REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

BORDER CONTROL USING HUMAN SHAPE DETECTION

Ali Abdullah Mohammed MOHAMMED

Master`s Thesis

Supervisor

Prof. Dr. OSMAN NURI UCAN

Istanbul, 2023

BORDER CONTROL USING HUMAN SHAPE DETECTION

Ali Abdullah Mohammed MOHAMMED

Information Technologies

Master`s Thesis

ALTINBAŞ UNIVERSITY

2023

The thesis titled BORDER CONTROL USING HUMAN SHAPE DETECTION prepared by ALI ABDULLAH MOHAMMED MOHAMMED and submitted on 10/04/2023 has been **accepted unanimously** for the degree of Master of Science in Information Technologies

Prof. Dr. OSMAN NURI UÇAN
Supervisor

Thesis Defense Jury Members:

Prof. Dr. OSMAN NURI UÇAN	Department of Electricaand Electronics Engineering , Altınbas University	_____
Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	Department of Comptuer Engineering , Altınbas University	_____
Asst. Prof. Dr. Serdar KARGIN	Department of Biomedical Engineering Arel University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master`s Thesis.

Submission date of the thesis to the Graduate Education Institute: ____/____/____

I hereby declare that all information presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Ali Abdullah Mohammed MOHAMMED

Signature

DEDICATION

First, I dedicate my dissertation to my beloved mother, my deceased father, who were the source of my inspiration and support, my brothers, my sisters, my teachers, and everyone who supported me.

ACKNOWLEDGEMENT

I would like to express my gratitude to my supervisor Asst. Prof. Dr. OSMAN NURI UCAN for all support during my study. I would also like to thank my family, friends and everyone who helped and supported me during my studies.

ABSTRACT

BORDER CONTROL USING HUMAN SHAPE DETECTION

MOHAMMED, Ali Abdullah
M.Sc., Information Technology, Altınbaş University,
Supervisor: Prof. Dr. Osman Nuri UÇAN
Date: April / 2023
Pages: 66

Face recognition is a well-known kind of biometric identification because it may achieve high levels of accuracy with minimum intrusion into the lives of those being recognized. There are several unique methods for handling the facial authentication procedure. For a lot of these processes, significant safety measures must be observed. In the automated border control (ABC) system the theoretical foundations of both machine learning and deep learning are examined in depth. Our primary interest lies in neural networks, and more specifically convolutional networks of neuronal connections. Additionally, we demonstrate how to train deep neural networks in accordance with the real world. We conducted tests utilizing face and object datasets comprised of photos acquired in a maritime setting. In these investigations, multiple techniques are used to train and evaluate the performance of deep CNNs for the classification of pictures. Rapid categorization is an absolute requirement for applications using real-time object recognition. Because of this, the computational performance of the models is measured by examining the time required to complete image categorization. After completing the trials, we will discuss the outcomes and compare them to past study findings. In addition, we address the limitations of the study and recommend other research avenues.

Keywords: CNN, ABC, ML, DL, MAE.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

GDP	:	Gross Domestic Product
AI	:	Artificial Intelligent
SVM	:	Support Vector Machine
NN	:	Neural Network
ANN	:	Artificial Neural Network
ML	:	Machine Learning
SDDE	:	Software Development Project Estimation
BPNN	:	Backpropagation Neural Networks
DCF	:	Discounted Cash-Flow
IDX	:	Indonesian Stock Index
JKSA	:	Jakarta Stock Exchange
MLP	:	Multi-layer Perceptron
KNN	:	K Nearest Neighbors
RF	:	Randon Forest
ROI	:	Return on Investment
MAE	:	Mean Absolute Error
RMSE	:	Root Mean Square Error
MAPE	:	Mean Absolute Percentage Error
LSTM	:	Long Short Time Memory

1. INTRODUCTION

1.1 BACKGROUND

Face recognition is a well-known kind of biometric identification because it may achieve high levels of accuracy with minimum intrusion into the lives of those being recognized. There are several unique methods for handling the facial authentication procedure. For a lot of these processes, significant safety measures must be observed. In the automated border control (ABC) system, for instance, the biometric function is utilized to monitor and maintain the integrity of the border crossing process. Although iris, fingerprint, and face recognition biometrics might all be considered components of I ABC systems, face recognition is often the only biometric used for passenger screening at airports. The ABC must compare a photo of a passenger's face taken at the destination with the photo of their face saved in their electronic machine-readable travel document in order to establish whether or not a traveler is who they claim to be (eMRTD). Numerous assaults or threats to the ABC systems are conceivable, including as identity theft or fraud, which is often referred to as spoofing.

