REPUBLIC OF TURKEY
ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY

INSTITUTE OF GRADUATE STUDIES
DEPARTMENT OF CYBER SECURITY

HUMAN ACTIVITY RECOGNITION USING DEEP CONVOLUTIONAL NEURAL NETWORK

ELİF KEVSER TOPUZ
MASTER OF SCIENCE

ADANA 2021

REPUBLIC OF TURKEY

ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

DEPARTMENT OF CYBER SECURITY

HUMAN ACTIVITY RECOGNITION USING DEEP CONVOLUTIONAL NEURAL NETWORK

ELIF KEVSER TOPUZ

MASTER OF SCIENCE

SUPERVISOR

Asst. Prof. Dr. YASİN KAYA

ADANA 2021

ABSTRACT

HUMAN ACTIVITY RECOGNITION USING DEEP CONVOLUTIONAL NEURAL NETWORK

Elif Kevser TOPUZ
Department of Cyber Security
Supervisor: Asst. Prof. Dr. Yasin KAYA
January 2021, 52 pages

Human activity recognition problems involve the processes of recording and analyzing a person's daily life to identify people's behavioral patterns. Various methods such as Video-Based activity recognition and Sensor-Based activity recognition have been developed to collect activity data. Studies have been carried out by recording activities such as walking, sitting, running, jumping, climbing stairs, climbing stairs by means of depth sensors, wearable (portable) devices, and RGB cameras. Systems have been developed especially in the fields of health, the internet of things, smart cities, transportation, and security with activity recognition. Activity-sensitive approaches have been developed for security monitoring and threat detection through activity recognition. By monitoring activity-based anomalies, potential threats can be identified, and appropriate precautions can be taken. Data collection has become extremely easy because of accelerometer, gyroscope, magnetometer sensors found in almost any smart device. Deep learning methods such as ANN, CNN, RNN, and LSTM are used to classify the obtained data.

In our research, we mainly focused on sensor-based activity recognition. Classification tasks were done using 1-D Convolution Neural Network feeding the raw data from the UCI-HAR dataset using accelerometer and gyroscope data. Raw data filtered using Median Filter. We didn't apply any mathematical or statistical feature extraction methods to the data. For the experimental results, we implemented 6, 7, and 12 classes activity recognition, and achieved accuracies of 96.95%, 95.03%, and 93.08%, respectively.

Keywords: Deep Learning, CNN, UCI-HAR, Human Activity Recognition, Sensor Based Activity Recognition

ÖZET

DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK İNSAN AKTİVİTESİ TANIMA

Elif Kevser TOPUZ
Siber Güvelik Anabilim Dalı
Danışman: Dr. Öğr. Üyesi Yasin KAYA
Ocak 2021, 52 sayfa

İnsan aktivitesi tanıma problemleri, insanların davranış kalıplarının belirlenmesi için bir kişinin günlük yaşamını kaydetme ve analiz etme süreçlerini içerir. Aktivite verilerini toplamak için video tabanlı aktivite tanıma ve sensör temelli aktivite tanıma gibi çeşitli yöntemler geliştirilmiştir. Derinlik sensörleri, giyilebilir (taşınabilir) cihazlar ve RGB kameralar aracılığıyla insanların yürüme, oturma, koşma, zıplama, merdiven inme, merdiven çıkma gibi aktivitelerini kayıt altına alınarak çalışmalar yapılmıştır. Aktivite tanıma ile özellikle sağlık, nesnelerin interneti, akıllı şehirler, taşımacılık ve güvenlik alanlarında sistemler geliştirilmiştir. Etkinlik tanıma yoluyla güvenlik izleme ve tehdit algılama için faaliyete duyarlı yaklaşımlar geliştirilmiştir. Faaliyet bazlı anormallikler izlenerek potansiyel tehditler tespit edilebilir ve uygun önlemler alınabilir. Hemen hemen birçok akıllı cihazda bulunan ivmeölçer, jiroskop ve manyetometre sensörleri sayesinde veri toplamak oldukça kolaylaşmıştır. Elde edilen verileri sınıflandırmak için ANN, CNN, RNN ve LSTM gibi derin öğrenme yöntemleri kullanılmaktadır.

Araştırmamızda esas olarak sensör tabanlı aktivite tanımaya odaklandık. İvmeölçer ve jiroskop verilerinden oluşan UCI-HAR veri setindeki ham veriler kullanılarak 1 boyutlu konvülasyonel sinir ağı ile sınıflandırmalar yapıldı. Datalar üzerinde matematiksel veya istatistiksel herhangi bir öznitelik çıkarımı yapılmadı. Ham veriler median filtreleme ile filtrelenmiştir. 6, 7 ve 12 aktivite üzerinde gerçekleştirilen deneylerde sırasıyla 96.95%, 95.03% ve 93.08% sınıflandırma doğruluğu elde edilmiştir.

Anahtar Kelimeler: Derin Öğrenme, CNN, UCI-HAR, İnsan Aktivitesi Tanıma, Sensör Tabanlı Aktivite Tanıma

I sincerely dedicate this thesis to my family who supported me everytime …

ACKNOWLEDGEMENTS

Thank you to my supervisor, Asst. Prof. Dr. Yasin KAYA, for providing guidance and feedback throughout this thesis. I also thank my friends and family for their support during my thesis studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

ANN	: Artifical Neural Network
CNN	: Convolutional Neural Network
ECG	: Electrocardiography
EMG	: Electromyography
FN	: False Negative
FP	: False Positive
GPS	: Global Positioning System
HAR	: Human Activity Recognition
KNN	: K-Nearest Neighbors
LR	: Learning Rate
LSTM	: Long Short-Term Memory
MLP	: Multilayer Perceptron
PAMAP2	: Physical Activity Monitoring Dataset
ReLU	: Rectified Linear Unit
RNN	: Recurrent Neural Network
SGD	: Stochastic Gradient Descent
SVM	: Support Vector Machine
TN	: True Negative
TP	: True Positive
UCI-HAR	: Human Activity Recognition Using Smartphones Dataset
WISDM	: Smartphone and Smartwatch Activity and Biometrics Dataset

1. INTRODUCTION

People’s behaviors, activities, ways of doing activities, and many issues related to this process have always been some of the main issues that attracted the attention of researchers. Recording, classifying and recognizing human activities have been one of the most remarkable research topics with the developments in detection and data processing technologies. In recent years, smart devices (smartphone, smartwatch, music player, etc.) which are found in almost every human include sensors such as accelerometer, gyroscope, and magnetometer. This made easier data collection and classification process. In this way, Human Activity Recognition (HAR) has been used in many areas such as health assessment, environmentally supported living systems, smart building, sports and exercise, building energy management, and security systems. HAR problem can be used to detect activity-based anomalies in daily life. Wrigkas et al. (Vrigkas, Nikou, & Kakadiaris, 2015) divided human activities into 6 categories according to their complexity. Behaviors are physical activities related to the psychological state and personality of the person (Martinez, Yannakakis, & Hallam, 2014). Actions are everything that happens in a particular situation. They describe the social actions of people (Lan, Sigal, & Mori, 2012). Gestures represent the primitive movements of one's body parts (Yang, Saleemi, & Shah, 2013). Atomic action represents a single movement of a single person (Ni, Moulin, Yang, & Yan, 2015). Group actions are described as actions performed by a group (Tran, Gala, Kakadiaris, & Shah, 2014). Human-to-object or human-to-human interactions are movements in which more than one person or object interacts (Patron-Perez, Marszalek, Reid, & Zisserman, 2012).

HAR problems include the processes of recording and analyzing a person's daily life to learn about people and behaviors. Various methods have been developed to collect and classify activity data. Das Antar et al. (Das Antar, Ahmed, & Ahad, 2019) examined the methods used for the activity recognition problem in two groups; video-based activity recognition (Bodor, Jackson, & Papanikolopoulos, 2003; Madabhushi & Aggarwal, 1999; Niu & Abdel-Mottaleb, 2005; Park et al., 2016; Robinovitch et al., 2013; Shieh & Huang, 2009) and sensor-based activity recognition (Asghari, Soleimani, & Nazerfard, 2020; He, Zhu, Su, & Tentzeris, 2020; J. Liu et al., 2020; L. Zhang et al., 2020). In our research, we mainly focused on sensor-based activity recognition. Sensors are devices used to convert physical quantities into electrical signals, such as heat, distance, shake, gravity, pressure, position, sound waves, and acceleration.

In our daily life, sensors are used in most of the places in daily life, most of the items we use such as smartphone, tablet, smart home systems, ships, cars, hospitals, smart home appliances, aircraft radar systems, and navigation systems. In recent studies, it was observed that sensor data provide more information about people. With sensors, diseases can be detected by monitoring the heartbeats and brain signals of people and inferences can be made about people and their feelings (Acharya, Oh, Hagiwara, Tan, & Adeli, 2018; Israel, Irvine, Cheng, Wiederhold, & Wiederhold, 2005; Jalal, Quaid, & Hasan, 2019). Activity classification can be performed by recording the activities of people with sensors (Clarkson, Mase, & Pentland, 2000).

Researchers basically divided sensor-based activity recognition into 3 categories; wearable sensor-based, environmental sensor-based, and smartphone sensor-based activity recognitions. Wearable sensors are smart sensors that track body movements. Biological sensors (ECG and EMG), smart watch are the main wearable sensors. Environmental sensors are sensors placed at a specific point in the environment. Radar, camera, temperature, proximity, and motion sensors are some of these sensors. Smartphone based sensors are sensors that can detect motion changes, changes in the environment and magnetic fields in mobile phones. Accelerometer, gyroscope, magnetometer, GPS, and proximity sensor are sensors found in many smartphones.

In the literature, human activity data were mostly recorded using accelerometer, gyroscope, and magnetometer sensors. While creating the PAMAP2 (Reiss & Stricker, 2012b) dataset, the data of 18 activities of the subjects were collected using accelerometer, gyroscope and magnetometer. It contains sensor data for 18 activities performed in daily life such as walking, sitting, driving, cycling, ironing, watching TV, and playing football. UniMiB SHAR (Micucci, Mobilio, & Napoletano, 2017) dataset consists of accelerometer sensors recorded from 30 subjects for use in fall detection. Accelerometer and gyroscope data of 51 subjects for 18 different activities were recorded using smart phones and smart watches, and WISDM (Weiss, 2019) dataset was created. UCI-HAR (Anguita, Ghio, Oneto, Parra, & Reyes-Ortiz, 2013a) dataset contains the gyroscope and accelerometer signals of the basic 6 activity data and also takes into account the signals that refer to the sitting-to-stand phase which called transition activities.

The second step is the classification of activities recorded by sensors. Artificial intelligence, machine learning, and deep learning methods are used to classify recorded activities. Support Vector Machine (SVM) (Anguita, Ghio, Oneto, Parra, & Reyes-Ortiz, 2012), Ada Boost, Decision Tree Classifiers (Reiss, Hendeby, & Stricker, 2013), Multi-Layer Neural Network and Fuzzy Logic (Kim & Helal, 2014), Multi-Layer Perceptron (MLP) (Walse, Dharaskar, & Thakare, 2016), AdaBoost.M1 algorithm, Random Forest, J.48, Naive Bayes (Walse, Dharaskar, & Thakare, 2017), Long-Short Term Memory (LSTM) (Eyobu & Han, 2018), Convolutional Neural Network (CNN) (Sikder, Chowdhury, Arif, & Nahid, 2019), LSTM-CNN (Xia, Huang, & Wang, 2020) and hybrid methods (Zhu, Chen, & Ye, 2020).

