MACHINE LEARNING OF ACCELEROGRAM DATA FOR ANALYSES, MODELING AND PREDICTION

MELİS ÇIKIŞ

SEPTEMBER 2022

ÇANKAYA UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

MASTER'S THESIS IN

ELECTRICAL AND ELECTRONICS ENGINEERING

MACHINE LEARNING OF ACCELEROGRAM DATA FOR ANALYSES, MODELING AND PREDICTION

MELİS ÇIKIŞ

SEPTEMBER 2022

ABSTRACT

MACHINE LEARNING OF ACCELEROGRAM DATA FOR ANALYSES, MODELING AND PREDICTION

ÇIKIŞ, Melis

Master of Science in Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Erdem AKAGÜNDÜZ

Co-Supervisor: Asst. Prof. Dr. Salih TİLEYLİOĞLU

September 2022, 89 pages

In this thesis, the earthquake epicenter coordinate prediction is provided by processing the accelerometer records recorded from the strong motion station using convolutional networks. Spectrogram-based false color representation of earthquake accelerometer records is proposed and its application in convolutional networks is discussed. Using more than forty-two thousand publicly available earthquake records, an epicenter cluster has been made with hundreds of thousands of 5-second false color spectrograms, and earthquakes in similar clusters were observed to produce similar impressions. With this epicenter clustering, the convolutional network is trained by using different records from different years. Using this trained network, it is aimed to predict the epicenter and depth information of any earthquake event. As a result of the trainings, it has been observed that the spectrograms created with single station accelerogram data can be used in convolutional networks and accelerograms have potential to detect epicenters.

Keywords: Earthquake Accelerograms, Epicenter Clustering, Convolutional Neural Networks, Spectrogram

ÖZ

ANALİZLER, MODELLEME VE TAHMİN İÇİN AKSELEROGRAM VERİLERİNİN MAKİNE ÖĞRENİMİ

ÇIKIŞ, Melis
Elektrik-Elektronik Mühendisliği Yüksek Lisans

Danışman: Doç. Erdem AKAGÜNDÜZ
Ortak Danışman Dr. Öğretim Üyesi Salih TİLEYLİOĞLU
Eylül 2022, 89 sayfa

Bu tezde, kuvvetli yer hareketi istasyonundan kaydedilmiş ivmeölçer kayıtlarının evrişimsel ağlarda işlenmesi ile merkez üssü konumu tahmini sağlanmıştır. Deprem ivmeölçer kayıtlarının spektrogram tabanlı sahte renk gösterimi önerilmiş ve bu gösterimin evrişimsel ağlarda uygulanması tartışılmıştır. Kamuya açık kırk iki binden fazla deprem kaydı kullanılarak, 5 saniyelik yüzbinlerce sahte renk spektrogramı ile bir merkez üssü kümelemesi yapılmış, ve benzer kümelerdeki depremlerin benzer gösterimler yarattığı gözlenmiştir. Elde edilen bu merkez üssü kümeleme ile farklı yıllara ait farklı kayıtlar kullanılarak evrişimsel ağ eğitilmiştir. Eğitilen bu ağ ile herhangi bir deprem olayına ait, merkez üssü ve derinlik bilgilerini tahmin etmek amaçlanmıştır. Eğitimler sonucunda, tek istasyondan kaydedilen ivmeölçer verileri ile yaratılan spektrogramların evrişimsel ağlarda kullanılabildiği ve ivmeölçer verilerinin merkez üssü tespit etmede potansiyeli olduğu gözlemlenmiştir.

Anahtar Kelimeler: Deprem İvmeölçer Kayıtlar, Merkez Üssü Kümeleme, Evrişimsel Ağlar, Spektrogram

ACKNOWLEDGEMENT

I would like to thank my advisor, Assoc. Prof. Dr. Erdem AKAGÜNDÜZ, for his continuous support and guidance in this study. He showed me the right way and stood by me every time I faced difficulties. Secondly, I would like to express my gratitude to my co-advisor Asst. Prof. Dr. Salih TİLEYLİOĞLU for helping me to explain the necessary concepts in the field of seismology. I have enjoyed various meetings and discussions with them. My special gratitude also goes to the rest of the thesis committee. Last but not the least, I would like to thank my friends and family for always being there for me, for their motivation and support.

The work carried out in this thesis is supported by TÜBİTAK as a part of the ongoing TÜBİTAK 1001 funded project, Project No.121M732, titled "Deep Learning and Machine Learning Based Dynamic Soil and Earthquake Parameter Estimation Using Strong Ground Motion Station Records".

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

SYMBOLS

km	: Kilometer
m	: Meter

ABBREVIATIONS

ABC	: Artificial Bee Colony
ACO	: Ant Colony Optimization
AFAD	: Disaster and Emergency Management Presidency
ANN	: Artificial Neural Networks
ANFIS	: Adaptive-Network Based Fuzzy Inference Systems
BDTIM	: Bölgesel Deprem-Tsunami İzleme ve Değerlendirme Merkezi
BOUN	: Boğaziçi Üniversitesi
CNN	: Convolutional Neural Networks
EEW	: Earthquake Early Warning
DFT	: Discrete Fourier Transform
FC	: Fully Connected
GA	: Genetic Algorithm
GLM	: Generalized Linear Model
GP	: Genetic Programming
KOERI	: Kandilli Rasathanesi ve Deprem Araştırma Enstitüsü
LPBoost	: Linear Programming Boosting
LR	: Learning Rate
LSTM	: Long Short-Term Memory
PNN	: Probabilistic Neural Network
PRNN	: Pattern Recognition Neural Network

PSO	: Particle Swarm Optimization
QPSO	: Quadratic Particle Swarm Optimization
RBF	: Radial Basis Function
ReLU	: Rectified Linear Unit
RMSE	: Root Mean Squared Error
RNN	: Recurrent Neural Networks
SGA	: Simple Genetic Algorithm
SGDM	: Stochastic Gradient Descent Momentum
STFT	: Short-Time Fourier Transform

CHAPTER I

INTRODUCTION

Like other natural disasters, earthquakes create a great danger to the built environment and, therefore, to people. Earthquakes occur due to sudden movement along faults, and the point where the fault first ruptures is called the focus or the hypocenter of the earthquake. Earthquake waves begin to radiate from hypocenter. This radiating is shown in Figure 1.

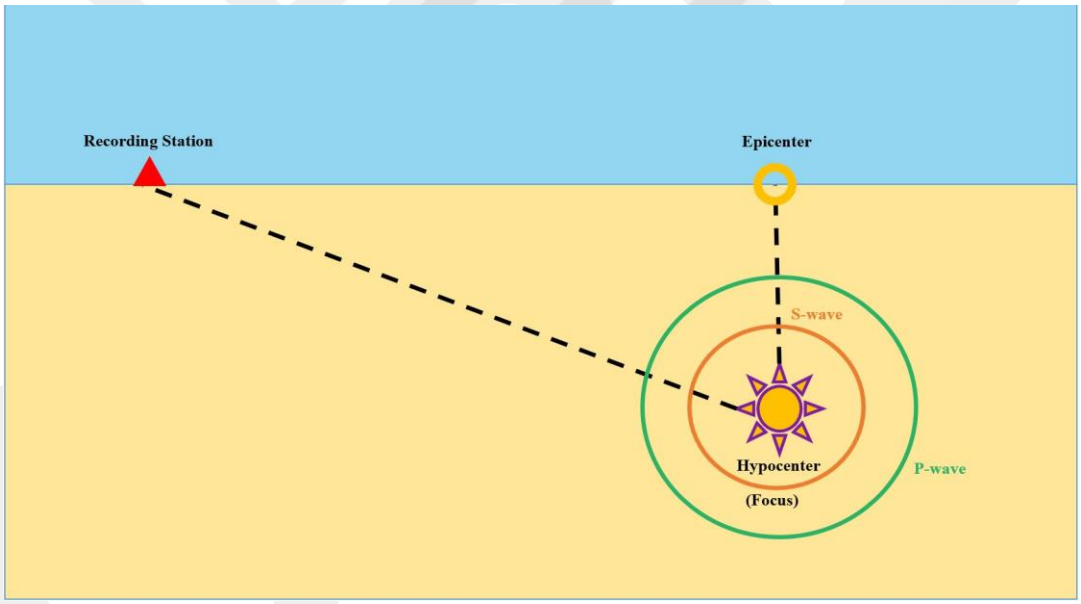


Figure 1: Wave propagation from a seismic source.

The focus is below the earth's surface, and its projection on the earth's surface is called the epicenter of the earthquake. Strong ground motions caused by earthquakes are measured by accelerographs at a station utilizing three perpendicular accelerometer devices.

Seismic waves can be broadly classified as body waves and surface waves. Body waves arrive the quickest to the surface and are of two types: Primary (P) and Secondary (S) waves. Primary (P) waves are the fastest moving earthquake waves and the first to reach the seismograph. Secondary (S) waves are the second fastest moving seismic waves. These waves are the second wave after the P waves to reach a seismograph from a distant earthquake and have a high destructive effect. Because S-waves come within the noise of P-waves, it is not always easy to accurately determine their arrival time. However, it is possible to observe such waves on seismograms recorded with three-component seismometers. The arrival times of the P and S waves contain important information such as the onset of the earthquake and the propagation of the earthquake signal [1]. An example seismic signal showing P and S waves is given in Figure 2.

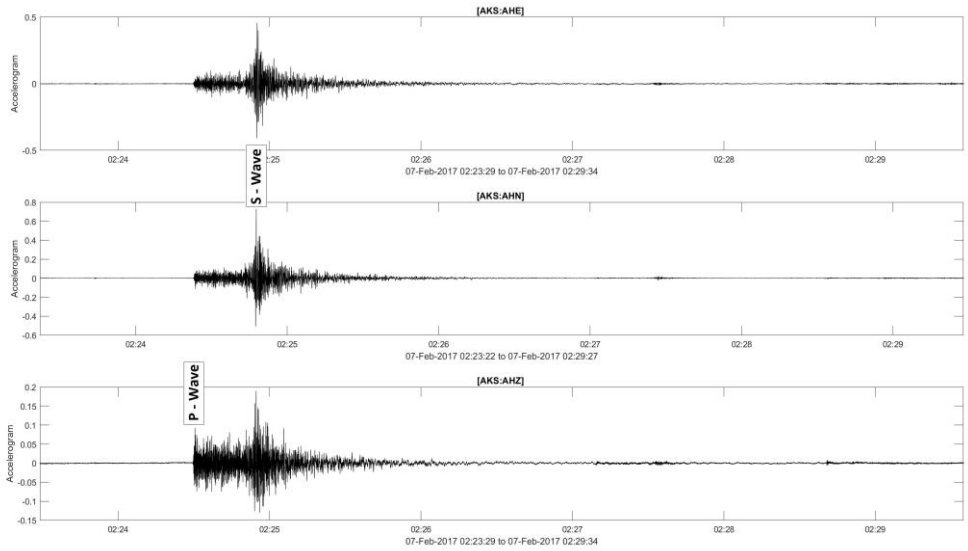


Figure 2: Example seismic P and S wave.

Earthquakes that may cause loss of life and property or in other words that affect people and their surroundings are termed as strong ground motion. Strong ground motions are measured with accelerographs (accelerometers) and are considered to have magnitudes greater than 3.5 [2]. The measured recordings are called the

accelerogram [3]. Earthquake magnitude is obtained as a result of calculations using the maximum amplitude and period value of earthquake motion recorded with a standard seismograph and instrument calibration functions [4]. It is defined as a measure of the energy released during an earthquake.

These data, which are recorded in different channels from the beginning of the earthquake to the end, are stored as earthquake accelerometer records. Earthquake records contain valuable parameters such as the peak ground acceleration of the earthquake at the recorded location, frequency content, and duration of the shaking. Earthquake stations and records also play a significant role in determining the epicenter of earthquakes. Velocity data obtained from at least three stations and measuring weak ground movements are generally used to determine the epicenter of earthquakes. In general, the basic parameters of earthquake (magnitude, epicenter coordinate etc.) are calculated depending on the amplitude and phase values in the velocity records, while acceleration records are mostly used in the preparation of intensity maps. In this thesis, it is proposed to calculate the parameters related to the earthquake using accelerometer records [5].

1.1 PROBLEM DEFINITION

Traditionally, in calculating the epicenter location of the earthquake, first, the distance from the seismic phases from the seismograms measuring velocity in at least three stations from the place where the time differences coincide best is determined. According to these distances, circles with radius equal to the distances determined for each station are drawn on a map that includes the stations used. An example illustration is given in Figure 1.1.

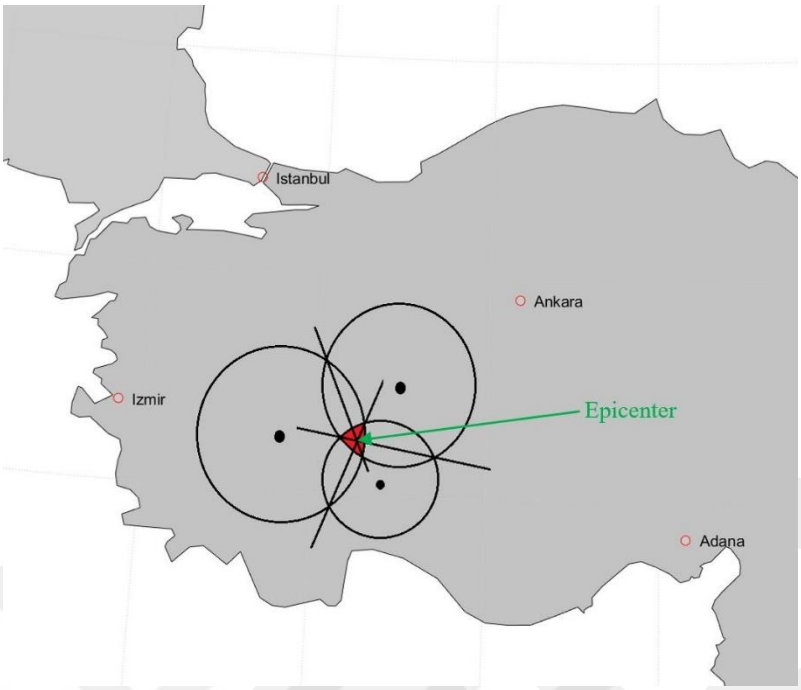


Figure 1.1: A process called triangulation uses seismic data from three locations to identify the epicenter of an earthquake.

The drawn circles intersect each other at two points and the earthquake center remains within the shaded area in Figure 1.1. In order to find the earthquake epicenter in this area, the intersection points of the intersecting circles are combined and the intersection point of these drawn lines gives the earthquake epicenter location [6]. As the number of stations used in this calculation increases, the amount of error in determining the epicenter location decreases. The concept of effective range cannot be mentioned for seismic devices. The stations used in this study can record records the larger the earthquake, the further away it is, so the range varies according to the magnitude of the earthquake.

In the event of a strong motion earthquake, a velocity sensor which is close to the fault cannot record due to its nature [5]. The accelerometer can also receive data. In this thesis, it is investigated to contribute to Earthquake Early Warning (EEW) systems by determining epicenter coordinate with accelerometer data. Accelerometer records are time series recorded mathematically from an inertial sensor [7]. The methods used to visualize time series such as sound and vibration and make them suitable for human examination may differ depending on the nature of the sensor.

Various studies have been carried out on these data using machine learning methods [8-11].

In this thesis, the accelerometer records of earthquake events recorded in different years in Turkey, which we clustered according to the epicenter, were converted into spectrogram-based three-channel false-color representation and made suitable for human analysis and the use of convolutional neural networks (CNN). In this way, RGB has officially become compatible with standard CNN structures with the display of earthquakes, and the applicability of convolutional network analysis methods used in computer vision to earthquake data has been ensured. As a result, it is aimed to predict the epicenter with the proposed spectrograms.

1.2 LITERATURE SURVEY

It is known that since the formation of the world, ground movements caused by earthquakes have occurred sequentially in seismically active regions [12]. When understanding seismic signals were essential and computational power was low, earthquake detection was done manually by examining waveforms.

Turkey is located in one of the most active earthquake zones in the world. There exists high risk of loss of lives and property in future earthquakes, as Turkey has had many destructive earthquakes in the past. Many researchers have studied different algorithms based on artificial intelligence techniques to foresee and predict future earthquake. The developed algorithms are meant to identify the low-magnitude signals that are often difficult to detect, even for humans, as they can be confused with noise.

In this section, a series of publications dealing with various earthquake prediction tasks with artificial intelligence techniques are reviewed. Artificial intelligence techniques encountered in the field of seismology are called genetic algorithms, artificial neural networks, and machine learning techniques.

1.2.1 Artificial Intelligence Optimization Techniques

Optimization is choosing the best among the possible alternatives under certain conditions in a problem [13]. Many algorithms have been proposed for optimization problems. When recent years are examined, artificial intelligence optimization

techniques such as Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) [14] have been encountered in studies with earthquake data.

In a study published in the World Journal of Modeling and Simulation in 2012, the Particle Swarm Optimization (PSO) algorithm was preferred to find the local earthquake location, in other words, the epicenter location where the earthquake occurred [15]. PSO is based on the movement of animals in search of food. In the algorithm, some particles are a swarm and each individual of the swarm. Particles adjust their position as the best position in the swarm, drawing on previous experience. Other particles update their movements according to the particle currently in the swarm's best place. This process is repeated until the goal is reached [16]. The proposed study uses longitude, latitude, and depth-based position vectors. It was concluded that with the Quadratic and standard PSO models, which are mathematically different from each other, PSO could calculate the epicenter location based on the inverse problem. Moreover, when comparing the two methods, it has been observed that it takes less time for QPSO to converge with better accuracy.

In a similar study carried out in the same years, Ant Colony Optimization (ACO) algorithm was preferred for earthquake prediction. ACO, a probability-based approach, provides the least computational overhead to find the optimum path [17]. In this study, the preferred ACO was compared with the results of the k-means clustering algorithm. Accordingly, it was observed that the ACO algorithm is a more effective method for future earthquake prediction studies [18].

Genetic Algorithm (GA), which has been applied to seismic data in the past to solve NP-hard problems, develops a permutation-based optimization method. It has been described in the literature as a powerful evolutionary strategy inspired by the fundamental principles of the biological universe [19]. After defining the variable type and the problem, the fitness to be optimized is defined. While solving the problem, GA is preferred if the number of parameters affecting the problem is high since it does not local but global research. On seismic data, GA is selected because it can manage uncertain and incomplete information. Today, many simple GA (SGA) variations are used in different experiments [20-22].

1.2.2 Artificial Neural Networks

Artificial Neural Networks (ANN) is the mathematical modeling of the human brain's learning process. ANN realizes the biological structure of neural networks in the brain, such as learning, creating new information, remembering, and discovering [23]. In recent years, different ANN algorithms have been used for various seismological tasks such as classifying events such as earthquakes or noise, phase selection [24], or time estimation. The methods used to accomplish these tasks range from machine learning algorithms [25] to Convolutional Neural Networks (CNN) [24] or Recurrent Neural Networks (RNN) [26].

Asim et al. implemented the earthquake prediction problem for different magnitudes with different machine learning algorithms [27-29]. In their first study (2017) [27], the problem was considered a binary classification in the Hindukush region. By creating a combination of Pattern Recognition Neural Network (PRNN) [30], RNN, Random Forest [31], and Linear Programming Boosting (LPBoost) [32] algorithms, earthquakes with earthquake magnitudes greater than 5.5 were predicted with seismicity indicators.

