REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

IMPROVING HEALTHCARE FACILITIES BY USING ARTIFICIAL INTELLIGENCE METHODS

Majid Mohammed Ridha ALTAWEEL

Master’s Thesis

Supervisor

Prof. Dr. Osman Nuri UÇAN

Istanbul, 2022

IMPROVING HEALTHCARE FACILITIES BY USING ARTIFICIAL INTELLIGENCE METHODS

Majid Mohammed Ridha ALTAWEEL

Electrical and Computer Engineering

Master's Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titled IMPROVING HEALTHCARE FACILITIES BY USING ARTIFICIAL INTELLIGENCE METHODS prepared by MAJID MOHAMMED RIDHA ALTAWEEL and submitted on 16/12/2022 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

Prof. Dr. Osman Nurı UÇAN
the Supervisor

Thesis Defense Committee Members:

Prof. Dr. Osman Nurı UÇAN	Department of Electrical and Electronics Engineering, Altınbaş University	__________________
Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	Department of Computer Engineering, Altınbaş University	__________________
Assoc. Prof. Dr. Adil Denız DURU	Department of Trainer Education Marmara University	__________________

I hereby declare that this thesis meets all format and submission requirements of a Master’s thesis.
Submission data of the thesis to Institute of Graduate Studies: ____/____/____

I hereby declare that all of the data/information that had been presented in the present graduation project has been obtained in full accordance with the ethical conduct and academic rules. I declare as well that all of the unoriginal materials and conclusions were cited in the text and all of the references that have been mentioned in the Reference List were cited in the text, and vice versa as required by above-mentioned conduct and rules.

Majid Mohammed Ridha ALTAWEEL

Signature

DEDICATION

To my Parents, the reason of what I have become today. To my sister and brothers. To my Dear wife, you have been my inspiration, and my soul mate. To my lovely Son or daughter if any and, special thanks to Prof. Dr. Osman Nuri UÇAN for his worth notes, support, and close follow up.

ABSTRACT

IMPROVING HEALTHCARE FACILITIES BY USING ARTIFICIAL INTELLIGENCE METHODS

ALTAWEEL, Majid Mohammed Ridha

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Prof. Dr. Osman Nurı UÇAN

Date: 12/2022

Page: 52

In order to determine the mobility of a wheelchair user and support them in leading an active life-style, it is crucial to monitor their movement. According to research, individuals with disabilities are more likely to lead sedentary lifestyle and health issues which go along with them, such as obesity, cardiovascular disease, and the occurrence of pressure ulcers [1]. Additionally, sedentary lifestyle was linked to the development of pressure ulcers as well as increased risks of obesity. The usage of external sensors, which should be acquired and maintained, is required by the technology now available for assessing wheelchair user movement data [2]. A wheelchair user could make use of current technology, like the smart mobile devices, for collecting and analyzing motion data, which will enhance how easily mobility data is maintained and evaluated. In the presented work, the creation of recurrent neural network (RNN) which has been trained with the use of information about wheelchair users obtained from the smart devices that have been attached to wheelchair user or the wheelchair is the main goal. As the majority of wheelchair users will have the access to smart devices that has the ability of capturing the data related to movement, this method of data collection has the advantage of not requiring the usage of additional equipment or sensors. The work discovered that utilizing an RNN, it has been possible to interpret smart device data in a useful way. RNN is used to evaluate the raw data and get insight into a wheelchair user's mobility. The overall amount of the time that has been spent moving, the lengthiest bout of the movement, and the number of bouts of movement are all included in the final analysis. Healthcare experts or wheelchair users might examine healthy lifestyle practices using the data that was produced.

Keywords: Artificial Intelligence (AI), Wheelchairs, Recurrent Neural Network (RNN), Mobility Assessment, Smart Devices.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

1. INTRODUCTION

1.1 BACKGROUND

People who lead inactive lifestyles are more likely to experience various health problems. Obesity, Cardiovascular disease (CVD), and hypertension are a few of such problems [3]. According to the World Health Organization (WHO), factors connected to physical inactivity account for about two million mortalities annually. Particularly wheelchair users are more prone to maintain sedentary lives that raise their risk for developing such diseases as well as other problems as the development of pressure ulcers [4]. Self-monitoring physical activities with real-time feedback utilizing sensor-based technologies aid weight loss in persons who are overweight or obese and are sedentary [5]. Self-monitoring was shown to boost people's physical activity data, but gathering and analyzing the data of the movement —particularly for the users of the wheelchair—is challenging to manually perform and typically needs using technology and sensors. Through using already-existing sensors in the smart devices that are carried by wheelchair users, such as smartphones or smartwatches, this work intends to gather as well as analyze the data of the movement from the users of wheelchairs with no additional sensors. By using algorithms of deep learning (DL) to categorize movement features of wheelchairs, this work will improve on existing techniques for gathering and evaluating the data of the wheelchair movement from the smart device sensors. Those DL algorithms will generate more precise data of movement and have an advantage of operating in real-time and enabling wheelchair users to receive rapid feedback.

1.1.1 Traditional Movement Analyses

The ability to record human movement for data analysis has advanced along with technology. A few of the first researchers that have utilized the photography for the measurement and evaluation of the human movements were [6] and [7] [8]. With sensor-based technologies, currently, there is a possibility for recording and analyzing human movement data in real time. Pedometers are examples of the devices that use electronics to track people's steps. The way pedometer function is by swinging a tiny metal pendulum that is connected to an electronic counting circuit to record an individual's step movement [9]. Due to their common characteristics of being compact, affordable, and integrated into other devices, those and comparable devices are particularly useful for obtaining movement data. The inbuilt sensors regarding a phone are used by fitness apps for smart phones like My FitnessPal and Google Fit to collect activity data. Those applications record the longitudinal, lateral, and vertical movements and evaluate steps using accelerometer and gyroscope [10]. Pedometers and other comparable devices collect helpful movement data when people are walking, yet they are not practical for those whose principal mode of transportation is a wheelchair. In contrast to an individual walking, whose hip motions are utilized in order to record data of movements, an individual operating a motorized wheelchair or propelling a manual wheelchair cannot be tracked in the same way.

1.1.2 Characteristics of Wheelchair Movement

The movements of a person using a wheelchair differ significantly from those of a person walking. Users of manual wheelchairs mostly propulsion themselves with their arms. While the body moves constantly throughout a regular walking cycle, wheelchair users could coast with momentum that had been provided by a push to continue the movement without the movement of any body parts. Two separate steps are involved in moving a wheelchair: the phase of propulsion, during which extremities are moved in order to create motion, and the period of recovery, during which the hands are not in close contact with wheels [11]. Depending on the severity of the handicap and the wheelchair user's skill, various propulsion patterns have been detected by certain research. Given that just the hand needed for pressing on the joystick to cause propulsion, powered wheelchair users could experience little to no limb movement when the chair is going. Due to such distinctions, wheelchair users cannot use conventional approaches for collecting and evaluating mobility data.

1.1.3 Movement Data from the Smart Devices

Various sensors are built into modern smart phones to gather movement data. Those sensors include gyroscopes, accelerometers, and GPS trackers. Smart phones with the GPS are normally accurate to within sixteen feet of open sky, with the precision dwindling in structures, beneath trees, and so on. [12]. Device movement is measured using gyroscopes and accelerometers. Those measures reflect not just the phone's movements, yet also the motions regarding the environment that the phone is in. Those movements include shakes, rotation, swings, and so on.

The movement data related to individuals using smart phones or movement data regarding wheelchair which a smart phone had been attached to can now be recorded thanks to this.

1.2 HEALTH ISSUES RELATED TO WHEELCHAIR USERS AND INACTIVITY

The necessity for the analysis of movement stems from knowledge that the users f wheelchairs who walk more frequently and for longer periods of time have a lower possibility to lead inactive life-styles compared to those that move for shorter intervals. Women who spent the most time sitting down were more likely to develop a number of fatal diseases, according to a 12-year study of female participants [13]. According to the work's findings, in the case when an individual spent a lot of time sitting down, their risk of disease has been higher than if they followed a healthy diet and exercise regularly. Maintaining an active lifestyle will help lessen the chance of developing such diseases even more for wheelchair users who are unable to utilize their lower extremities.

1.2.1 Cardiovascular Disease, Diabetes, and Obesity

CVD is a catch-all phrase for a variety of illnesses that have an impact on the heart. Issues with the heart's rhythm, coronary artery disease, and heart abnormalities are a few of the diseases that come within this category. The phrase "cardiovascular disease" is typically applied to describe ailments involving obstructed blood vessels that are either blocked or narrowed and cause a heart attack. The main cause of this is typically an accumulation of the fatty plaques in the arteries that block the flow of the blood to the tissues and organs. Which represents the most frequent CVD cause and is brought on by issues that include inactivity, poor diet, smoking, and being overweight. The users of the wheelchairs have a higher possibility to be sedentary, which increases their risk of developing such disease. Wheelchair users are more likely to become obese due to a lack of exercise. The average American obesity is becoming more obese, according to researches, yet persons with disabilities are particularly at risk. According to estimates, between 25 and 31% of adults with disabilities are obese, compared to 15 to 19% of adults without disabilities [14]. Obesity is related to high risks for other illnesses, such as the high blood pressure, diabetes, and specific cancer types.

An increase in fatty tissue, lack of exercise-related weight growth, and the way that an individual with disabilities body processes insulin all raise their risk for developing diabetes. Also, diabetes happens in the case when blood glucose levels are too high because pancreas is no longer producing a sufficient amount of the insulin to help glucose reach cells of the body. Diabetes-related high blood glucose raises the risk of additional health concerns such as vision problems, kidney disease, and stroke. All diseases mentioned are linked, among other things, to sedentary habits and inactivity. Active lifestyles have been found to help lower the risk of developing certain diseases, even if it might not be feasible to totally decrease the risk of developing such diseases. This is why it's crucial to accurately track wheelchair users' movement data. Wheelchair users who keeps track of their movements and gives their healthcare doctors this information will be able to evaluate their existing way of life and determine whether any data are required for ensuring healthy amounts of activity. When using a manual wheelchair, the act of propulsion might burn up to 120 calories in 30 minutes, which is three times more than when using a motorized wheelchair [15]. Moving more often is linked to higher levels of physical activity, even though movement alone is insufficient to evaluate a person's lifestyle. The mobility data could be useful in establishing how much individual's movement contributes to their overall lifestyle when used in collaboration with a healthcare practitioner.

1.2.2 Pressure Ulcer Formation

Wheelchair users, particularly those who utilize powered wheelchairs, run the risk of getting pressure ulcers along with the hazards related to sedentary lifestyles. Long-term pressure on the skin damages underlying tissues and the skin and leads to pressure ulcers. In addition, pressure ulcers are far more prone to form in powered wheelchair users, a lot of them lack feeling in lower bodies [16]. Exercises tailored to ease tension on the lower body and lessen pressure on particular places are mostly effective in preventing pressure ulcers. Conventional pressure ulcer prevention methods rely heavily on assessments via medical professionals and are not suitable for usage at home by people who are wheelchair-bound [17]. Additionally, there is pricey equipment for automatic workouts that is mostly utilized in hospitals and involves moving a person's bed or seat on its own to help prevent pressure ulcers. Through enabling wheelchair users to automatically track the time amount that is spent exercising, the presented work will also try to bridge the gap in the prevention of the pressure ulcer for those who use them. This data will allow users to determine whether they are spending enough time exercising, and they might utilize it with a healthcare professional for additional direction.

2. ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING (ML)

2.1 BACKGROUND

Artificial intelligence (AI) is needed to analyze the features of vast datasets that the sensors offer. Data that the sensors provide is noisy and inconsistent. RNNs in particular, which use AI, are capable of quickly analyzing data sequence from the sensors and producing insightful results [18].

2.1.1 AI Overview

Algorithms with the AI are those which use input data to learn from and improve upon themselves. The way they differ from conventional algorithms is that they don't adhere to rigid guidelines or provide predetermined results. AI algorithms gather data from various sources, analyze it, and then generate responses depending on this analysis. The advantage of adopting AI instead of other algorithms of the data processing is their capacity to work with the types od data which are noisy or inconsistent from time to time. Large data sets can now be instantaneously categorized by AI algorithms, eliminating the necessity for the laborious manual classifications.

2.1.2 Introduction to ML

ML can be defined as a branch of AI where a system learns how to get better on its own without being specifically programmed to do that. The way that the algorithms of machine learning learn is through creating models out of the sample data.

2.2 MACHINE LEARNING

In order to build a mathematical model that may be utilized for making decision or predictions based on the input data without being instructed what to do explicitly, the approaches of ML utilize training sets of data. The more data ML algorithms are trained on, and the greater number of the predictions that they make, the more accurate they become. ML algorithms are employed in a variety of applications, including self-driving cars, image recognition, computer vision, and speech recognition [19]. ML algorithms' objective is taking in a lot of the sensor data for analyzing wheelchair movement and learn the way for classifying the data for giving properties regarding a wheelchair's movement.