In the field of security, studies are carried out to identify people from human activities and to identify their identity (G. Li, Huang, & Xu, 2017)(Topuz & Kaya, 2019). As a result of detailed analysis of the data received from wearable sensors by using learning methods, it has been seen that human identification and authentication systems can be developed (Gholamrezaii & Taghi Almodarresi, 2019). The authors made predictions about the age ranges of people from walking data from the 3D accelerometer sensor (Terrier & Reynard, 2015). In the study conducted in 2005, the signals produced by the sensors embedded in the portable device were used (Ailisto, Lindholm, Mantyjarvi, Vildjiounaite, & Makela, 2005). Nowlan's study on identity identification in 2009 performed the identification of individuals using the biometric features of human gait with the help of a sensor consisting of an accelerometer and gyroscope. The movements of the individuals were determined in three different ways with 30 degrees difference in 6 different ways. The collected data were evaluated with the K-Nearest Neighbor algorithm and 95% success rate was achieved (Nowlan, 2009). In addition to the individual studies conducted by the researchers, big companies carried out similar studies. Google developed a project called ABACUS in 2016 to secure personal data on mobile devices. In this project, they developed a system, classifying people according to walking data by using the human activity recognition (HAR) method with the data they collected from 1500 people from different continents. The researchers used the outputs of deep learning methods (such as CNN and RNN) to extract the features from the data recorded and achieved the highest accuracy rate of 69.41% (Neverova et al., 2016). In 2017, a system called iWalk was developed to authenticate the mobile user. This system collects data based on accelerometers placed on smartphones and uses the HAR method to determine whether the current user is the owner of the phone. The application decides whether the user owns the phone within a certain period of time. If it decides the phone doesn't own it, it will perceive it as lost and generate alerts. The proposed method achieved an average correct detection rate of 97.80% (G. Li et al., 2017).

The studies show that recognizing events using the sensors of smartphones or a similar device is feasible. There is a need to develop approaches that provide accurate and reliable performance for HAR. Moreover, the performance of these models needs to be increased to use in real-world applications. This motivated us to develop a rapid and accurate approach to HAR. In literature, a lot of experiments have been applied for the HAR problem. But most of them used sensor data after signal processing and some complex features calculated from the signal using mathematical and statistical methods. Researchers used basic activities such as running, walking, etc. for HAR problems. If we want to recognize human activities or behaviors more detailed, we should examine complex postural transition activities. In our model, we used raw data in the UCI-HAR dataset. Experimental results were calculated using 6, 7, and 12 activities including postural transition activities. In the study, deep learning hyper parameters were investigated in order to obtain the best results. Experiments were conducted to determine the effect of the classification results of using different sensor data individually and together. As a result of experimental tests, it was seen that the proposed model achieved promising results.

2. LITERATURE REVIEW

This research focuses on smartphone sensor-based HAR using UCI-HAR dataset. There are many datasets for the HAR in literature, especially gyroscope and accelerometer sensor data. Some of these databases are PAMAP2 (Reiss & Stricker, 2012a), UCI-HAR(Anguita, Ghio, Oneto, Parra, & Reyes-Ortiz, 2013b), WISDM (Weiss, Yoneda, & Hayajneh, 2019). The data includes signal data obtained by collecting sensor signals such as accelerometer, gyroscope, and magnetometer of smart devices. Mostly, the researchers have used basic activities for HAR, except postural transition activities.

In the study conducted by Kırkan et al. (Kırkan, Samet;Karabey, Işıl;Bayındır, 2016), accelerometer sensors were used found in modern smartphones. Many classification algorithms were applied to the experiments performed with 10 different activities, including standing, sitting, lying, walking, running, eating, getting up with a lift, getting off with an elevator, driving, and cycling. The classification rate of 96% was achieved in the Random Forest classification algorithm. Tuan Dinh Le & Chung Van Nguyen (Le & Van Nguyen, 2015) conducted experiments with the UCI-HAR dataset using various machine learning methods. They achieved a 96.18% classification rate using the Decision Tree algorithm. In a study implementing LSTM in 2018, researchers achieved 90% accuracy using the UCI-HAR dataset (Zhao, Yang, Chevalier, Xu, & Zhang, 2018). Researchers obtained accuracy rate of 97% with their Divide and Conquer Based 1D CNN model (Cho & Yoon, 2018). Researchers (Hassan, Uddin, Mohamed, & Almogren, 2018) proposed a model based on Deep Belief Neural Network, and they made experiments with 12 activities from UCI- HAR dataset. They achieved accuracy and error rate of 95.85% and 4.14%, respectively. Myo et al. (Myo, Wettayaprasit, & Aiyarak, 2019) used ANN to classify UCI-HAR data and achieved 98.32% accuracy rate. Bilal et al. (Bilal, Shaikh, Arif, & Wyne, 2019) applied different feature extraction methods to UCI-HAR dataset and WISDM dataset to classify activities. They implemented MLP, SVM, and LR classification methods. They added a post-learning data engineering method to their model and achieved an accuracy rate of 99% for both datasets. In 2019, Irvine et al. (Irvine, Nugent, Zhang, Wang, & Ng, 2020) collected 24 activities (data from daily activities such as brushing teeth, going to the toilet, washing dishes, etc.) data from 246 people. Participants' data were recorded three times a day, in the morning, noon and evening, for a week. In the study conducted to improve the dataset. Machine learning was performed using NN and as a result, an 80.39% success rate was achieved. In 2019 (Wu & Zhang, 2019), researchers applied 2 different encoding algorithms to the UCI-HAR dataset: Gramian Angular Field encoding algorithm ramian Summation Angular Field (GASF) and Gramian Difference Angular field (GADF). Later, they applied the 8-layer Deep CNN model that they proposed to 4 different groups, which they created from the only accelerometer and gyroscope data. They achieved their best performance of 94% using GADF applied accelerometer data. Fan et al. (Fan, Jia, & Jia, 2019) applied 3 feature extraction algorithms to the dataset: Fisher score, ReliefF, and Chi square. They reached 99% success rate using the Decision Tree algorithm. Shah et al.(Zaki, 2020) used Logistic Regression, Gradient Boosting, Random Forest, Gaussian NB, and KNN algorithms and UCI-HAR dataset. They obtained accuracy rate of 96.20% using Logistic Regression. In 2019, researchers (M. Zhang, 2019) achieved 95.38% accuracy rate using LSTM network. Using UCI-HAR, Motion Sense, and MobiAct dataset, researchers classified activities using KNN, ResNet, and SVM. The accuracy calculated as 96.46% for the UCI-HAR dataset (Garlisi, 2019). T-WaveNet, one of the tree structured neural networks, was used in the study conducted with OPPORTUNITY (OPPOR), UCI-HAR, BCICIV2a, and NinaPro DB1 datasets. As a result of the study performed on four different datasets, 97% performance was achieved for the UCI-HAR dataset for 6 activities (M. Liu, Zeng, Lai, & Xu, 2020). CNN and RNN models proposed in the study with MotionSense and UCI-HAR datasets. As a result of the tests of the models, 90% classification performance was obtained for 6 activities for the UCI-HAR dataset (Taghanaki & Etemad, 2020). Researcher (Arigbabu, 2020) achieved an accuracy rate of 97.4% using their Recurrrent Convolutional Neural Network (RCN) model. Tufek et al. (Tufek et al., 2020) achieved 90% performance rate using CONV1D in their experiments. They used CNN and LSTM to calculate experimental results on the UCI-HAR dataset. The researchers used raw data. Kun Xia et al. (Xia et al., 2020) tested the hybrid LSTM-CNN model and used similar datasets such as UCI-HAR and WISDM. They achieved a 95.78% classification accuracy rate for the UCI-HAR dataset. In another study, the researchers (Ramachandran & Pang, 2020) developed a CNN model using WISDM and UCI-HAR datasets. In this study, they implemented their model with 6 activities in the UCI-HAR dataset, they achieved a 94% success rate using transfer learning for the CNN model.

3. MATERIALS AND METHODS

HAR process consists of 4 stages; data collection, data preprocessing, deep learning, and activity classification. Figure 3.1.1 shows these stages. The data obtained from the subjects are sent to the pre-processing step, passed through the Median filtering and windowing steps, and prepared for the deep learning step. In the deep learning step, after the data is trained, tests are performed to classify the activities.

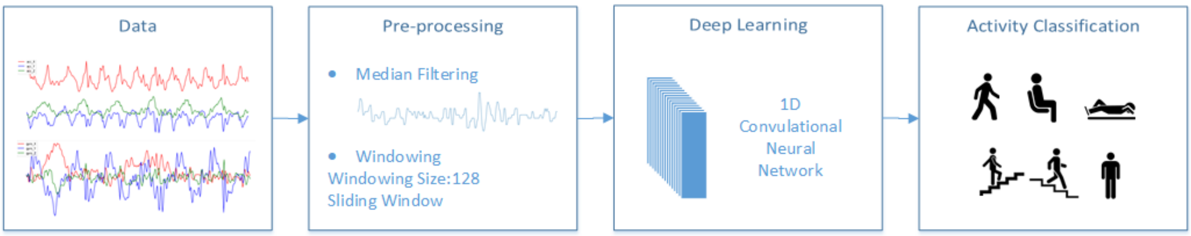

Figure 3.1.1: Activity recognition steps: Data Collection, Data Preprocessing, Deep Learning and Activity Classification

In this study, we used the UCI-HAR dataset. The dataset contains triaxial angular velocity and linear acceleration signals received from the smartphone's gyroscope and accelerometer sensors. The sampling rate of the signals for both sensors is 50 Hz. Collected signals are processed with 3rd order median filtering to reduce noise. Data was transposed after windowing with 128-size sliding windows and we got one-dimensional data. To make the experiments, the data were divided into groups according to the number of activities and the sensor data included. To find optimum values of the learning rate, optimizer, batch size, epoch, output unit size of convolutional layer and hidden unit size of convolutional layer. The experimental tests were evaluated with the same test users. While finding optimum values, we have only evaluated according to accuracy and loss values. Training and testing processes were carried out with a deep learning model written in python language. Keras and Tensorflow libraries are used to construct the model. Our evaluation metrics were F1 Score, Precision, Sensitivity, Specificity, Accuracy, and Loss value.

3.1. Dataset

Basically, the human activity recognition process consists of the steps in Figure 3.1.1. The data collection step represents obtaining data. UCI-HAR dataset was used in our experiments. This dataset contains the accelerometer and gyroscope sensor data of 30 users. The researchers recorded the data in a controlled laboratory environment at a frequency of 50 Hz (per 20 seconds). They recorded the data of each participant in two different ways. In the first stage, a waist bag was placed on the user's waist and the mobile device was placed inside the bag, in the second stage, the waist bag was attached to the chest height or where the user wanted.