In another study [28], in which Pakistan's earthquake data were used, the combination of GP [33] and AdaBoost [34] algorithms were used while predicting earthquakes magnitude greater than 5. Before this article, this combination had not been used in earthquake prediction problems before. Another innovation of the approach is the simultaneous use and calculation method of seismicity indicators, based on the idea of gaining maximum information about the geological features of the observed regions (rather than selecting parameters separately for each area). Another study [29], it was aimed to predict earthquakes with a magnitude of at least 5, within the next 15 days, with a combination of different machine learning algorithms. In each step, the algorithm uses the information obtained by learning the previous one.

ANN models have also been used to predict the seismic response of structures and to understand damage conditions. In a study [35], ANN model designed to detect seismic damage of building structures was first trained with undamaged data, then trained with data from the damaged state, and became the reference to measure

structural damage with the change in the level of estimation error between the two states.

As the intensity of the earthquake increases, the damage to people and the environment increases. Most studies have predicted earthquakes with earthquake magnitudes above a particular value (threshold). These studies vary the basis of the architectures which they use. For example, which was considered a classification problem in [36], seismicity indicators were used as input values and fed into the Recursive Neural Network [37]. The proposed methods are used to predict the magnitude of the largest earthquake (within 0.5) in a predefined region in the next month. Again by Panakkat and Adeli [38], this problem was realized with Probabilistic Neural Network (PNN) [39], and it was concluded that RNN is effective in predicting large earthquakes and PNN is more effective in small and moderate earthquakes. The eight seismicity indicator sets proposed by Panakkat and Adeli have been used in various studies by researchers from all over the world.

In some studies, the magnitude range was determined, and the PNN algorithm was used to predict earthquakes in that magnitude range [40]. A probability model for future earthquakes is constructed using a PNN approach. Probability is calculated for next 5 days for the earthquakes that happen below the threshold value of the magnitude and larger earthquakes that happen more than 5 day time in fixed interval [41]. Similarly, the ANN structure is used to calculate this in Chile region. Another study using artificial neural networks [42] aimed to predict the start times of earthquakes with earthquake magnitudes greater than 6.

In earthquake magnitude estimation studies [27, 43, 44], which are seen as a binary classification problem, different inputs and architectures/structures are used. In the approaches, we can observe that the input values are mostly seismicity indicators. There are also studies where the image (256x256) which the former using 12 physics-based features, and each geographic cell of a seismicity map is used as the input value in the predictions made according to the earthquake magnitude. Larger earthquake magnitude can cause thousands of death and economic loss. In the study [43], which is proposed to prevent losses and treated as a binary problem, if the magnitude of the

largest earthquake in the next 30 days is six or greater, it is marked with "1". Likewise, negative samples are marked with "0".

Another area of study is prediction of aftershocks after an earthquake. Aftershock prediction is also considered a binary classification problem [44]. This classification was made by determining the probability of containing or not aftershocks. Formulated as a large-scale binary classification problem to correctly classify every 5 km x 5 km x 5 km grid cell in the volume around each main shock, using 75% of the 130,000 data as training and 25% as testing.

A fault detection study was conducted with 256x256 images solved with U-Net. Here, fault detection is performed pixel by pixel [45] with the semantic segmentation problem [46]. Earthquake prediction, which has not been handled only as a classification problem, has been processed with the Generalized Linear Model (GLM) algorithm [47] as a regression problem in order to predict the magnitudes of earthquake events in the next seven days using seismicity indicators [48]. Similar to this thesis, the Supervised Radial Basis Function (RBF) [49] network and the ANFIS [50] model, made by Zamani et al. [51] in 2012, were applied because they showed effectiveness in classification and prediction problems. The results of this study showed that RBF neural networks and ANFIS models could be suitable tools for the accurate prediction of the epicentral area as well as for the time of occurrence of strong earthquakes in active seismogenic places.

There are two different output values in the study with RNN in 2009 [52] to predict the location and time of medium and large-scale future earthquakes and seismic indicators are used as inputs. The first is 1 or 0, indicating that an earthquake of a certain threshold magnitude has occurred or not, and the second is a prediction of the epicenter location and time of occurrence. In the case of epicenter location prediction, the prediction error is calculated as the geographic distance between the recorded location and the predicted location of earthquakes in the test dataset. The best result was obtained by using recurrent neural network yielded which an error 63 km in distance.

As stated earlier, earthquakes occur due to the energy released by the sudden movement along faults. As the faults move, energy is released in the form of heat and

seismic. The seismic waves are divided into two: body and surface waves. P (primary) and S (secondary) waves, which are body waves, are waves that occur in the earth's crust. P waves are the first waves to reach the earthquake stations. S waves arrive later, but they are the waves that create a destructive effect (mainshock) [53]. The need for accurate earthquake locations to distinguish between different competing physical processes in the post-seismic period required even more accurate estimates of the arrival time of the seismic phase. In order to increase the accuracy of the earthquake location, seismic phase arrival time are marked a part of the study. Also, as the number of data increased, manual methods to find P and S waves became insufficient due to the required time and resource investment. To overcome this, Woollam et al. [54] carried out a study in 2019 in which they predicted P and S waves and noise with convolutional neural network [55] architecture. As a result of this study, the P wave improved by 0.23 seconds and the S wave by 0.336 seconds compared to manual methods. In another study carried out for the prediction of the arrival time of S waves [56], 3-channel accelerometer data was used as the input value. Here, a methodology is presented with Long Short-Term Memory (LSTM) [57] to classify earthquake vibrations from near source or far source within one second of P wave detection. In another study, the CNN + RNN structure was used, this time using 3-channel accelerometer data from the dataset in which P and S waves were labeled. In this study, the detection problem is formulated as a sequence learning where inputs from a time series are mapped to a time series probability outputs, and separate predicts are made for each sample [58].

A different study was carried out to determine whether the recorded from smartphone accelerometer signals belong to humans or earthquakes. This study considers recognizing human activities and earthquake vibration a time series forecasting problem. The smartphone accelerometer records the acceleration caused by vibrations in a time series format. The data consists of acceleration value and timestamp in three axes (X, Y, and Z). Each set of acceleration values is labeled with the vibration source, e.g., standing, walking, running, upstairs, and downstairs. These labeled examples are then used as training input for the RNN - LSTM network [59].

1.3 RESEARCH QUESTIONS

In this thesis, it is aimed to answer the following scientific research questions:

- Is a spectrogram-based false color representation suitable for earthquake signal processing in convolutional networks?
- Can earthquake epicenter coordinate be found using accelerometer data like velocity data?
- Is it possible to find earthquake epicenter locations using accelerogram data recorded from single strong motion station?

CHAPTER II

DATA COLLECTION

2.1 ACCELEROGRAM RECORDS

When an earthquake occurs, seismic waves radiating from the epicenter produce vibrations on the ground, bedrock, or soft ground layers at various periods. This period varies according to the earthquake's magnitude, location, distance, and site properties [60]. It measures the movements in three perpendicular directions, two horizontally and one vertically, utilizing a three-axis accelerometer at an earthquake station. These movements are recorded from the beginning to the end of the earthquake. The most basic method to study strong ground motion is to analyze accelerometer records.

The earthquake accelerometer data used in this study which was recorded in Turkey can be accessed publicly on the website of the T.C. Disaster, and Emergency Management Presidency Ministry of Internal Affairs (AFAD) [61], Earthquake Department, and the server shared on the Boğaziçi University (BOUN) Kandilli Observatory and Earthquake Research Institute (KOERI) Regional Earthquake-Tsunami Monitoring and Evaluation Center (BDTIM) website [62]. In these records, it is possible to access all necessary information about the occurrence of an earthquake, including the location, time, epicenter coordinate, depth, magnitude, station number recording the earthquake, the coordinate of that station, and acceleration records recorded in all directions.

In Table 2.1, the records of a sample earthquake obtained from the AFAD website and the information about the station are given. In each record, the size of the sampled three-channel accelerometer data differs depending on the time between the start and end moments of the respective earthquake event.

Table 2.1: Example Earthquake Record from AFAD Dataset

Place	MUGLA MILAS 2 DERINCE DAM		
Earthquake Date	2017/08/14 07:15:07		
Epicenter Coordinates	37.11760N-27.70510E		
Earthquake Depth (km)	6.97		
Earthquake Magnitude	3.1		
Station ID	4822		
Station Coordinates	37.44170N-27.64600E		
Station Altitude (m)	128		
Record Type	GeoSig gmsplus		
Record Serial No	101468		
Record Time	14/08/2017 07:15:04.000000		
Number of Data	8490		
Sampling Interval (sec)	0.01		
RAW PGA VALUES (gal)	(N-S) 0.514297 (E-W) 0.931399 (U-D) 1.274957		
Accelerogram Records	(N-S) 0.001559 …	(E-W) 0.001819 …	(U-D) -0.008576 …

Figure 2.1 shows the seismic wave state of the accelerometer records of a sample earthquake obtained from AFAD dataset. Seismic waves radiate in all directions away from the source in different types due to the breaking and faulting that creates the earthquake. Body waves, one of the waves released during an earthquake, spread from the source to all directions and move within the ground. Body waves, are divided into two Pressure (or Primary) and Shear (or Secondary) waves, depending on their speed of movement on the ground, their movement waveform and the order in which they appear [63]. The section gives more detailed information and analysis of the dataset used.

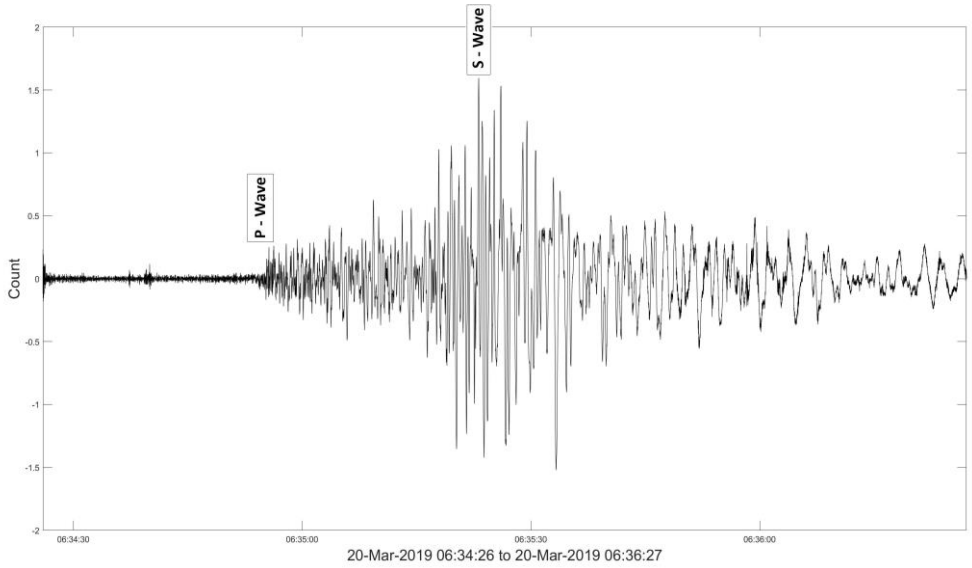


Figure 2.1: Arrival times of P and S waves on the east-west component and seismic wave representation of sample accelerometer data from the three-component recording made at the RBG earthquake recording station of AFAD of the aftershock of the M=5.5 magnitude Manisa earthquake on 20 March 2019. As can be seen from the figure, the P wave is the first to arrive at the surface during an earthquake.

2.2 DATASET

2.2.1 AFAD Records

In the dataset published by AFAD open to the public, there are 42858 text-formatted records between 1980 and 2018. One of the aims of this thesis is to propose a spectrogram-based earthquake signal representation suitable for use in convolutional networks. This proposed representation is planned to be used in the estimation of the epicenter with accelerometer records obtained from a single station in the continuation of this thesis. Therefore, earthquakes with close epicenters were clustered using the K-medoids clustering method [64]. The clustered form of the station and epicenter locations of these records by the K-medoids method is shown on the maps in the Figure 2.2.

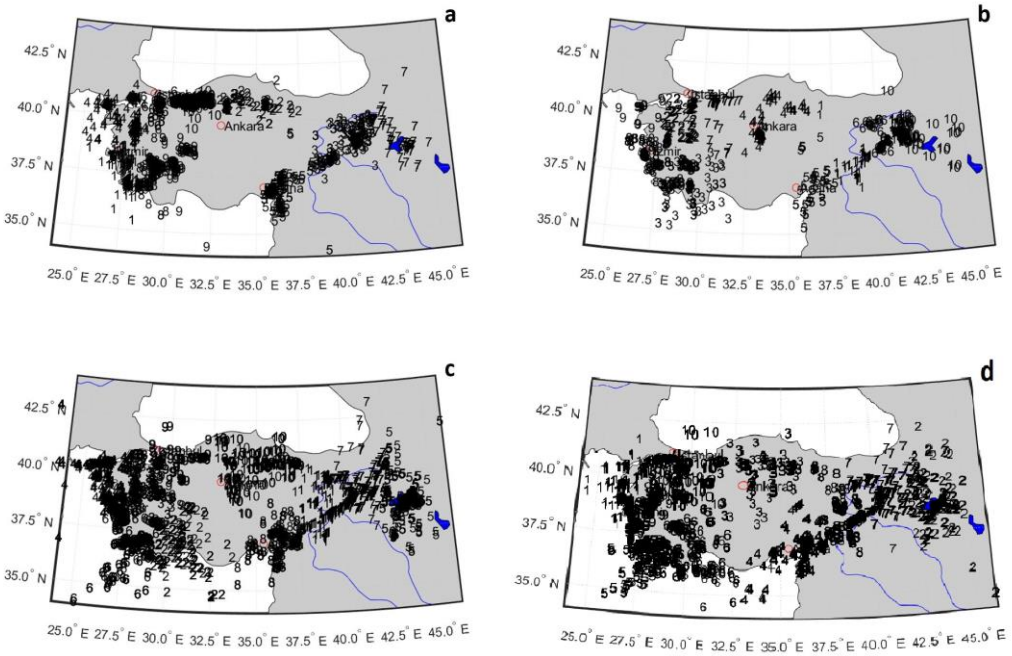


Figure 2.2: Turkey earthquake events recorded by AFAD between the years 1980 and 2018 were clustered in 10 groups using the K-medoids method according to their epicenter coordinates and shown on the map, a: 1980-2005 years, b: 2005-2010 years, c: 2010-2015 years, d: 2015-2018 years.

The fact that two earthquakes are close to each other does not always indicate that they are similar in geological terms. However, accelerometer data recorded by a station as a result of an earthquake event is undoubtedly highly correlated with the earthquake's location [65]. Therefore, the purpose of this statistical clustering is to first observe, through examination, that earthquakes with similar epicenters create similar signals and then to enable this representation to predict the epicenter from a single station via convolutional networks. Thus, instead of approaching the problem with the whole dataset, a region-by-region approach was performed.

At the beginning of the study, earthquakes with relatively close epicenter coordinates were clustered with the K-medoids method.

In K-medoids clustering, which is based on finding k representative objects representing various structural features of the data [66], the representative object is called the medoid and is the closest point to the center of the cluster. When dividing a group of objects into k clusters, the main goal is to find clusters where objects that are

very similar to each other coexist and objects from different clusters are unique from each other. Kaufman and Rousseeuw developed the most widely used K-medoids algorithm in 1987 [67]. The representative object is the most central object of the set, which minimizes the average distance from other objects. Therefore, this division method is applied based on the logic of reducing the sum of the uniqueness between each object and its reference point.

Since the goal is to find k objects, it is called the K-medoids method. After k representative objects are detected, k clusters are created by assigning each object to the closest representative. As shown in Figure 2.3, each representative object is replaced with a non-representative object in the following steps, shifting until the quality of the clustering is improved. This quality is evaluated using the average uniqueness cost function between the object and the representative object of the cluster it belongs to [68].

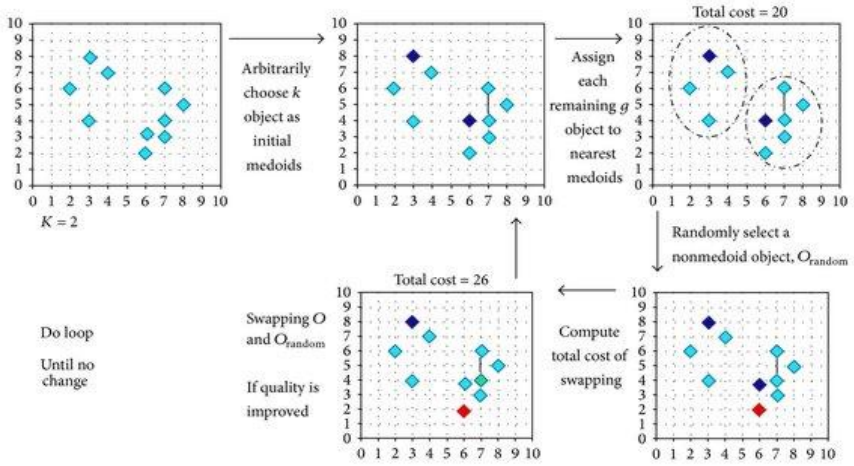


Figure 2.3: K-medoids Clustering [69]

Of the 42858 aforementioned records, 28365 records were recorded after 2015. In this study, records after 2015 were used because they are the majority of all records and because they were recorded in higher quality. For clustering purposes, a 28365x28365 geographic distance matrix (cost function) was created using the geographic locations of the epicenters of the earthquakes that occurred between the years 2015 and 2018. This distance matrix was fed into the K-medoids algorithm, and

the related earthquakes were divided into ten different clusters. These clusters (as numbers) are shown in Figure 2.2 (d).

2.2.2 Kandilli Observatory and Earthquake Research Institute Records

The proposed trainings were made with the data shared publicly by AFAD. After these trainings, the publicly shared data by the Kandilli Observatory were considered to increase the amount of the dataset. The data shared with the public by AFAD and Kandilli Observatory were recorded by stations in different regions. Both institutions have their own strong motion stations.

In the dataset published by the Kandilli Observatory, there are 36252 accelerometer records in SAC format between 2011 and 2022. The sample seismic wave of these records and the three-component record of the AKS earthquake recording station of the M=5.2 magnitude Çanakkale earthquake on February 7, 2017, are given in Figure 2.4.

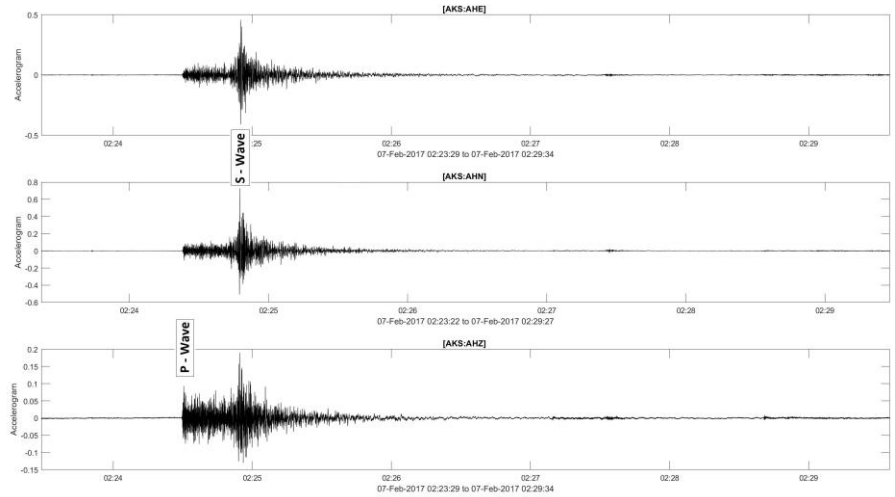


Figure 2.4: Three-component recording seismic wave display of the AKS earthquake recording station of the M=5.2 magnitude Çanakkale earthquake on 7 February 2017.

The sample seismic wave shown in Figure 2.4 is the arrival times and wave appearance properties of P and S waves on the three-component recording of the aftershock of the M=5.2 Çanakkale earthquake in 2017 at the AKS earthquake

recording station. The S-wave amplitude ratio of the north-south component is two times greater than the amplitude in the vertical.