2.2.1 ML Approaches

There are numerous ML algorithm varieties that vary depending upon how they classify data, the kinds of data they deal with, and the issues they are intended to address. Algorithms of various types learn in various ways, leading to various application uses. Every kind of algorithm was taken into consideration, and the choice was ultimately decided depending on the data which would be accessible for the specific problem statement. In specifically, devices read sensor data from it and output it in raw form. Data will be timestamped as well as arranged sequentially. Due to the available sensors' sensitivity to slight vibrations and limitations in accuracy, the data will possibly include noise. Additionally, a training set will be made accessible with raw data that has been accurately classified as static or moving. With this information, many algorithms of learning have been indicated, taking into account which algorithms are appropriate for the work. Algorithms for the unsupervised learning are a type that is taken into consideration.

On datasets in which the input data is unlabeled, unsupervised algorithms are employed. The algorithm builds a model that looks for existing structures in data and extracts rules for the structuring of new data depending upon those structures. For instance, the K-means clustering algorithm [20] might take a raw dataset then divide it to objects depending upon the distance between every data point as well as other points.

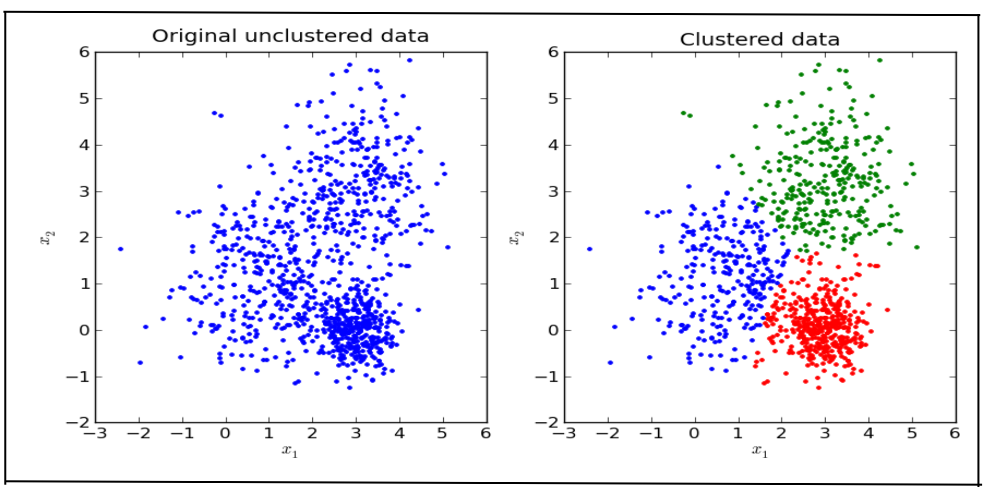

Figure 2.1: K-means clustering example [20]

Although this could theoretically be possible to arrange the raw sensor data according to the intensity related to sensor readings, it might sacrifice data's sequential ordering and make additional analyses challenging. Another kind of algorithm under consideration is the reinforcement learning algorithm. In addition, reinforcement learning algorithms learn to maximize reward for some specific task [21]. According to metrics set forth via the programmer, this algorithm attempts at making a set of the decisions and it’s either penalized or rewarded for such decisions. The algorithm gradually learns which course of set will result in the highest reward. Due to the difficulty in representing data as collection of the decisions with the performance-based metrics, such type of learning isn't adequately suited to the analysis of sensor data. Algorithms for reinforcement learning were therefore not taken into account for this work. The final category of algorithm taken into consideration is supervised learning algorithms. They learn knowledge though using a group of the training data in which the input data are mapped to accurate output data [22]. Iterative training of supervised learning algorithms produces results depending on at least one input and an objective function. In the event that the output doesn’t match training data, the goal function is changed, and the procedure is repeated. Which keeps on until either objective function fails to improve in accuracy after a predetermined number of iterations, or the objective function reaches the set accuracy. Due of the learning algorithm's access to a training set, this method of learning is appropriate for the given problem. This work's topic has been selected for supervised learning since it can use a training set, learn from previous experiments, and work with sequential data.

2.3 MSUPERVISED LEARNING

There are several supervised learning implementations to take into consideration, even though supervised learning has been chosen as optimal contender for this work. There isn’t any optimum algorithm regarding all of the problem statements, according to the "No Free Lunch" theorem [23]. A number of criteria, such as the complexity and quantity of training data, training data set variance, the noise amount in output values, and so on, must be taken into consideration when selecting a suitable algorithm of supervised learning for some certain problem statement. When deciding on the best algorithm, a few prior research that used ML for the analysis have been taken into account.

2.3.1 K-Nearest Neighbor

For instance, KNN was utilized in a 2013 study on wheelchair mobility features [24]. KNN organizes the data by measuring distance between each point in a collection of data points. The group had used a KNN classification approach, which assigns every point a category depending upon the vast majority of the nearby points. The group was successful in reducing noise in raw sensor data through employing such supervised algorithm.

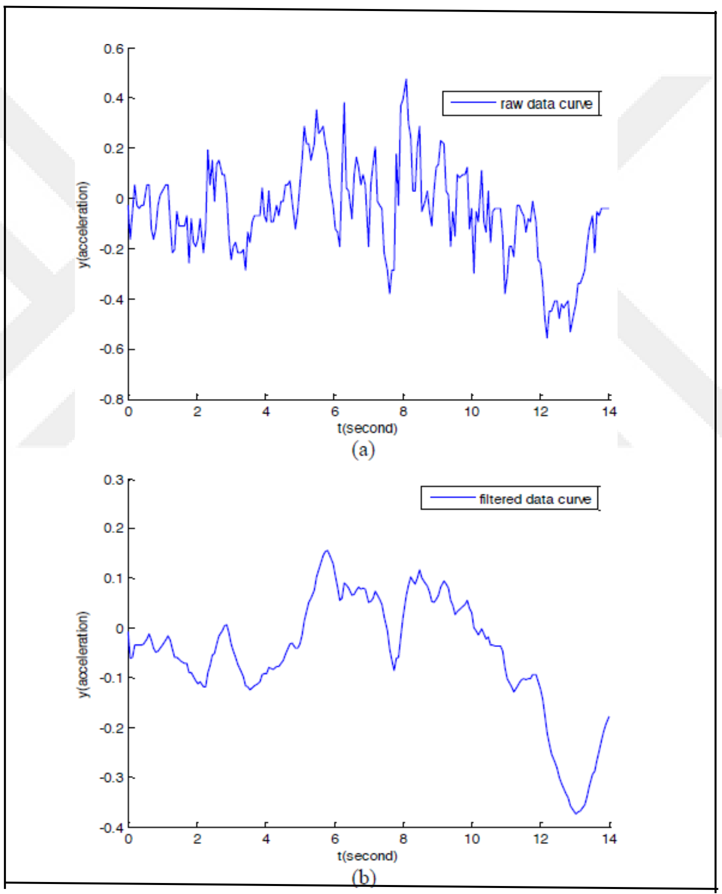

Figure 2.2: KNN smoothed data [24]

The group after that makes an approximation of a wheelchair's movements using smoothed data from the gyroscope and accelerometer. Wheelchair movements in batches gathered and examined following the movements have been conducted were successfully classified by this work. Yet, other than to make points regarding each data point's nearby points, which have been sequentially sorted in the referenced study, KNN does not account for the sequential ordering related to the data. But data that had been gathered for the present study is time-stamped, allowing it to be utilized with a learning algorithm which incorporates memory. Also, the algorithm utilized in the cited paper has been just capable to be utilized after the data has been gathered in batches, while the algorithm had to relearn as well as smooth curve regarding every one of the data batches. An algorithm which could be taught in advance and classify the data as it's received from sensors is required for a purpose of obtaining real-time data analysis with regard to wheelchair movements.

2.3.2 Linear Regression

In a different work that examined wheelchair movements, a research team have obtained sensor data and put data through algorithm of curve fitting to create regression curve depending upon data of sensors [25]. In order to generate a predicted output value, linear regression algorithms construct linear equation from a group of the input values. An easy regression equation can take the following form:

$$y = B_0 + B_1 x$$

In which y represent the predicted output depending on the input value x, the coefficients B0, and B1. A machine learned linear regression will alter coefficients in an effort to generate a line which best matches data. It is possible to utilize more input variables to possibly boost the linear regression algorithm's accuracy, yet it must be noted that utilizing an excessive number of the inputs may overfit the data and reduce the algorithm's effectiveness on new data. The group created a line with peaks in data, representing periods of the wheelchair motions using an algorithm of linear regression on updated sensor data version.

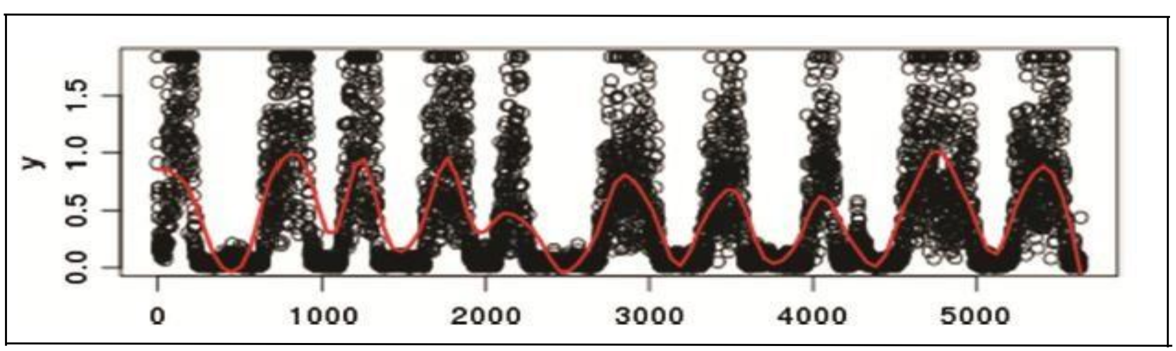

Figure 2.3: Linear regression curve fitting [26]

This error of wheelchair movement analysis was mainly effective, it did make a few errors, with an average error of 19%, in accurately estimating the length of movement in each bout. This error occurred because the length of a bout was approximated using the updated data set's valleys and peaks. In order to create more precise bout duration readings for the investigation, a more complicated learning algorithm that can consider the time-stamped data will be taken into consideration.

2.3.3 Neural Networks

Biological neural networks (NNs), such as those in the human brain, provide the basis for artificial neural networks (ANNs), which can be defined as supervised learning algorithms. They function through assembling a network of neurons, each of which generates an output after receiving at least one weighted input.

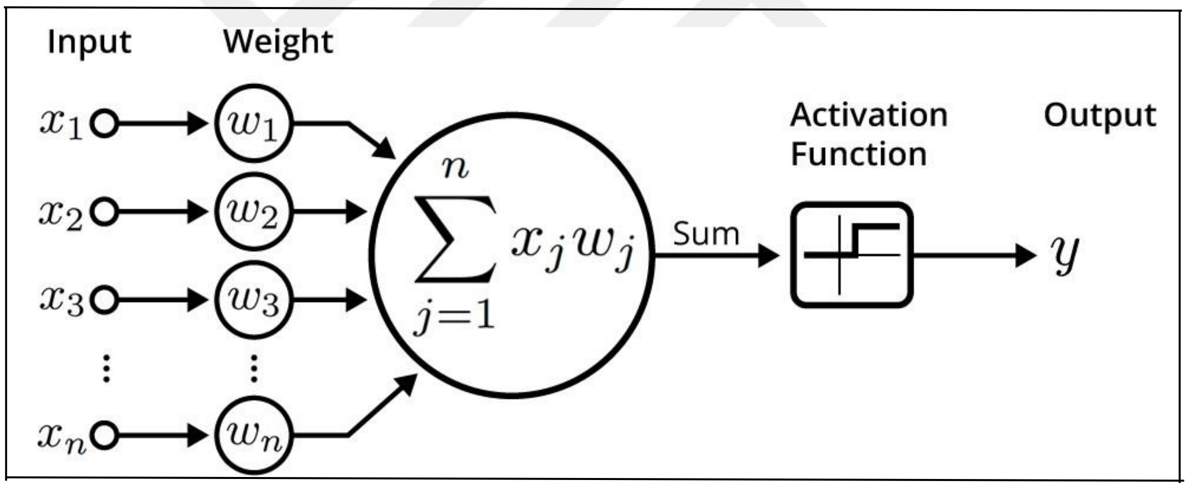

Figure 2.4: Artificial neuron [27]

Neurons are organized into layers, each of which has an output layer, an input layer, and typically at least one hidden layer. Also, input layer receives information from original source, and this information is after that sent down the hidden layers in which it is acted on through internal weighing system which modifies values and generates output. To improve accuracy, NNs learn weight values of hidden layers through putting to comparison the outputted values to the training values. Backpropagation is used to make this modification, in which the output error is returned via the network to change weights. One advantage of a NN is that it could swiftly classify new data in real-time after initial network, negating the need for additional network. This is advantageous for this work because it enables the network to automatically process new sensor data and classify it for real-time movement analyses. Neurons can utilize the activation functions along with the real-time analysis in order to help with the task of data classification. This enables the output to be converted into binary classifications regarding stationary or moving using sigmoid functions. There are many widely used NNs, with feedforward NNs being the most often used. In such networks, which are the most fundamental kind of NN, data only moves forward in the network. Another type of NN, convolutional neural networks (CNNs), are employed mostly in operations like image classification [24].