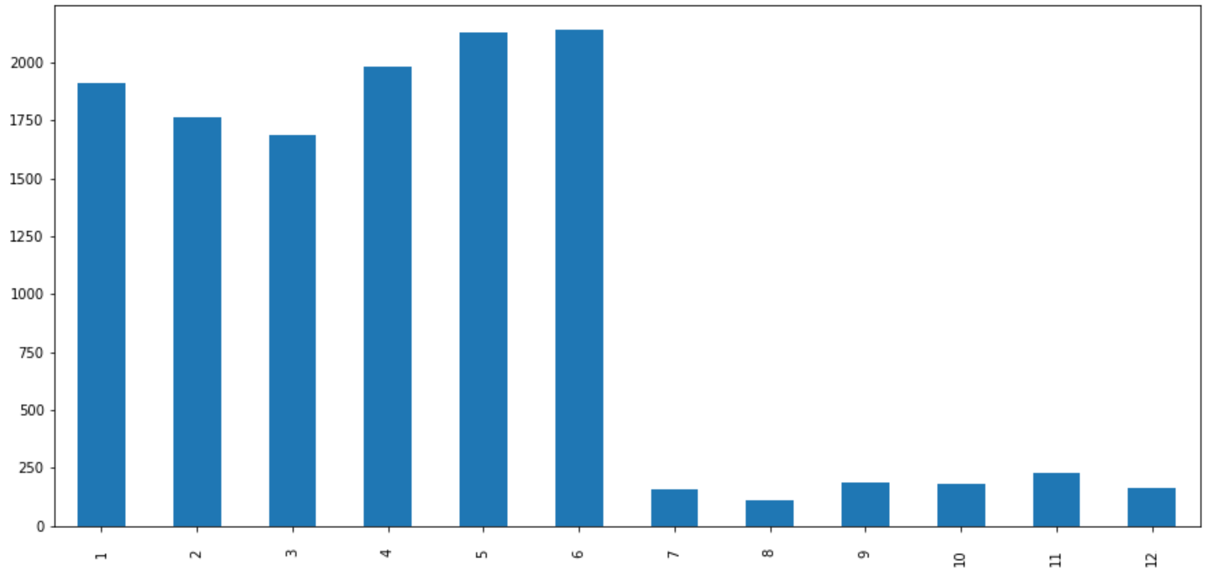

Figure 3.1.2: Activity Numbers

While gathering data, users were recorded with a camera to label activities. Figure 3.1.2 shows the number of labeled samples of 12 activities. Table 3.1.1 shows activities and their labels. 6 basic activities (walking, walking upstairs, walking downstairs, sitting, standing, lying) were recorded for each subject, also postural transition data of 6 activities are also labeled (stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, lie-to-stand.). To conducting the experiments, the data divided into groups according to the number of activities and the sensor data included.

Table 3.1.1: Activity Labels

Labales for Activity					
12 activities		7 activities		6 activities	
Id	Label	Id	Label	Id	Label
1	Walking	1	Walking	1	Walking
2	Walking upstairs	2	Walking upstairs	2	Walking upstairs
3	Walking downstairs	3	Walking downstairs	3	Walking downstairs
4	Sitting	4	Sitting	4	Sitting
5	Standing	5	Standing	5	Standing
6	Lying	6	Lying	6	Lying
7	Stand-to-Sit	7	Postural Transitions		
8	Sit-to-Stand				
9	Sit-to-Lie				
10	Lie-to-Sit				
11	Stand-to-Lie				
12	Lie-to-Stand				

Table 3.1.2: Data Groups

Type	Activity Number	Sensor
1	6	Acc+Gyro
2	12	Acc+Gyro
3	7	Acc+Gyro
4	6	Acc
5	12	Acc
6	7	Acc
7	6	Gyro
8	12	Gyro
9	7	Gyro

Table 3.1.2 demonstrates data groups: Type 1 contains, gyroscope and accelerometer data, Type 4, only accelerometer data, and type 7 only gyroscope data, for 6 activities. Type 2, gyroscope and accelerometer data, Type 5, only accelerometer data, and type 8, only contains gyroscope data, for 12 activities. Type 3 contains, gyroscope and accelerometer data, Type 6, only accelerometer data, and type 9 only gyroscope data, for 7 activities.

As seen in Table 3.1.2, the activity numbers were set as 6, 7, and 12. Groups containing 6 activities include the data of walking, walking upstairs, walking downstairs, sitting, lie and standing activities. Groups containing 12 activities include 6 transition activities besides the basic 6 activities. Groups containing 7 activities classified 6 transition activities with a single label. In other words, it includes 6 basic activities and 1 postural transition activities.

3.2. Pre-processing

The dataset contains triaxial angular velocity and linear acceleration signals received from the gyroscope and accelerometer sensors of a smartphone. The sampling rate of the signals for both sensors is 50 Hz. Collected signals were processed with 3rd order median filtering to reduce noise. Data was transposed after windowing with 128-size sliding windows and one-dimensional data created for the next steps.

3.2.1. Median Filtering

The noise of the signal occurs during data recording or collecting. This noise terribly affects the signal processing operations. Thus, it must be eliminated to decrease these effects. The median filter is a non-linear digital filtering method that used to remove noise from a signal or image. Noise reduction is a specific pre-processing stage to increase the results of later stages.

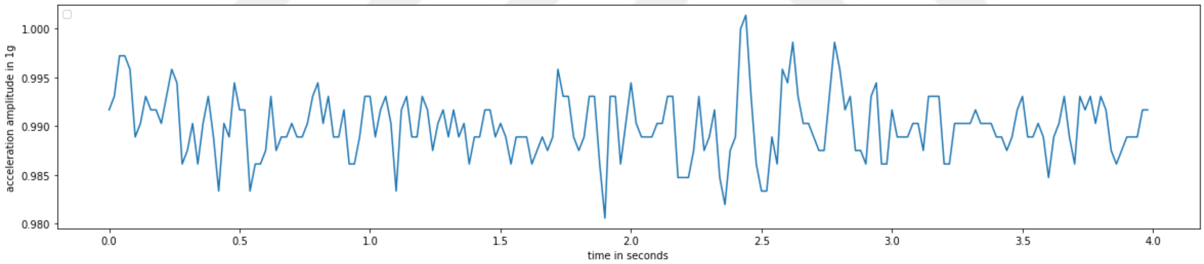

Figure 3.2.1: Normal signal example for 4 second

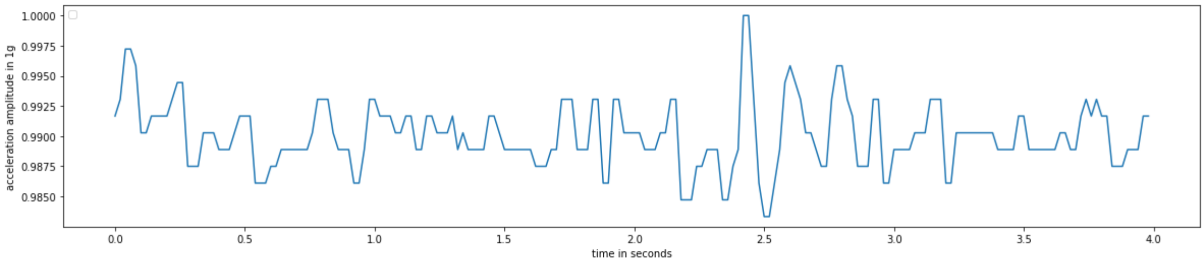

Figure 3.2.2: Median filtered signal for same 4 second

The main idea is to remove the noise from the data by replacing each input value with the median of other neighboring input values. Figure 3.2.1 shows the original signal and Figure 3.2.2 shows the same signal after passing through the median filter.

$$\text{Median}(\text{L}+, \text{L}-) = \text{L} \tag{1}$$

For each value in the selected column, the filter selects the values that come one before and after the column. The value in the output column is the median of these three values in 3^{rd} median filtering. So, the median will be the same for the first column and second column. Thus, the median of the L+ and L- will be L as you can see in Formula 1.

3.2.2. Windowing Operation

At this stage, we prepared the data using 128 size sliding windows which had a 50% overlap. This overlap means that the starting point of the next window is shifted left by 64 samples. 6 * 128(2.56 seconds) shaped windows were created to represent one action. Figure 3.2.3 demonstrates windowing operation stages.

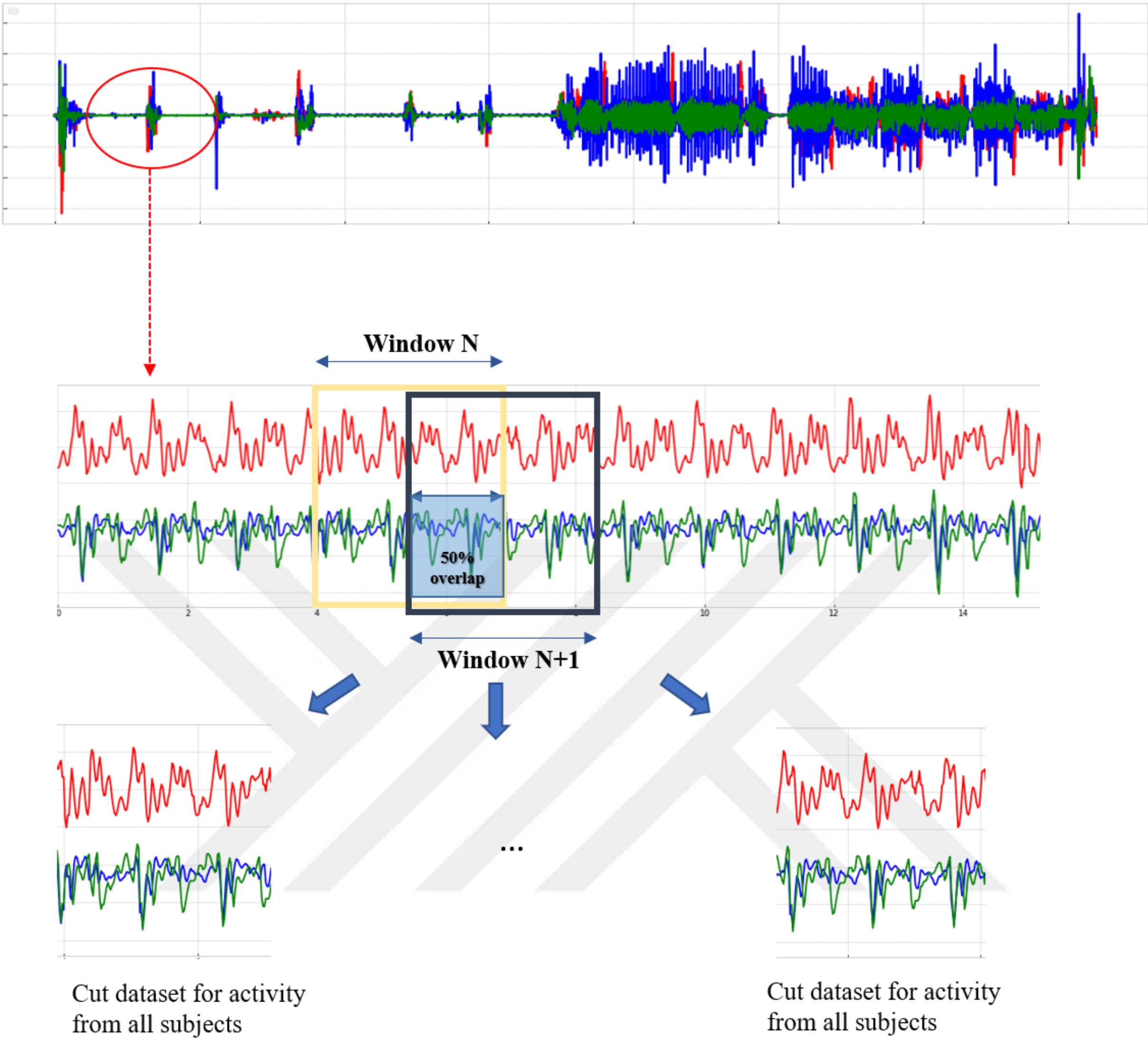

Figure 3.2.3: Windowing Operation

Table 3.2.1: Windowed data in x, y, z-axis for accelerometer and gyroscope sensors

	acc_X	acc_Y	acc_Z	gyro_X	gyro_Y	gyro_Z
0	0.918056	-0.112500	0.509722	-0.054978	-0.069639	-0.030849
1	0.911111	-0.093056	0.537500	-0.012523	0.019242	-0.038485
2	0.881944	-0.086111	0.513889	-0.023518	0.276417	0.006414
…						
125	0.934722	-0.138889	0.031944	-0.045815	0.490830	-0.380264
126	0.923611	-0.144444	0.056944	-0.143553	-0.041539	-0.656680
127	0.923611	-0.144444	0.056944	-0.227853	0.091019	-0.982882

Table 3.2.1 shows the windowed data samples. The first three columns contain the values of the accelerometer sensor and the last three columns contain the values of the gyroscope sensor in the x, y, and z-axis. After the windowed data were created, the data were transposed as in **Hata! Yer işareti başvurusu geçersiz.**.