As applied to the AFAD data set, the accelerometer records of the Kandilli Observatory BDTIM, the station, and epicenter positions coordinates by the K-medoids method are shown on the maps in Figure 2.5.

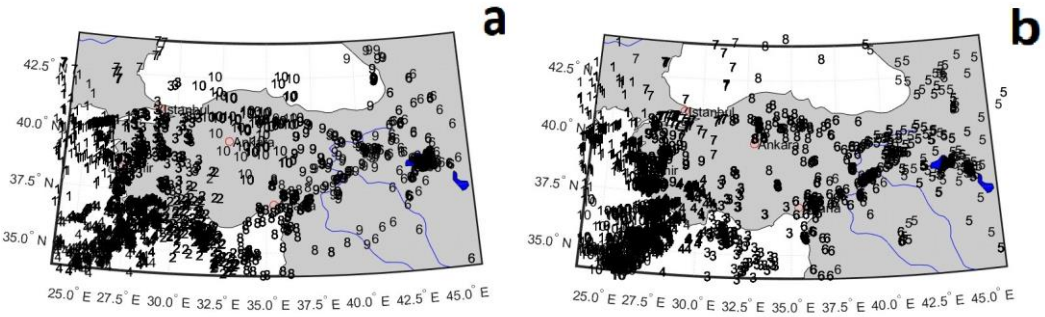


Figure 2.5: Turkey earthquake events recorded between the years 2011 and 2022 belonging to the Kandilli Observatory were clustered in 10 groups using the k-medoids method according to their epicenters and shown on the map, a: 2011-2016 years, b: 2017-2022 years.

At the beginning of the study, earthquakes with epicenters close to each other were clustered with the K-medoids [64] method. 23890 records belong after 2017. Recordings after 2017 were used because they were the majority of all records and because they were recorded in higher quality. For clustering purposes, a 23890x23890 geographic distance matrix (cost function) was created using the geographic locations of the epicenters of the earthquakes that took place between the years 2017-2022. This distance matrix was fed into the K-medoids algorithm, and the related earthquakes were divided into ten different clusters. These clusters (as numbers) are shown in Figure 2.5.

In this thesis, a dataset was created from two different databases recorded from different stations in Turkey. With these accelerometer data belonging to AFAD and Kandilli Observatory, clustering was done according to different years. AFAD dataset was used in the mentioned experiments. In the TÜBİTAK project, which is a continuation of the study, experiments will also be carried out with Kandilli Observatory data.

CHAPTER III

METHODOLOGY

3.1 PROBLEM SETTING

When earthquakes occur, features of that earthquake such as recorded accelerometer data, station coordinates, magnitude, and depth are recorded. The epicenter coordinate is one of them. The epicenter is the point on the place closest to the focus point [70]. The epicenter is also the point where the earthquake is most damaged or felt most strongly. This study aims to process epicenters in convolutional networks with accelerometer data recorded from a single station according to the information obtained from earthquake records.

In this study, a spectrogram-based false-color representation of the accelerometer records was made, and its usability was demonstrated for human examination and application in convolutional networks.

Earthquake records were taken from the public catalogs of AFAD, which included 42858 records; these records took place between 1990 and 2018 at the beginning of the study. Considering the quality of the records and the number of data among these records, a geographic distance matrix of 28365x28365 was created with 28365 records between the years 2015-2018, and clustering of 10 sets was made using the K-medoids clustering method to observe that earthquakes with similar epicenters create similar signals.

As a result of the clustering, the 3-channel accelerometer data of cluster 7, which contains 2133 records, was transferred to a two-dimensional time/frequency representation. The proposed spectrogram-based false color representation is given in Figure 3.1.

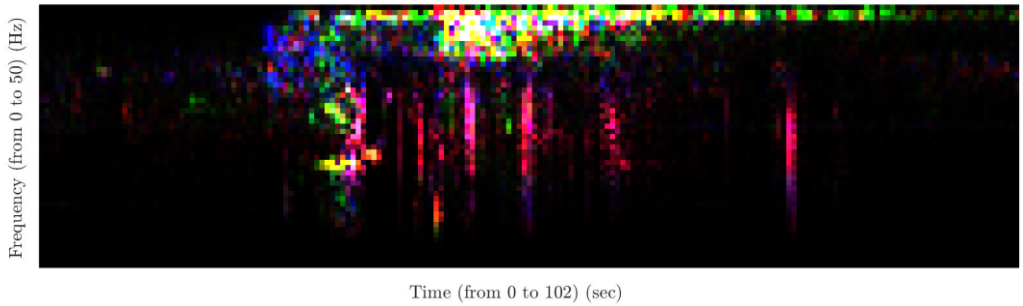


Figure 3.1: The proposed spectrogram-based false color representation

A spectrogram is a visual representation of the frequency spectrum of a signal as it changes with time. The common format of spectrograms is graphics with two geometric dimensions, with one axis representing time and the other axis frequency. The third axis contains the amplitude of a particular frequency at a specific time. This is characterized by the intensity or color of each point in the image [71].

The aim is to benefit human review by visualizing the records and primarily for feature analysis for deep learning methods. These created spectrograms were fed into the proposed CNN + LSTM network. The results demonstrate the efficient and reliable performance of the proposed model.

3.2 SUPERVISED LEARNING

In this case, the epicenter prediction is modeled as a supervised learning regression problem that includes earthquake recordings with a time series as input.

Regression is a familiar supervised learning approach that involves the prediction of a continuous label. Supervised learning is used after the model is trained to make predictions on test input data that the model has not seen before and without target information for test data. These problems can be any data type such as time series, image, or have any size and any number of input variables [72].

3.3 ACCELEROGRAM DATA IN THE SPECTRAL DOMAIN

Accelerometer records are multidimensional time series signals. Therefore, for the representation of such data, it is sufficient to refer to the time signal representations in the literature [73]. This study aims to make an epicenter prediction

from a single station through convolutional networks with these visuals transferred from the 3-channel accelerometer data to be visualized to a two-dimensional time/frequency representation.

It is conceivable that visualizing a signal with a method such as false color would be of no benefit other than human examination. Because the data called "signal" is just an input for a learning system, it has been seen that the visualizable signal can be helpful in both human analysis and feature analysis, which can be applied to some deep learning methods [74]. Convolutional encoders are trained to extract features from 2D-3D inputs in general. That's why converting the signal obtained from the accelerometer to false color spectrogram transfers the problem from time series prediction to a 2D regression problem. 2D regression problems are more suitable for CNNs than time series prediction.

The scope of this study is to make the multi-channel accelerometer data 2D through spectrograms, to express different channels with false color, to show the parts of the signal at different time intervals as fixed 2D, false picture parts, and with these images transferred to a two-dimensional time/frequency display. It can be summarized as a single-station epicenter prediction via convolutional networks.

Records from different earthquakes have different durations. For the data used in this study, the duration of the earthquakes between 2015 and 2018 observed to have durations ranging from 5 seconds to 300 seconds. Therefore, in this thesis, while the earthquake signal was converted to a spectrogram, each earthquake signal was divided into 5-second segments.

The Fourier transform was used to create the spectrograms. The Fourier transform is conceptually similar to the Taylor Series, which is a set of polynomials whose function is added together [75].

The Fourier Transform transforms a time function, into a frequency function. It uses Discrete Fourier Transform (DFT) to convert from time domain to frequency domain:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-j\omega t} dt \tag{3.1}$$

When using the Inverse Fourier Transform for frequency domain to time domain conversion:

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)e^{j\omega t}d\omega \tag{3.2}$$

Short-Time Fourier Transform (STFT), often visualized as a spectrogram, ensures a level of consistency time domain and frequency domain and is used to analyze time-varying signals. As the resolution in the frequency domain increases, the resolution in the time domain decreases.

STFT is repeatedly perform DFT to the overlapping window parts. The frequency measurement becomes more precise as the time window size controls the relationship between temporal and frequency resolution increases. The smaller the time window, the less accurate the frequency measurement.

STFT is defined as:

$$F_m(\omega) = \sum_{n=\infty}^{\infty} f(n)\omega(n-m)e^{-j\omega n} \tag{3.3}$$

where:

$f(n) = Input\ signal\ at\ time\ n$

$\omega(n) = Window\ function$

$F_m(\omega) = DTFT\ of\ the\ window$

Therefore, each spectrogram is 5 seconds long, but each earthquake consists of a different number of 5-second long spectrograms depending on the length of its accelerometer data. While applying the Short-Time Fourier Transform, the one-second window size and 50% overlap values were used as parameters. This method divides each earthquake into equal-length segments and applies the Fourier Transform separately.

Since the sampling interval of all the data obtained after 2015 in the AFAD dataset is 0.01 seconds, one-second windows carry 100 sample numbers. Therefore, the highest frequency value that the spectrograms could sample corresponded to 50Hz vibrations.

The suggested earthquake signal representation is given in Figure 3.2. Firstly, spectrograms of the records taken from three vertically positioned accelerometers for this earthquake event were created separately, using the above parameters. Then, these two-dimensional three spectrogram channels (N-S, E-W, U-D, respectively) were placed in 3 color channels of an RGB picture, and the representation in Figure 3.2 was obtained. Since such an earthquake event lasts longer than 30 seconds, there are six pieces of 5 seconds. When these parts are combined, a signal representative of the whole earthquake is obtained.

The illustration in Figure 3.2 facilitates human study and contains invaluable clues. For example, the first thing to notice is that in the first seconds of the earthquake, there is a movement between 10-20 Hz at right angles to the ground surface (vertical axis, blue channel). Then the earthquake intensified in all directions. This denotation with false color gives information about the order of the earthquake depending on the station coordinate. It, therefore, includes appropriate semantic features for a machine learning system.

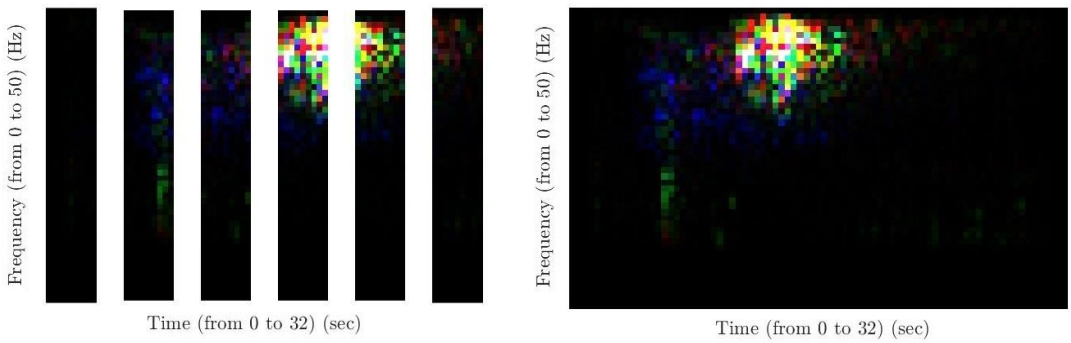


Figure 3.2: 5-second size fragmented (left) and combined (right) images of the proposed spectrogram-based false color representation are shown.

For the data set of Kandilli Observatory, when the duration of the earthquakes between the years 2017 and 2022 was examined, it was observed that they were dispersed in a wide range from 20 seconds to 900 seconds.

For the data set shared by the Kandilli Observatory, each earthquake signal was used in 5-second segments while the earthquake signal was converted into a spectrogram. Figure 3.3 shows that three two-dimensional spectrogram channels (N-S, E-W, and U-D, respectively) are placed in 3 color channels of an RGB image.

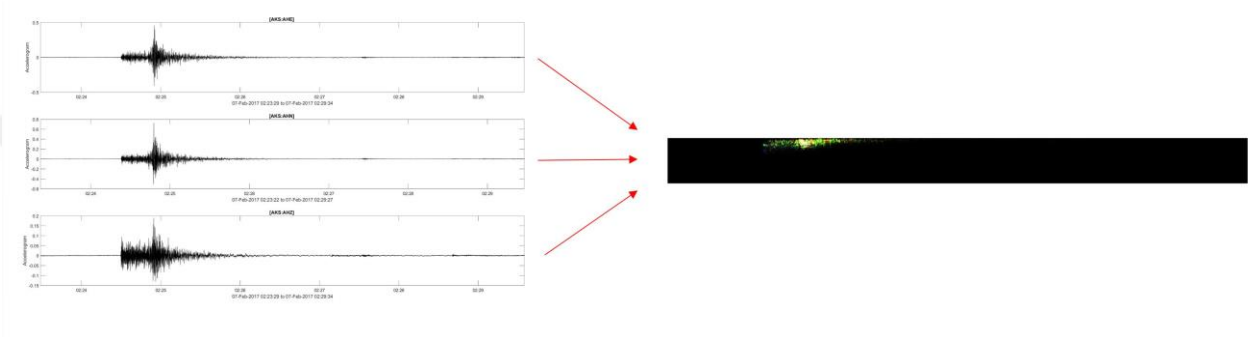


Figure 3.3: Conversion of Waveform to the proposed spectrogram-based false color representation

The earthquake signal representation suggested in Figure 3.3 is given with the data set of the Kandilli Observatory. Since this earthquake event lasted more than 300 seconds, 61 fragments of 5 seconds have occurred. When these parts are combined, a signal representative of the whole earthquake is obtained. The false color representation applied to the AFAD data was also applied to the Kandilli Observatory BDTİM dataset.

3.3.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning algorithm that uses images as input but is also successful in long-time series prediction. This algorithm, which captures the features in the inputs with different operations, consists of different layers. Various processes are applied to the input passing through the convolutional, pooling, and fully connected layers, which is made to enter the deep learning model.

The convolutional layer considers the spatial correlation between convolution filters and input variables. Convolution filters are a set of weight matrices that perform the convolution operation shown in equation 3.4 across each pixel in an image.

$$y(t) = x(t) \otimes h(t) = \int_{-\infty}^{+\infty} x(\tau) h(t-\tau) d\tau \quad (3.4)$$

Where $x(t)$ and $h(t)$ continuous time signals.

As shared in Equation 3.4, the given filter size is slid over the given image. As a result, a new matrix feature map is created. The parameters learned in this proposed algorithm are the values in these filters. The model constantly updates these values and begins to extract features even better. Apart from this, there are other filters used [76].

The stride value is determined by how many pixels the filter floating on the image will shift on the image. For example, in a convolution operation with a stride value of one, this operation is done by skipping only one pixel. As the stride value increases, the number of skipped pixels will increase so that the resulting feature map will be smaller.

After applying the filter to the image, the image size becomes smaller than the original image size. In the padding process used to prevent this, zeros are added to all four sides of the image as a frame around the padding value. The padding value can be increased according to the filter size so that the image to be obtained is the desired size.

Activation functions are needed to teach complex world data such as images and sounds to artificial neural networks. The activation process is a non-linear transformation over the input signal. The output of this transformation is sent as input to the next neuron layer. ReLU (Rectified Linear Unit), one of the commonly used activation functions, was used in this study. ReLU is a practical function in a multilayer neural network so that some neurons in the network are active and activation has a sparse load. The equation of ReLU makes the network work faster because it is the

simple max function between given input and zero. ReLU activation formula is given in 3.5.

$$f(x) = max\ (0, x) \tag{3.5}$$

Nonlinear functions such as ReLU are preferred to prevent the model from learning negative values or not being able to grasp some features due to these negative values.

Another layer used for downsampling, such as the Convolution layer, is the pooling layer. In this layer, where no learning process takes place, the aim is to reduce the size based on width and height by keeping the number of channels of the input matrix constant. This reduces the processing power required and focuses on more essential features by ignoring unnecessary features.

This kernel again slides over the image in the pooling layer, which has a kernel (filter) as in the convolutional layer. But instead of the convolutional operation, it applies the determined pooling technique this time. MaxPooling layer is preferred in this study; a new image matrix with a reduced size is created by choosing the maximum value among the pixel values at the point where the filter crosses over the image matrix.

While all the operations have done so far are performed on matrices, single plane vectors are given as input to the Fully Connected (FC) layer, where learning takes place. Flatten layers are used for single plane vector transformation with the matrix. It is given to FCs as input, which becomes a single plane vector, and the learning process begins.

Normalization is a preprocessing technique used to standardize data. Batch Normalization, on the other hand, is a normalization operation between the layers of a neural network. It normalizes in mini-batches, not the whole data. It facilitates learning by allowing to accelerate training and use higher learning rates. It is usually located between the convolution layer and the activation layer [77].

The dropout layer is used to prevent the model from being overfitted [78]. It allows neurons to be ignored during the training phase of certain randomly selected neuron sets. This provides an increase in performance [79].

Finally, another layer used in this study, the LSTM (Long Short-Term Memory) layer, is a particular type of artificial RNN (Recurrent Neural Network) especially suitable for learning long-term dependencies. LSTMs were born as a solution to the vanishing gradient problem experienced in RNNs. Vanishing gradient is an unavoidable problem for RNN due to its architecture, where weights disappear with back-propagation. LSTMs have internal mechanisms called gates that can regulate the flow of information.

As the cell state goes on its journey, information get's added or removed to the cell state via gates. This LSTM concept is shown in Figure 3.4. LSTM uses these mechanisms to keep or forget information (forget). Instead of hidden states in RNN, LSTMs have cell states. Cell states are related to gates in LSTM. Gates and information are added or removed as the cell state continues its journey.

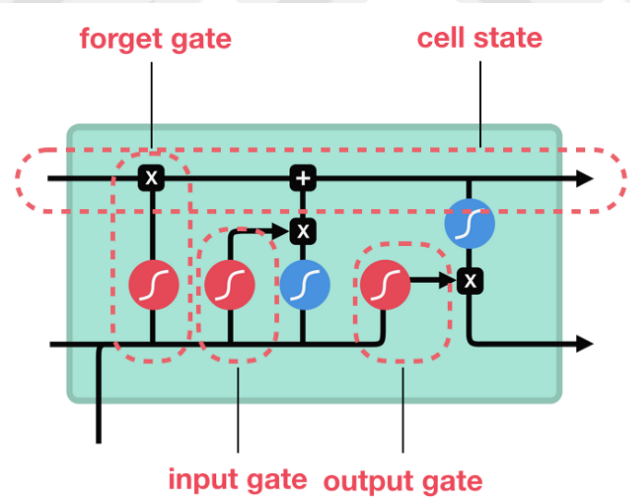


Figure 3.4: Long-Short Term Memory (LSTM) Architecture [80].

Cell states work as long-term memory, while hidden states work as short-term memory. The tanh and sigmoid functions found in the LSTM gates are, respectively equations 3.6 and 3.7:

$$-1 < tanh\,(x) < +1 \quad (3.6)$$

$$0 \leq sigmoid(x) < +1 \quad (3.7)$$

The sigmoid function is used to update or forget. Any number multiplied by 0 will cause values to be lost or "forgotten" because it is 0. On the other hand, any number multiplied by 1 remains the same, or "kept," because it is itself.

The forget gate decides what information is forgotten or kept. Information from the previous hidden state and information from the current input is passed through the sigmoid function. The closer the values are to 0, the closer they are to forget, and the closer to 1 they are to keep. Gate outputs are vectorial, not scalar. That's why cell state updates are element-wise.

The input gate is used to update the cell state. Passing the previous hidden state and current input to the sigmoid function to decide which values to update, the input gate also passes the hidden state and current input to the tanh function, which does the overwhelming task of arranging the values between -1 and 1 to organize the network.

Then the tanh output is multiplied by the sigmoid output, which decides which of the information from the tanh output will be kept in the sigmoid output.

The cell state forget vector is multiplied pointwise to calculate the cell state. If this multiplication is done with values close to 0, it may decrease the values in the cell state. Then, pointwise addition is made, which updates the cell state to the new values that the neural network finds relevant to the output from the input gate.