2.3.4 Recurrent Neural Networks (RNN)

RNNs are an NN type, which excel at processing sequential data. RNNs conduct tasks sequentially, in which one node's output depends upon prior calculations, whereas a typical NNs believes that all inputs are fully independent.

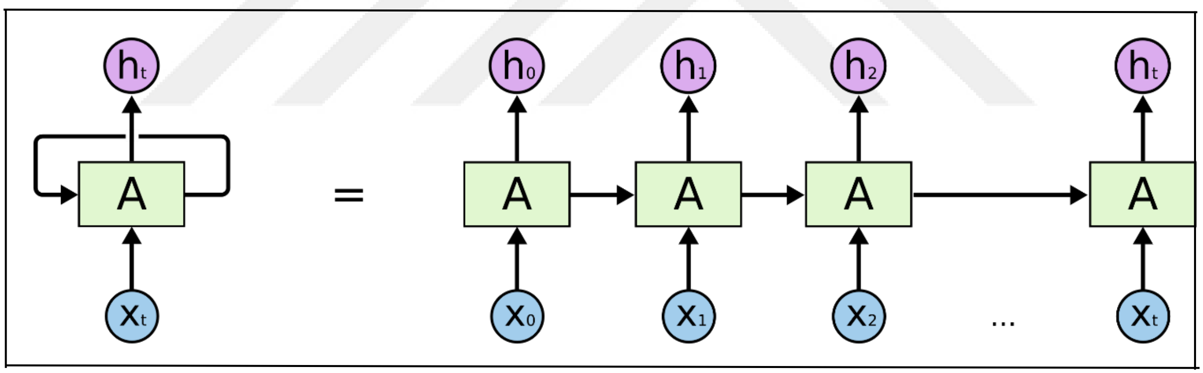

Figure 2.5: Unrolled RNN [20]

As can be seen in the above figure, the data in an RNN has been looped so that certain data from each one of neurons is sent back to neuron. As a result, the network effectively becomes a collection of similar networks, each one with unique outputs and inputs which pass information to the following network. The capacity to transmit temporal information is important for this work's objectives since the current state related to a wheelchair's movement features affects its subsequent state. Friction, the chair's arms propulsion of the wheel, or the motor's acceleration all have an impact on the chair's movement. For example, in the case when the accelerometer detects a high value of acceleration in x direction, it's possible that the following value that is detected by a sensor will be comparable, possibly lower or higher based on the wheelchair's current state of motion.

2.3.5 LSTM Networks

RNNs are further refined in LSTM networks. RNNs might theoretically retain memories of prior inputs, but in practice they typically struggle to maintain long-term dependence [28]. By employing a module with some different structure compared to conventional modules of the RNN, LSTM networks address this problem. LSTM modules contain four interacting network layers as opposed to the typical single function structure regarding conventional RNN modules.

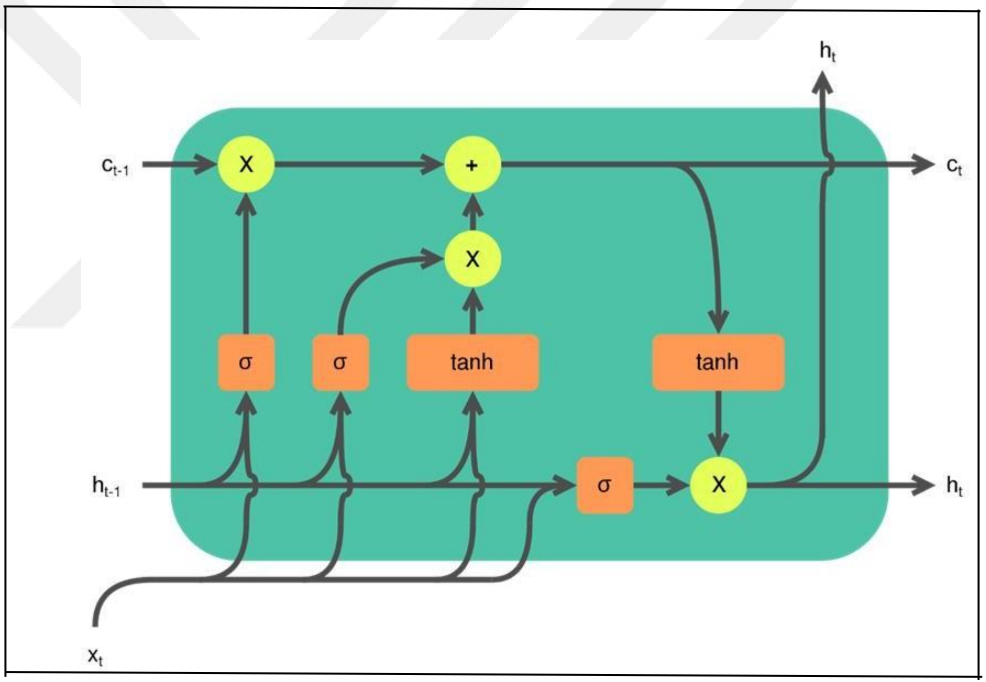

Figure 2.6: LSTM cell structure [25]

Cells are a common name for LSTM modules. The network's whole structure is represented by the cell state ct, to which the LSTM removes or adds data at each cell. Determining whether information at cell state must be forgotten or kept is the 1st LSTM stage. "Forget gate layer" represents the first layer of LSTM. It is a sigmoid layer that produces a value in the range from 0 to 1 after taking input vector of the current cell xt and hidden state vector from previous cell $ht-1$. A 0 means "completely forget this state," whereas 1 means "completely remember that state." This is in which information of current movement state is forgotten or remembered in this work. The cell might forget moving state which has been transferred into the new cell in the case when it appears that a wheelchair isn't moving anymore. The two cell layers below update any information that was retained from old cell state to new cell state. "Input gate layer" sigmoid layer generates a list of the inputs to b updated, and "tan h layer" creates a vector containing new potential values. For updating the cell state, these are joined. This could be new with the newly discovered movement state being substituted for the existing movement state in memory. The output regarding the cell must be made before updating the hidden state ht. The current cell state is multiplied by sigmoid gate's output so that only the pertinent portions related to cell state have been transformed as well as pushed to the output, and a sigmoid layer control that portions regarding cell state to output. The current state of the cell would be pushed through tan (h) function in order to obtain values in the range of -1 to 1. In order to transition into the following cell state, this is in which relevant information regarding the properties of the movement could be found. The data used for this work's analysis fit well with LSTM networks. They take into account the data's context and the wheelchair's current state, and after training, they might rapidly classify new data in real-time. It will take certain time to train LSTM network, but once it is, it may be utilized for all subsequent movements. For such reasons, the work will train an RNN on wheelchair movement characteristics using LSTM networks.

2.4 K-NEAREST NEIGHBOR

The first classifier used was KNN—a non-parametric learning algorithm, meaning no underlying assumptions are made about the distribution of data [13]. Although the method was first developed in the 1950s, widespread use occurred in the 1960s after increased computing power was available. Given today’s computing power, this method is widely used for pattern recognition.

The KNN method is based on human learning by analogy, comparing data points with those that are in the immediate vicinity. Each data point resides in an n-dimensional space. KNN calculates distances and searches for the K-closest points in the n-dimensional sample space to determine similarity between data points. Many distance metrics, including Minkowski, Euclidean, and

Manhattan, are available. This study takes into account Euclidean distances regarding two objects described via n attributes. Euclidean distance between X_1\ and\ X_2\ points\ has been defined as

$$d(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \tag{1}$$

The algorithm works by determining a test set of correctly labeled training data in a form of D=(X_i,y_i), where X_i\in\mathbb{R}^n is the measured input data, and y_i\in\mathbb{R} is the corresponding class label. For each point in test dataset T, distances are calculated to every point in the training set using (1). The closest K points for each test point are selected, and the most frequent class label among the K points is chosen as the estimate for the class label of the test point. When choosing between two classes, an odd K should be used to avoid a tie situation.

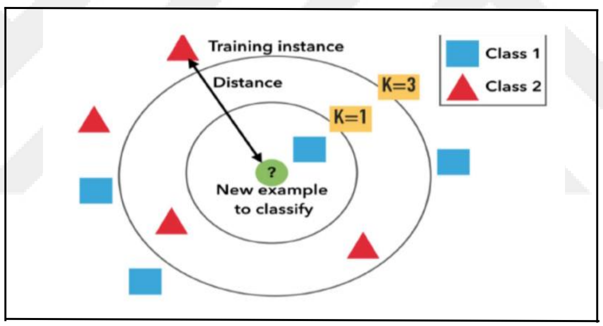

Figure 2.7: K-nearest neighbor algorithm visualization [30].

The KNN classifier is simple, yet powerful, requiring no training other than having correctly labeled available data. This simplicity can be a significant advantage, as training the data can be the most time-consuming activity of many classification techniques. However, simplicity comes at a cost when classifying new points, as the distance to every training point must be calculated. Hence, if k=1 and D\ is a dataset with |D| points, the number of computations required is O\left(|D|\right). Utilizing search trees can decrease the number of computations to O(\log{\left|D\right|}). The amount of calculations necessary is independent regarding the size of D thanks to parallel implementation, which could further decrease running time to a constant, O(1).

2.5 SUPPORT VECTOR MACHINE

The next classification technique under review was SVM [13]. This technique was first presented in 1992 by Vladimir Vaprik et al., although the groundwork had been available since the 1960s. Training times for SVMs can be slow, albeit for many applications results are highly accurate because SVM can map complex nonlinear decision boundaries.

The idea behind any SVM is leveraging to translate the dataset into higher dimensions, use a nonlinear mapping. When done properly, increasing dimensionality leads to increased separation between classes. The goal of SVM is locating the hyperplane which divides the classes (i.e., a decision boundary). Support vectors (i.e., essential training points) and margins are used. Overfitting is much less likely to occur in SVMs when compared with other methods. The learned model can be compactly described using the support vectors. SVMs are commonly used in a variety of areas.

To illustrate how SVMs work, a linearly separable case is shown below in Figure 2.3.

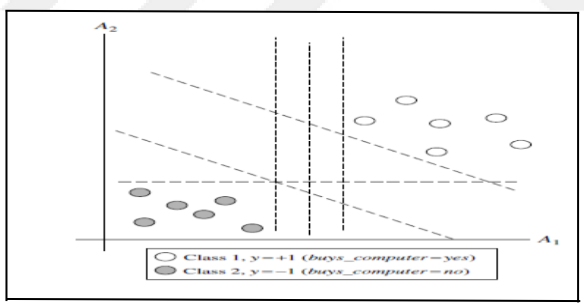

Figure 2.8: Linearly separable training data set example [13].

In the figure above, there are an endless number of potential linear choice boundaries. The task is choosing the best one. The line proves to be a plane or hyperplane in higher dimensions. To solve this problem, SVM searches for the maximum margin hyperplane. Figure 2.4 demonstrates the concept.