Table 3.2.2: Example from transposed windowed data

	0	1	2	…	125	126	127
acc_X	1.020833	1.025000	1.020833		1.022222	1.022222	1.023611
acc_Y	-0.125000	-0.125000	-0.125000		-0.119444	-0.120833	-0.125000
acc_Z	0.105556	0.101389	0.104167		0.094444	0.087500	0.091667
gyro_X	-0.002749	-0.000305	0.012217		0.031765	0.031460	0.021380
gyro_Y	-0.004276	-0.002138	0.000916		0.002443	-0.000305	0.003971
gyro_Z	0.002749	0.006109	-0.007330		-0.003665	-0.001527	-0.000305

After the data was transposed, 12637 windows with size 128 * 6 were created. As a result, combining all windows, 12637 * 770 size dataset was created. The last two columns were subject Id and action class label.

3.3. Deep Learning Methodologies Used in The Study

Deep learning enables us to train the artificial intelligence that can predict the results of a given dataset and to work with larger data and more complex artificial intelligence algorithms. Deep learning includes Artificial Neural Networks (ANN) that imitate the human brain and can be used in many areas such as visual object recognition, speech recognition, and signal classification. ANN is an information processing technology that analyzes existing data and creates new information from this data. A typical ANN consists of the input layer containing one or more nodes (neurons), the hidden layer, and the output layer. The ANN architecture with the structures of the neurons is shown in Figure 3.3.1. Each link between neurons is associated with a numerical value called weight. In this way, the effects of the data on neurons can be adjusted as desired. Before the data coming to the artificial neuron reaches the next neuron, it is multiplied by the weights of the connections and transmitted to the activation function. A neural network without the activation function will have limited learning. Learning of the neural network in nonlinear situations is provided by using the activation function. Basically, the input layer used to detect the properties of data (signals, pictures etc.) that are sent to the network and the output layer calculates result values. Hidden layers are connections between the input and output layers, each designed to produce a specific output for an intended result (Okwu & Tartibu, 2021).

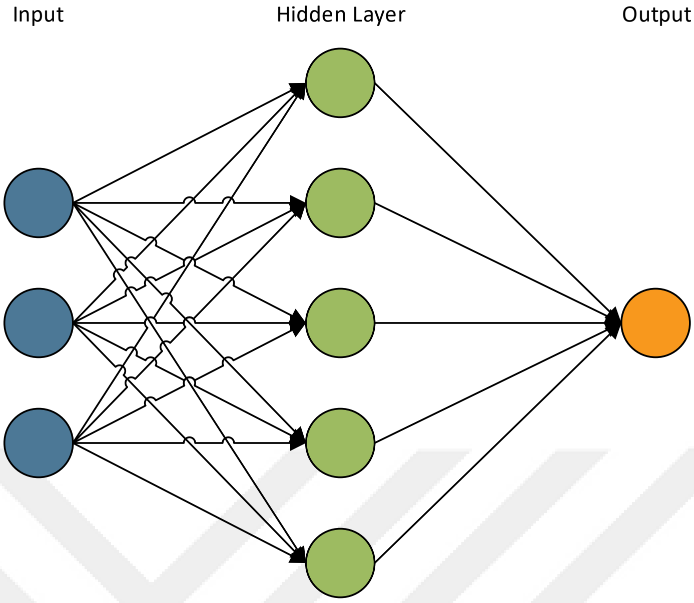

Figure 3.3.1: Basic ANN Structure

3.3.1. Convolutional Neural Network

CNN is a neural network which generally used for classification based on image processing problems. It achieves very high success in problems such as image classification and object detection (Simonyan & Zisserman, 2015)(Ioffe & Szegedy, 2015) using 2D CNN. In addition, 1D CNN often used in classification problems based on sensor data and achieved high performance (Burger, Qian, Schiele, & Helms, 2020) (F. Li, Shirahama, Nisar, Köping, & Grzegorzek, 2018) (Moradi, Aghapour, & Shirbandi, 2019). CNNs consist of neurons organized in three dimensions. It processes the height, width, and depth. Unlike standard neural networks. Neurons in any layer depend on only one region of the preceding layer.

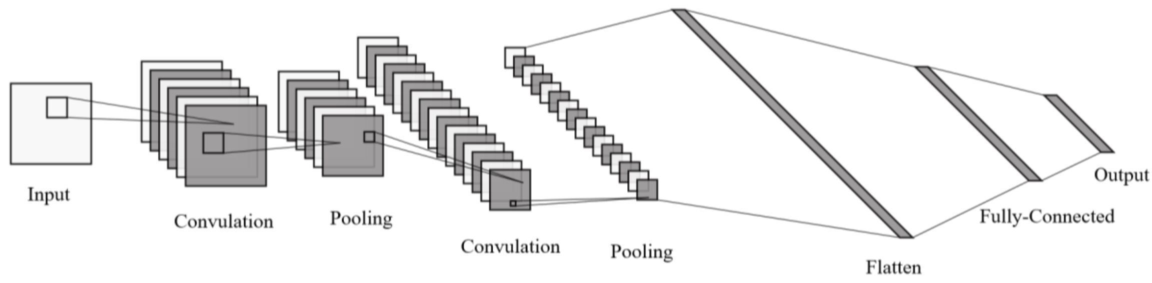

Figure 3.3.2: Basic CNN Layers

Basic CNN structure consists of three layers; convolutional layers, pooling layers, and fully connected layers. The structure of a typical CNN is demonstrated in Figure 3.3.2. The convolational layer used to detect the properties of the signal or image and send results to the network. The pooling layer reduces the number of weights and checks the availability of incoming data. The flattening layer prepares the data from the pooling layer for the fully connected layer. The fully connected layer is a classical neural network layer that used in classification.

3.3.1.1. Convolutional Layer

The convolutional layers start with the calculation of the scalar product of the weights and input data. The output values of the neurons to which the input data are connected be determined by calculating. This layer performs the normalization of the data and using an activation function nonlinear data definition of properties to the neural network.

Batch normalization is one of the techniques that can be used to normalize the activations in the intermediate layers of deep neural networks. In addition to providing a regulating effect, it also shows a resistance to the extinction gradient of the deep neural network during training. In this way, it reduces the training time and allows the model to perform better.

Activation functions help normalize the output of each neuron between 1 and 0 or -1 and 1. The purpose of the activation functions is to adjust the weight and threshold (bias) values.

Relu, Sigmoid, Tanh, and Softmax activation functions are among the most frequently used functions. Rectified Linear Unit (ReLU) returns the input directly as an output if the input is positive. If the input is negative, it returns zero. ReLU is more effective than other functions because all neurons are not activated at the same time and a certain number of neurons are activated. Thus, this reduces computation load.

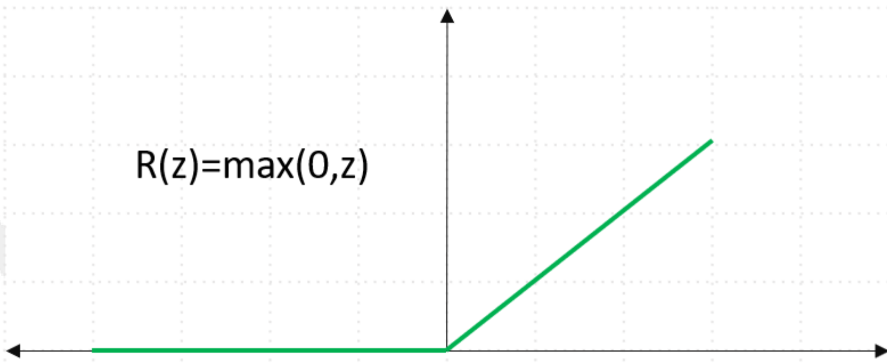

Figure 3.3.3: ReLU

In some cases, the gradient value is zero because the weights and bias values cannot be updated during the backpropagation step in the training of NN (Sharma, Sharma, & Athaiya, 2020). The sigmoid function maps input values between the range 0 and 1. This can be treated as probabilities of data points of a particular class and used for binary classification. Unlike the sigmoid function, the softmax function can be used for classification problems with more than two classes (Milosevic, 2020). The softmax function is a combination of multiple sigmoid functions.

3.3.1.2. Pooling Layer

Generally, pooling is used after the convolution layer to reduce the size of the output of the convolution layer and reduce the complexity in subsequent layers. It can be thought of as the decreasing resolution of images for image processing. Pooling does not affect the number of filters. Maximum pooling is one of the most common types of pooling methods. It continues by taking the maximum value from the pool formed (Milosevic, 2020).

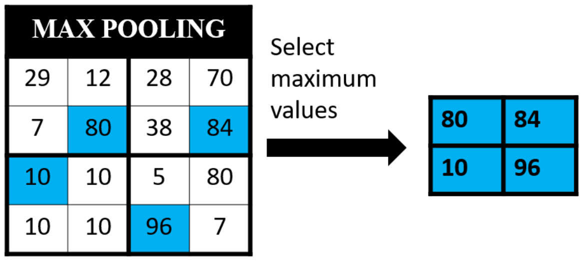

Figure 3.3.4: Maximum Pooling

In **Hata! Başvuru kaynağı bulunamadı.**, there is a 4x4 matrix representing the first input. W hen a 2x2 filter is passed over our input, a new output is created by taking the maximum values in the regions where the filter is applied.

3.3.1.3. Flattening Layer

The depth of the output of the pooling layers can be more than one. The Flatten layer combines the output from the convolution layers to create a flat structure. Thus, it can be used as input to the fully connected layer. (Milosevic, 2020).

3.3.1.4. Fully Connected Layer

The fully connected layer is the layer where the learning process takes place. Data from the flattening layer is trained here. This layer applies the working logic in classical neural networks. The properties are kept in the neurons in the layers and the learning process is carried out by changing the weights and biases (Milosevic, 2020).

3.3.2. Optimizers

Optimization algorithms are used to minimize the error rate. Adam optimizer is one of the stochastic gradient descent-based optimization algorithms. Optimization algorithms such as Adam, adamax, stochastic gradient descent, adagrad, adadelta are frequently used algorithms. Adam optimizer is more suitable for problems with big data and unstable targets. It is also a more applicable and efficient algorithm in practice. The learning rate is a setting parameter that determines the step size in each iteration in an optimization algorithm.

3.3.3. Proposed CNN Model

Since the activity classification problem is a one-dimensional time series problem, a one-dimensional convolutional neural network (1DConv) was used in our study. The layers and parameters used in our proposed deep NN model are shown in Table 3.3.1.