The output gate decides what the next hidden state will be. Hidden states containing information about previous inputs are also used for prediction. In the output gate, the hidden state and current input first pass through a sigmoid function. Then the newly updated cell state passes through the tanh function. The tanh output is multiplied by the sigmoid output to decide what information the hidden state should carry. The result of this operation gives the new hidden state. The new cell state and the new hidden state are moved to the next time step.

The proposed network within the scope of this study is shared in Figure 3.5.

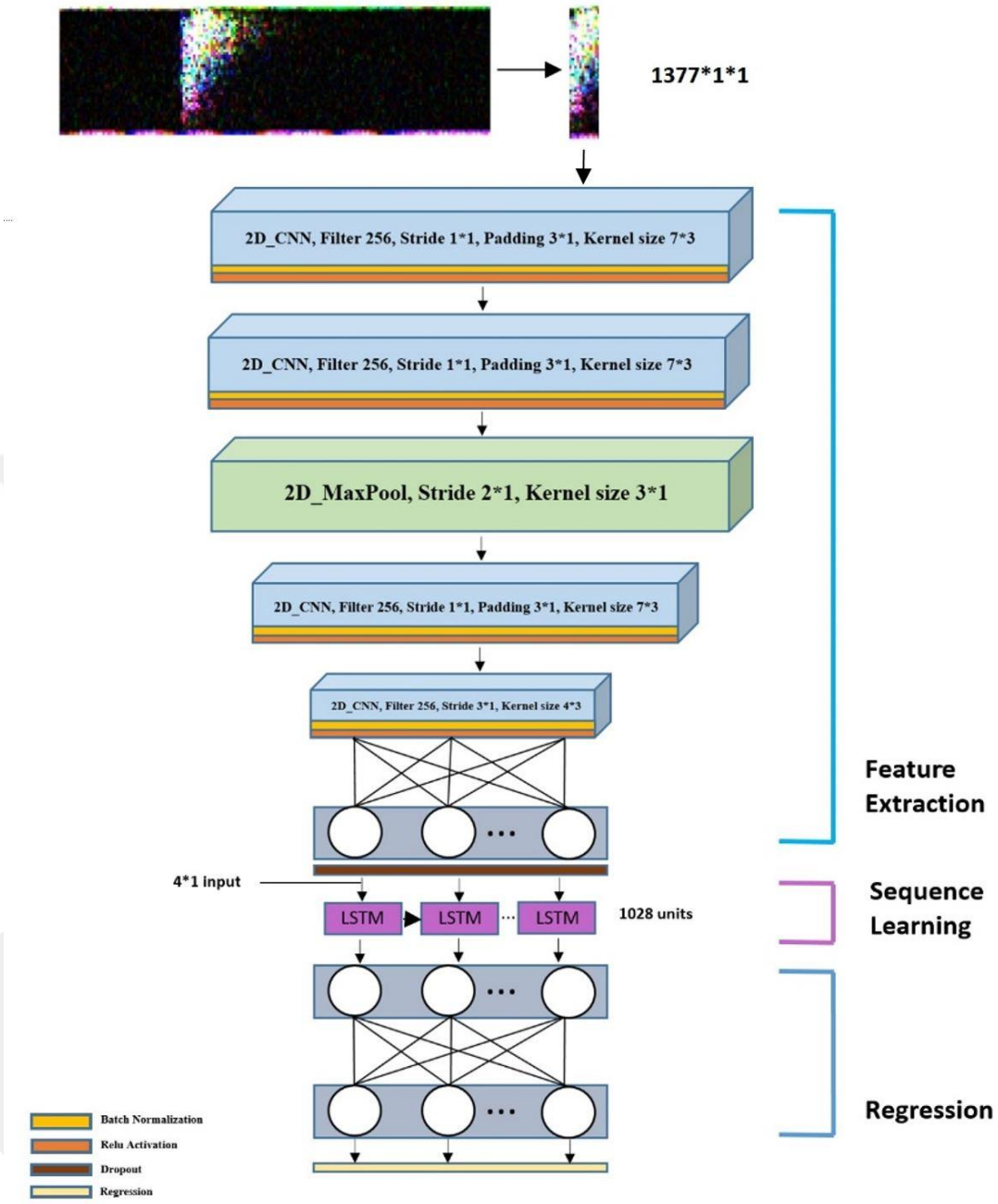


Figure 3.5: Recommended CNN + LSTM Architecture

The spectrogram with the highest energy of the 3-channel accelerometer (N, E, Z components) data is fed as input to this network structure. The station coordinate

and altitude information added to the formed high-level features are connected with fully connected layers.

The station location is included in the network after the high-level features are created. At the same time, the time difference between the marked P and S waves is added here as a second input. The P wave is the first wave recorded by the station, followed by the S wave.

After the spectrogram input comes to the convolutional blocks consisting of the convolutional layer, the activation layer, and the sampling layer, after the two inputs were combined and all the nodes were connected in the fully connected layers, the LSTM layer was added to the CNN network, and the learning and sequence data modeling was performed. RNN is used in related input sequence problems and can take inputs of different lengths. RNNs are designed to process sequential data such as seismograms. An earthquake recording is a time series signal. Signal length depends on several factors and are not the same for even the same earthquake recorded at different stations. High-level features consisting of full link layers are fed to LSTM. As the output, the epicenter coordinate and depth of the earthquake event were learned with a regression layer.

3.3.2 Neural Network Architecture

The first input of the neural network is the spectrogram, which represents the 2-dimensional time-frequency information of the three-channel accelerometer data. The waveforms of the three-channel components (N, E, Z) of the accelerometer data were converted to spectrograms, as shown in Figure 3.5. Afterward, the station information of the related earthquake event, the height, and the time difference between the P and S waves were added to the network after the features were created. The convolutional network visualized in architectural Figure 3.5 can be divided into two parts: the concatenation layer up to the encoder, which extracts relevant information known as feature maps, and the full link layers and LSTM layer which acts as a regression to identify the event.

Four convolutional blocks are used to extract features from the two-dimensional spectrogram data. Spectrograms are fed into convolution layers to

generate feature maps. These features were used in the fully connected layer to learn only the resulting features before other inputs were added. Station information and other inputs, which have more understandable meanings, are fed after the resulting attributes. After combining two different inputs, earthquake spectrograms with time series problems were provided to the LSTM layer. It is then fed into its fully connected layers to perform the learning. While choosing the proposed architecture, the studies in the literature were taken as an example [24, 58, 81].

A 2d convolutional operation is described by:

$$Y = B + \sum(W + 1) \tag{3.8}$$

where:

$Y = Output\ Target$

$B = Bias$

$W = Weights$

$I = Input\ Data$

The number of tensor output sizes for each layer is given by:

$$O = \frac{I - K + 2P}{S} + 1 \tag{3.9}$$

where:

$O = Size\ (width) of\ output\ image$

$I = Size\ (width) of\ input\ image$

$K = Size\ (width) of\ kernels\ used\ in\ the\ Conv.Layer$

$N = Number\ of\ kernels$

$S = Stride\ of\ the\ convolution\ operation$

$P = Padding$

The number of parameters for each conv layer is computed by:

$$W_c = K^2 * C * N,\ B_c = N,\ P_c = W_c + B_c \quad (3.10)$$

where:

$W_c = Number\ of\ weight$

$B_c = Number\ of\ bias$

$P_c = Number\ of\ parameters$

$N = Number\ of\ kernels$

$K = Size\ (width) of\ kernels$

$C = Number\ of\ channels$

The number of tensor output sizes for MaxPool layer is given by:

$$O = \frac{I - P_s}{S} + 1 \quad (3.11)$$

where:

$O = Size\ (width) of\ output\ image$

$I = Size\ (width) of\ input\ image$

$S = Stride\ of\ the\ convolution\ operation$

P_s = *Pool size*

And the number of parameters of a fully connected layer to another fully connected layer is:

$$W_f f = F_{-1} * F,\ B_f f = F,\ P_f f = W_f f + B_f f \quad (3.12)$$

where:

$W_f f = Number\ of\ weight$

$B_f f = Number\ of\ bias$

$P_f f = Number\ of\ parameters$

$F = Number\ of\ neurons$

$F_{-1} = Number\ of\ neurons\ in\ the\ previous\ FC\ layer$

Rectified Linear Unit (ReLU) activation computes:

$$ReLU(x) = max\ (0, x) \tag{3.13}$$

The number of parameters of a LSTM layer is:

$$W_f = (4 * H) * I_{-1},\ W_f = (4 * H) * H,\ B_r = (4 * H) \tag{3.14}$$

where:

$W_f = Input\ weights$

$B_r = Number\ of\ bias$

$W_r = Recurrent\ weights$

$H = Number\ of\ hidden\ units$

$I_{-1} = Previous\ layer\ input$

A detailed example of the CNN + LSTM architecture used is given. The spectrogram of the 51x9x3 3-channel accelerometer data was converted into a column vector. The station coordinate (2x1), altitude, and the time difference between the P and S waves, marked in the earthquake records before, were added to the 1377x1x1 input, respectively. These inputs are split into spectrograms for high-level feature extraction and other information added to the input containing more meaningful information. The mini-batch size is explained assuming inputs of size 500 and 1381x1x1:

1. The first convolution block takes 51x9x3 3-channel spectrograms as input and applies 256 filters of 7x3 size to them (stride = 1x1, padding = 3x1). It is followed by batch normalization and ReLU layer. The number of outputs for Conv2d is 51x9x256, and there are 16,384 Conv2d parameters and 512 BatchNorm parameters.

2. The other convolution block applies 256 filters of size 7x3 (stride = 1x1, padding = 3x1). Next, 256 batch normalizations are applied. Finally, 2d max pooling is applied (kernel size = 3x1, stride = 2x1). The number of output for Conv2d is 51x9x256, followed by MaxPool2d at 25x9x256. There are 1,376,512 Conv2d parameters and 512 BatchNorm parameters.

3. The third convolution block applies 256 filters of size 7x3 (stride = 1x1, padding = 3x1). Next comes the ReLU and batch normalization layers. Batch normalization is a common normalization procedure for optimizing the performance of hidden layers. The Conv2d output is 25x9x256, and there are 1,376,512 Conv2d parameters and 512 BatchNorm parameters.

4. The last convolution block applies 256 filters of size 4x3 (stride = 3x1, padding = 0). It is followed by batch normalization and ReLU layer. The Conv2d output is 8x7x256, and there are 786,688 parameters for Conv2D and 512 parameters for BatchNorm.

Spectrograms are fed into the convolution block to generate the feature map. These high-level features are first provided to the fully connected layer to learn the spectrograms within themselves.

5. The first fully connected layer has 1024 neurons. Dropout, a normalization technique, is applied during training. The dropout parameter is set to 50%. The linear operation output size is 1x1x1024 with 14,681,088 parameters.

In the next step, the high-level features from the spectrograms and the station coordinate (2x1), altitude, and P-S wave difference inputs of the earthquake were combined with the concatenation layer. Before being introduced to the fully connected layer, the feature map and split-second inputs are flattened and combined horizontally. Before being introduced to fully connected layers, these combined feature maps and inputs were fed to the LSTM layer. Epicenter prediction is an RNN problem since each earthquake data has a different length and is ultimately a time series signal.

6. The layer is LSTM layer with 1028 hidden states. There are 4,227,136 input and recurrent weights and 4112 biases for the LSTM layer.

7. The second fully connected layer is connected with 1024 nodes to another layer. There are 1,053,696 parameters for a fully connected layer.

8. And the last fully connected layer is connected with three nodes. There are 3,075 parameters for a fully connected layer.

9. The output layer is a regression layer. As a result of the learning, 2x1 epicenter values and 1x1 depth values of the earthquake were suggested as the output of this layer.

The best results were obtained with this proposed architecture. In this study, different architectures are tested and discussed in the following sections.

3.3.3 Training the Network

In training a deep learning model, it is necessary to define a set of hyperparameters that significantly impact performance. These hyperparameters describe how to calculate loss from supervised data, how much this loss should affect the weights of the network, and how long the model will be trained. In this section, the definitions and justifications of the preferred hyper-parameters in the proposed network are discussed in detail. The hyperparameters used are taken as examples from the literature [82, 83, 84, 85]. Optimum hyperparameter results were obtained by trying different hyperparameters for the architecture and the problem.

Hyper-parameters define how to calculate the error between the prediction of the model and the ground truth. Since the datasets used to train the models are too large to be stored in memory, they are divided into smaller groups called mini-batches, and the loss is calculated for each. In the proposed network, the mini-batch size was used as 500. Various batch sizes ranging from 8 to 1024 were tested within this network. Each mini-batch will update the weight values of the network with calculated loss values using stochastic gradient descent algorithms. Combining the smaller batch sizes with the stochastic gradient descent updates took longer because it couldn't use the existing GPU. Larger batch sizes have been found to take advantage of the GPU more efficiently and are faster.

Since the approximation output is scalar values, the back-propagation error is calculated using the Root Mean Squared Error (RMSE) to train the model and update the weight. The equation of the RMSE function used is shared in 3.15.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2} \quad (3.15)$$

where:

$y_i = Ground\ Truth$

$\widehat{y_i} = Prediction\ from\ the\ model$

Another preferred function as a loss function is F1-score. The main reason for using the F1-score value is not to make an incorrect model selection in data sets that are not evenly distributed. In addition, F1-score was preferred since a measurement metric that would include not only False Negative or False Positive but also all error costs was needed. The equation of the F1-score function used is shared in 3.16.

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3.16)$$

After the loss function selection, the optimization algorithm was decided. An optimization algorithm helps the neural network learn faster by determining how to update the weight of the network and changing the learning rate depending on the current state of the network. Some optimization algorithms use a component called Momentum to accumulate the computed gradient for past iterations of the network and update the weights of the network-based not only on the current loss but also on the direction of all previous gradients.

Many optimization algorithms can be used to update the model state. Analysis can be done to select the appropriate optimizer for each model, but since this is a very comprehensive study and out of the scope of this study, all models in this project were evaluated with the same optimization algorithm, the Stochastic Gradient Descent Momentum (SGDM) optimizer. The SGD with momentum algorithm has become one of the most common optimizers for [86], where it can be used for various problems.

While the model is being trained, not all of the data is trained at the same time. As stated earlier, the datasets used to train the model are divided into mini-batches because they are large. Thanks to these mini-batches, the data is included in the training in parts. After the first batch is trained, the model's performance is tested, and the weights are updated with back-propagation according to this success. This process is repeated at each training step to try to calculate the most appropriate weight values for the model. The epoch for each training step is the last parameter to choose.

The models have been trained for 500 epochs maximum. Various epoch sizes from 10 to 2000 have been tested under this model. In addition, if the convergence point is reached before 500 epochs, the training process is stopped because it will not improve the model's performance much after this point.

In Table 3.1, the hyperparameters defining the training process of the proposed model are summarized. After the training configuration was described, each model was analyzed and modified according to its performance results.

Table 3.1: Training hyperparameters configuration

Loss Function	RMSE, F1-score
Optimizer	Sgdm with learning rate $3e^{-7}$
Max. Epoch	500

The deep learning model generalizes any input within the model and can learn a number of features that allow it to analyze. In order to evaluate the generalization of the developed model, it is necessary to separate the dataset into separate groups with a set of examples from which the model will learn and another group with invisible examples. The cross-validation preferred in this study is a statistical (resampling) method used to evaluate the machine learning model's performance on data that has not been seen before, as objectively and accurately as possible and tests the results of the analysis on an independent data set. In this method, the dataset is randomly divided into three sets: the train set to be used to learn the patterns in the data, the validation set containing the samples that are not seen to be evaluated during the training process, i.e., for hyperparameter optimization, and the test set, which has the samples not included in any of the previous sets, to be used to evaluate the model after training.

In this study, a cross-validation algorithm was applied for the selected data set in each experiment. The percentage distribution of this method used for each data set is shown in Table 3.2.

Table 3.2: Cross-Validation Distribution

Train Set	70%
Validation Set	15%
Test Set	15%

3.3.4 Transfer Learning

Transfer learning is reusing features, weights, etc., obtained from a pre-trained neural network model [87]. Normally, independent learning from scratch is performed for each problem. However, this method is preferred because it is possible and advantageous to use some information learned from some problems in other problems. The weights of the trained models contain much information about that problem and its solution. Therefore, fine-tuning using this information reduces the training time for the new model.

Transfer learning allows working on smaller data sets. Creating large-scale datasets is time-consuming. Fine-tuning pre-trained models with transfer learning ensures high performance using fewer data.

In the proposed CNN + LSTM model, the outputs of the pre-trained neural network (feature, weights, etc.) similar to the CNN model are used. Unlike this proposed model, this pre-trained model does not include P and S wave inputs. However, it had a higher number of data than the proposed model. In the pre-trained model, 2133 cluster 7 data shared in the Dataset section were used. In the proposed model, less amount of data is used since P and S waves marked in the data set are included as input. If transfer learning had not been performed with this pre-trained model, which also differs in hyperparameters, it was observed that the performance achieved in the proposed model would be much less. These results are shared in detail in the Experiments and Results section.

CHAPTER IV

EXPERIMENTAL RESULTS

4.1 EXPERIMENT SETUP

Several experiments were conducted with the ultimate goal of creating a generalized prediction model. Studies carried out in this context tested the performance of the methods proposed in chapter 3. This section presents these studies and reviews these results.

At the beginning of these experiments, the data set publicly published by AFAD was used, as mentioned in chapter 2. In this data set, there are 42858 three-component accelerometer records in text format between the years 1980 and 2018. This study aims to predict the epicenter coordinate of the earthquake using accelerometer data from earthquake records. To observe the relationship between these two components of earthquakes, firstly, K-medoids clustering was applied to all dataset according to the epicenter coordinate. Due to a large amount of data, this clustering process was updated to be 10 clusters in each clustering process between the years 1980-2005, 2005-2010, 2010-2015, and 2015-2018. This clustering process is shared in Figure 2.2.

Since the beginning of the study, this problem has been seen as a time series prediction problem. In order to give the current data as an input to a convolutional neural network, first of all, time/frequency transformation was performed on these 3-component accelerometer data, and it was turned into a spectrogram, which is a two-dimensional false color representation of [65], which was proposed in another study.

In Figure 4.1, a visualized version of an example accelerometer recording is given after spectrogram transformation. While the spectrogram conversion suggested in Figure 4.1 was realized, it went through the processes mentioned in section 3.3. After these processes, each 3-component accelerometer data had separate spectrograms of 5 seconds, equal to the earthquake length. In the visualization in

Figure 2.2 (a), each segment will belong to 5 seconds of the accelerometer recording. However, there will be a 5-second segment equal to the length of the relevant earthquake, and the visualization on the right shows the spectrogram transformation of the accelerometer data for the entire sample earthquake.