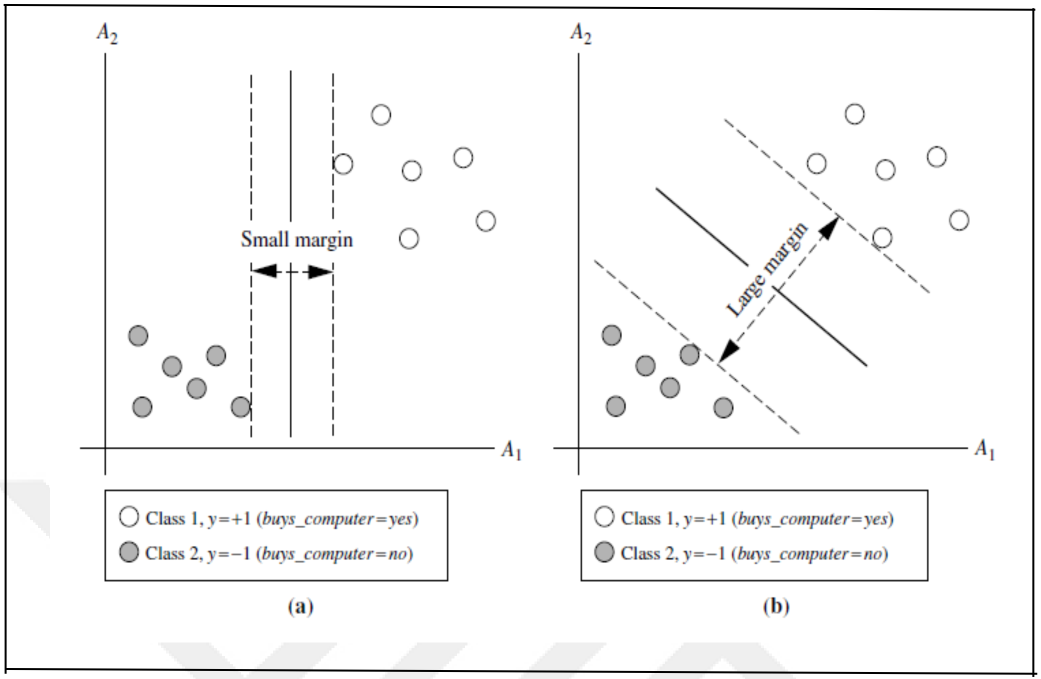

Figure 2.9: Two possible decision boundaries and their associated margins [13].

The figure above demonstrates that the margin is the distance orthogonal to the decision boundary to the closest points of the two classes. Mathematically, the decision boundary as shown below

$$W \cdot X + b = 0 \tag{2.2}$$

where $W \in \mathbb{R}^n$ represent a weight vector, and b is a scalar bias. In two dimensions, this reduces to

$$w_1 x_1 + w_2 x_2 + b = 0 \tag{2.3}$$

The sides of the margins can then be defined by

$$y_i = \begin{cases} +1, w_1 x_1 + w_2 x_2 + b \geq 1 \\ -1, w_1 x_1 + w_2 x_2 + b \leq -1 \end{cases} \tag{2.4}$$

The test points above or below the boundaries are sorted into the respective classes. The distance from the decision boundary to either margin hyperplane is \frac{1}{|\left|\mathbit{W}\right||}. Because both margin hyperplanes are equidistant to the decision boundary, total width of the margin is \frac{2} {|\left|\mathbit{W}\right||}. Support vectors are the points in the training data set that lie on the margin hyperplane and can compactly describe the SVM. The decision boundary is orthogonal to the line connecting the support vectors, and the margin width is the magnitude of the distance between the points. To solve this problem and find the maximum margin hyperplane and support vectors, a Lagrangian formulation is used to convert the problem into a constrained convex quadratic optimization. Exact details for this process are beyond the scope of this work, although they can be found in [13].

When the training data is linearly inseparable, a nonlinear transformation into a higher dimensional space is required. Kernel functions are used for transforming the data. These nonlinear SVMs compute the same decision boundaries as corresponding.

There are no general rules for choosing a kernel function to maximize the accuracy of the classifier. Typically, the choice does not make a significant difference. The training process for SVMs means that the SVM always finds the global minimum.

2.6 ARTIFICIAL NEURAL NETWORKS

Neural networks [13] is nothing but a collection of interconnected input-output units known as neurons, each having its own weight and activation algorithm. A neuron collection connected in complex ways can "learn" by adjusting the weights of each neuron connection to model a complicated nonlinear function, mapping input data to output data. Neural networks were originally the domain of psychologists and neurobiologists and served as a tool for comparing a computational analogy to a real neuron. Recently, however, the field has proliferated in the data science community. Neural networks typically require large data sets and long training times, which can limit the scope of their applications. When feasible, though, few machine learning techniques are as flexible or powerful.

Several other advantages to neural network classifiers include a high tolerance to noise in the capacity to classify new patterns, as well as the training data. Also, very little knowledge on relationships between classes and attributes is needed, as relationships are automatically trained into the network. Neural networks are much better suited for continuous valued data when compared with other classifiers, such as decision trees. They have seen a wide variety of real-world applications, from recognizing handwritten characters to pathology and diagnostics in medicine, as well as for training computers to pronounce English words more naturally. Due to neural network's inherent parallel structure, the training process time can be drastically decreased by utilizing parallelization and GPUs. Recently, new techniques have been developed to extract rules from trained neural networks, allowing us to gain new insights into the situation, as well as to determine exactly what neural networks learn to recognize. An example of a fully connected NN is depicted by Fig. 2.5.

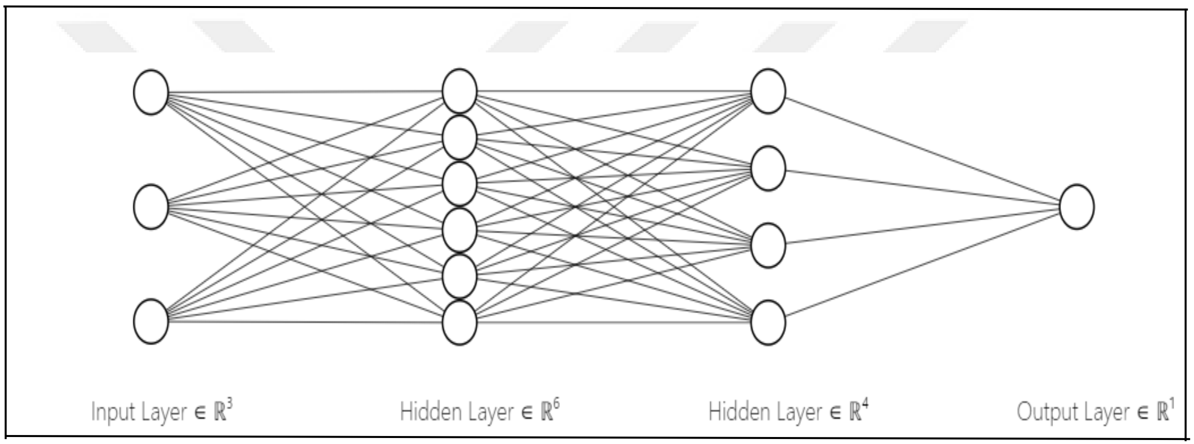

Figure 2.10: Neural network fully connected architecture.[41].

The example above shows three inputs, four neurons in the second hidden layer, six neurons in the first hidden layer, and 1 output neuron to determine the output class. There are a variety of neural network topologies and training techniques to choose from. The emphasis of this thesis is on a multi-layer feedforward network that is completely linked. Backpropagation is the most widely used network training technique. The network weights are first set at random numbers. Backpropagation trains the network by processing data points in the training dataset repeatedly and comparing the output to the true class. The method modifies neuron weights after each data point to reduce the prediction's mean-squared error. Starting with the output layer and moving backwards, changes are made. Convergence is not guaranteed, although weights will ultimately converge in general.

Given a neuron, j, which is located in an output or hidden layer, the net input, I_j, to neuron j can be given by:

$$I_j = \sum_i w_{ij} O_i + \theta_j \tag{2.5}$$

here w ij represents the weight of the link to the ith neuron on the previous layer, O i denotes the output of the preceding layer's ith neuron, and _j is the neuron bias. For variable neuron activity, the bias term works as a threshold. Each neuron receives the net input from all neurons in the layer before applying an activation function. Many different activation functions are utilized, with sigmoid, hyperbolic tangent, and rectified linear unit being the most prevalent (ReLU).

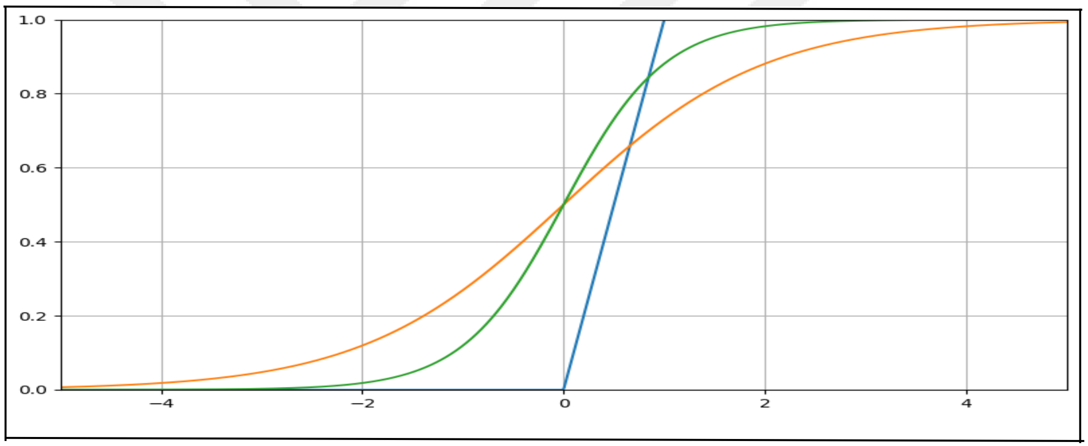

Figure 2.11: Common activation functions.

The mistake of the network's prediction is transmitted backwards by updating weights and biases in each layer as you travel backwards. For neuron j in the output layer, error is determined by the equation.

$$e_j = O_j(1 - O_j)(T_j - O_j) \tag{2.6}$$

where O_j is the actual output of the neuron, and T_j is the known true class of the training point. One should note that O_j\left(1-O_j\right) is the derivative of sigmoid function.

The mistake of a neuron j in hidden layers is written as

$$e_j = O_j(1 - O_j)\sum_k e_k w_{jk} \tag{2.7}$$

2.7 RELATED WORK

Gesture recognition has seen a recent proliferation of research. This thesis utilized 3D gyroscope and accelerometer-based gesture recognition. Three main strategies have been proposed by the research community: 1) probabilistic signal modeling, 2) temporal warping, and 3) machine learning-based classification.

In the probabilistic gesture recognition approach, both discrete [14-16] and continuous HMMs [17] have been widely employed. To create discrete HMMs, Kela et al. [16] use gesture velocity profiles (dHMM). Raw data must first be clustered in order to reduce dimensionality and build a feature codebook. The second step is to generate a discrete HMM with the use of vector codebook index sequences. From eight movements done by 37 people, a 96.1 percent correct recognition rate was obtained using five HMM states and a codebook. Pylvänäinen [17] proposed a continuous HMM (cHMM)-based probabilistic gesture recognition system that obtained a 96.76 percent recognition rate on a data-set of 20 samples for 10 gestures that have been performed by seven individuals.

The second approach [18-20] uses a set of reference motions to direct temporal warping samples. Liu et al. [19] claimed that utilizing pre-processed signal data and Dynamic Time Warping (DTW), they were able to obtain 93.5 percent gesture recognition and 88 percent user identification rates.

The third strategy [21-25] employs machine learning-based classifiers. Hoffman et al. [21] employed an AdaBoost-based linear classifier to obtain a user-independent 98 percent recognition rate for 13 gestures that have been carried out by 17 people.

Sung-Jung Cho et al. [22] utilized a two-stage recognition algorithm consisting of a Bayesian belief network, followed by an SVM for confusing cases. Results showed a very high recognition rate, with 100 users the system achieving an average recognition rate of 96.9% on 11 gestures.

This classifier worked on accelerometer data alone and was installed in Samsung cell phones, starting in 2005.

Wu et al. [23] suggested an SVM-based classifier. To synthesize final feature vectors, each gesture was split for each time segment, statistical measures (e.g., mean, energy, entropy, standard deviation, and correlation) were calculated. The detection rate for 12 movements made by ten participants was 95.21 percent, greater than the DTW results.

A method depending on the bidirectional long-short-term memory RNNs was suggested by Lefebvre et al. [24] in recent research (BLSTM-RNN). The BLSTM-RNN classifier achieved a recognition rate of 95.57 percent after being trained on data from 22 people executing 14 gestures.

A convolutional neural network (CNN) with a particular topology, as well as a combination of 1-D convolutional networks, averaging, and max-pooling procedures, is used in this technique was suggested by Stefan Duffner et al. [25]. The study categorized the fixed-length input matrix, which was made up of normalized sensor data, directly. The ConvNet presented in [24] met or exceeded the approaches in [21-23], indicating that neural networks may be used to recognize gestures.

Sojeong Ha and Seungjin Choi [26] used CNN to analyze data from ten volunteers who made 12 motions while wearing three accelerometers (e.g., right wrist, chest, left ankle) and two gyroscopes (e.g., left ankle, right wrist). The new CNNs correctly identified 91.94 percent of the test data. 10,000 motions were utilized in this research to teach participants, demonstrating that more training is necessary to detect more gestures. Given the purpose of identifying a single gesture, precision is required. Casey Cole et al. [27] utilized an Apple Watch's accelerometer and an ANN for recognizing the gesture of smoking a cigarette at an accuracy of 85-95%. The goal of the study was providing a way to recognize smoking and intervene or reduce the number of cigarettes consumed per day. This investigation is similar to this thesis in that improving health outcomes was the primary motivator. With a similar goal of smoking cessation, Abhinav Parate et al. [28] used multiple IMUs, trajectory-based methods, and a probabilistic model to detect smoking gestures at 95.7% accuracy.