Table 3.3.1: Proposed CNN Model

Layer (type)	Output Shape	Param #
Conv1D	(None, 768, 256)	2816
Batch_normalization	(None, 768, 256)	1024
Activation	(None, 768, 256)	0
MaxPooling1D	(None, 384, 256)	0
Conv1D	(None, 384, 128)	327808
Batch_normalization	(None, 384, 128)	512
Activation	(None, 384, 128)	0
MaxPooling1D	(None, 192, 128)	0
Conv1D	(None, 192, 128)	163968
Batch_normalization	(None, 192, 128)	512
Activation	(None, 192, 128)	0
MaxPooling1D	(None, 96, 128)	0
Conv1D	(None, 96, 64)	81984
Batch_normalization	(None, 96, 64)	256
Activation	(None, 96, 64)	0
Flatten	(None, 3072)	0
Dense	(None, 1024)	3146752
Activation	(None, 1024)	0
Batch_normalization	(None, 1024)	4096
Dropout	(None, 1024)	0
Dense	(None, 2048)	2099200
Activation	(None, 2048)	0
Batch_normalization	(None, 2048)	8192
Dropout	(None, 2048)	0
Dense	(None, 7)	14343
Batch_normalization	(None, 7)	28
Activation	(None, 7)	0
Total params: 5,851,491		
Trainable params: 5,844,181		
Non-trainable params: 7,310		

The proposed model consists of 4 convolutional layers, 1 flatten and 3 fully connected layers. Hyperparameter optimization performed to minimize the loss function and determine the best parameters. The next section shows hyperparameter optimization results. We implemented experimental tests according to different learning rates, batch sizes, optimizers, and epoch size. To construct an effective CNN model, we tried model optimization techniques that determine the size and depth of the convolution layers.

4. RESULTS AND DISCUSSIONS

We used the UCI-HAR dataset to evaluate the proposed model. In the tests, the data were divided into training (%70) and test (%30) subsets. These subsets were subject dependent, so the subjects used in training were completely different from test subjects. Training and testing processes were carried out with a deep learning model written in python language. Keras and Tensorflow libraries were used in the model. Our evaluation metrics were F1 Score, Precision, Sensitivity, Specificity, Accuracy, and Loss value. We have only used Type 2 which includes all activities for hyperparameter optimization.

4.1. Evaluation Metrics

Many metrics are used to measure the performance of classification models. Some of these are accuracy, loss, F1 score, sensitivity, precision, and specificity. In this study, the results are metrics such as accuracy, loss, f1 score, sensitivity, and specificity. Accuracy is the number of predictions where the predicted values are equal to the actual values. So, it is the ratio of correct predictions to all predictions as you can see in Formula (2).

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2)$$

Loss is also known as the cost function. It provides a more detailed perspective on how accurate the model is performing. Unlike accuracy, the value is not a percentage. It is equal to the sum of errors made for each sample in training and testing sets. Loss is often used in the training process to find the "best" parameter values for the model. The aim of the training process is to minimize this value. The most common loss functions are log loss and cross-entropy loss (gives the same result when calculating error rates between 0 and 1).

Sensitivity (Formula 3) represents the percentages of correctly defined positive values and specificity (Formula 5) correctly defined negative values in the dataset. In the measurement of a dataset, accuracy is the proximity of the measurements to a certain value. Precision (Formula 4) is defined as the proximity of the measurements made to each other. The terms "positive" and "negative" do not represent the utility of any value, but its presence or absence. F1 Score value shows the harmonic mean of Precision and Recall (Sensitivity) values and its formula shown in (6).

$$\text{Recall(Sensitivity)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{Presicion} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{5}$$

$$\text{F1 Score} = 2 * \frac{\text{Presicion} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

- True Positive (TP) is the case where the output of the algorithm outputs YES when the actual state is YES,
- False Positive (FP) is the case where the output of the algorithm outputs NO when the actual state is YES.
- True Negative (TN) is the case where the output of the algorithm outputs NO when the actual state is NO,
- False Negative (FN) is the case where the output of the algorithm YES when the actual state is NO.

4.2. Hyperparameter Optimization

The data were divided into train data and test data for hyperparameter tuning. 70% of the users selected as train data and the remaining 30% are selected as test data. The same users (2, 4, 9, 12, 13, 26, 18, 28, and 24 are test users.) (Anguita et al., 2013b) used in our tests to compare our results with the other researchers' results. This made the comparison of our results to the results of the other studies more fairly. Optimization was made for the learning rate, optimizer, batch size, epoch, dept of the CNN layers, and output unit size of CNN layers. While finding optimum values, we evaluated according to accuracy and loss values.

Updating parameters in deep learning works by deriving backward derivatives (backpropagation) and multiplying these derivatives by the parameter called the learning rate. Since finding the best learning rate (LR), some values in different ranges were tested. If the learning rate is high, it causes overfit (excessive memorization), if the learning rate is small, the training takes too long because the training proceeds with small steps.

Table 4.2.1: Accuracy for different learning rates

Learning Rate	Accuracy	Loss
0.1	0.9279	0.0659
0.01	0.9171	0.0776
0.001	**0.9308**	**0.0465**

We first tried different LRs and the best learning rate value determined as 0.001. As you can see in Table 4.2.1 when the LR was set as 0.1, the accuracy decreased to 92.79%. When LR was determined as 0.01 and an accuracy rate of 91.71% was obtained. We achieved the highest accuracy rate of 93.08% using LR as 0.001.

In deep learning, since it is difficult to train large datasets, the data need to be divided into small groups (batches). The training cycle is done on small batches selected, so it can be possible to train the deep learning model more efficiently. The batch size parameter determined when designing the model, means how much data processes at the same time.

Table 4.2.2: Model perofmance calculated using different batch sizes

Batch size	Accuracy	Loss
64	**0.9308**	0.0465
128	0.9131	0.0561
256	0.9221	0.0392
512	0.9308	0.0418

Table 4.2.2 shows the results of the test performed with different batch size values. The results show that 64 and 256 batch size values gave the best results. The results were remarkably close to each other. The batch size value was determined as 64 to ensure that the dataset did not give memorization results, especially for postural transitions. In order not to make the batch size value too large, the 64-batch size value is chosen.

Optimization algorithms such as Adam, Adamax, Stochastic Gradient Descent (SGD), Adagrad, Rmsprop are methods used to minimize the error rate. There are some differences between these optimization algorithms in terms of speed and performance. To find the best optimization algorithm for our model, tests were performed using Adam, Adamax, Adagrad, SGD, and Rmsprop optimizers.

Table 4.2.3: Model perofmance achieved using different optimizers

Optimizer	Accuracy	Loss
Adam	**0.9308**	0.0465
Adamax	0.9300	0.0418
AdaGrad	0.8791	0.1119
SGD	0.8316	0.1329
RMSprop	0.9292	0.0544

Table 4.2.3 shows the results of the tests. Especially, the results of the tests were performed using the Adam and Adamax algorithms obtained remarkably close results to each other. According to test results, the best result was achieved using Adam optimization algorithm with a small difference.

Table 4.2.4: Accuracy for different output unit sizes

Hidden unit size	Accuracy	Loss
64-128	0.9342	0.0478
128-256	0.9340	0.0490
256-512	0.9342	0.0472
512-1024	**0.9371**	0.0479
1024-2048	0.9308	**0.0465**

Table 4.2.4 shows the results of different output unit sizes. It can be seen from the table that the accuracy and loss values are very close to each other. Thus, the lowest loss value was selected to develop the deep learning model. Using 1024-2048 hidden unit size for fully connected layers, we achieved the best test accuracy and the smallest loss value.

Table 4.2.5: Accuracy for different dept of hidden unit sizes

Depth of Hidden unit size	Accuracy	Loss
32-64-128	0.9276	0.0551
64-128-256	0.9255	0.0320
32-64-128-256	0.9290	0.0526
32-64-128-256-512-1024-2048-4096	0.9221	0.0518
2048-1024-512-256-128-64	0.9168	0.0513
1024-512-256-128-64	0.7767	0.1594
512-256-128-64	0.9245	0.0408
256-128-64	0.9271	0.0369
128-64	0.9255	0.0397
64-128-128-256	0.9345	0.0412
256-128-128-64	**0.9308**	0.0465

To investigate the depth of our model, we conducted tests with hidden unit size values of convolutional layers. In Table 4.2.5, the tried unit size values are shown. It was seen that the 256-128-128-64 value gave the highest accuracy result.

How much the model is trained is an important issue. If the data is not trained enough or overtrained, the data is memorized. Thus, this issue needs to be balanced. Usually, when the epoch increases, accuracy will increase, but after a certain value, the data starts to be memorized. All previous tests were performed with 100 epochs. To determine the best number of epochs, the training was run at 250 epochs to observe the change in accuracy and loss values.

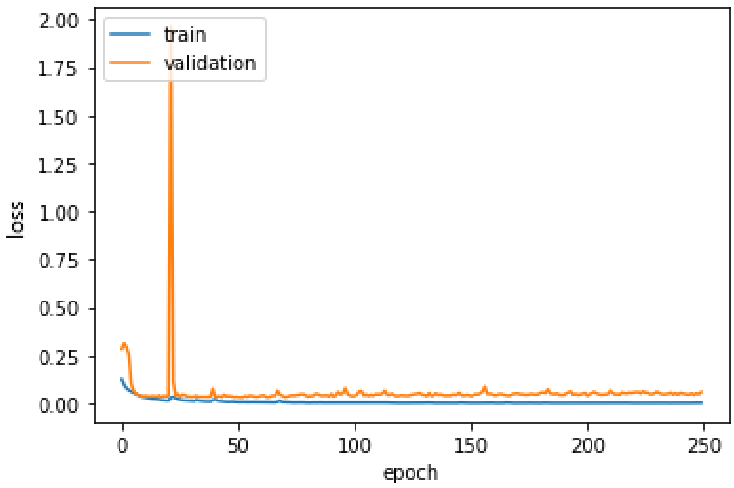

Figure 4.2.1: Loss over epochs

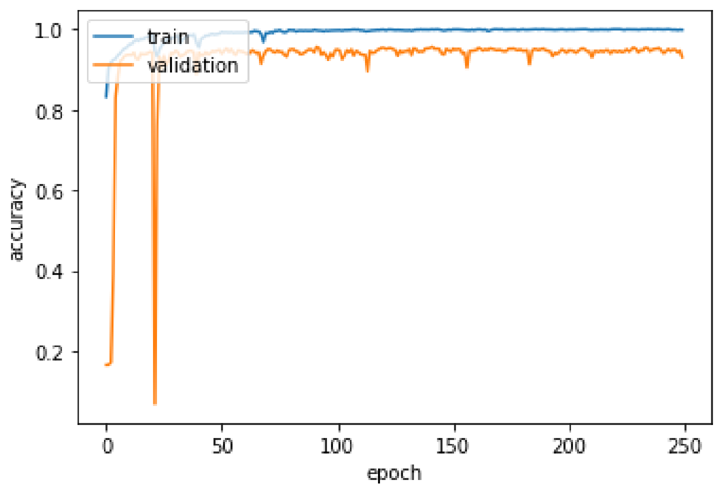

Figure 4.2.2: Accuracy over epochs

In our experiments, when we try to find true epoch size, 20 percent of the training data was selected as validation data. **Hata! Başvuru kaynağı bulunamadı.** shows the change of loss v alues of train and validation data according to epochs, and **Hata! Başvuru kaynağı bulunamadı.** demonstrates the change of accuracy values according to epochs. As you can see in **Hata! Başvuru kaynağı bulunamadı.**, sharply decreased in the loss values of the validation d ata between 0-50 epochs, and the accuracy value increased sharply in **Hata! Başvuru kaynağı bulunamadı.**. After the 100th epoch, the loss and accuracy values are almost stable, so our epoch value is set to 100.

After parameter optimization, we achieved best values for hidden unit size, batch size, optimization algoritm and learning rate parameters. Best parameters we found; batch size=64, optimizer=Adam, learning rate=0.001, hidden unit size=1024-2048, and epochs=100.

4.3. Experimental HAR Results

In the experiments, tests were carried out with the proposed 7-layer deep CNN model. Evaluation metrics are F1 score, sensitivity, specificity, and accuracy. Table 4.3.1 shows HAR resulsts.

