turkey

Özgenel (2018), *Crack Detectıon Wıth Deep Learnıng: An Exemplary Study Of Data Desıgn In Archıtecture,* **5**, Turkey

Paker (2019), *Deep Learnıng For Bıologıcal Sequences,***1,** Turkey

Pala (2019), *A Comparatıve Study of Deep Learnıng Based Face Recognıtıon Algorıthms For Vıdeo Under Adverse Condıtıons*, **5**, Turkey

Rasheed (2019), *Object Detectıon From Images Usıng Deep Learnıng,* **2**, Turkey

Razavi, S., Yalcin, H. (2017). *Using convolutionalneural networks for plant classification. In Signal Processing and Communications Applications Conference (SIU)*, 2017 25th (pp. 1-4). IEEE.

Sorkun (2018), *Tıme Serıes Forecastıng On Solar Radıatıon Usıng Deep Learnıng,* **2**, Turkey

Soydaş (2019), *Aırcraft Detectıon From Large Scale Remote Sensıng Images Wıth Deep Learnıng Technıques,* **23,** Turkey

Süberk (2019), "*Deep Learnıng Technıques For Buıldıng Densıty Estımatıon From Remotely Sensed Imagery", **2**, Turkey

Şeker, A., Yüksek, A. G. (2017). "*Stacked Auto-encoder Method for Fabric Fault Detection*." Cumhuriyet University Faculty of Arts and Sciences Journal of Science,**342-354**.

Tanimu (2019), "*Gpr Raw-Data Analysıs To Detect Crack Vıa Wavelets And Deep Learnıng Methods",* **3**, Turkey

Tienin (2019), "*Cloud Coveragepredıctıonwıthdeep Learnıng",* **1**, Turkey

Uçar, A., Bingöl, M. S. (2018). "*Evaluating graphics processing cards using deep learning"*, Mühendislik Dergisi 9(1), **39-49**.

Üner (2019), Deepside: *"Predicting Drug Side Effects With Deep Learning"*, **1,** Turkey

Yalçın, H. (2016). "*Human activity recognition using deep belief networks. In Signal Processing and Communication Application Conference (SIU)",* Turkey

Yetiş (2019), "*Auto-Conversıon From 2d Drawıng to 3d Model Wıth Deep Learnıng",* **5**, Turkey

Wang et. al(2017) . "*Compressive buckling strength of extruded aluminum alloy I-section columns with fixed-pinned end conditions"*, China

Yetkin, A. E., Hamamcı, A. (2016). *Data Augmentation for Head Pose Estimation From MRI*

https://keras.io/why-use-keras/#keras-prioritizes-developer-experience

Wald, Abraham (June 1945). *"Sequential Tests of Statistical Hypotheses". The Annals of Mathematical Statistics. **16**..*

Berger, James (2008). Sequential Analysis. *The New Palgrave Dictionary of Economics, 2nd Ed. **438–439.***

Kenneth J. Arrow, David Blackwell and M.A. Girshick (1949). *"Bayes and minimax solutions of sequential decision problems". Econometrica. **17***

Randell, Brian *(1980), "The Colossus", A History of Computing in the Twentieth Century, **30***

Turnbull, Bruce (2000).*Group sequential methods with applications to clinical trials. Chapman & Hall.*

O'Brien, Peter C. Fleming, Thomas R. (1979). *"A Multiple Testing Procedure for Clinical Trials". Biometrics. **35***

Wang, Samuel K.; Tsiatis, Anastasios A. (1987). *"Approximately Optimal One-Parameter Boundaries for Group Sequential Trials". Biometrics. **43***

*Demets, David L.; Lan, K. K. Gordon (1994). "Interim analysis: The alpha spending function approach". Statistics in Medicine. **13** (13–14): 1341–1352.*

Diederik Kingma, Jimmy Ba (2014), *On the Convergence of Adam and Beyond*

Håkan Gadd (2006). *Evaluation: definitions, methods and models, ITPS, Swedish Institute For Growth Policy Studies Studentplan 3, SE-831 40 Östersund, Sweden*

J. Andreas Graefe; *Scott Armstrong; Randall J. Jones, Jr.; Alfred G. Cuzán (2010).* "Combining forecasts for predicting U.S. Presidential Election outcomes"*Surface.* Turkey