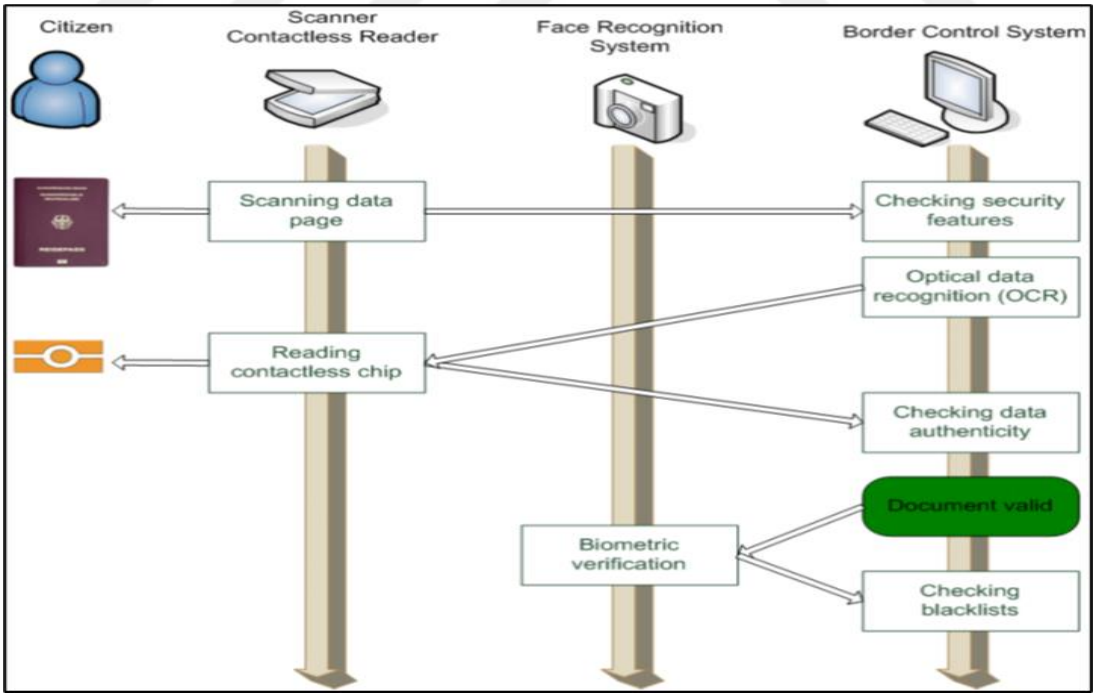


Figure 1.1: Automated Border Control System ABC [2].

the most dangerous cyber security risks. When authenticating a people identify, the same morphing processes used for passport images are used. As part of the morphing attack, a changed image of the legitimate owner's ID card (accomplice) and an altered image of the surrogate's or impostor's ID card are both stored in the eMRTD (criminal). The next phase of the system is to determine whether or not the passenger is who they claim to be. The effectiveness of the technique depends on the ability to compare the photograph taken on-site (ABC) with the image contained inside the eMRTD, which may have been altered. Since there is a great degree of resemblance between the face of the criminal and the morphed criminal accomplice picture, if an ABC does not include a MAD module, the typical verification outcome will be acceptance. The morphing technique was first used in the creative business, where it was used to create jaw-dropping visual effects in a variety of media including films, television shows, and ads. Initially, everything had to be completed manually; but, as soon as the first ground-breaking algorithms were devised, the necessary tasks started to be rapidly mechanized. It is vital to remember that differentiating two digitally blended faces may be difficult for even the most skilled professionals. Therefore, the approach shifted from its prior purpose as an aesthetic tool to an arsenal of ridicule. Given that ABC systems are vulnerable to assaults from applications that rely on face recognition, the morphing of facial images might be considered a serious concern. the results of the NIST Face Recognition Vendor Test MORPH indicate that the submitted MAD approaches lack robustness and performance when considering unknown and hard corpora. [4] In contrast, in addition to facial recognition, additional biometric features such as fingerprints and even the iris have been studied in morphing assaults. Face morphing is the attack that may do the most harm to ABC systems and is also the most difficult to recognize, hence it is the primary focus of our research. According to the European Border and Coast Guard Agency, the extensive deployment of ABC systems at airports over the last several years has increased study and attention to the potentially various risks (such as a presentation assault). This has resulted in a heightened focus on these possible hazards (FRONTEX). As the paradigm is difficult to detect, these assaults stimulate the development of algorithms for presenting attack detection (PAD) and morphing attack detection. Detection of morphing attacks is very crucial (MAD). In this paper, we provide a novel technique for detecting morphing attacks that use a reverse de-morphing strategy based on convolutional neural networks. This strategy was devised by the study's authors. In the sections that follow, we will examine the several ways in which this effort differs from past ones.