Table 4.3.1: Human Activity Classification Results

Type	F1 Score	Sensitivity	Specificity	Accuracy
1	**97%**	**97%**	**99,37%**	**96,95%**
2	**81%**	**80%**	**99,34%**	**93,08%**
3	**95%**	**95%**	**99,17%**	**95,03%**
4	92%	92%	98,31%	91,65%
5	63%	70%	98,16%	79,73%
6	91%	91%	98,50%	91,10%
7	79%	80%	95,80%	79,01%
8	72%	70%	98,16%	81,18%
9	78%	79%	96,32%	78,41%

When the results were examined in detail, it was observed that the results with the only accelerometer and only gyroscope data between Types 4-9 were lower than Type 1, Type 2, and Type 3. When the sensitivity and F1 score values of these results were examined, it was seen that the dataset performed lower in some results. These lower achievements are also reflected in accuracy values. However, accuracy values and other determinant values also show high performance for Type 1, Type 2, and Type 3.

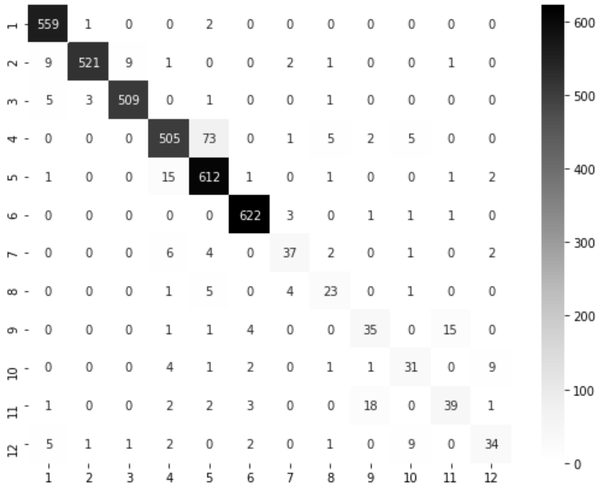

Figure 4.3.1: Confusion Matrix for Type 2

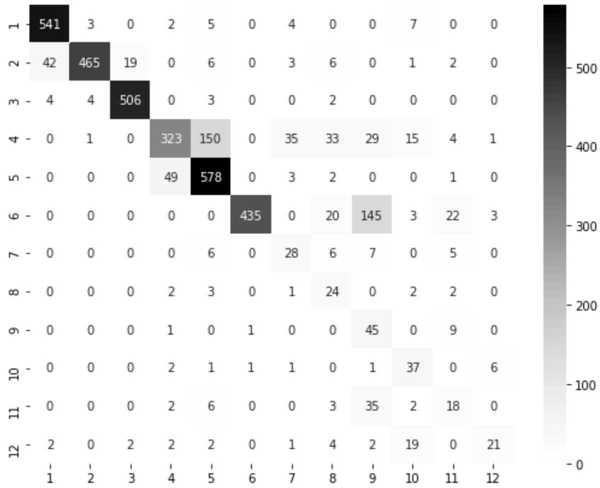

Figure 4.3.2: Confusion Matrix for Type 5

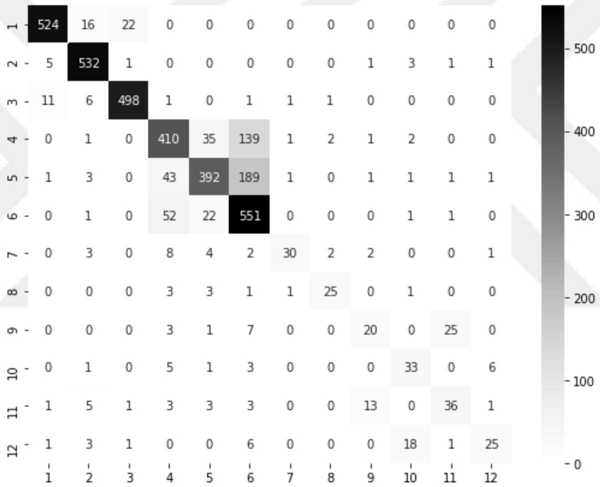

Figure 4.3.3: Confusion Matrix for Type 8

Figure 4.3.1 shows the confusion matrix of Type 2 which includes gyroscope and accelerometer data of 12 activities. Figure 4.3.2 shows the estimation of only accelerometer data, and Figure 4.3.3 shows the estimation of only the gyroscope data for 12 activities. In Figure 4.3.1, it is observed that the false predictions are distributed homogeneously, and the accuracy value is 93,08% for Type2. In Figure 4.3.2 and Figure 4.3.3, it is seen that some incorrect predictions are over-repeated for certain activities. When looking at only accelerometer or only gyroscope data, the activities were predicted accurately with lower performance, but higher performances were obtained when accelerometer and gyroscope data were evaluated together.

Table 4.3.2: Detailed Results for Type 2 (12 activities)

Activity ID	F1 Score	Sensitivity	Specificity
1	0.98	0.99	0,99
2	0.97	0.96	0,99
3	0.98	0.98	0,99
4	0.90	0.85	0,98
5	0.92	0.97	0,97
6	0.99	0.99	0,99
7	0.75	0.71	0,99
8	0.67	0.68	0,99
9	0.62	0.62	0,99
10	0.64	0.63	0,99
11	0.63	0.59	0,99
12	0.66	0.62	0,99

We achieved a 79,73% accuracy rate for Type 5 and the confusion matrix is shown in Figure 4.3.2. For the gyroscope, we achieved an 81,18% accuracy rate. If we examine **Table 4.3.2**, the correct prediction rate is less in activities between 6-12, which are transition activities, compared to other activities. Thus, postural transition activities are more complex to recognize.

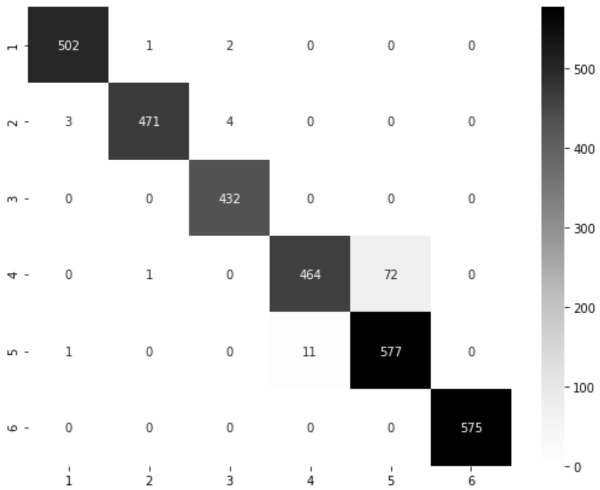

Figure 4.3.4: Confusion Matrix for Type 1

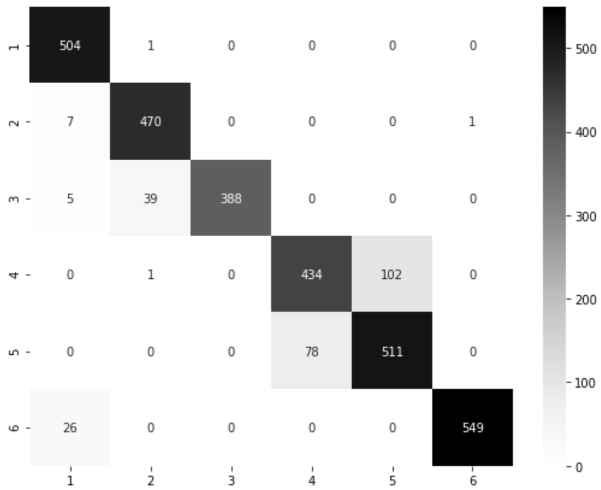

Figure 4.3.5: Confusion Matrix for Type 4

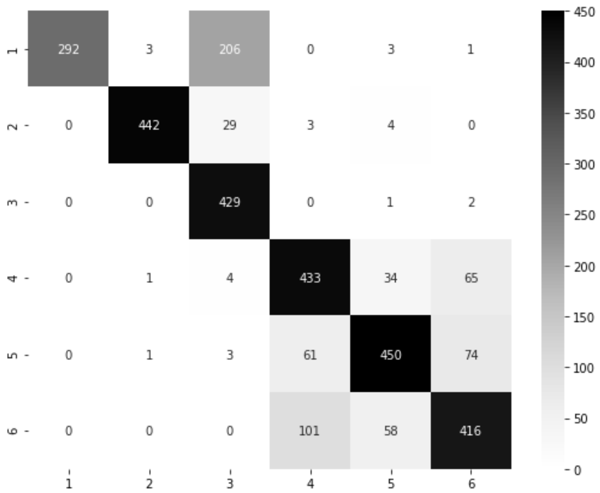

Figure 4.3.6: Confusion Matrix for Type 7

Figure 4.3.4 shows the confusion matrix of gyroscope and accelerometer data of 6 activities (Type 1) and we achieved an accuracy rate of 96,95%. Figure 4.3.5 shows the confusion matrix of Type 4 (only accelerometer data), and we obtained an accuracy rate of 79.01%. The confusion matrix of Type 7 (only the gyroscope data) shown in Figure 4.3.6 and an accuracy rate of 79.01% was achieved.

Table 4.3.3: Detailed results for Type 1

Activity ID	F1 Score	Sensitivity	Specificity
1	0.99	0.99	0,99
2	0.99	0.99	0,99
3	0.99	1.00	0,99
4	0.92	0.86	0,99
5	0.93	0.98	0,97
6	1.00	1.00	1.00

Table 4.3.3 shows the values obtained from the test results for 6 activities. When the table is examined in detail, it is observed that the results of some activities are quite successful. In addition, it was observed that some wrong predictions were made consistently for the same activity. Although the sensitivity and precision values may seem low for some activities, the overall accuracy value is acceptable.

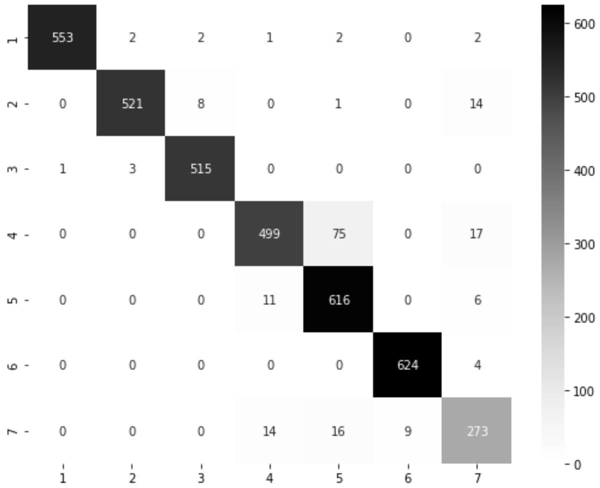

Figure 4.3.7: Confusion Matrix for Type 3

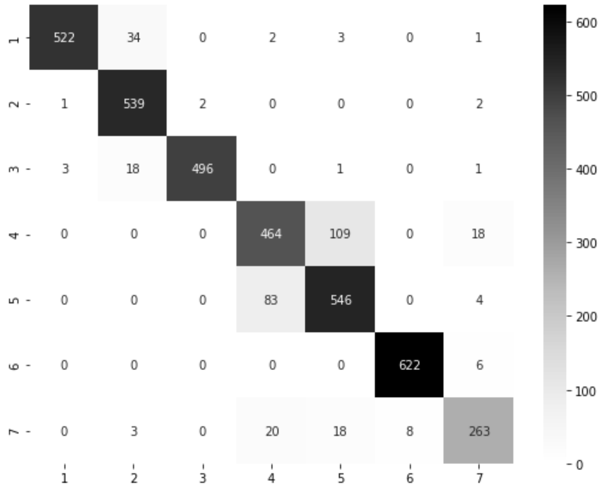

Figure 4.3.8: Confusion Matrix for Type 6

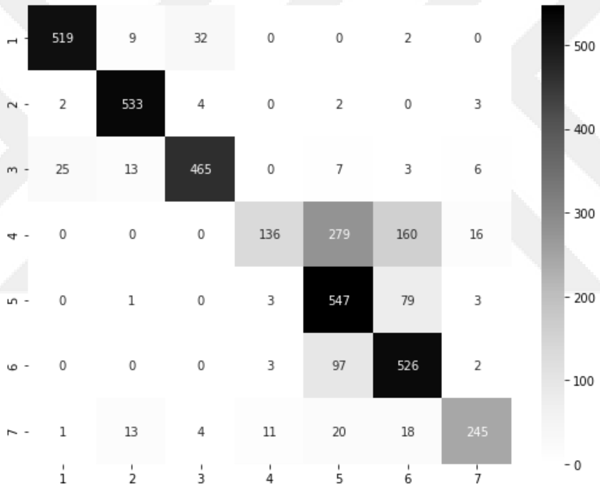

Figure 4.3.9: Confusion Matrix for Type 9

1: Walking, 2: Walking_Upstairs, 3: Walking_Downstairs, 4: Sitting, 5: Standing, 6: Liying, 7: Postural transitions. Figure 4.3.7 shows the confusion matrix of gyroscope and accelerometer data of 7 activities. Figure 4.3.8 shows an estimation of only accelerometer data, and Figure 4.3.9 shows an estimation of only gyroscope data, for 7 activities, and detailed results shown in Table 4.3.1.