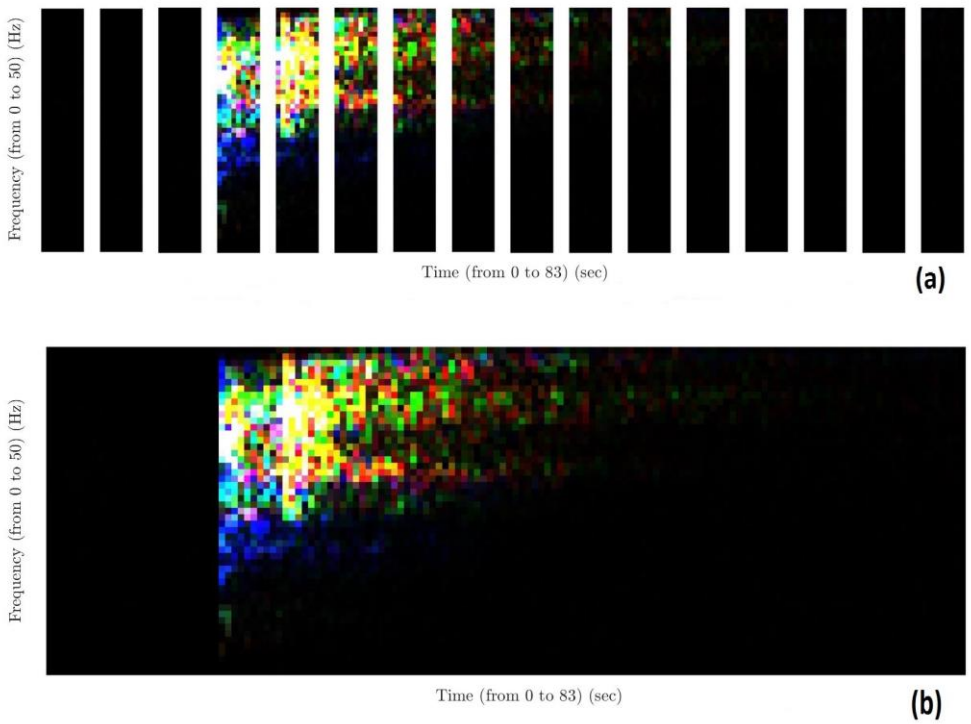


Figure 4.1: Sample spectrogram display from the data set publicly published by AFAD. In the spectrogram-based false color representation of the Adıyaman earthquake with a magnitude of M=3.7 that took place on January 3, 2015, it is seen that the earthquake lasted longer than one minute. It is observed that the earthquake moves in all three axes after the first seconds. (a) 5-second segmental representation of the proposed spectrogram-based false color representation, (b) Combined representation of the proposed spectrogram-based false color representation.

At this stage, the data were divided into three different groups, as mentioned in chapter 3. These are the training set containing 70% of the dataset, the 15% validation set, and the remaining 15% test set.

The dataset created for the training was composed of subsets selected from 250 to 19855 earthquake data. A series of training were made with the created subsets. Because the goal is to create a generalized prediction model, the model that requires

the least number of training examples is preferred over other models. Many experiments have been done for hyperparameter, architecture, and data preprocessing decisions. The best models were selected manually from the experimental results by examining different train scenarios for the model trained in each subset.

4.1.1 Computational Setup

All data collection and data processing processes of AFAD data are written in Matlab [88] programming language on a Windows 10 Pro desktop with Intel(R) Core(TM) i5-9400F 2.90GHz CPU and GeForce RTX 2060 SUPER GPU. Data collection and processing processes of Kandilli data are written in Matlab [88] programming language on a Windows 10 Pro desktop with Intel(R) Xeon(R) W-2255 3.70GHz CPU and two RTX A5000. The training of CNN models, a cluster, and testing performance is done using a Windows 10 Pro desktop with Intel(R) Core(TM) i5-9400F 2.90GHz CPU.

4.2 ANALYSIS AND DISCUSSION

4.2.1 Relation on a Single Station

A number of analyses were made in order to see the relationship between the data shared by AFAD with the public and the proposed problem. From the shared data, it has been observed that some information (altitude, magnitude etc) about the earthquake that took place in the records before 2015 is missing. Therefore, these analyses were made using data recorded starting from 2015 to 2018, as the recording quality was higher. First, it was observed that 28365 data were recorded from 820 different station numbers.

Figure 4.2 shows how many earthquake events are recorded in each of these 820 stations.

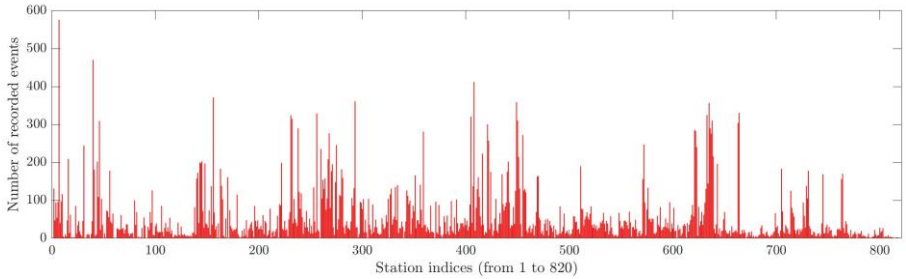


Figure 4.2: The number of earthquake events in each of 820 different stations, recorded by AFAD's publicly released 28365 earthquake data between 2015 and 2018.

In order to observe the relationship of the three-component accelerometer data with the epicenter coordinate, it was observed how many different epicenter coordinates contained earthquake events at each of the 820 other recording stations. There are several aspects to consider when performing this analysis. First of all, more than one earthquake event can be recorded at a station, as shown in Figure 4.2. In the analysis to be made, it was observed how many different epicenters a station recorded earthquakes.

The clustering method was used to perform this analysis. Instead of the more preferred K-means clustering algorithm, the K-medoids algorithm was preferred, as mentioned in chapter 2. The reason why the K-means algorithm is not selected is that the distances to be used are not Euclidean distances. Instead, the distance between the epicenter coordinates of the bilateral earthquake events was calculated for each of the 28365 data for 2015-2018, which were shared publicly by AFAD and included higher recording quality. Thus, a distance matrix of 28365x28365 was obtained, in which each earthquake had the epicenter coordinates of each earthquake. The calculated distance matrix is given as input to the K-medoids clustering algorithm. As a result, earthquake centers in the country were determined as much as the number of clustering to be done. Clustered form of 28365 data divided into ten different clusters with the K-medoids algorithm applied for this purpose is given in Figure 2.2 (d).

The first experiments were carried out with 28365 aggregated data. These initial experiments were trained on a different network than the neural network model proposed in the section. Although there are a few differences between the two models, the differences in the model in which the first experiments took place; After the first

fully connected layer, there is a ReLU layer, there is no LSTM layer, and there are ReLU and dropout layers in the second fully connected layer.

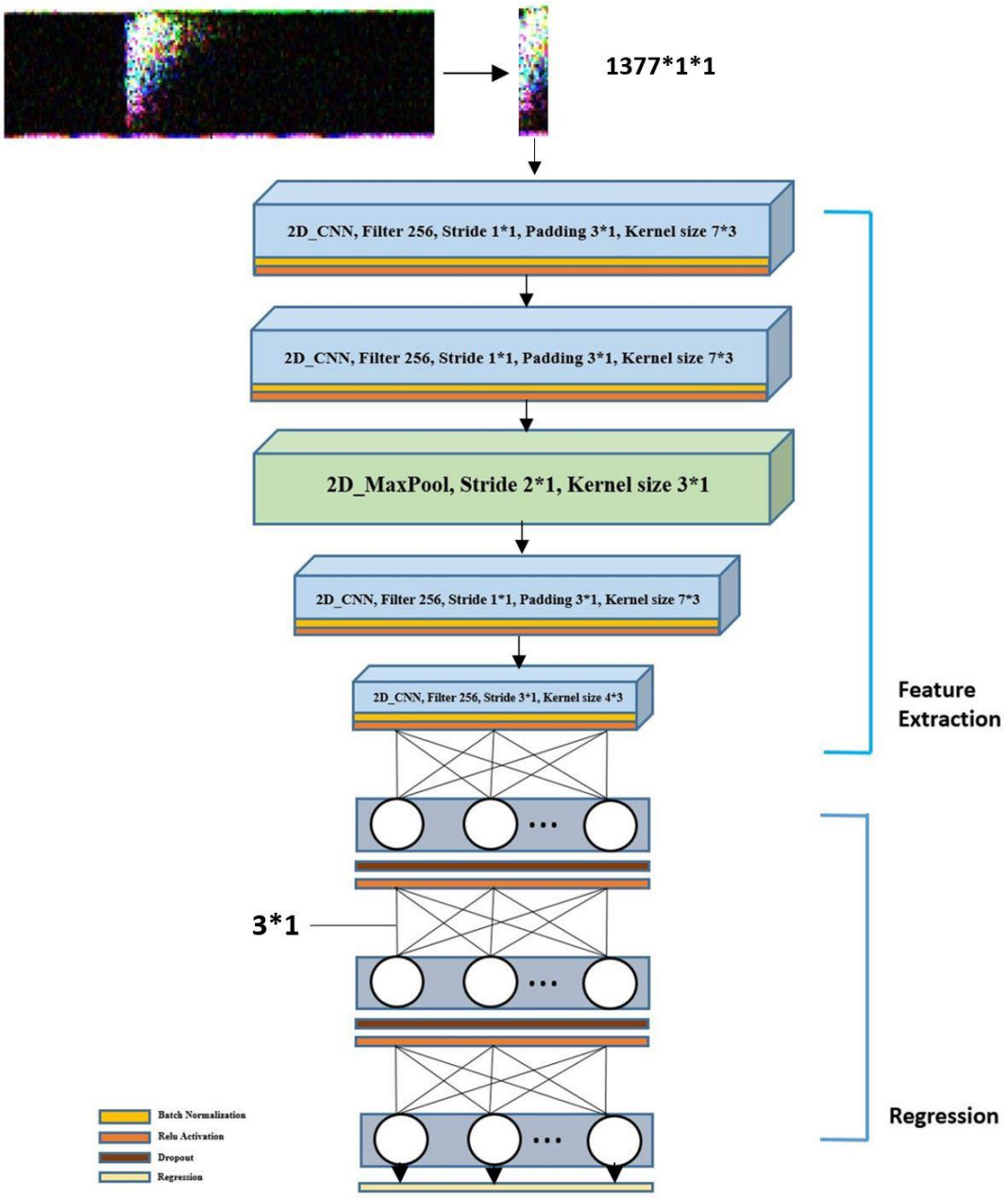


Figure 4.3: First Proposed Architecture

Overfitted results were observed in the continuation of the experiments. The main reason for validation accuracy decreasing is seen as data, and the data do not allow us to generalize the problem sufficiently. In the experiments, 3-component

accelerometer data were converted into 2D false color representations, and one-second 9-step signals starting from the highest energy point of these spectrograms were used. The non-overlapping region next to it was also taken to avoid the problem of shifting in the signal, but when the results obtained were examined, it was observed that these procedures were not sufficient.

In order to prevent this situation, instead of taking a 9 area starting from the peak region of the earthquake, a total of 11 51x9x3 spectrograms were prepared; let's call the peak region of the earthquake n, starting from n-5 and fully overlapping up to n+5.

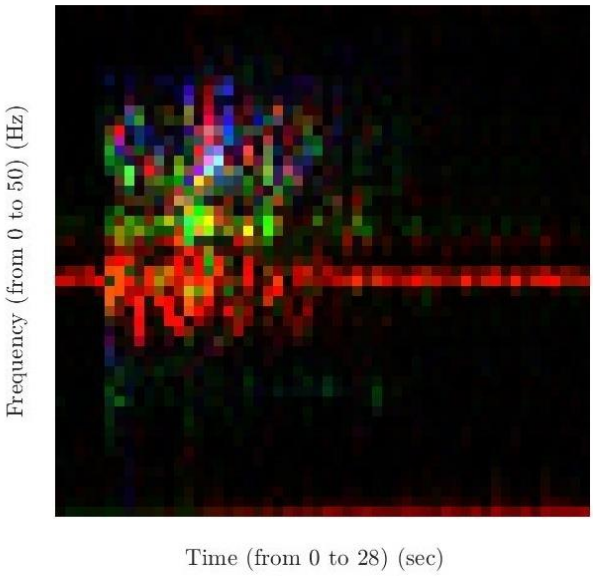


Figure 4.4: Representation of proposed spectrogram

It was observed that the localization error decreased by 50% when 28365 earthquake events, which were fully overlapped, were trained with the neural network model proposed for the first experiment. Separate experiments were carried out for each cluster in parallel with data starting from the highest energy and with data augmentation.

It has been observed that the performance is getting better with fully overlapping. According to the results obtained here, the dataset will be continued with fully overlapping or with maximum energy. The hyperparameters were updated, and

the model results, which were trained with maximum energy spectrograms and 200 epochs, are given in Figure 4.5. It was observed that the localization error decreased by 49%, and the performance can be improved by fine-tuning with babysitting without playing with the dataset and without using fully overlapping.

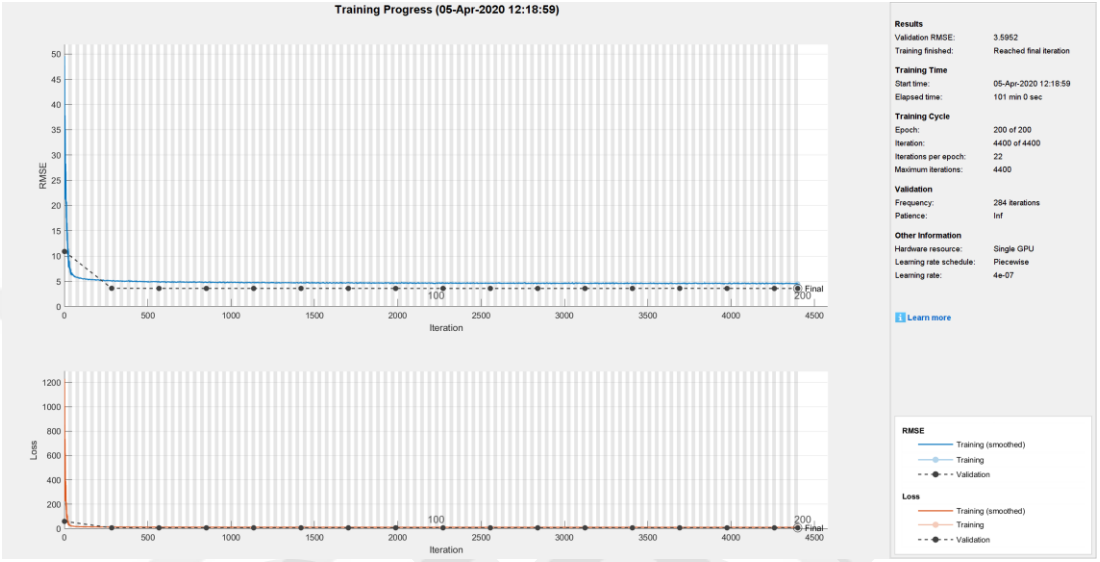


Figure 4.5: 200 epoch training results of 28365 data with maximum energy spectrograms.

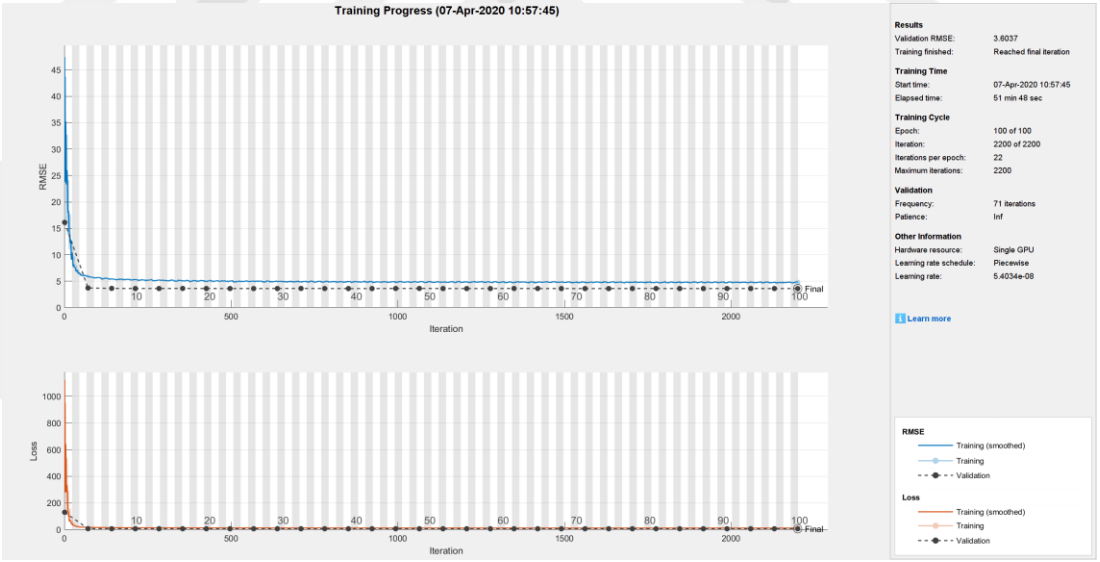


Figure 4.6: The result of 100 epoch training with LR drop rate of 28365 data with maximum energy spectrograms.

In the train shared in Figure 4.5, it was seen that learning stopped at ten epochs. To change this, training have been made that decreased the LR hyperparameter given

at the beginning every five epochs. LR drop rate was applied in 100 epoch training in Figure 4.6. Thus, as LR drops, progress will be seen again; it is aimed to solve the learning stop problem seen in Figure 4.5. When Figure 4.5 and Figure 4.6 were compared, it was seen that the result did not change. In both pieces of training, learning work is completed in 10 epochs. The result is that we have data or network capacity-based walls, not an LR or optimization-based wall, while learning all the data.

With the decision taken here, no changes were made to the overall data. Figure 4.8 shows train results of 2495 data belonging to cluster 9 with 2000 epochs. Compared to 28365 pieces of data between the years 2015-2018, much less data is encountered. However, it took a long time to learn. As mentioned in Figure 4.5 and Figure 4.6, while learning with 28365 data was completed in 10 epochs, the cluster-wise analysis showed that learning took up to 450 epochs in Figure 4.7.

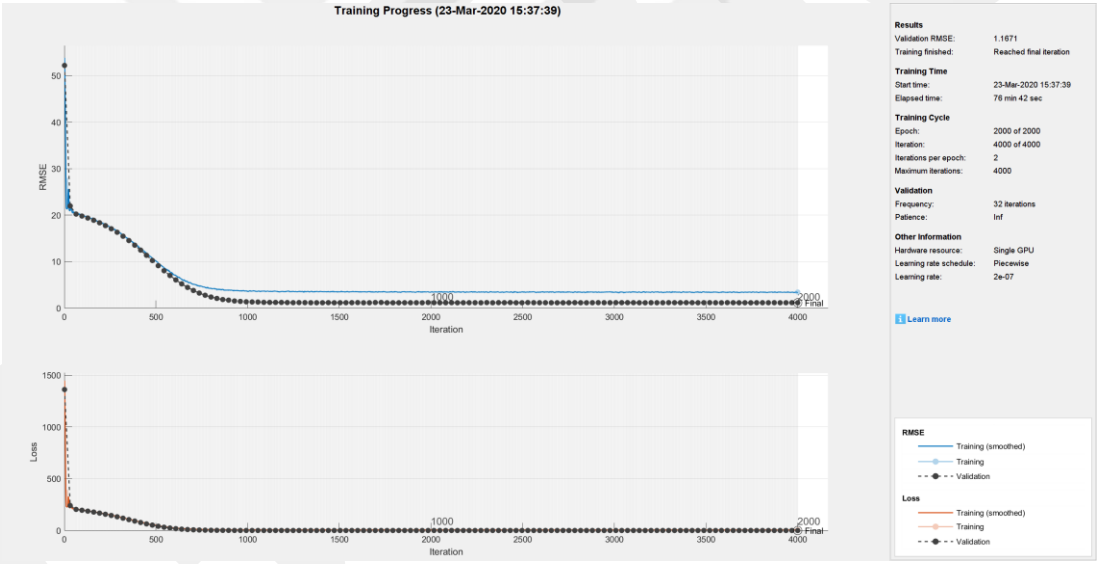


Figure 4.7: Training result of maximum energy spectrograms of 2495 cluster 9 with 2000 epochs.

In experiments with data from 2015-2018, learning stopped in ten epochs due to the complexity of the data and the lack of network capacity. However, cluster-wise experiment results were found to be more successful. The problem continued to be studied as a cluster problem. To verify this, experiments were repeated by selecting only stations close to the epicenter cluster center in cluster. For each cluster, separate

experiments were carried out by taking only earthquake events at stations 300 km, 200 km, and 100 km close to that epicenter cluster center.