3. RESEARCH PURPOSE

3.1 PURPOSE STATEMENT

Using the already available technology regarding the wheelchair user is less expensive and more straightforward than developing new technologies to analyze the movements and maneuvers of the wheelchair as well as assisting in the needed exercises in order to support healthy lifestyle, like the sensors that are mounted on chair's wheels or automatic seat adjustment hardware. The goal of the presented work is to build an application to help the users of wheelchairs and the medical professionals to evaluate the mobility of the wheelchair user by utilizing the current technology regarding wheelchair users, specifically their smart watch or smartphone devices equipped with the accelerometers in combination with the RNN [29].

3.2 METHOD

This work investigates if it is feasible to collect the data from a smart watch or smartphone and train a RNN to interpret this data for detecting bouts of movement so as to evaluate the mobility of a wheelchair user. The lengthiest bout, total number of bouts, and the overall time amount that has been spent moving are utilized in the presented work to evaluate the wheelchair user's mobility.

3.2.1 Technology and Equipment Used

Powered wheelchairs provided almost all data that were utilized in the presented work. Two Android 11.0-powered devices, the Google Pixel 2 and Google Nexus 5, provided the accelerometer data. The phone was either put flat on inclined table that has been mounted to wheelchair or installed in a phone holder that has been linked to wheelchair arm. Fossil Sports Smart Watch (DW9F2), which runs Wear OS v. 2.5.0, also provided some data.

3.2.2Python Development Environment

Data pre-processing as well as RNN programs have been created with the use of Google's Colab notebook development environment and Python 3.6.7. Developers may combine executable code and rich text formatting in a single document thanks to the Colab development environment. Libraries are available for importing and storing files in Google Colab as well. The comma separated data files that contain accelerometer data were read using library of Python Pandas 1.1.0. Data frames that are 2D labeled data structures, were also built using Pandas. The timestamp data and labeled rotational axis data that had been collected from smart devices are stored in such data frames. In order to apply the functions to matrices that are required for the proper accelerometer data formatting, the library of Python Numpy 1.19.0 has been utilized. Colab notebook's graphs have been made using the Seaborn and MatPlotLib libraries. RNN was developed using the Tensorflow 2.3.0 library.

3.2.3 Method for Movement Data Gathering

The accelerometer regarding the users' smart devices was used to gather movement data from wheelchair users for measuring their mobility. To collect data, two applications were made: one for an Android-OS smartphone and the other for a WearOS-enabled smart watch. The programs' features were remarkably comparable. The SensorManager class was used by the applications to connect to device's accelerometer. The rate of data polling has been set to be the same as Android UI mode's refresh rate (14-16 Hz). With little negative effects on battery life of devices, the polling rate delivered data sufficiently quick to account for wheelchair movements.

The sensors' data output is a 3D vector that excludes gravity and represents acceleration along each axis. For later processing, a csv file containing this data and the millisecond timestamp was created.

3.2.4 Factors Affecting Data

Numerous potential influences on the experiment's accuracy were noted, and appropriate mitigation approaches have been put into place. There has been at least one risk-mitigation strategy developed for each recognized risk factor. Training dataset, the RNN algorithm accuracy as a result, or a misrepresentation of population that is being studied might all be adversely impacted by factors. For creating a customized workout program, a healthcare expert will utilize the results regarding such experiment to evaluate the mobility of a wheelchair user, so it was crucial that risks were sufficiently reduced. Various wheelchair designs, whether manual or powered, will generate various accelerometer data types, which might have an impact on the experiment's results. Only powered wheelchair data have been used in such experiment; on the other hand, the RNN that emerged from this has been tested against the manual wheelchair data, and results will be described further in the present work. Considering the fact that the wheelchair users move differently from each other, the data has been gathered from numerous people who were in charge of the powered chair. The experiment's results can potentially be impacted by the device's placement, the accuracy of the accelerometer, and the type of device utilized. Various devices have been evaluated with various wheelchair placements in an effort to reduce such risks. The Google Pixel 2 and Google Nexus 5 are two distinct phones with differing hardware. The hardware variations between such two phones guarantee that the sensor's accuracy won't have an impact on the outcomes. To guarantee that the results would not be harmed by varied device orientations, the device was additionally positioned with the use of several holders. The devices have been placed in a phone that holds mount affixed to the chair arm and on a table that has been mounted to the chair. Those many mounting options made sure that the device can move and be positioned in various ways that might be normal for a device that has been installed on a wheelchair.

3.3 EXPERIMENTAL STUDY: DATA ANALYSIS USING RNN

With an exception of acceleration due to gravity, the device accelerometer measures the device's acceleration along each axis. The sensor gauges the device's applied acceleration. Through utilizing the relation to measure forces that have been applied on actual sensor, F_s:

$$A_d = -g - \sum F_s/mass$$

In which g denotes the acceleration because of gravity. With regard to linear acceleration, impact of the gravity has been accounted for and this formula could be simplified into:

$$A_d = -\sum F_s/mass$$

Simply said, such equation is a negative summation of all of the forces acting on sensor divided by device's mass [22].

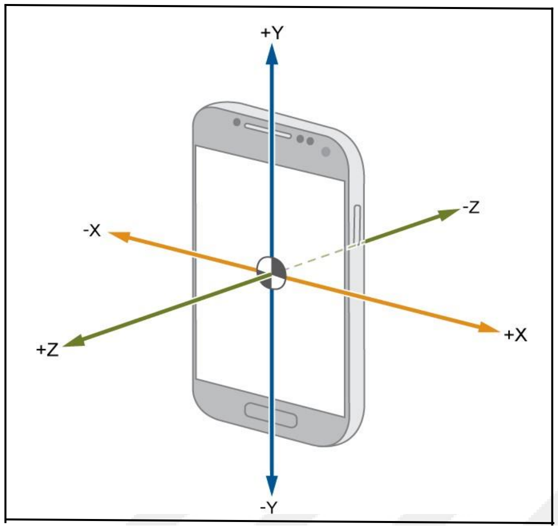

Figure 3.1: Linear acceleration axes [23]

Devices' acceleration data was gathered in raw format. The file's data each comprised information for the y, x, and z axes as well as a timestamp expressed in milliseconds since Jan. 1st, 1970 UTC. The data that had been collected could be shown as an acceleration sequence:

$$D_C = \{c_1, c_2, \ldots, c_n\}$$

where $c_i = \langle a_i^x, a_i^y, a_i^z, t_i \rangle$ with $(1 \le i \le n$ and $t_1 \le t_i \le t_n)$ and a_i^x, a_i^y, a_i^z represent

The accelerations of device that had been measured along every axis, at the time t_i.

Raw acceleration data has been converted to a group of jerk data for extracting useful information from the data. Jerk can be defined as the derivative of action regarding time:

3.3.1 Python RNN Programs

Two Python programs have been made for analyzing the data. For organizing and cleaning raw data, a program for data preprocessing was made. A program for processing RNNs was made to generate RNN, train it by the labeled data, check its accuracy compared to labeled data, run the algorithm on the unlabelled data-sets, and evaluate results.

3.3.2 Data Pre-processing Program

The purpose of the program of the data pre-processing has been applying the transformation to raw data as well as cleaning the data for removing errors that might have happened owing to sensor noise. With the use of files library included in google.colab suite, the data is manually entered into the program. A data frame has been created with 4 columns: 1 for every axis of acceleration and 1 for time-stamp. Following data loading to program, accelerations would be converted to series of jerk values with corresponding timestamps.

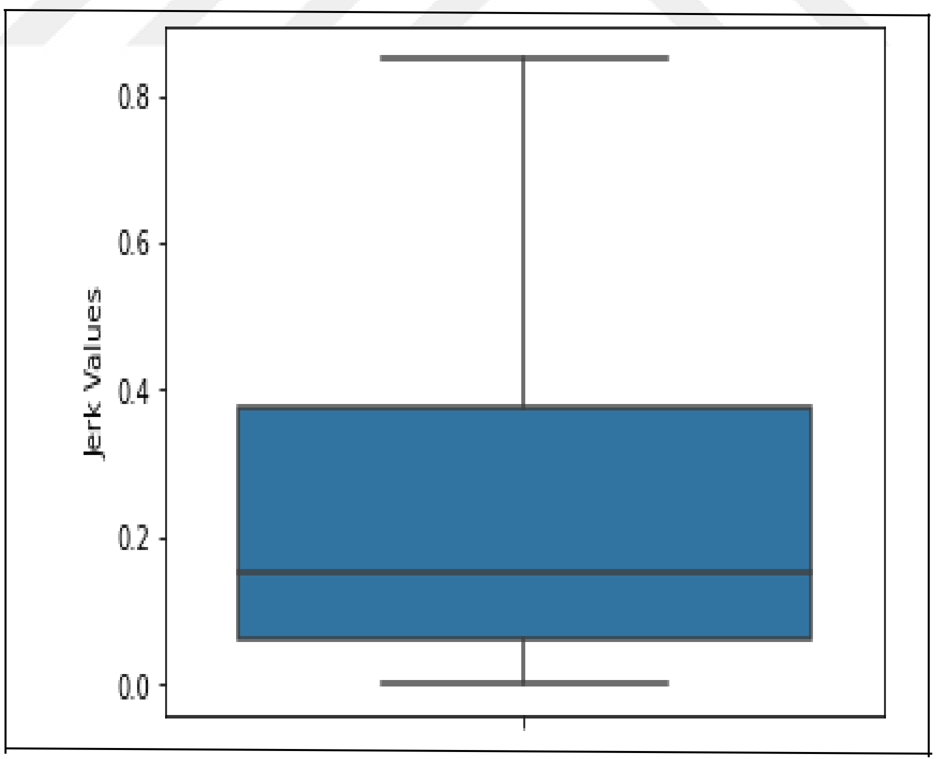

Figure 3.2: Example boxplot that shows data that had been gathered from a set of the bouts

Figure 8 depicts the calculation of inner quartile range, the lower and upper quartiles regarding each set of data for removing extreme data outliers which might probably be brought on by sensor errors. As there could never be a value below zero, zero is used to establish the lower quartile, and any values that are found above upper quartile have been substituted by upper quartile's value. Acceleration along at least one axes could increase as the wheelchair moved, increasing the jerk value. The wheelchair is stationary if the value is low.

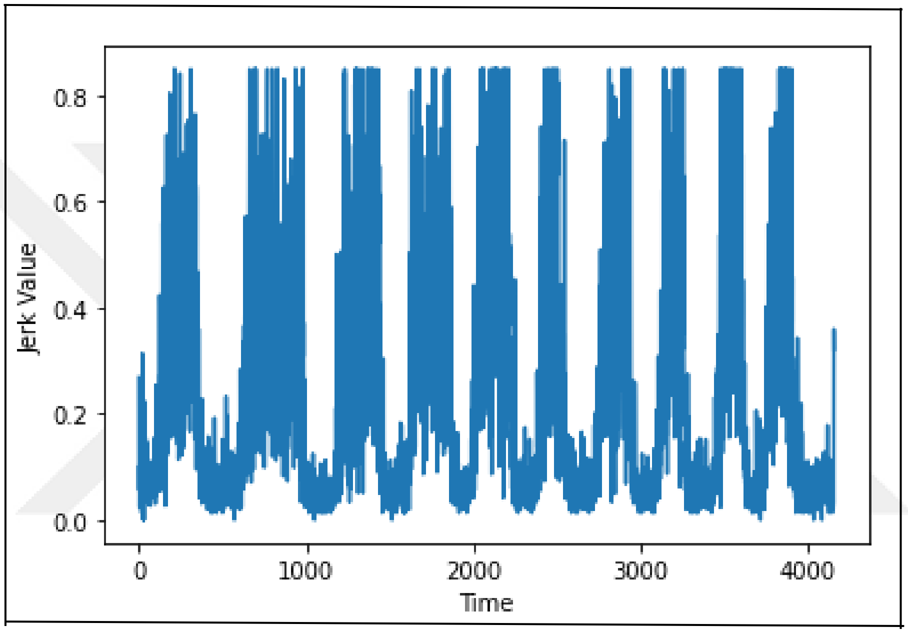

Figure 3.3: Graph of all of the adjusted jerk values for one dataset

The noise in sensor data, as depicted in Figure 3.3, makes it challenging to analyze. The jerk data shows clear patterns, yet the variety makes it challenging to pinpoint exactly where the movement begins and ends. It has been possible to perform manual analysis for determining approximate threshold to utilize in order to specify if those jerk value must be classified as moving or otherwise; in such case, the 0.25 value has been utilized. Datasets that have been utilized for the training include correct labels for every one of the sensor readings.