Table 4.3.4: Detailed Results for Type 3

Activity ID	F1 Score	Sensitivity	Specificity
1	0.99	0.98	0,99
2	0.97	0.96	0,99
3	0.99	0.99	0,99
4	0.89	0.84	0,99
5	0.92	0.97	0,97
6	0.99	0.99	0,99
7	0.87	0.88	0,98

Table 4.3.4 shows the values obtained from the test results for 7 activities. In general, it was observed that false predictions were made for the same activity, as observed in 6-activity and 12-activity results. Some of the sensitivity and precision values decreased due to incorrect predictions made for the same activities. As a result, the overall accuracy is acceptable as seen in Table 4.3.1.

4.4. Discussion

In literature, a lot of experiments have been applied for the HAR problem but most of them used sensor data after processing the signal and extracting some features from the signal. While researchers doing experiments, they used basic activities such as running, walking, etc. If we want to recognize activities or human behavior, we should research postural transition activities.

Table 4.4.1: The performance comparison with other studies

	Method	Accuracy	Feature Extraction	Activity Number
(Tufek et al., 2020)	1D CNN (5 Layer)	94%	✗	6
(Xia et al., 2020)	LSTM-CNN	95.78%	✓	6
(Arigbabu, 2020)	RCN	97.4%	✓	6
(Cho & Yoon, 2018)	Divide and Conquer-Based 1D CNN	97%	✓	6
(Hassan et al., 2018)	Deep Belief Network	95.45%	✓	12
(Zhao et al., 2018)	LSTM	90%	✓	6
(Ramachandran & Pang, 2020)	CNN+Transfer Learning	94%	✓	6
Proposed Model	1D CNN	96,95%	✗	6
Proposed Model	1D CNN	95.03%	✗	7
Proposed Model	1D CNN	93.08%	✗	12

Some studies, their results, and used methodologies are shown in Table 4.4.1. Most of them used 6 basic activities from the UCI-HAR dataset to recognize human activities and also applied feature extraction to data. (Tufek et al., 2020) used raw data signals for 6 activities from the UCI-HAR dataset. The researchers divided the data into 75% train 25% test data. Their proposed 1D CNN model consists of 2 convolution layers, a max-pooling layer, a fully connected layer, and 2 dropout layers. They use ReLu as the activation function. Arigbau (Arigbabu, 2020) allocated 10% of the UCI-HAR dataset for testing and randomly selected the test data, and achieved 97% performance rate using the 2-layer RCN model. By creating a total of 8 LSTM cells, Zhao et al. (Zhao et al., 2018), proposed a model with 2 hidden layers and determined 70% of the dataset as training data and they obtained an accuracy rate of 90%. Xia et al. (Xia et al., 2020) proposed a 2-layer model with 64 neurons and achieved 95.78% success rate in their model. Cho et al. proposed the Divide and Conquer-Based approach and they obtained 97% accuracy rate using six activities from the UCI-HAR dataset (Cho & Yoon, 2018). Hassan et al. (Hassan et al., 2018) used median and low-pass Butterworth filters on the dataset. They applied statistical feature extraction methods such as mean, median, and correlation to the signals and obtained 561 features. We used raw data in our experiments. We achieved an accuracy rate of 96,95% using our CNN model. We also applied our model to 7 activity classes and 12 activity classes; results are shown in Table 4.3.1. Although we didn't use any complex feature extraction and pre-processing steps, our model achieved promising results. The lack of complex pre-processing and feature extraction steps increases the usability of the model for real-time applications.

5. CONCLUSIONS

HAR process basically consists of 3 steps as data collection, pre-processing, and classification. In the pre-processing step, median filtering was applied to the raw data in the UCI-HAR Dataset. Then, the data was transposed after the windowing step was performed using 128 sized floating windows, and the dataset with 12637 rows and 770 columns were created. Data were divided into groups according to the number of activities and the sensor data included. Groups containing 6 activities include the data of walking, walking upstairs, walking downstairs, sitting, lie and standing activities. Groups containing 12 activities include 6 transition activities besides the basic 6 activities. Groups containing 7 activities includes 6 basic activities and 1 postural transition activities. Since the activity classification problem is a one-dimensional time series problem, 1D CNN was used in our study. Comparisons were made for the learning rate, the optimizer, batch size, and output unit size. The same users were tested to make the comparison fairly (2, 4,9,12,13,26,18,28, and 24 are test users). Thus, 30% of the users were selected as test data and the remaining 70% were used as training data. The best parameters are batch size is 64, the optimizer is Adam, the learning rate is 0.001, hidden unit size is 1024-2048, and epochs is 100.

We didn't apply any mathematical or statistical feature extraction methods to the data. For the experimental results, we implemented 6, 7, and 12 classes activity recognition, and achieved accuracies of 96.95%, 95.03%, and 93.08%, respectively. When we used only accelerometer or gyroscope data, we got low accuracies on our experiments While combining two sensors data, we achieved more acceptable results. As you can see in Table 4.3.1, we obtained highest accuracies for Type 1, Type 2 and Type 3.

6. RECOMMENDATIONS

When the confusion matrices of the algorithms applied in the Experiments and Results section is examined, it is seen that while the algorithms generally predict an activity, their false predictions are on the same activities instead of displaying a general distribution. Therefore, the similarities between the signal data of these activities should be examined and development should be achieved on the same dataset with new studies on this subject. In our future studies, activity recognition problems can be used to detect activity-based anomalies in daily life and real-life applications can be developed.

REFERENCES

Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., & Adeli, H. (2018). Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Computers in Biology and Medicine*. Retrieved from https://doi.org/10.1016/j.compbiomed.2017.09.017

Ailisto, H. J., Lindholm, M., Mantyjarvi, J., Vildjiounaite, E., & Makela, S.-M. (2005). Identifying people from gait pattern with accelerometers. *Biometric Technology for Human Identification II*, 5779(March 2005), 7. Retrieved from https://doi.org/10.1117/12.603331

Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2012). Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7657 LNCS, 216–223. Retrieved from https://doi.org/10.1007/978-3-642-35395-6_30

Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013a). A public domain dataset for human activity recognition using smartphones. *ESANN 2013 Proceedings, 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, (April), 437–442.

Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013b). A public domain dataset for human activity recognition using smartphones. In *ESANN 2013 proceedings, 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

Arigbabu, O. A. (2020). Entropy Decision Fusion for Smartphone Sensor based Human Activity Recognition. Retrieved from http://arxiv.org/abs/2006.00367

Asghari, P., Soleimani, E., & Nazerfard, E. (2020). Online human activity recognition employing hierarchical hidden Markov models. *Journal of Ambient Intelligence and Humanized Computing*, 11(3), 1141–1152. Retrieved from https://doi.org/10.1007/s12652-019-01380-5

Bilal, M., Shaikh, F. K., Arif, M., & Wyne, M. F. (2019). A revised framework of machine learning application for optimal activity recognition. *Cluster Computing*, 22(May), 7257–7273. Retrieved from https://doi.org/10.1007/s10586-017-1212-x

Bodor, R., Jackson, B., & Papanikolopoulos, N. (2003). Vision-Based Human Tracking and Activity Recognition. In *11th Mediterranean Conference on Control and Automation - MED'03.*

Burger, A., Qian, C., Schiele, G., & Helms, D. (2020). An Embedded CNN Implementation for On-Device ECG Analysis. Retrieved from https://doi.org/10.1109/percomworkshops48775.2020.9156260

Cho, H., & Yoon, S. M. (2018). Divide and conquer-based 1D CNN human activity recognition using test data sharpening. *Sensors (Switzerland)*, 18(4), 1–24. Retrieved from https://doi.org/10.3390/s18041055

Clarkson, B., Mase, K., & Pentland, A. (2000). Recognizing user context via wearable sensors. *International Symposium on Wearable Computers, Digest of Papers*. Retrieved from https://doi.org/10.1109/iswc.2000.888467

Das Antar, A., Ahmed, M., & Ahad, M. A. R. (2019). Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: A review. *2019 Joint 8th International Conference on Informatics, Electronics and Vision, ICIEV 2019 and 3rd International Conference on Imaging, Vision and Pattern Recognition, IcIVPR 2019 with International Conference on Activity and Behavior Computing, ABC 2019*, (February 2020), 134–139. Retrieved from https://doi.org/10.1109/ICIEV.2019.8858508

Eyobu, O. S., & Han, D. S. (2018). Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network. *Sensors (Switzerland)*, 18(9), 1–26. Retrieved from https://doi.org/10.3390/s18092892

Fan, S., Jia, Y., & Jia, C. (2019). A feature selection and classification method for activity recognition based on an inertial sensing unit. *Information (Switzerland)*. Retrieved from https://doi.org/10.3390/info10100290

Garlisi, D. (2019). *Human Activities Recognition Using Accelerometer and Gyroscope* (Vol. 3). Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-030-34255-5

Gholamrezaii, M., & Taghi Almodarresi, S. M. (2019). Human Activity Recognition Using 2D Convolutional Neural Networks. *ICEE 2019 - 27th Iranian Conference on Electrical Engineering*, (August), 1682–1686. Retrieved from https://doi.org/10.1109/IranianCEE.2019.8786578

Hassan, M. M., Uddin, M. Z., Mohamed, A., & Almogren, A. (2018). A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, 81, 307–313. Retrieved from https://doi.org/10.1016/j.future.2017.11.029

He, X., Zhu, J., Su, W., & Tentzeris, M. M. (2020). RFID Based Non-Contact Human Activity Detection Exploiting Cross Polarization. *IEEE Access*, 8. Retrieved from https://doi.org/10.1109/ACCESS.2020.2979080

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *32nd International Conference on Machine Learning, ICML 2015*.