In Figure 2.2 (d), it is observed how many different places each different station can sense earthquakes separately. When cluster 1 was examined according to the figure, it was revealed that 2023 distant earthquakes were felt at 200 km from the station distance to the cluster center belonging to the western region of Turkey. Looking at Cluster 3, it has been observed that Turkey mostly includes the northern part of Turkey as well as the Central Anatolian region. The desired result here is to prepare a database that can sense (record) earthquakes in different locations and simultaneously be sufficient for a relatively large number of earthquakes. In the epicenter clustering grouped according to the station distance, the best results were encountered at a distance of 200 km. The best result was obtained in cluster 7 with a localization error of 80 km, with 2133 pieces of data at a 200 km distance to the station.

More localization error was tried to be reduced by fine-tuning with the suggested data. Before the architectural change, the accuracy of the predicted coordinate was observed as a result of the trainings made with 2133 cluster 7 data used in the analysis. In order to see this, epicenter coordinates, station coordinates, and prediction results of 200 km data belonging to cluster 7 are drawn on the Turkey map. The maps of the prediction results drawn separately for each earthquake event in Figures 4.8, 4.9, and 4.10 are shared below.

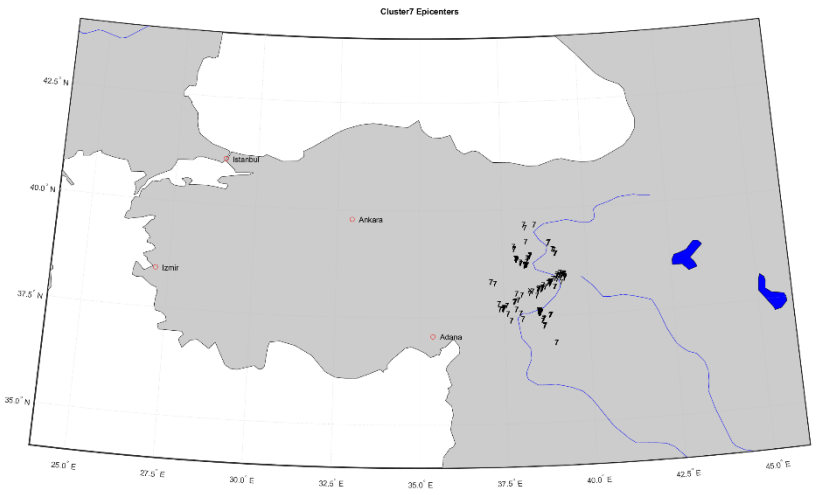


Figure 4.8: The epicenter coordinates of earthquake events belonging to 2133 cluster 7 clusters with 200 km station distance.

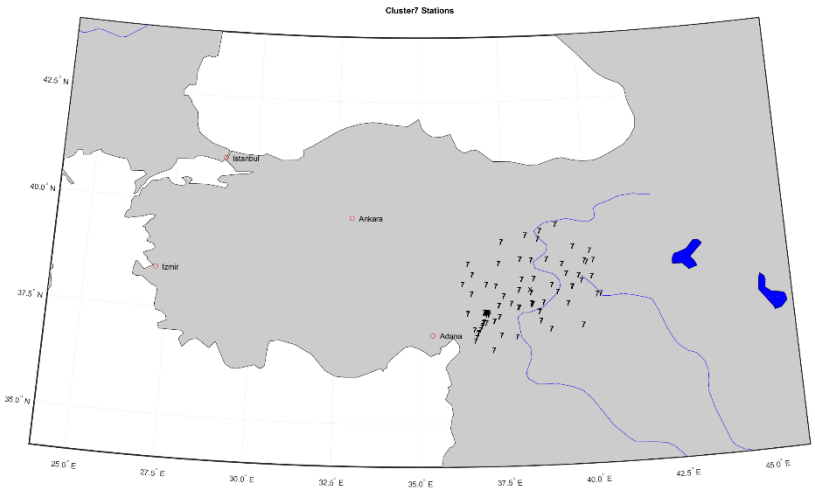


Figure 4.9: Station coordinates of earthquake events belonging to 2133 cluster 7 with 200 km station distance.

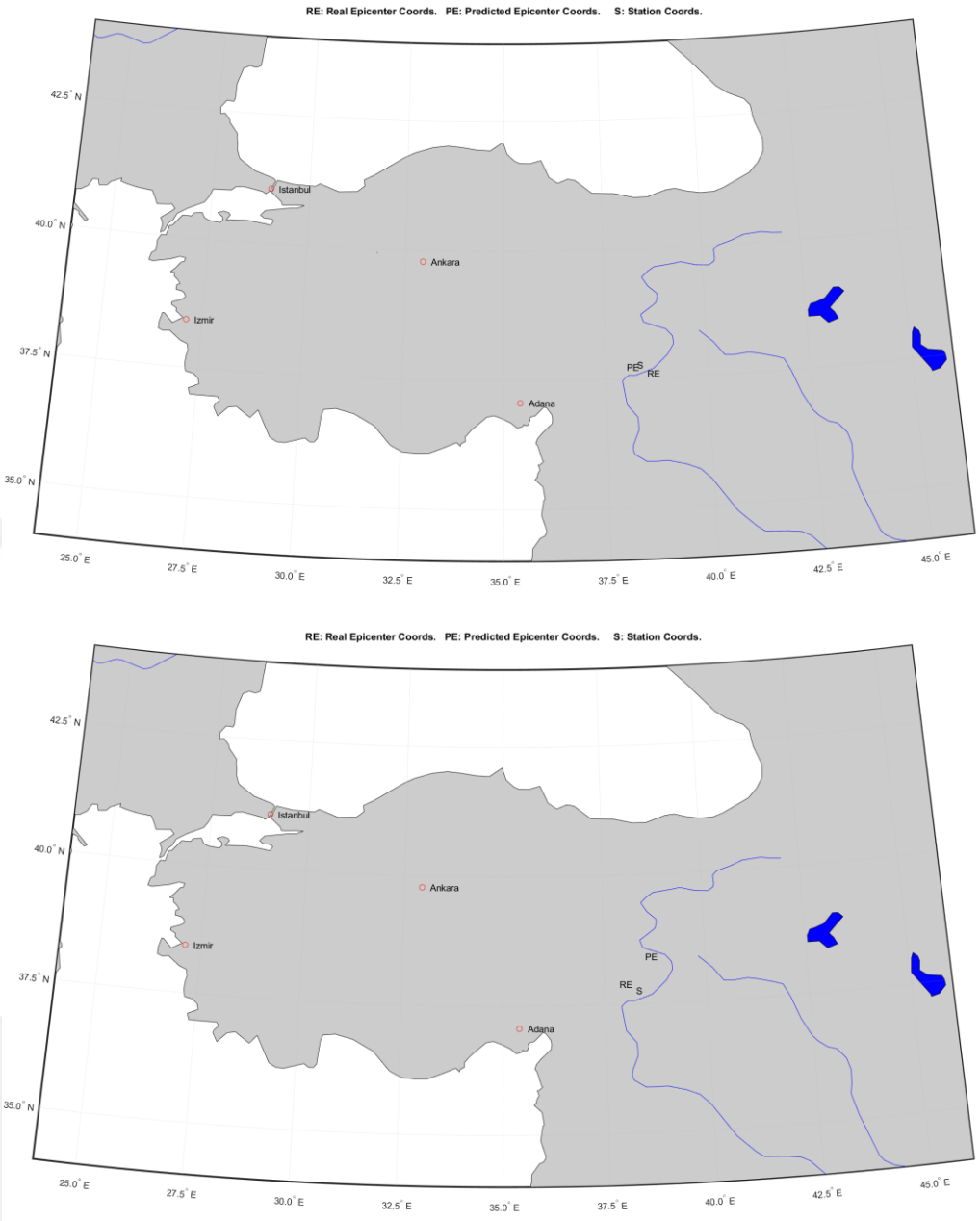


Figure 4.10: In the training made with the earthquake events belonging to 2133 cluster 7 with a distance of 200 km, the sample of the validation set predicted epicenter coordinates (PE), real epicenter coordinates (RE) and station coordinates (S).

Figure 4.10 summarizes the prediction results. In the left figure, the distance between the ground truth and the predicted epicenter is 47 km, while in the right figure, the distance between the ground truth and the predicted epicenter is 141 km. In order to find an answer to the question of whether it is related to the depth of the earthquake

event between the predicted epicenter distances with ground truth, ground truth, and depth histogram analysis were performed with the validation set of 2133 data belonging to the proposed cluster 7, this histogram is given in Figure 4.11.

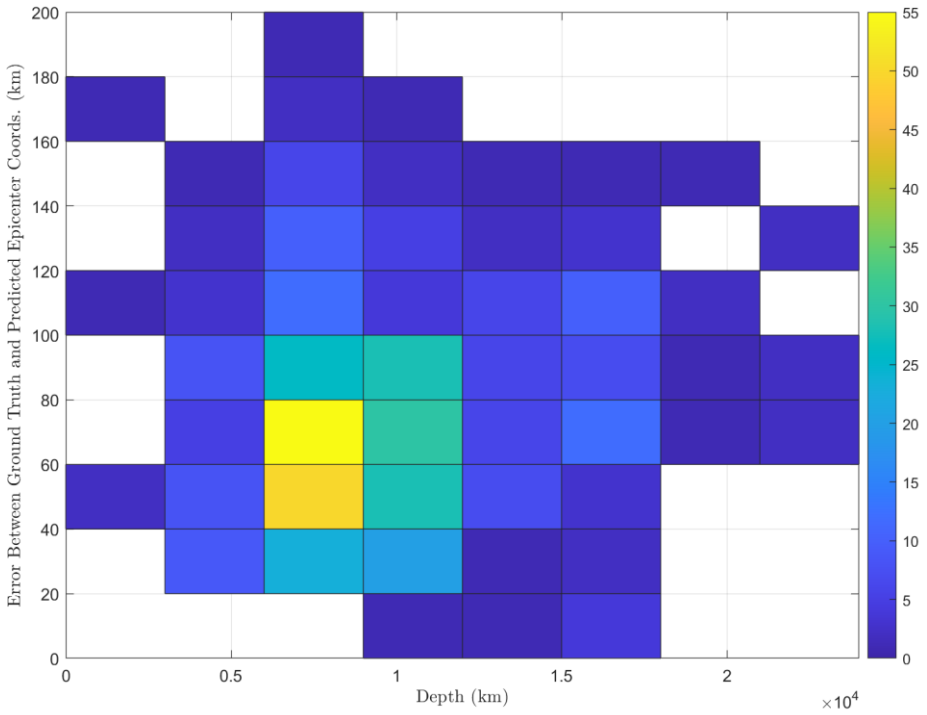


Figure 4.11: Earthquake depth relationship between ground truth of validation set and predicted epicenter coordinates error in training with 2133 cluster 7 earthquake events with 200 km station distance.

As shown in Figure 4.11, for earthquakes having a focal depth less than 15000 km, the epicenter prediction error is lower. There are 427 data in the validation set analyzed. There are only 53 data with depths greater than 15000 km in these data. It is predicted that removing 53 data from the already small dataset will not change the performance. It has been observed that at depths less than 15000 km, the error value is up to 100 km. This error value is not sufficient for successful performance, so predicting the epicenter coordinate does not have a direct relationship with the depth.

Experiments made in this section are intended to show that epicenter coordinate is related to station distance. In order to reach this conclusion, the experiments carried out before and after the earthquake data are explained. It was decided to change the

network model with 2133 earthquake events belonging to cluster 7, which includes the epicenter at a distance of 200 km from the station. The analysis for the model proposal is explained in the next section.

4.2.2 Primary and Secondary Waves with CNN

While continuing with the maximum spectrogram data of the earthquake event, P and S wave markings were made for 325 of 2133 data belonging to cluster 7 at a 200 km station distance. The time between these marked waves was added after the first FC layer with the station coordinate and altitude values as the second split input.

It is predicted that P and S waves will positively affect performance. In order to increase the performance success, instead of taking 5 seconds from the maximum energy spectrogram, new spectrograms were created by taking 2 seconds before and 3 seconds after the marked P wave, and S wave start times. The spectrograms of the S wave and the training result with a localization error of 827 km are given in Figure 4.12, with 148 initially marked data belonging to cluster 7 at a distance of 200 km from the station.

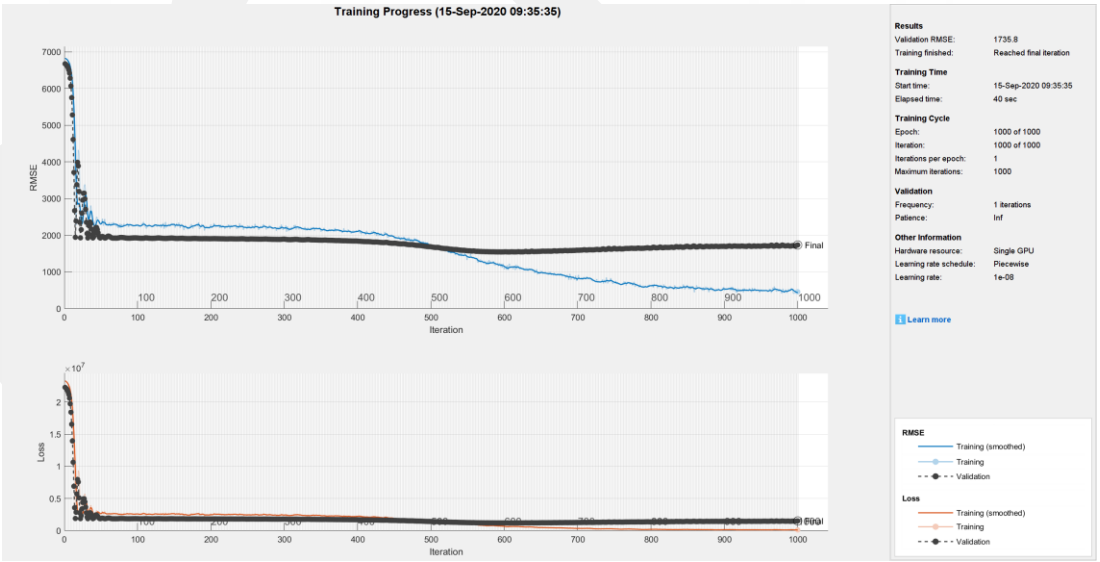


Figure 4.12: S wave spectrogram training result with 148 P and S wave marked data of cluster 7 at 200 km station distance.

Performance could not be improved with fine-tuning processes. The model seen in Figure 4.12 is overfitting. This showed that it learned the features in the training

set used but could not generalize it to other data. Since the P wave shows the earthquake's starting point, it was not used in the experiments, considering that it would not provide sufficient interpretation in the prediction of the epicenter coordinate.

4.2.3 Partial Transfer Learning

In addition to the model proposed in the Relation on a Single Station section, the time difference of P and S waves mentioned in the P and S waves with the CNN section gives information about how far the earthquake is from the epicenter. The aim of the architecture proposed here is to find the P and S phase moments of earthquakes. Therefore, this architecture will be built on top of the previous model through transfer learning. Each architecture to be produced includes structures transferred from the previous one. Therefore, each model constitutes a phase of the final, intended model described in the next section.

In addition to the time difference input between the P and S phases proposed in the previous section, here, the P and S phase onset moments are also predicted to fuse with the maximum energy spectrogram data in the fully connected layer. The raw spectrogram data from the model initially proposed in this section, obtained from the model trained in the previous stage and transferred to this architecture, is fed to the convolutional encoder.

In order to train this architecture, P and S wave markings are used. Therefore, the number of data that this architecture can be trained on is limited. P and S waves were marked for 325 out of 2133 pieces of data belonging to cluster 7 at the recommended 200 km station distance in the previous stage. In order to overcome this data constraint, the transfer learning method was used. The model output to be obtained at this stage is still the same. The aim is to predict the epicenter and depth of the earthquake event in question (more successfully than in the previous stage). Besides this, another objective is to reveal and develop a general encoder structure that we train using all available data. This encoder structure will continue to be transferred to future models.

Before the raw spectrogram data and the second inputs are fused in the fully connected layer, transfer learning is suggested with the model calculated with the localization error of 80 km, with the best results. In addition, the partial transfer is proposed for 1024x1027 weights except for the P and S phase moments added at this stage in the second fully connected layer.

While applying the partial transfer method, attention was paid to training the layers without freezing. The layer to be partially transferred was initialized, then the columns to be transferred were transferred before training. It has been observed that the Matlab [88] platform used in the study, the deep learning toolbox, performs the initialize function in the training phase. In order to prevent this, custom weight and bias initializer functions have been written for the layer where the partial transfer method will be applied.

The result of the train made with this proposed method is given in Figure 4.13. No partial transfer effect was observed in this train, whose localization error was calculated as 1141 km. In order to control the partial transfer weights, learnable parameters in columns 1028 and 1029 were initialized as 0. However, it was observed that the weights changed in the train made.

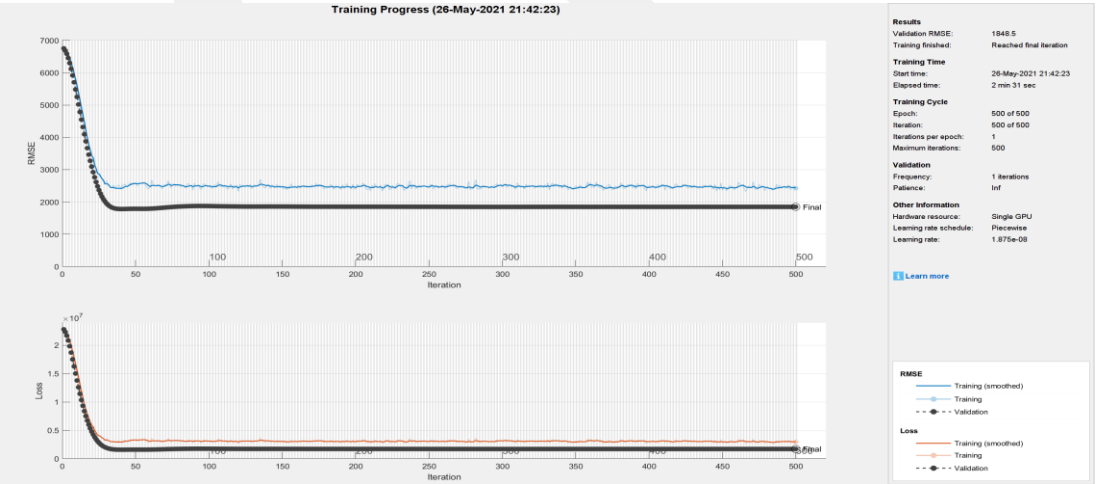


Figure 4.13: Partial transfer learning made training result with 325 P and S wave marked data.

Regularization can be done in order to prevent overfitting seen in Figure 4.13. However, high weights have been reduced for regularization in the proposed

architecture, and dropout layers have been used. Localization error could not be reduced by fine-tuning. The performance could not be made more successful with the transfer learning effect. However, since the number of data is limited, a different architecture proposal was made by continuing with transfer learning.

4.2.4 CNN + LSTM with Spectrogram

Earthquake events are time series signals with different lengths. In the proposed study, the accelerometer data belonging to the earthquakes were transferred to the time-frequency representation and turned into a spectrogram. Predicting the intended epicenter and depth with accelerometer data is a time series problem. Since earthquakes have different lengths from each other, it is a more practical problem for RNN. Sufficient improvement could not be made to the architecture proposed in the previous section, and a new architectural proposal was made in this section.