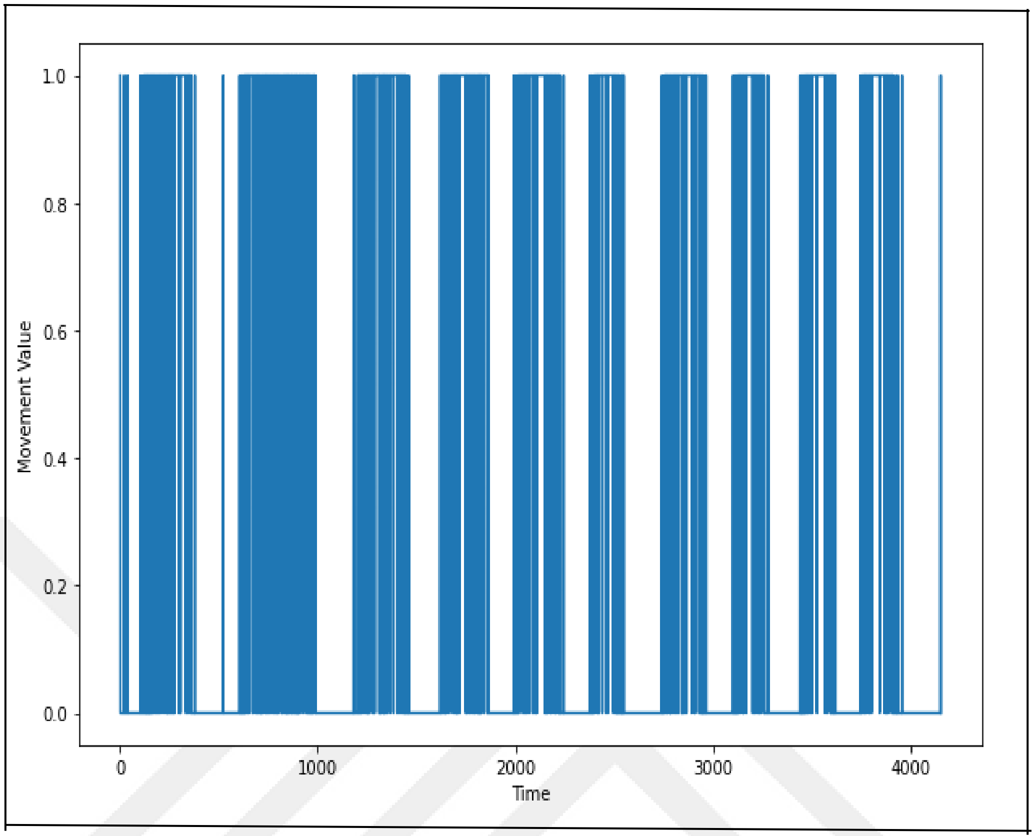

Figure 3.4: Movement Data after the application of the threshold

The threshold was used to apply min-max scalar to data, producing a dataset in which all values are labelled as one or zero as illustrated in Figure 3.4. In such dataset, the patterns are more distinct, but there remains too much noise to conduct a thorough analysis. The set has been divided into groups of 2.10 in set to additionally refine it so that it could be utilized as RNN training dataset. In each one of the groups, the label with the highest frequency was chosen as label for the whole group. This modification produced a dataset that has very distinct movement bouts. There have occasionally been noise values which have made it to final training dataset, therefore a small algorithm was created for additionally cleaning data and checking whether any values were misclassified by looking at the neighbors regarding each value. Since movement bouts last long enough for numerous sensor outputs to be available in every one of the bouts and sensor data is gathered quickly, it is indicated that any single value which differs from the rest of the set will be identified as having been classified incorrectly and will be automatically updated in order to reflect correct value.

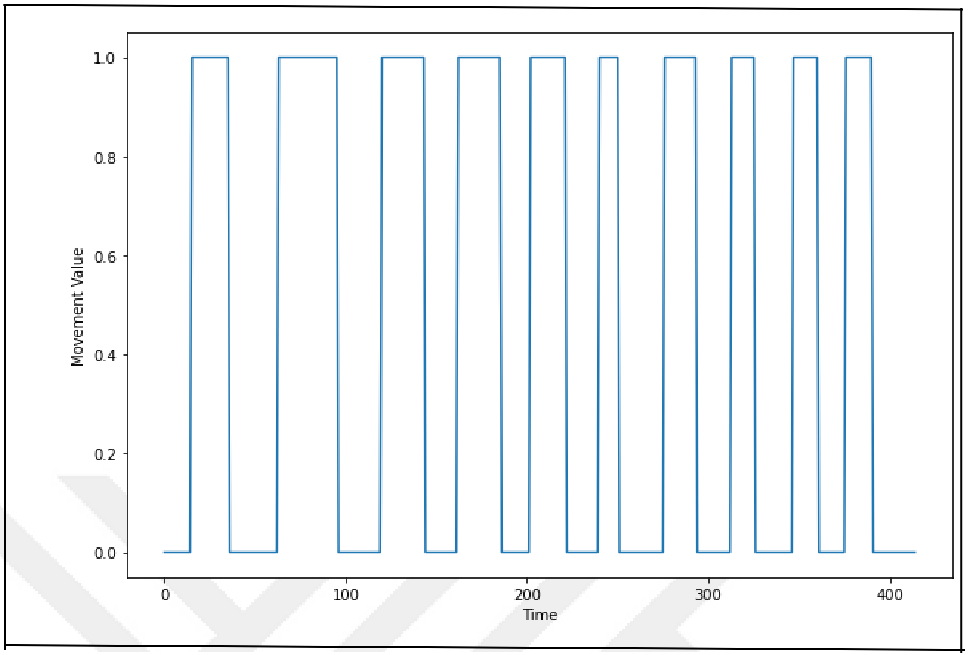

Figure 3.5: Correctly labeled training dataset

The dataset is after that utilized for the training of the RNN, which will subsequently have the ability of the automatic identification of the sensor data as it's received from sensors. Data must be transformed from one value of the movement to a set of the features with regard to RNN LSTM cells as the last stage in the training data preparation process. A new data frame has been formed with 10 values, one value for every one of the movement values in each one of the ten groups, which each contain ten components with the same label. A total of 12 columns, one for each weight, make up the final data frame format.

$x_0, x_1, \ldots, x_9$ and column for label to train RNN, and time-stamp.

3.3.3 RNN Data Processing Program

Preprocessing program's cleaned training data are used by RNN program for the training of the network, which is after that tested using a different set of cleaned data. Taking a validation set and separating it from the initial training data is the first stage in this procedure. In this work, 100 rows from the training dataset have been selected to serve as the validation set. Each one of the layers in the sequential model-configured RNN has one output tensor and one input tensor. Also, RNN is configured up for monitoring its value loss and terminate the algorithm early when it reaches a value where it stops getting better. RNN is built with two layers: a time distributed dense layer with sigmoid function for a purpose of ensuring that the output is either zero or one to represent stationary or moving, and LSTM layer with ten LSTM units, the input of which includes ten data items. Once the RNN was set up, it is tested against a different test set after being trained with the validation and training sets to ensure that the algorithm is accurate.

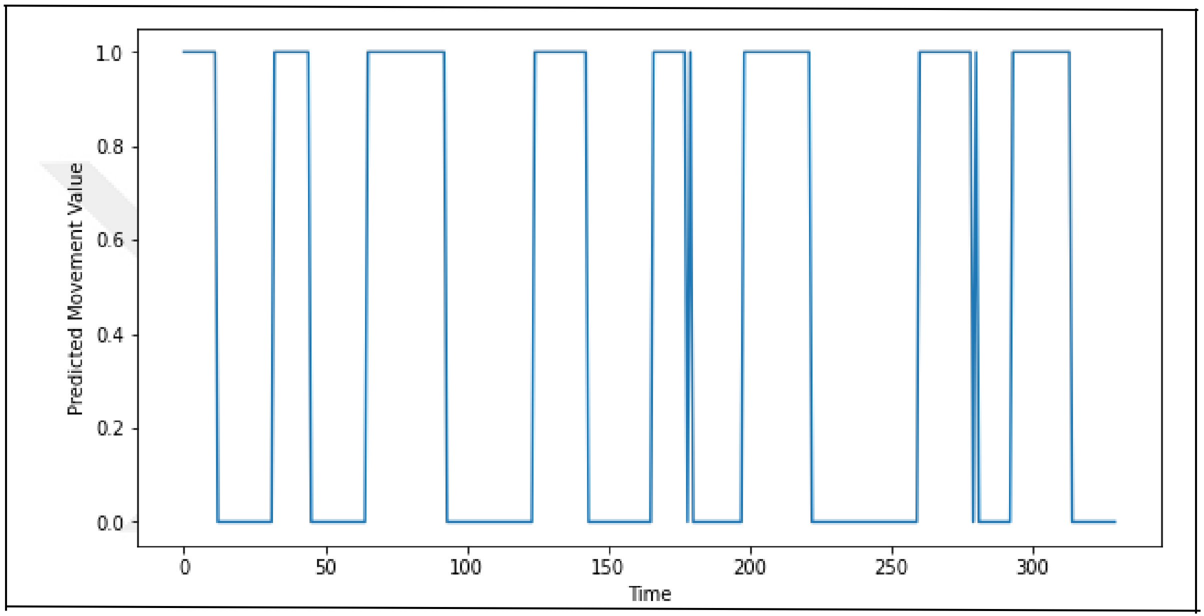

Figure 3.6: Example of Output Prediction dataset from RNN

3.3.4 Collecting the Time Data for Analyses

Associated timestamp data has been utilized for the purpose of collecting movement data after RNN algorithm has been run and a prediction set has been obtained. Every run of values has a difference between first and last timestamp which is collected as well as classified as moving or stationary, depending upon run's predicted value. Using such information, the overall amount of time spent during the whole session has been calculated through summing all of time samples that have been collected. It is also possible to deduce from the gathered data the overall time amount that is spent moving, the lengthiest bout of movement, and the total number of the movement bouts.

3.4 EXPERIMENTAL STUDY: GATHERING THE DATA FROM SMART WATCHES

A research was conducted on collecting data of movement from a wheelchair user's smart watch along with data of movement from android phone affixed to powered wheelchair. This enables for the collection of both wheelchair motion data and individual arm movements. In the case when motion could be recorded in this way, manual wheelchair users will be able to do so while tracking movement with their pushing motions. Additionally, it could enable users of powered and manual wheelchairs to monitor their movements during exercises in order to compile information on the frequency and duration of their use.

3.4.1 Android Studio

Two applications have been made for this work. A partner app for Android phone and a smart watch app for collecting movement data The watch makes a connection to the watch and locally saves output. The app of smart watch was created utilizing the same structure as the one that has been used for the earlier study. The buttons and screens presented to the user are exhibited in Figure 3; the Sensor Manager begins collecting sensor data in the case where a user pushes a button on watch face and stops in a case where a user hits the button once more.

Figure 3.7: Screens of Smart Watch Application

Sensor data has been transferred to the phone app, in which it is locally saved. Through connecting the phone to a PC and transferring the files, it is simple to retrieve the data.

3.4.2 Differences in the Accelerometer Data

With regard to the present work, movement data has been gathered as a person in a powered wheelchair engaged in a variety of activities meant to reduce the development of pressure ulcers. According to the exercise, the accelerometer data on each axis fluctuated and did not resemble the raw data gathered for the prior work. Yet, the discrepancies in the data have been less obvious after the transformation of raw acceleration data to a collection of the jerk data. Although lengths of movement had varied, there have been clearly distinct patterns indicating when the device was moving and when it has been stationary.

3.4.3 Analysis of the Data with RNN

Although interpretation of this data will differ depending on RNN algorithm's prediction, RNN from prior work has still been able to assess this dataset. For the purposes of the present work, periods when the device is stationary signify that an exercise has been carried out, whilst movement periods signify a change in the exercise. A button on the watch was used by the users performing the exercises for starting and stopping data collection, which caused movement at the end and beginning of dataset that has been unrelated to exercises.