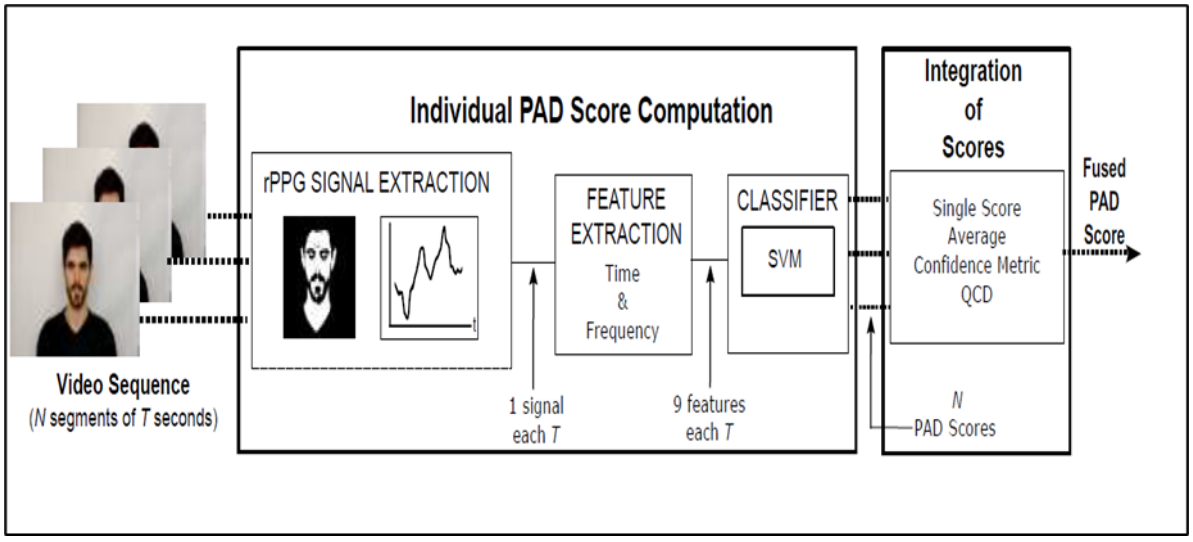


Figure 1.2: Individual Presented Attack Detection PAD [5].

1.2 PROBLEM STATEMENT

The false acceptance rate (FAR) and the false rejection rate (FRR) are two fundamental indicators for face recognition systems that are crucial for decision-making by authorities (FRR). In assessing the FAR of facial recognition systems, it is common to use imposters that require no effort whatsoever. It is probable that the previous methods of measuring FAR by using zero-effort impostors may not accurately reflect the actual number of erroneous acceptances in a realistic border control situation. Theoretically, anybody, regardless of their financial means, is capable of launching an assault against the system using a variety of technical and physical methods. Some individuals opt to wear masks, apply cosmetics, allow their facial hair grow out, or even undergo cosmetic surgery in order to disguise their true identity. Others prefer to establish a false identity via the use of a computer program or another technical approach. According to the claims made by a business that offers 3D face recognition, the issue of detecting identical twins, which has plagued this technology for a long time, has been resolved. Therefore, in order to get a more accurate estimate of FAR, it is essential to investigate possible assaults, their impact on FAR, and the resources required to execute them. Due to this investigation, authorities and other end-users of face recognition technology will be able to make more informed evaluations and become more aware of the problems, allowing them to take the required corrective measures.

1.3 THESIS JUSTIFICATION

Globalization is a process of deepening economic, social, cultural and political integration. This economic level has made the application of financial resources a complex task for investors. This is due to the amount of information coming from various parts of the world. In general, every company or individual investor, when applying their resources, thinks about at least two aspects: return and risk. With an adequate level of information and knowledge of the financial market, it is possible, for a certain level of return, to reduce exposure to risk. Therefore, an attempt to carry out a price forecast in the stock market can bring great benefits to investors, by increasing the level of information about the financial market, minimizing exposure to financial risk. In this sense, a computational technique called Artificial Neural Networks (ANNs) can be applied. The Artificial Neural Network (ANN) simulates on computers the functioning of the human brain in a simplified way. It has the ability to recognize patterns, identify regularities, deal with noisy, incomplete or inaccurate data and preview non-linear systems, which makes its application interesting in the financial market [4]. The use of ANNs can help investors in the choice of assets as it offers the possibility of predicting the behavior of stock prices in the future and thus subsidizing the decisions to buy and/or sell securities. In this context, this article aims to develop an artificial neural network capable of predicting the price accordingly and, with that, explore the ability of neural networks the research problem is the question: how the following artificial neural networks can be applied in the stock price prediction process.