Irvine, N., Nugent, C., Zhang, S., Wang, H., & Ng, W. W. Y. (2020). Neural network ensembles for sensor-based human activity recognition within smart environments. *Sensors (Switzerland)*. Retrieved from https://doi.org/10.3390/s20010216

Israel, S. A., Irvine, J. M., Cheng, A., Wiederhold, M. D., & Wiederhold, B. K. (2005). ECG to identify individuals. *Pattern Recognition*, 38(1), 133–142. Retrieved from https://doi.org/10.1016/j.patcog.2004.05.014

Jalal, A., Quaid, M. A. K., & Hasan, A. S. (2019). Wearable sensor-based human behavior understanding and recognition in daily life for smart environments. *Proceedings - 2018 International Conference on Frontiers of Information Technology, FIT 2018*, (November), 105–110. Retrieved from https://doi.org/10.1109/FIT.2018.00026

Kim, E., & Helal, S. (2014). Training-free fuzzy logic based human activity recognition. *Journal of Information Processing Systems*, 10(3), 335–354. Retrieved from https://doi.org/10.3745/JIPS.04.0005

Kırkan, Samet;Karabey, Işıl;Bayındır, L. (2016). Pantolon Cebinde Ta şı nan Ak ı ll ı Telefonun İ vmeölçer Verilerini Kullanarak Aktivite Tan ı ma, (October 2015).

Lan, T., Sigal, L., & Mori, G. (2012). Social roles in hierarchical models for human activity recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Retrieved from https://doi.org/10.1109/CVPR.2012.6247821

Le, T. D., & Van Nguyen, C. (2015). Human activity recognition by smartphone. In *Proceedings of 2015 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science, NICS 2015*. Retrieved from https://doi.org/10.1109/NICS.2015.7302194

Li, F., Shirahama, K., Nisar, M. A., Köping, L., & Grzegorzek, M. (2018). Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors (Switzerland)*. Retrieved from https://doi.org/10.3390/s18020679

Li, G., Huang, L., & Xu, H. (2017). iWalk: Let your smartphone remember you. *Proceedings - 2017 4th International Conference on Information Science and Control Engineering, ICISCE 2017*, 414–418. Retrieved from https://doi.org/10.1109/ICISCE.2017.93

Liu, J., Spakowicz, D. J., Hoyd, R., Ash, G. I., Lou, S., Lee, D., … Gerstein, M. (2020). Bayesian Structural Time Series for Mobile Health and Sensor Data : A Flexible Modeling Framework for Evaluating Interventions, 1–12.

Liu, M., Zeng, A., Lai, Q., & Xu, Q. (2020). T-WaveNet: Tree-Structured Wavelet Neural Network for Sensor-Based Time Series Analysis. Retrieved from http://arxiv.org/abs/2012.05456

Madabhushi, A., & Aggarwal, J. K. (1999). A Bayesian approach to human activity recognition. In *Proceedings - 2nd IEEE International Workshop on Visual Surveillance, VS 1999*. Retrieved from https://doi.org/10.1109/VS.1999.780265

Martinez, H. P., Yannakakis, G. N., & Hallam, J. (2014). Don't classify ratings of affect; Rank Them! *IEEE Transactions on Affective Computing*. Retrieved from https://doi.org/10.1109/TAFFC.2014.2352268

Micucci, D., Mobilio, M., & Napoletano, P. (2017). UniMiB SHAR: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences (Switzerland)*, 7(10). Retrieved from https://doi.org/10.3390/app7101101

Milosevic, N. (2020). *Introduction to Convolutional Neural Networks*. *Introduction to Convolutional Neural Networks*. Retrieved from https://doi.org/10.1007/978-1-4842-5648-0

Moradi, B., Aghapour, M., & Shirbandi, A. (2019). Compare of Machine Learning And Deep Learning Approaches for Human Activity Recognition.

Myo, W. W., Wettayaprasit, W., & Aiyarak, P. (2019). Designing classifier for human activity recognition using artificial neural network. *2019 IEEE 4th International Conference on Computer and Communication Systems, ICCCS 2019*, (May), 81–85. Retrieved from https://doi.org/10.1109/CCOMS.2019.8821638

Neverova, N., Wolf, C., Lacey, G., Fridman, L. E. X., Chandra, D., Barbello, B., & Taylor, G. (2016). Learning Human Identity From Motion Patterns. *IEEE Access*, 4, 1810–1820. Retrieved from https://doi.org/10.1109/ACCESS.2016.2557846

Ni, B., Moulin, P., Yang, X., & Yan, S. (2015). Motion Part Regularization: Improving action recognition via trajectory group selection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Retrieved from https://doi.org/10.1109/CVPR.2015.7298993

Niu, F., & Abdel-Mottaleb, M. (2005). HMM-based segmentation and recognition of human activities from video sequences. In *IEEE International Conference on Multimedia and Expo, ICME 2005*. Retrieved from https://doi.org/10.1109/ICME.2005.1521545

Nowlan, M. F. (2009). Human Identification via Gait Recognition Using Accelerometer Gyro Forces Michael Fitzgerald Nowlan CPSC-536-Networked Embedded Systems and Sensor Networks Professor Savvides Fall 2009 2 . RELATED WORK AND HUMAN GAIT.

Okwu, M. O., & Tartibu, L. K. (2021). Artificial Neural Network. In *Studies in Computational Intelligence*. Retrieved from https://doi.org/10.1007/978-3-030-61111-8_14

Park, S. U., Park, J. H., Al-Masni, M. A., Al-Antari, M. A., Uddin, M. Z., & Kim, T. S. (2016). A Depth Camera-based Human Activity Recognition via Deep Learning Recurrent Neural Network for Health and Social Care Services. In *Procedia Computer Science*. Retrieved from https://doi.org/10.1016/j.procs.2016.09.126

Patron-Perez, A., Marszalek, M., Reid, I., & Zisserman, A. (2012). Structured learning of human interactions in TV shows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Retrieved from https://doi.org/10.1109/TPAMI.2012.24

Ramachandran, K., & Pang, J. (2020). Transfer Learning Technique for Human Activity Recognition based on Smartphone Data, 11(1), 1–17.

Reiss, A., Hendeby, G., & Stricker, D. (2013). A competitive approach for human activity recognition on smartphones. In *ESANN 2013 proceedings, 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

Reiss, A., & Stricker, D. (2012a). Creating and benchmarking a new dataset for physical activity monitoring. In *ACM International Conference Proceeding Series*. Retrieved from https://doi.org/10.1145/2413097.2413148

Reiss, A., & Stricker, D. (2012b). Introducing a new benchmarked dataset for activity monitoring. *Proceedings - International Symposium on Wearable Computers, ISWC,* 108–109. Retrieved from https://doi.org/10.1109/ISWC.2012.13

Robinovitch, S. N., Feldman, F., Yang, Y., Schonnop, R., Leung, P. M., Sarraf, T., … Loughi, M. (2013). Video capture of the circumstances of falls in elderly people residing in long-term care: An observational study. *The Lancet*. Retrieved from https://doi.org/10.1016/S0140-6736(12)61263-X

Sharma, S., Sharma, S., & Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology*. Retrieved from https://doi.org/10.33564/ijeast.2020.v04i12.054

Shieh, W. Y., & Huang, J. C. (2009). Speedup the multi-camera video-surveillance system for elder falling detection. In *Proceedings - 2009 International Conference on Embedded Software and Systems, ICESS 2009*. Retrieved from https://doi.org/10.1109/ICESS.2009.62

Sikder, N., Chowdhury, M. S., Arif, A. S. M., & Nahid, A. Al. (2019). Human activity recognition using multichannel convolutional neural network. In *2019 5th International Conference on Advances in Electrical Engineering, ICAEE 2019*. Retrieved from https://doi.org/10.1109/ICAEE48663.2019.8975649

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Taghanaki, S. R., & Etemad, A. (2020). Self-supervised Wearable-based Activity Recognition by Learning to Forecast Motion. Retrieved from http://arxiv.org/abs/2010.13713

Terrier, P., & Reynard, F. (2015). Effect of age on the variability and stability of gait: A cross-sectional treadmill study in healthy individuals between 20 and 69 years of age. *Gait and Posture,* 41(1), 170–174. Retrieved from https://doi.org/10.1016/j.gaitpost.2014.09.024

Topuz, E. K., & Kaya, Y. (2019). B iyolojik Sinyaller ve Taşınabilir Sensörlerden Kişi Tanıma : Bir Literatür Taraması Identity Recognition from Wearable Sensors and Biological Signals : A Literature Review.

Tran, K. N., Gala, A., Kakadiaris, I. A., & Shah, S. K. (2014). Activity analysis in crowded environments using social cues for group discovery and human interaction modeling. *Pattern Recognition Letters*. Retrieved from https://doi.org/10.1016/j.patrec.2013.09.015

Tufek, N., Yalcin, M., Altintas, M., Kalaoglu, F., Li, Y., & Bahadir, S. K. (2020). Human Action Recognition Using Deep Learning Methods on Limited Sensory Data. *IEEE Sensors Journal*, 20(6). Retrieved from https://doi.org/10.1109/JSEN.2019.2956901

Vrigkas, M., Nikou, C., & Kakadiaris, I. A. (2015). A review of human activity recognition methods. *Frontiers Robotics AI*. Retrieved from https://doi.org/10.3389/frobt.2015.00028

Walse, K. H., Dharaskar, R. V., & Thakare, V. M. (2016). PCA based optimal ANN classifiers for human activity recognition using mobile sensors data. In *Smart Innovation, Systems and Technologies*. Retrieved from https://doi.org/10.1007/978-3-319-30933-0_43

Walse, K. H., Dharaskar, R. V., & Thakare, V. M. (2017). A study on the effect of adaptive boosting on performance of classifiers for human activity recognition. In *Advances in Intelligent Systems and Computing*. Retrieved from https://doi.org/10.1007/978-981-10-1678-3_41

Weiss, G. M. (2019). WISDM Smartphone and Smartwatch Activity and Biometrics Dataset. *UCI Machine Learning Repository: WISDM Smartphone and Smartwatch Activity and Biometrics Dataset Data Set*.

Weiss, G. M., Yoneda, K., & Hayajneh, T. (2019). Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living. *IEEE Access*. Retrieved from https://doi.org/10.1109/ACCESS.2019.2940729

Wu, W., & Zhang, Y. (2019). Activity recognition from mobile phone using deep CNN. *Chinese Control Conference, CCC*, 2019-July, 7786–7790. Retrieved from https://doi.org/10.23919/ChiCC.2019.8865142

Xia, K., Huang, J., & Wang, H. (2020). LSTM-CNN Architecture for Human Activity Recognition. *IEEE Access*, 8, 56855–56866. Retrieved from https://doi.org/10.1109/ACCESS.2020.2982225

Yang, Y., Saleemi, I., & Shah, M. (2013). Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Retrieved from https://doi.org/10.1109/TPAMI.2012.253

Zaki, Z. (2020). Logistic Regression Based Human Activities Recognition. *Journal of Mechanics of Continua and Mathematical Sciences*, 15(4). Retrieved from https://doi.org/10.26782/jmcms.2020.04.00018

Zhang, L., Hua, Y., Cotton, S. L., Yoo, S. K., Da Silva, C. R. C. M., & Scanlon, W. G. (2020). An RSS-Based Classification of User Equipment Usage in Indoor Millimeter Wave Wireless Networks Using Machine Learning. *IEEE Access*, 8. Retrieved from https://doi.org/10.1109/ACCESS.2020.2966123

Zhang, M. (2019). Gait activity authentication using LSTM neural networks with smartphone sensors. *Proceedings - 2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2019*, (2018), 456–461. Retrieved from https://doi.org/10.1109/MSN48538.2019.00092

Zhao, Y., Yang, R., Chevalier, G., Xu, X., & Zhang, Z. (2018). Deep Residual Bidir-LSTM for Human Activity Recognition Using Wearable Sensors. *Mathematical Problems in Engineering*, 2018. Retrieved from https://doi.org/10.1155/2018/7316954

Zhu, J., Chen, H., & Ye, W. (2020). A Hybrid CNN-LSTM Network for the Classification of Human Activities Based on Micro-Doppler Radar. *IEEE Access*, 8. Retrieved from https://doi.org/10.1109/ACCESS.2020.2971064