3.5 DEEP LEARNING (DL)

This algorithm, which represents a well-known algorithm of ML, was used in the suggested identification system and its performance was put to comparison with that of other algorithms used in this work to determine which algorithm provided the best accuracy for recognizing and identifying individuals. A lazy algorithm for limitless learning is KNN. There is no hypothesis on the distribution regarding the underlying data, which is how the KNN limitless appears. In another sense, the model framework developed from the dataset will be extremely helpful in practice because the majority of well-known real-world datasets do not meet mathematical theoretical assumptions. The KNN approach is a lazy algorithm for building models since it doesn't need any training data points. The training phase takes less time since the testing phase makes use of all the training data. The testing phase, in comparison, is slower and demands more memory and time compared to the training phase. In the worst situation, the KNN algorithm takes longer to scan all data points, necessitating additional memory space to store the training data. It is possible to obtain the KNN algorithm for both regression and classification problems. Figure 3.10 provides a clearer illustration of the KNN classification steps. By reducing difficult high-level results like data representation through hierarchical teaching analysis, DL models outperform standard ML in terms of accuracy and speed. DL thus significantly enhances object recognition, object detection, and object classification. A computational model has many processing levels (i.e. nonlinear transformation) utilized for learning different idea-level information's representation. Learn the feature extractor hierarchy. From the results of the preceding level, every hierarchy level extracts features. The capability to choose ideal qualities is the main goal infrastructure of DL. Detailed learning based upon the data and computer-compressed data. Which is really what AI requires, particularly when there is a huge amount of data available. As seen in Figure 3.2, DL is a NN with at least three hidden levels. There were multiple hidden levels in the NN. DL is performed using neural networks that have at least three hidden levels integrated into them or sequential levels. In DLL, CNNs are more prevalent. The model is most likely made up of numerous layers, each of which has an output which serves as an input for the layer above it. The supplied data image is transformed to any number of the hidden levels, for instance, when a dog image is provided at an improved level. It is a CNN that has been created by Google DeepMind team and derived from old Chinese in ancient Chinese game of Go, which is thought by many to demonstrate learning tree and the best humans on the planet. We've defeated several groups and won them. Less complex NN with multiple interconnected node layers.

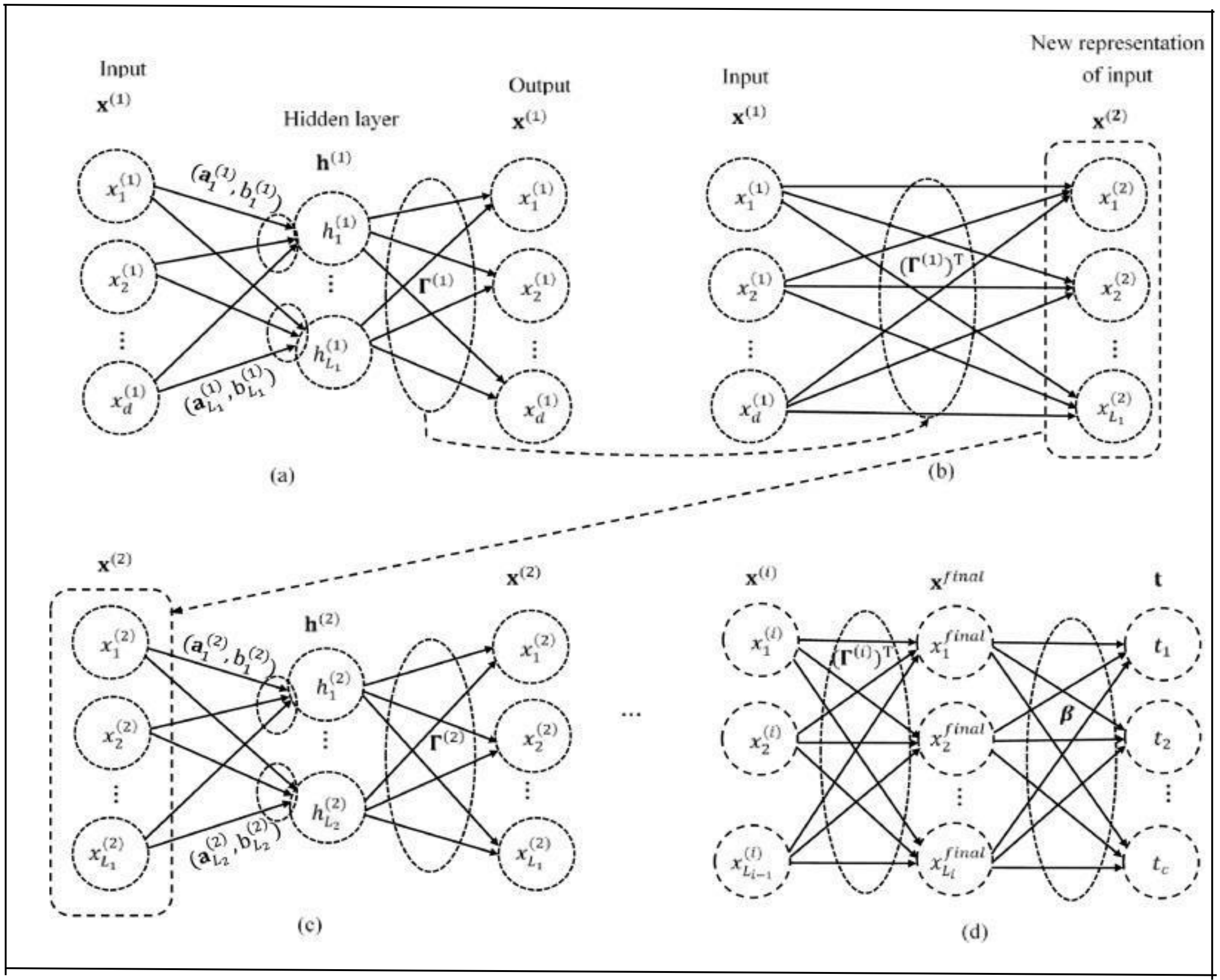

Figure 3.8: DL for the ELM

3.6 CONVOLUTIONAL NEURAL NETWORKS (CNNS)

There are numerous varieties of ML algorithms, which vary depending on how they classify data, the kinds of data they deal with, and the issues they are intended to address. Algorithms of various types learn in various ways, leading to various application uses. Every kind of algorithm was taken into information, and the decision was ultimately chosen depending on the data that would be accessible for the specific problem statement. A device will read sensor data from it and output it in the raw form. Data will be timestamped then arranged in a sequential order. Due to the available sensors' sensitivity to slight vibrations and limitations in accuracy, data will potentially include noise. Additionally, a training set will be made accessible with raw data that has been accurately classified as static or moving. With this in mind, various learning algorithms have been discussed, taking into account the algorithms that are appropriate for the research. Particularly DNNs are establishing various types of captures and outperforming various approaches to problems with computer vision and pattern recognition. The effectiveness of such frameworks depends on their ability to construct object representations without human interference. Given enough training data and tests, DNNs are capable of teaching two-level concepts, including top-level abstracted ideas and bottom-level image features. As of today, CNNs first appear in 1970. For various classification and detection targets, it is hardly ever used. It has been demonstrated that they're quite effective, and it's frequently used in fields including image recognition, segmentation, classification, and Appling object detection. They can quickly learn from the image data, removing the need for the task of the feature extraction from the components. Later, Yann LeCun developed CNNs that were far more efficient for tasks requiring handwriting recognition and conducted document recognition using stochastic gradients depending on gradients. Generally, CNNs are quite good at categorizing objects. Figure 3.3 depicts a straightforward CNNs construction. DL has the disadvantage of requiring very large data-sets for the training to recognize objects.

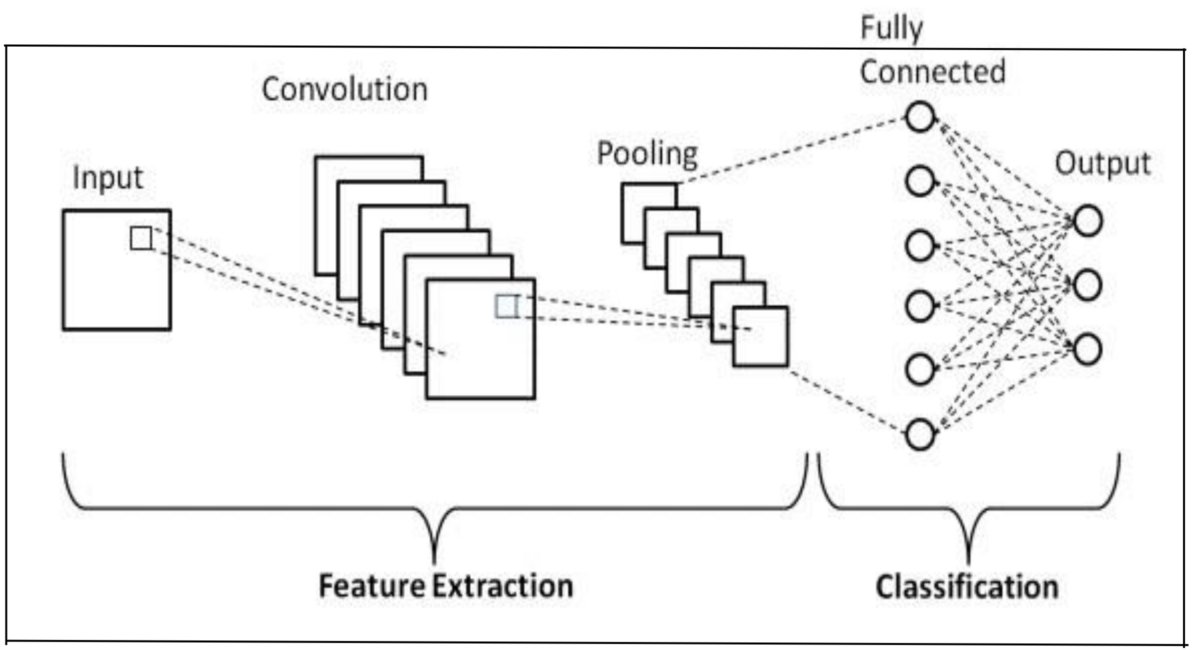

Figure 3.9: Extraction of CNN features

4. RESULTS AND DISCUSSION

4.1 RESULTS FROM EXPERIMENTAL STUDY

4.1.1 Accuracy of RNN on the Powered Wheelchair Movements

Five separate movement data sets from a smartphone which has been in a phone holder that has been attached to chair arm were used in the work to try to train the data set. Every time a new dataset has been used in order to train RNN, it has been checked against the other datasets, and its accuracy was trained. The work found that the highest accuracy was 99% and lowest accuracy was 96%.

Table 4.1: Accuracy that has been obtained on the tested datasets

File ID	No. of data items in file	Accuracy of RNN
1	4,158	99.0%
2	4,378	97.0%
3	4,074	98.0%
4	4,433	97.0%
5	6,656	96.0%

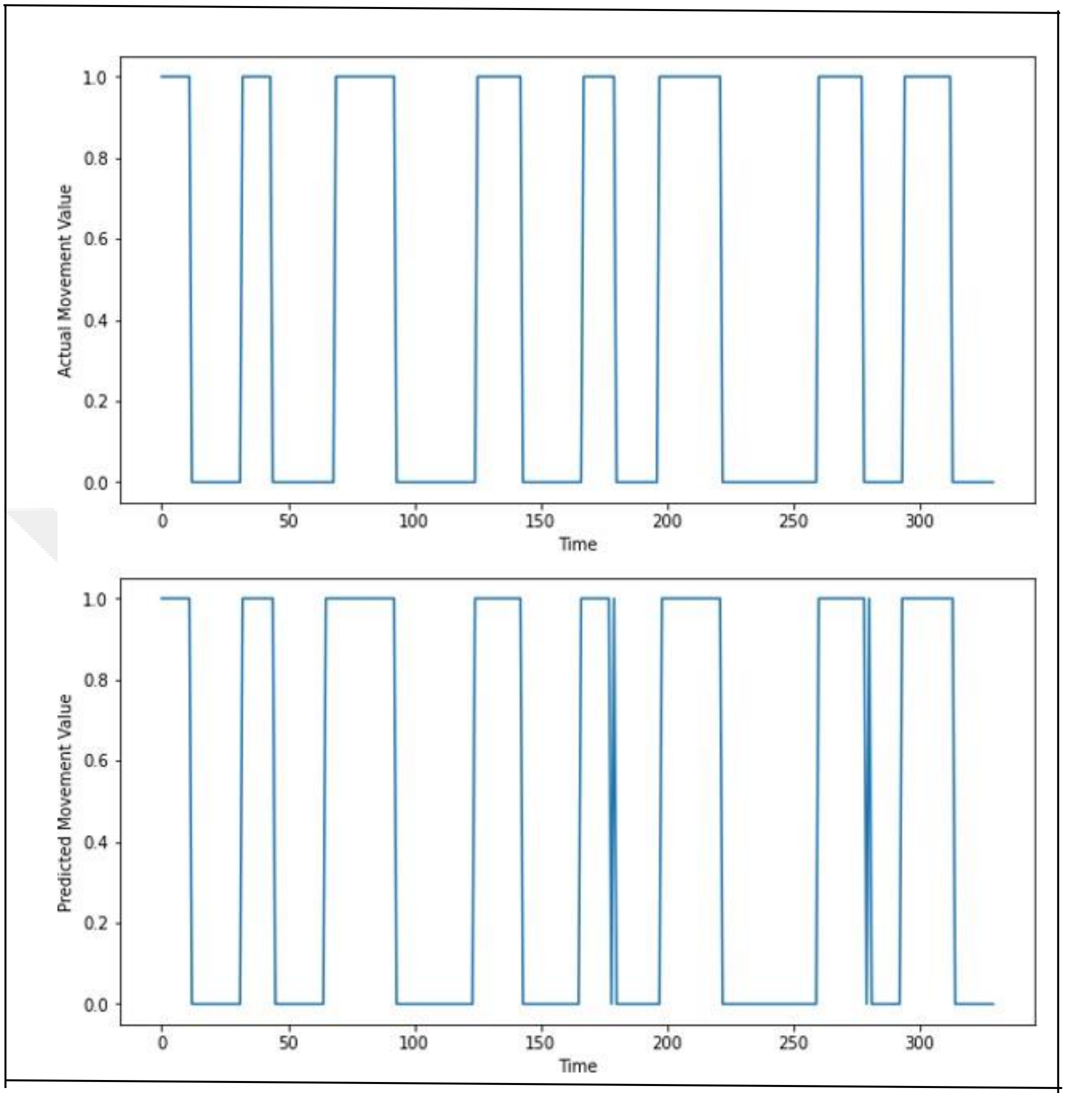

Figure 4.1: Example of actual vs. predicted movement data

The end or beginning of a bout of movement was the area in which RNN frequently failed in correctly predicting movement value, as it has been illustrated in Figure 4.1. Accelerometer data being near or at threshold value at the end or start of the movement, as accelerations start to decrease or increase, explains why this is the case. For each test set, the loss and validation loss have been both noted and graphed.