The maximum energy spectrogram data belonging to 325 data and the encoder stage fusing with the station coordinate, altitude, and P-S phase difference in the fully connected layer, the result of the training carried out with 2133 data belonging to Cluster 7 with 200 km station distance, was transferred with the 80 km localization error model. After that, P and S phase moments were added next to the Second inputs. In the architecture proposed here, six different inputs fused after the formation of high-level features are fed to LSTM first.

Localization error approached 72 km in the proposed architecture with 325 data. The best-performing training result is shared in Figure 4.14.

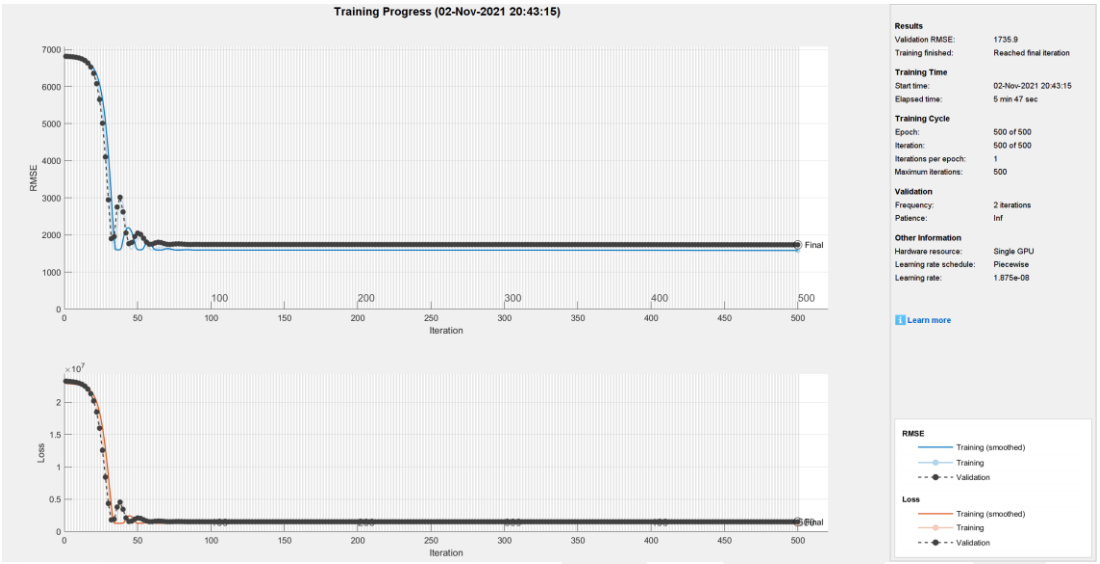
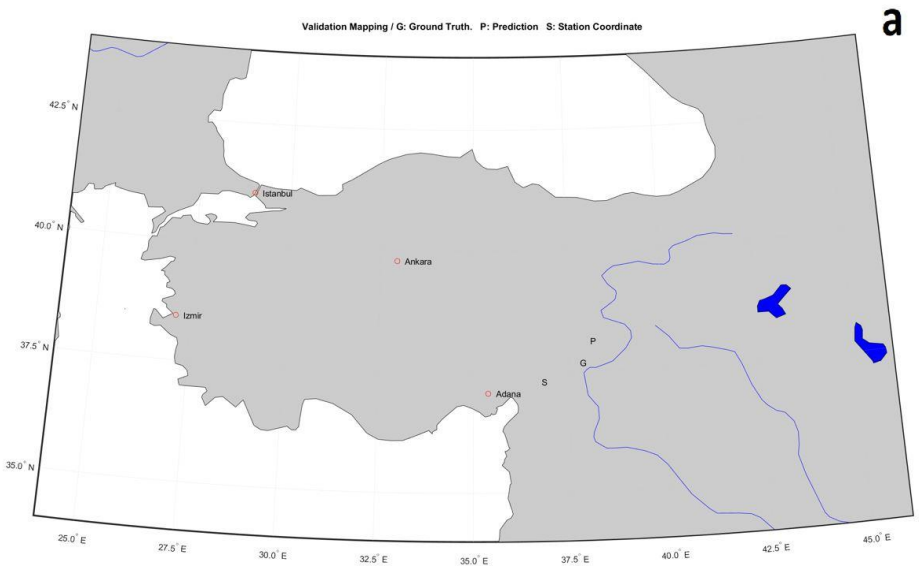
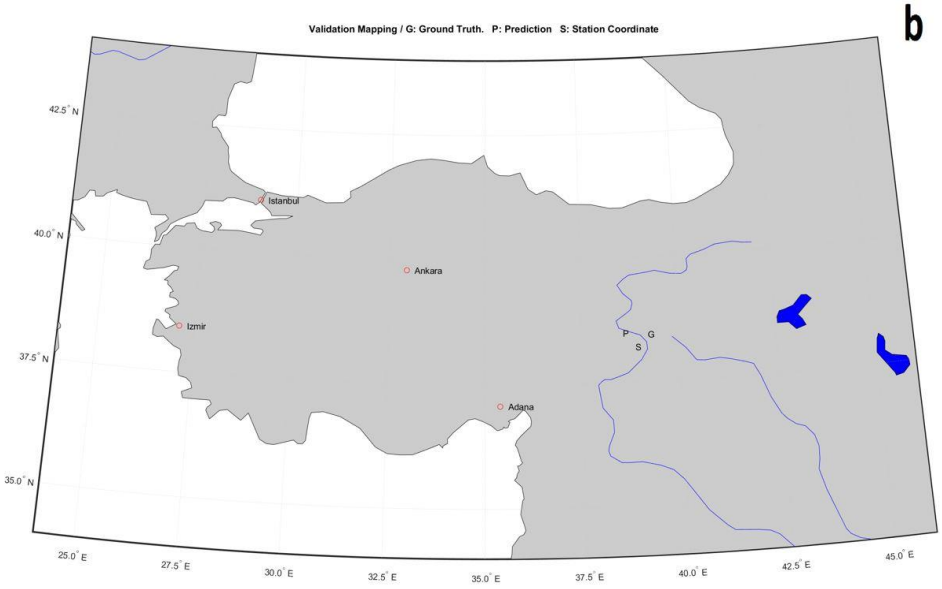


Figure 4.14: CNN + LSTM architecture with 325 data training result.

Very different results were obtained with CNN + LSTM before the partial transfer in the epicenter prediction architectural proposal for earthquake events with time series problems. In order to verify the problem approach, the station coordinate, ground truth, and predicted epicenter values were plotted on the Turkey map for the localization error 72 km results with the CNN + LSTM architecture suggested in this section.


a
Validation Mapping / G: Ground Truth. P: Prediction S: Station Coordinate
42.5° N
40.0° N
37.5° N
35.0° N
Istanbul
Ankara
Izmir
Adana
P
G
S
25.0° E
27.5° E
30.0° E
32.5° E
35.0° E
37.5° E
40.0° E
42.5° E
45.0° E


b
Validation Mapping / G: Ground Truth. P: Prediction S: Station Coordinate
42.5° N
40.0° N
37.5° N
35.0° N
Istanbul
Ankara
Izmir
Adana
P
G
S
25.0° E
27.5° E
30.0° E
32.5° E
35.0° E
37.5° E
40.0° E
42.5° E
45.0° E

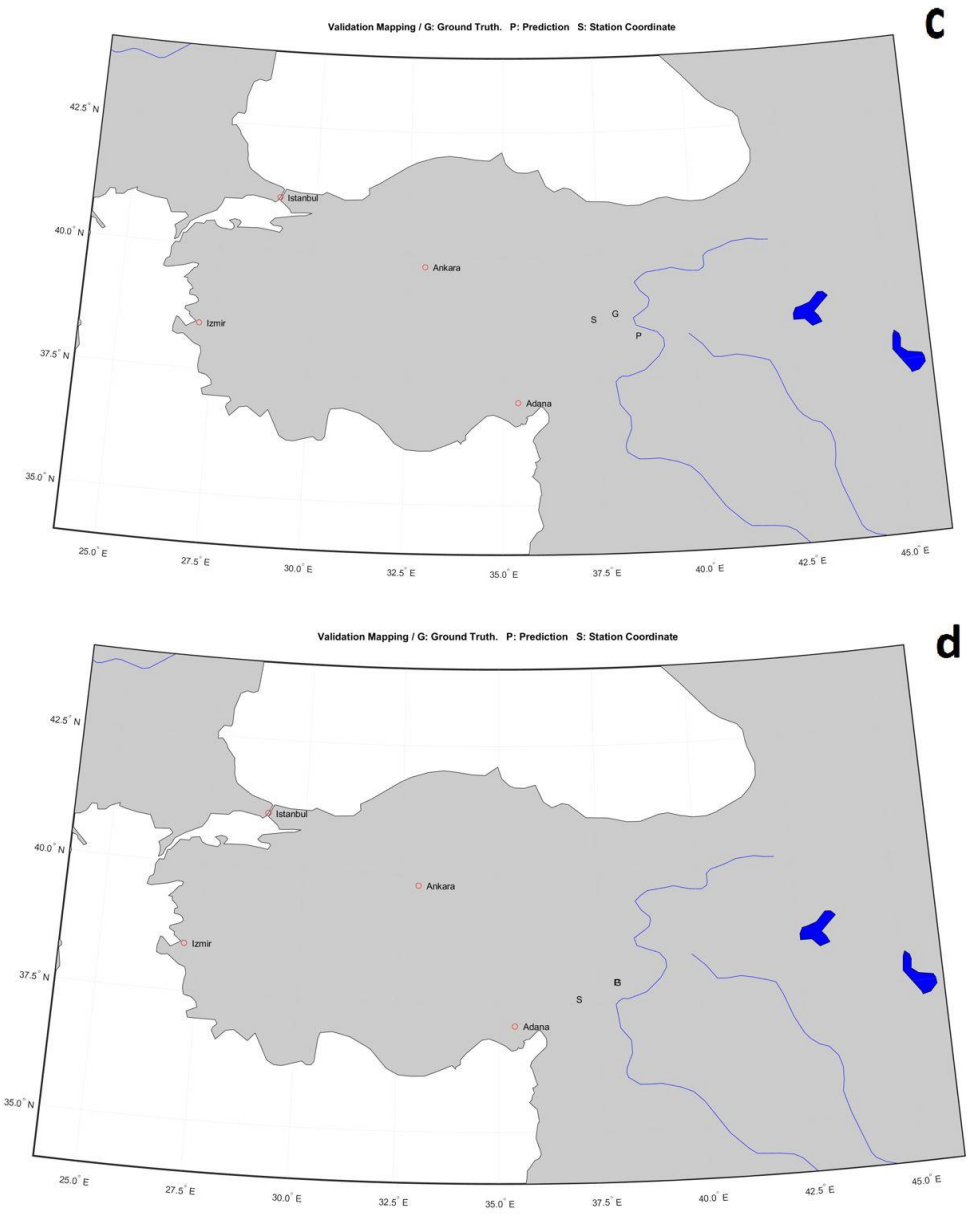


Figure 4.15: CNN + LSTM architect's best training result of the validation set on the map of Turkey predicted epicenter (P), ground truth (G) and station coordinates (S) notation.

In Figure 4.15, the verification of the place predicted by the data of 65 validation sets of training made with 325 data is given. Here, the localization error result is 72 km, and the training result seen in Figure 4.14 has answered whether the proposed architecture overfits the station coordinate.

In order to understand the relationship between ground truth and predicted value, station altitude histograms where the difference between ground truth and the predicted value is less than or more than 50 km were created.

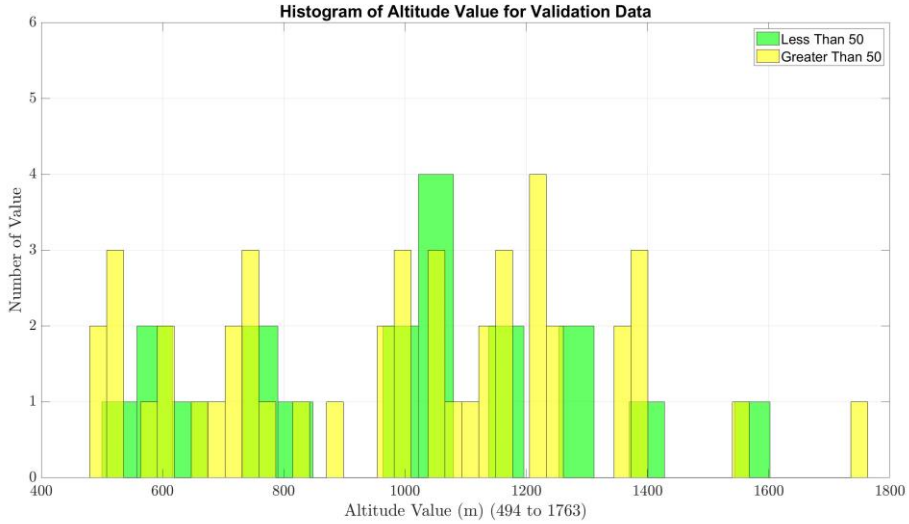


Figure 4.16: The altitude relationship between the ground truth and predicted value of the validation set belonging to the best training result of the CNN+ LSTM architect and earthquake events where the value is less than or more than 50 km.

As seen in Figure 4.16, there is no distinction between earthquake events where the difference between ground truth and the predicted value is less than or greater than 50 km. Error values are distributed in such a way that they are at each altitude value.

In order to prevent the overfitting seen in Figure 4.14, the location of the LSTM layer in the proposed architecture has been moved to the last fully connected layer. The performance of the trains realized here is given in Figure 4.17.

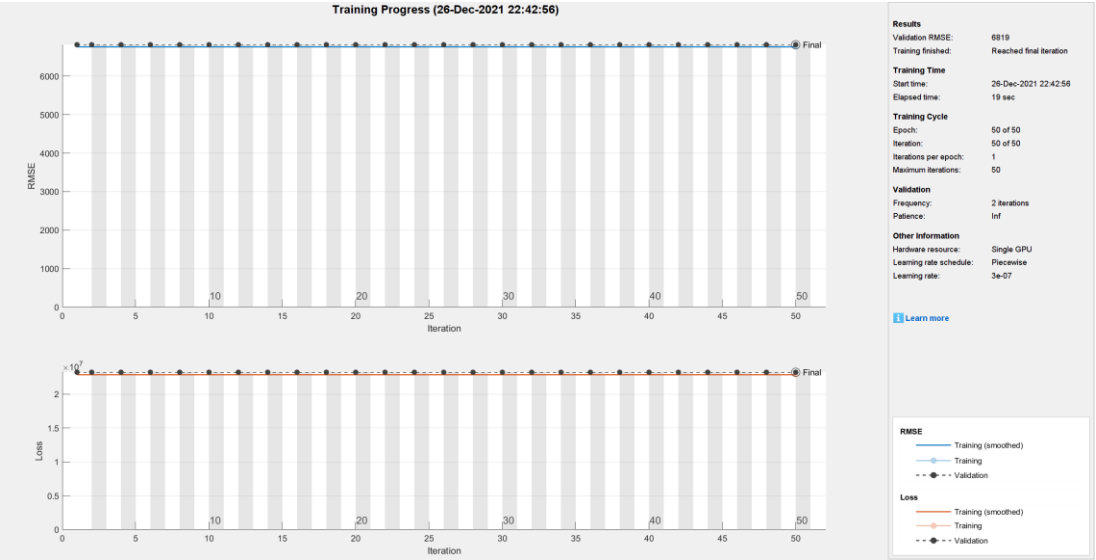


Figure 4.17: Training result with LSTM layer attached architecture after fully connected layers.

It was deemed appropriate to feed the LSTM layer after all the inputs suggested in the model were combined and after gaining meaning. However, the calculated localization error value of the training result shared in Figure 4.17 is 5823 km. It has been observed that although there is a learning drop rate in training, it does not drop, and learning does not occur. Since the intended result was not achieved with this proposed architecture, the LSTM layer was pulled back after the first fully connected layer, just after the inputs were fused.

In the proposed new model, analyses were made with transfer learning. While the localization error is 80 km, no transfer learning is recommended. When transfer learning is not made to convolutional blocks but only to the fully connected layer where the inputs are fused, the localization error is calculated as 74 km. However, when transfer learning was applied only to the last fully connected layer, the localization error encountered was 64 km.

In the proposed model, the entire earthquake event is used as a sequential input instead of the maximum energy spectrogram of the earthquake event. The lengths of earthquake events are different from each other. The spectrograms of each earthquake event have been adjusted to the same size in order to sequentially turn the input and feed it into the proposed model. It was observed that the average duration of 325 earthquake events in which P and S phases were marked was 14 seconds. The

spectrogram lengths of 325 earthquake events were fixed at 16 seconds. While stabilizing, earthquakes lasting longer than 16 seconds were cropped, and short-lasting earthquakes were zero padded.

It has been observed that the localization error is 64 km; that is, when the train is made with the model parameters that give the best results, an out of GPU memory error is received due to the amount of input size. For this reason, the model hyper-parameters were fine-tuned. As a result of the trainings, it was observed that the lowest localization error calculated between the predicted epicenter coordinate and ground truth values was 5704 km.

In summarize, the model structure and the corresponding parameters used in the different datasets are provided in Table 5.

Table 5: Details of the prediction models used for different structures.

Network Architecture	**Hyper-parameters**	**RMSE**	**Distance (km)**	**F1-Score**
Conv Block 4 Fc {1024}, ReLU Cat Layer Fc {1024}, ReLU Fc {3} Regression *(No. of Events: 28360)*	Optimizer: SGDM Learning Rate: 4e-07 Max. Epoch: 200	3.60	273	0.511
Conv Block 4 Fc {1024}, ReLU Cat Layer Fc {1024}, ReLU Fc {3} Regression *(No. of Events: 2495, Cluster 9)*	Optimizer: SGDM Learning Rate: 4e-07 Max. Epoch: 2000	1.17	99.9727	0.65

Table 5 Cont.

Conv Block 4 Fc {1024}, ReLU Table 5 Cont. Cat Layer Fc {1024}, ReLU Fc {3} Regression *(No. of Events: 2133, Cluster 7)*	Optimizer: SGDM Learning Rate: 4e-07 Max. Epoch: 1000	0.90	80.3699	0.76
Conv Block 4 Fc {1024}, ReLU Cat Layer Fc {1024}, ReLU Fc {3} Regression *(No. of Events: 325, marked P and S waves) + Partial Transfer*	Optimizer: SGDM Learning Rate: 3e-07 Max. Epoch: 500	1848.44	1141.2626	0.62
Conv Block 4 Fc {1024}, Cat Layer LSTM {1029} Fc {1024}, Fc {3} Regression *(No. of Events: 325, marked P and S waves) + Partial Transfer*	Optimizer: SGDM Learning Rate: 3e-07 Max. Epoch: 500	0.60	63	0.78
Total run time	2776 GPU hours			

In addition, a total of 686 separate trainings were made in this study with different neural networks and different parameters to optimize the results. These trainings took a total of 2776 GPU hours.

CHAPTER V

CONCLUSIONS

In this thesis, for earthquake epicenter coordinate prediction, first of all, the seismic dataset of Turkey, which AFAD publicly publishes, was collected. A spectrogram-based false color representation of earthquake accelerometer records has been studied. The proposed indication is not only convenient for human analysis but also suitable for earthquake signal processing in convolutional networks.

It has been observed that the accelerometer data of earthquake events that took place at close epicenters and recorded at stations close to each other created similar two-dimensional false-color pictures in this representation. This is a strong clue that this relationship can be learned if earthquake events in the same cluster (in other words, belonging to close epicenters) are fed into the convolutional network.

A convolutional network is trained using hundreds of thousands of 5-second false color spectrograms obtained from more than 40 thousand earthquake events in the dataset. This professional network is aimed to predict the epicenter and depth information of an earthquake event. Experiments have been done on a system that can calculate the epicenter of an earthquake using single station data using convolutional neural networks, and the results confirming the hypothesis have been observed.

First, experiments were trained on the network shown in Figure 3.5. Two different inputs to this network structure are fed from two different layers. In the first layer, the spectrogram data with the highest energy of any earthquake was fed, while in the first fully connected layer, the station location, altitude and the arrival time difference between the P and S phases were fed. As the output, the epicenter coordinate of the earthquake event and the earthquake depth were learned with a regression layer.

In the experiments, networks were trained separately for each different K-medoid cluster of 2015. In each experiment, 70% of the selected subset of the dataset was reserved for learning, 15% for hyperparameter optimization (validation), and 15% for testing. As a result of the cross-validation experiments, it was seen that the epicenter coordinate could be calculated with an average error of around 80 km, and

the depth with an average error of around 800 m. As it has been observed that the spectrogram based false color representation facilitates human examination of accelerometer data, it is concluded that this representation is suitable for convolutional networks. It has also been observed that accelerometer data is associated with the epicenter. Thus, accelerogram records will benefit EEW systems by using it to calculate epicenter coordinates. In the continuation of the thesis, the epicenter coordinate 64 km localization error was observed as the closest result with the studies carried out on the architectures working with holistic earthquake event instead of fixed spectrograms using RNN.

The studies are not sufficient to make sense of the earthquake signal and to predict the epicenter location, but they are effective. When the obtained result is compared with the studies in the literature, it is seen that the results are approximated [52]. There are studies in the literature with better results using similar techniques in this study [89]. The attention mechanism will be used in the rest of the study to improve the results.

It has been observed that the inputs in the first steps cannot be adequately represented in long sequences with RNN, so using the encoder decoder model of LSTM is not very effective. In the continuation of the thesis, it is foreseen to use the transformer and attention mechanism, which replaces the recurrence models. Thus, no matter how long the number of steps is, all hidden states from the encoder are used and the performance of the model increases. It has been observed that the Attention mechanism improves the performance on seismic data. With the attention-based mechanism, a network is proposed where 8 seismic indicators are used as inputs and earthquake or not earthquake information is given as output [90]. When the proposed attention-based LSTM network and LSTM network accuracy values are compared, Attention-Based LSTM architecture is observed to be 4% more successful. In addition, in a study where hypocenter and origin time were calculated as outputs, the attention mechanism was again preferred [91]. In the study [92], which also included P phase and S phase as output, it was observed that the performance increased by using earthquake records with station and epicenter distances up to 300 km and attention

mechanism with noises. In the light of similar studies shared, the studies will be continued by adding the attention mechanism.

REFERENCES

[1] Goldsmith, Mike, 'Seismic waves', Waves: A Very Short Introduction, Very Short Introductions (Oxford, 2018; online edn, Oxford Academic, 22 Nov. 2018), https://doi.org/10.1093/actrade/9780198803782.003.0004

[2] Archuleta, Ralph, Fletcher, Jesse and Shakal, Ahmed, 2014. Strong Motion Recording in the United States. *AGU Fall Meeting Abstracts*, [online] Available at: https://ui.adsabs.harvard.edu/abs/2014AGUFM.S13F..03A [Accessed 28 August 2022].

[3] Matic, Aleksandar, Osmani, Venet, & Mayora, Oscar (2012). Speech activity detection using accelerometer. *2012 Annual International Conference Of The IEEE Engineering In Medicine And Biology Society*. doi: 10.1109/embc.2012.6346377

[4] Başkanlığı, T., 2022. Depremin Büyüklüğü Ve Şiddeti Aynı Kavramlar Mıdır?. [online] Afad.gov.tr. Available at: <https://www.afad.gov.tr/depremin-buyuklugu-ve-siddeti-ayni-kavramlar-midir > [Accessed 15 October 2019].

[5] Stein, Seth, & Wysession, Michael (2009). *An introduction to seismology, earthquakes, and earth structure*. Blackwell Puslishing. 2003

[6] Havskov, Jens, Bormann, Peter and Schweitzer, Johannes, 2011. Seismic source location.

[7] Mayrhofer, Rene, & Gellersen, Hans (2009). Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices. *IEEE Transactions On Mobile Computing*, 8(6), 792-806. doi: 10.1109/tmc.2009.51

[8] Alimoradi Arzhang, Beck James L. (2015), "Machine-Learning Methods for Earthquake Ground Motion Analysis and Simulation", *Journal of Engineering Mechanics*, Volume 141, Issue 4.

[9] Housner George W., Jennings Paul. C. (1964), Generation of artificial earthquakes, *Proc. ASCE*, Volume 90, pp. 113-150.

[10] Jennings Paul C., Housner George W., Tsai Chia-Nan (1969), "Simulated earthquake motions for design purposes", Proc. of the Fourth World Conference on Earthquake Engineering, pp. 145-160.

[11] Saragoni G. Rodolfo, Hart Gary (1973), Simulation of artificial earthquakes, *Earthquake Engineering & Structural Dynamics*, Volume 2, pp. 249-267.

[12] Zeybek, Abdulhakim, Madabhushi, Gopal, & Pelecanos, Loizos (2020). Seismic response of partially saturated soils beneath shallow foundations under sequential ground motions. *Bulletin Of Earthquake Engineering*, 18(5), 1987-2002. doi: 10.1007/s10518-020-00792-5.

[13] Yang, Xing-She and Deb, Suash, 2015. Cuckoo Search for Optimization and Computational Intelligence. *Encyclopedia of Information Science and Technology*, Third Edition, pp.133-142.

[14] Karaboga, Dervis and Basturk, Bahriye, 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), pp.459-471.

[15] Deep Kusum, Yadav, Anupam, Kumar, Improving local and regional earthquake locations using an advance inversion Technique: Particle swarm optimization. *World Journal of Modelling and Simulation*, 8, 135–141 (2012)

[16] Eberhart, Russell and Kennedy, James, n.d. A new optimizer using particle swarm theory. MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science.

[17] Medium. 2020. The inspiration of an Ant colony optimization. [online] Available at: <https://towardsdatascience.com/the-inspiration-of-an-ant-colony-optimization-f377568ea03f> [Accessed 26 April 2020].

[18] HOU, Jing-wei, KONG, Yun-geng and SUN, Jiu-lin, 2013. Application of ant colony algorithm for parameter optimization of water demand prediction model. *Journal of Computer Applications*, 32(10), pp.2952-2955.

[19] Medium. 2017. Introduction to Genetic Algorithms — Including Example Code. [online] Available at: <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3> [Accessed 8 July 2017].

[20] Kao, Ching-Yun, Chung, Jen-Kuang & Yeh, Yeong-Tein (2010). A comparative study of the least squares method and the genetic algorithm in deducing peak ground acceleration attenuation relationships. *TAO: Terrestrial, Atmospheric and Oceanic Sciences*, 21(6), 8.

[21] Nicknam, Aahmad, Abbasnia, Reza, Bozorgnasab, Mohzen, & Eslamian, Yasser (2010). Synthesizing strong motion using empirical Green's function and genetic algorithm approach. *Journal of earthquake engineering*, 14(4), 512-526.

[22] Akhoondzadeh, Mehdi (2013). Genetic algorithm for TEC seismo-ionospheric anomalies detection around the time of the Solomon (Mw= 8.0) earthquake of 06 February 2013. *Advances in Space Research*, 52(4), 581-590.

[23] Yang, Guangyu Robert and Wang, Xiao-Jing, 2020. Artificial Neural Networks for Neuroscientists: A Primer. *Neuron*, 107(6), pp.1048-1070.

[24] Perol, Thibaut, Gharbi, Michael and Denolle, Marine, 2018. Convolutional neural network for earthquake detection and location. Science Advances, 4(2).

[25] Friedman, Jerome H. (2001) Greedy function approximation: a gradient boosting machine, *Annals of statistics*, pp. 1189–1232.

[26] LeCun, Yann, Bengio, Yoshua and Hinton, Geoffrey (2015). Deep learning, *Nature*, 521(7553), pp. 436–444.

[27] Asim, Khawaja, Martínez-Álvarez, Francisco, Basit, Abdul and Iqbal, Turab. Earthquake magnitude prediction in Hindukush region using machine learning techniques, *Natural Hazards*, vol. 85, pp. 471-486, 2017.

[28] Asim, Khawaja, Idris, Adnan, Iqbal, Turab, and Martínez-Álvarez, Francisco. Seismic indicators based earthquake predictor system using Genetic Programming and AdaBoost classification, *Soil Dynamics and Earthquake Engineering*, vol. 111, pp. 1–7, 2018.

[29] Asim, Khawaja, Iqbal, Turab, Idris, Adnan and Martínez-Álvarez, Francisco. Earthquake prediction model using support vector regressor and hybrid neural networks, *PLOS ONE*, vol. 13, 2018.

[30] Pao, Yohhan. "Adaptive pattern recognition and neural networks." (1989).

[31] Liaw, Andy, and Matthew Wiener. Classification and regression by randomForest, R news 2.3 (2002): 18-22.

[32] Saffari, Amir, et al. Online multi-class LPBoost. 2010 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Ieee, 2010.

[33] Suganuma, Masanori, Shinichi Shirakawa, and Tomoharu Nagao. A genetic programming approach to designing convolutional neural network architectures. *Proceedings of the Genetic and Evolutionary Computation Conference*. 2017.

[34] Hastie, Trevor, et al. Multi-class adaboost. *Statistics and its Interface 2.3* (2009): 349-360.

[35] Hung, Shih-Lin, Wen, Chih-Min and Tu, Tran Thanh (2003) A neural network approach for structural identification and diagnosis of a building from seismic response data. *Earthquake Engineering and Structural Dynamics* 32(2): 187–206.

[36] Panakkat, Ashif and Adeli, Hojjat. Neural network models for earthquake magnitude prediction using multiple seismicity indicators. *International Journal of Neural Systems*, vol. 17(1), pp. 13–33, 2007.

[37] Graves, Alex. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013). https://doi.org/10.48550/arXiv.1308.0850

[38] Adeli, Hojjat and Panakkat, Ashif, A probabilistic neural network for earthquake magnitude prediction, *Neural Networks*, vol. 22(7), pp. 1018–1024, 2009.

[39] Specht, Donald F. Probabilistic neural networks. *Neural networks* 3.1 (1990): 109-118.

[40] Bowen, Hayden, et al. Lateral spreading in the Canterbury earthquakes–Observations and empirical prediction methods. *Proceedings, 15th World Conference on Earthquake Engineering*. 2012.

[41] Reyes, Jorge, and Víctor H. Cárdenas. A Chilean seismic regionalization through a Kohonen neural network. *Neural Computing and Applications* 19.7 (2010): 1081-1087.

[42] Bodri, Bertalan. A neural-network model for earthquake occurrence. *Journal of Geodynamics* 32.3 (2001): 289-310.

[43] Huang, Jipan, et al. Large earthquake magnitude prediction in Taiwan based on deep learning neural network. *Neural Network World* 28.2 (2018): 149-160.

[44] DeVries, Phoebe, Viégas, Fernanda, Wattenberg, Martin and Meade, Brendan. Deep learning of aftershock patterns following large earthquakes, *Nature,* vol. 560, pp. 632–634, 2018.

[45] Li, Shengrong, et al. Seismic fault detection using an encoder–decoder convolutional neural network with a small training set. *Journal of Geophysics and Engineering* 16.1 (2019): 175-189.

[46] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[47] Kumar, Arun, Jeffrey Naughton, and Jignesh M. Patel. Learning generalized linear models over normalized data. *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. 2015.

[48] Asencio-Cortés, Gualberto, Morales-Esteban, Antonio and Martínez-Álvarez, Francisco. Earthquake Prediction in California Using Regression Algorithms and Cloud-based Big Data Infrastructure, *Computers & Geosciences*, vol. 115, pp. 198-210, 2018.

[49] Orr, Mark JL. Introduction to radial basis function networks. (1996).

[50] Jang, Jyh-Shing Roger. ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics* 23.3 (1993): 665-685.

[51] Zamani, Ahmad, Mohammad Reza Sorbi, and Ali Akbar Safavi. Application of neural network and ANFIS model for earthquake occurrence in Iran. *Earth Science Informatics* 6.2 (2013): 71-85.

[52] Panakkat, Ashif, and Hojjat Adeli. Recurrent neural network for approximate earthquake time and location prediction using multiple seismicity indicators. *Computer-Aided Civil and Infrastructure Engineering* 24.4 (2009): 280-292.

[53] Bullen, Keith Edward, Bullen, Keith A. and Bolt, Bruce A. (1985). An introduction to the theory of seismology. Cambridge university press.

[54] Woollam, Jack, et al. Convolutional neural network for seismic phase classification, performance demonstration over a local seismic network. Seismological Research Letters 90.2A (2019): 491-502.

[55] O'Shea, Keiron, and Ryan Nash. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458 (2015).

[56] Kuyuk, H. Serdar, and Ohno Susumu. Real-Time Classification of Earthquake using Deep Learning. Procedia Computer Science 140 (2018): 298-305.

[57] Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney. LSTM neural networks for language modeling. Thirteenth annual conference of the international speech communication association. 2012.

[58] Mousavi, S. Mostafa, et al. CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection. Scientific reports 9.1 (2019): 1-14.

[59] Kusumo, Budiarianto Suryo, et al. Recognizing Human Activities and Earthquake Vibration from Smartphone Accelerometers using LSTM Algorithm. *2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*. IEEE, 2018.

[60] Kramer, Steven L. (1996). Geotechnical earthquake engineering.

[61] n.d. [online] Available at: https://deprem.afad.gov.tr/ [Accessed 12 November 2019].

[62] Koeri.boun.edu.tr. n.d. B.Ü. KRDAE Bölgesel Deprem-Tsunami İzleme ve Değerlendirme Merkezi. [online] Available at: <http://www.koeri.boun.edu.tr/sismo/2/tr/> [Accessed 3 February 2022].

[63] Acikders.ankara.edu.tr. n.d. [online] Available at: <https://acikders.ankara.edu.tr/pluginfile.php/119309/mod_resource/content/0/Ders_04_Sismik_Dalgalar_%28Cisim_Dalgalari_1%29.pdf> [Accessed 7 May 2022].

[64] Jin, Xin and Han, Jiawei, 2017. K-Medoids Clustering. *Encyclopedia of Machine Learning and Data Mining*, pp.697-700.

[65] Cikis, Melis, Tileyoglu, Salih and Akagunduz, Erdem, 2020. Representing Earthquake Accelerogram Records for CNN Utilization. *2020 28th Signal Processing and Communications Applications Conference (SIU).*

[66] Kaufman, Leonard, Hopke, Philip and Rousseeuw, Peter, 1988. Using a Parallel Computer System for Statistical Resampling Methods. *Computational Statistics Quarterly*, 2, pp.129-141.

[67] Kaufman, Leonard, & Rousseeuw, Peter (2009). Finding groups in data: an introduction to cluster analysis. *John Wiley & Sons.*

[68] Park, Hae-Sang and Jun, Chi-Hyuck, 2009. A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications*, 36(2), pp.3336-3341.

[69] Choi, Jaewon and Kwon, Hyuk-Jun, 2015. The Information Filtering of Gene Network for Chronic Diseases: Social Network Perspective. *International Journal of Distributed Sensor Networks*, 11(9), p.736569.

[70] Bergoeing, Jean Pierre, 2015. Structural Geomorphology. Geomorphology of Central America, pp.1-12.

[71] Kovitvongsa Kathryn E., Lobel Phillip S. (2009) Convenient fish acousticdata collection in the digital age. In: Pollock NW (ed) Diving for Science 2009. Proceedings of the American Academy of Underwater Sciences 28th Symposium, Dauphin Island, AL, pp 43–57.

[72] Johnson, Paul A., Rouet-Leduc, Bertrand, Pyrak-Nolte, Laura, 2021. Laboratory earthquake forecasting: A machine learning competition. *Proceedings of the National Academy of Sciences*, 118(5).

[73] Spanos, Pol D. and Giaralis, Agathoklis 2007. Time–frequency representation of earthquake accelerograms and inelastic structural response records using the adaptive chirplet decomposition and empirical mode decomposition. *Soil Dynamics and Earthquake Engineering*, 27(7), pp.675-689.

[74] Nguyen, Anh Mai, Yosinski, Jason and Clune, Jeff, 2016. Understanding Innovation Engines: Automated Creativity and Improved Stochastic Optimization via Deep Learning. *Evolutionary Computation*, 24(3), pp.545-572.

[75] Lu, Wenkai and Li, Fangyu, 2013. Seismic spectral decomposition using deconvolutive short-time Fourier transform spectrogram. GEOPHYSICS, 78(2), pp.V43-V51.

[76] Basu, Mitra, 2002. Gaussian-based edge-detection methods-a survey. *IEEE Transactions on Systems, Man and Cybernetics*, Part C (Applications and Reviews), 32(3), pp.252-260.

[77] Ioffe, Sergey and Szegedy, Christian (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning, in Proceedings of Machine Learning Research* 37:448-456.

[78] Dietterich, Tom (1995). Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3), 326-327.

[79] Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, & Salakhutdinov, Ruslan (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929-1958.

[80] Medium. 2018. Illustrated Guide to LSTM's and GRU's: A step by step explanation. [online] Available at: <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21> [Accessed 24 September 2018].

[81] Houbaer, Maikel, 2022. Comparing multichannel mixed CNN-RNN to individual models for earthquake prediction.

[82] Bergstra, James and Bengio, Yoshua. Random search for hyper-parameter optimization. *J. of Machine Learning Research*, 13(1):281–305, Feb 2012.

[83] Bergstra, James, Bardenet, Remi, Bengio, Yoshua and Kegl, Balazs. Algorithms for hyper-parameter optimization. In advances in Neural Information Processing Systems 24, pages 2546–2554. *Curran Associates, Inc.*, 2011.

[84] Domhan, Tabias, Springenberg, Jost Tobias and Hutter, Frank. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. *In Proc. of the Twenty-Fourth Int. Joint Conf. on Artificial Intelligence, IJCAI*, pages 3460–3468, July 2015.

[85] Young, Steven R. et al. Optimizing deep learning hyper-parameters through an evolutionary algorithm. In Proc. of the Workshop on Machine Learning in High-Performance Computing Environments, *MLHPC '15*, pages 4:1–4:5, New York, NY, USA, 2015. ACM.

[86] Ruder, Sebastian, 2022. An overview of gradient descent optimization algorithms.

[87] Ribani, Ricardo and Marengoni, Mauricio, 2019. A Survey of Transfer Learning for Convolutional Neural Networks. 2019 32nd SIBGRAPI Conference on Graphics, *Patterns and Images Tutorials* (SIBGRAPI-T).

[88] MATLAB. (2019). version R2019b. Natick, Massachusetts: The MathWorks Inc.

[89] Ristea, Nicolae-Catalin and Radoi, Anamaria, 2022. Complex Neural Networks for Estimating Epicentral Distance, Depth, and Magnitude of Seismic Waves. *IEEE Geoscience and Remote Sensing Letters*, 19, pp.1-5.

[90] Banna, Md. Hasan Al, Ghosh, Tapotosh and Nahian Md. Jabel Al, 2021. Attention-Based Bi-Directional Long-Short Term Memory Network for Earthquake Prediction. IEEE Access, 9, pp.56589-56603.

[91] Chin, Tai-Lin, Chen, Kuan-Yu, Chen, Da-Yi and Wang, Te-Hsiu, 2022. An Attention-Based Hypocenter Estimator for Earthquake Localization. *IEEE Transactions on Geoscience and Remote Sensing*, 60, pp.1-10.

[92] Mousavi, S. Mostafa, Ellsworth, William, Zhu, Weigiang, Chuang, Lindsay and Beroza, Gregory, 2020. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature Communications*, 11(1).