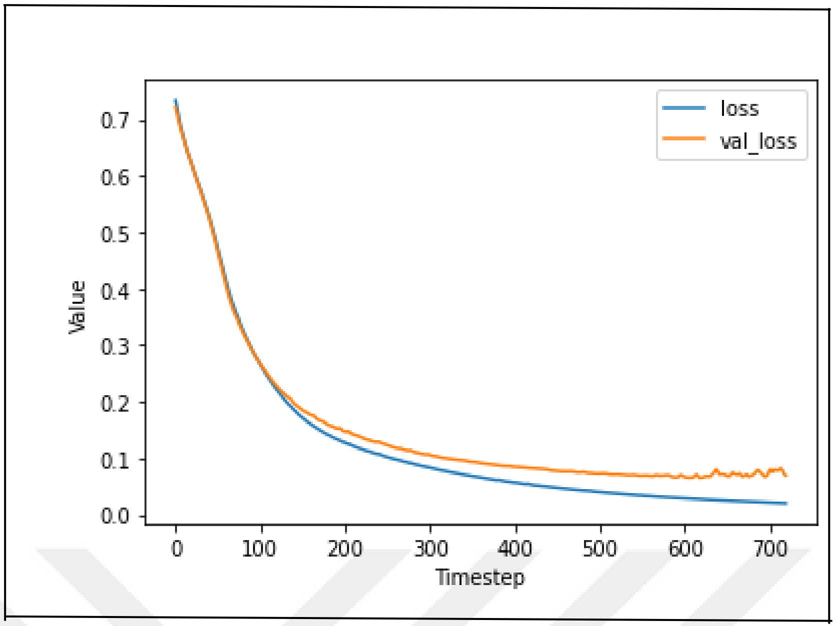

Figure 4.2: loss and validation loss from the RNN

The value of validation loss is connected to loss in validation set, while loss value is connected to the loss in test data. In a case when RNN no longer lowers, value of validation loss is utilized in order to initiate an early stop, yielding final value of the.

4.1.2 RNN Accuracy on Smart Watch Movements

With an average accuracy of 87% in a case where analyzing data from smart watch, RNN's accuracy has been less exact.

Table 4.2: Accuracy that has been obtained on the tested datasets

File ID	No. of data items in file	Accuracy of the RNN
1	2,559	85.0%
2	2,859	84.0%
3	1,469	90.0%
4	2,365	90.0%
5	2,070	89.0%

Since a person might experience certain degree of movement in their hands and wrists during a bout of exercise because of the pressure of the wrist, movements for smart watch have been more challenging to precisely predict. A user having to press stop and start buttons on a watch had caused some movement to be recorded at the beginning and the end of every one of the sensor collection sessions. As the phone was mounted, this did not present a problem, yet hitting the watch's button typically caused certain movement to be recorded. Even though the findings from the watch are less exact, they are still helpful because the inaccuracies might just have an impact on determining the beginning and end of every one of the exercises, not the amount of exercise data recorded.

4.2 ANALYSIS OF THE RESULTS

Experimental investigations have demonstrated that an AI that has been properly trained can analyze wheelchair users' movements. Without the requirement for additional installation or expenditure, the experiment's results were as accurate as those achieved using wheelchairs equipped with additional equipment. An individual could utilize the data to evaluate their own lifestyle or give it to a medical information to help them evaluate their general health. Even though accuracy of smart watch application study has been lower than that of the smartphone application study, the findings contained some valuable information. These findings might help people time their exercises correctly and, when combined with the findings from the first application, can improve evaluations of general health and activity levels.

5. CONCLUSION AND FUTURE RESEARCH DIRECTION

In the present work, we have proposed a method for evaluating a wheelchair user's mobility on a quantitative level. Our method avoids the expense of obtaining and maintaining extra data sensors by merely using the wheelchair user's own smartwatch or smartphone for collecting wheelchair maneuvering data. We have created methods for processing and analyzing sensor data. We have created a RNN, which can categorize wheelchair maneuvering data effectively and deliver insightful analytical findings on the mobility of wheelchair user. As manual wheelchair users make up the bulk of wheelchair population, we will expand our research's scope in the following phase.

5.1 APPLICATIONS OF THIS STUDY TO THE MANUAL WHEELCHAIR USERS

The application of the smart watches might be utilized for collecting movement data from manual wheelchair users without requiring phone to be in any way attached to the chair. Due to the fact that acceleration transformation to separate jerk values is less concerned with the sort of movement than it is with presence of the movements, the data of the movement might be produced by movements of wheelchair user moving the wheels. Additionally, a chair's physical movement could continue to add motion, resulting in accelerations from a chair that the sensors might still be able to detect even if the user's hands were immobile at the start or finish of a wheel push.

5.1.1 Smart Watch Application Continuation

The functionality of the smart watches can be increased to make sure people exercise properly. To more accurately determine which exercises are carried out and for how long, a new RNN may be trained so as to distinguish distinctive motions that are involved in every one of the activities. Each one of the exercises involves a new range of movements and an acceleration in another direction. A set of accelerations could be used to accurately identify which exercise is being performed if the data of acceleration is recorded without being transformed into jerk data and an RNN has been trained for every one of the exercises.

REFERENCES

[1] Y. Sun, W. Dong, and X. Yu, “Effects of coolant temperature coupled with controlling strategies on Particulate Number Emissions in GDI engine under Idle Stage,” *Fuel*, vol. 225, pp. 1–9, 2018.

[2] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, “Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries,” CA. Cancer J. Clin., vol. 68, no. 6, pp. 394–424, 2018.

[3] W. H. Organization, WHO position paper on mammography screening. World Health Organization, 2014.

[4] A. Chetlen, J. Mack, and T. Chan, “Breast cancer screening controversies: Who, when, why, and how?, vol. 40, no. 2, pp. 279–282, 2016.

[5] A. Chekkoury et al., “Automated malignancy detection in breast histopathological images,” in Medical Imaging 2012: Computer-Aided Diagnosis, 2012, vol. 8315, p. 831515.

[6] I. Pollanen, B. Braithwaite, T. Ikonen, H. Niska, K. Haataja, P. Toivanen, and T. Tolonen, “Computer-aided breast cancer histopathological diagnosis: Comparative analysis of three DTOCS-based features: SW-DTOCS, SW-WDTOCS and SW-3-4-DTOCS,” 2014 4th International Conference on Image Processing Theory, Tools and Applications (IPTA), 2014.

[7] F. F. Ting, Y. J. Tan, and K. S. Sim, “Convolutional neural network improvement for breast cancer classification,” Expert Syst. Appl., vol. 120, pp. 103–115, 2019.

[8] H. Ravishankar et al., “Understanding the mechanisms of deep transfer learning for medical images,” arXiv Prepr. arXiv1704.06040, 2017.

[9] S. Lu, Z. Lu, and Y.-D. Zhang, “Pathological brain detection based on Alexnet and Transfer Learning,” Journal of Computational Science, vol. 30, pp. 41–47, 2019.

[10] Y. Chen et al., “A feature-free 30-disease pathological brain detection system by linear regression classifier,” CNS Neurol. Disord. Targets (Formerly Curr. Drug Targets-CNS Neurol. Disord., vol. 16, no. 1, pp. 5–10, 2017.

[11] S. Wang, P. Phillips, A. Liu, and S. Du, “Tea category identification using computer vision and generalized eigenvalue proximal SVM,” *Fundam. Informaticae*, vol. 151, no. 1–4, pp. 325–339, 2017.

[12] M. M. Rahman, Y. Ghasemi, E. Suley, Y. Zhou, S. Wang, and J. Rogers, "Machine Learning Based Computer Aided Diagnosis of Breast Cancer Utilizing Anthropometric and Clinical Features," Irbm, vol. 42, no. 4. pp. 215–226, doi: 10.1016/j.irbm.2020.05.005.2021.

[13] N. I. R. Yassin, S. Omran, E. M. F. El Houby, and H. Allam, "Machine learning techniques for breast cancer computer aided diagnosis using different image modalities: A systematic review," *Computer Methods and Programs in Biomedicine*, vol. 156. pp. 25–45, doi: 10.1016/j.cmpb. 12.012.2018.

[14] R. Yan et al., "Breast cancer histopathological image classification using a hybrid deep neural network," Methods, vol. 173. pp. 52–60, doi: 10.1016/j.ymeth. 06.014.2020.

[15] N. Harbeck et al., "Breast cancer," Nat. Rev. Dis. Prim., vol. 5, no. 1, pp. 1–31,

[16] A. G. Waks and E. P. Winer, "Breast cancer treatment: a review," *Jama, vol*. 321, no. 3, pp. 288–300, 2019.

[17] O. J. Scully, B.-H. Bay, G. Yip, and Y. Yu, "Breast cancer metastasis," *Cancer Genomics Proteomics*, vol. 9, no. 5, pp. 311–320, 2012.

[18] W. J. Allard et al., "Tumor cells circulate in the peripheral blood of all major carcinomas but not in healthy subjects or patients with nonmalignant diseases," *Clin. cancer Res*., vol. 10, no. 20, pp. 6897–6904, 2004.

[19] A. van de Stolpe, K. Pantel, S. Sleijfer, L. W. Terstappen, and J. M. J. Den Toonder, "Circulating tumor cell isolation and diagnostics: toward routine clinical use." *AACR*, 2011.

[20] N. Houssami and P. Skaane, "Overview of the evidence on digital breast tomosynthesis in breast cancer detection," *The Breast*, vol. 22, no. 2, pp. 101–108, 2013.

[21] S. W. Fu, L. Chen, and Y. Man, "miRNA biomarkers in breast cancer detection and management," *J. Cancer*, vol. 2, p. 116, 2011.

[22] S. G. Kandlikar et al., "Infrared imaging technology for breast cancer detection–Current status, protocols and new directions," *Int. J. Heat Mass Transf.*, vol. 108, pp. 2303–2320, 2017.

[23] K. Ganesan, U. R. Acharya, C. K. Chua, L. C. Min, K. T. Abraham, and K.-H. Ng, "Computer-aided breast cancer detection using mammograms: a review," *IEEE Rev. Biomed. Eng.*, vol. 6, pp. 77–98, 2012.

[24] R. Guo, G. Lu, B. Qin, and B. Fei, "Ultrasound imaging technologies for breast cancer detection and management: a review," *Ultrasound Med*. Biol., vol. 44, no. 1, pp. 37–70, 2018.

[25] S. J. Mambou, P. Maresova, O. Krejcar, A. Selamat, and K. Kuca, "Breast cancer detection using infrared thermal imaging and a deep learning model," *Sensors,* vol. 18, no. 9, p. 2799, 2018.

[26] K. Suzuki, "A review of computer-aided diagnosis in thoracic and colonic imaging," *Quant. Imaging Med.* Surg., vol. 2, no. 3, p. 163, 2012.

[27] A. F. Agarap, "On breast cancer detection,"Proceedings of the 2nd International Conference on Machine Learning and Soft Computing - *ICMLSC* '18, 2018.

[28] M. Karabatak, "A new classifier for breast cancer detection based on Naïve Bayesian," Measurement, vol. 72, pp. 32–36, 2015.

[29] *R. Mormont, P. Geurts, and R. Marée, "Comparison of deep transfer learning strategies for digital pathology," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 2262–2271.2018.*