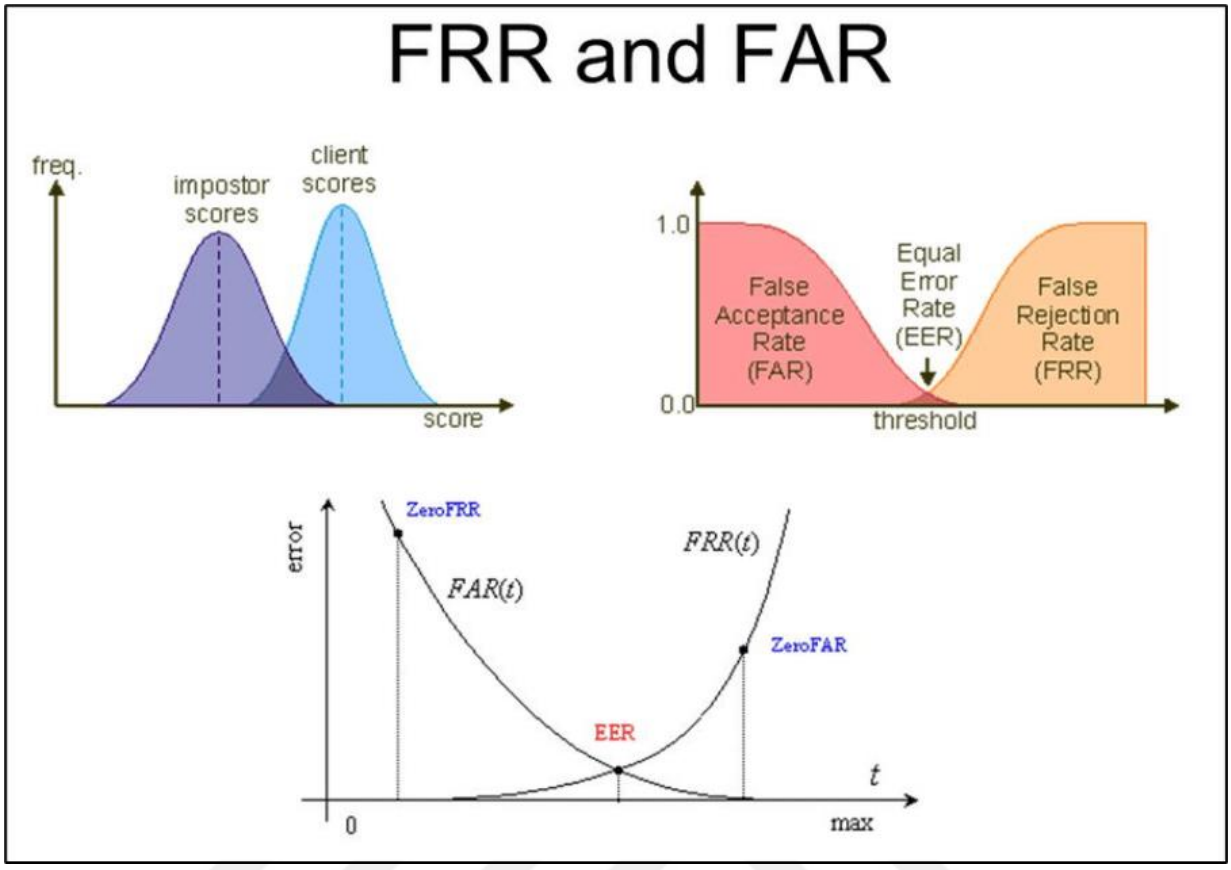


Figure 1.3: FAR and FRR in Human Shape Detection [7].

1.4 OBJECTIVES

The basic goal of installing video surveillance systems in any environment, regardless of whether it is public or private, is to increase the level of safety experienced by those who are there. These systems may be able to foresee potentially risky actions such as running and fighting, which may result in a decreased possibility that these activities would take place. Reduce the likelihood that terrorist attacks will be carried out.

1.5 CONTRIBUTION

This thesis, which investigates the object identification component of the border control system, focuses on research and experimentation employing the most contemporary object recognition approaches. "Object recognition component of the border control system" is the subject of this thesis. Throughout the thesis, the theoretical foundations of both machine learning and deep learning are examined in depth. Our primary interest lies in neural

networks, and more specifically convolutional networks of neuronal connections. Additionally, we demonstrate how to train deep neural networks in accordance with the real world. We conducted tests utilizing face and object datasets comprised of photos acquired in a maritime setting. In these investigations, multiple techniques are used to train and evaluate the performance of deep CNNs for the classification of pictures. Rapid categorization is an absolute requirement for applications using real-time object recognition. Because of this, the computational performance of the models is measured by examining the time required to complete image categorization. After completing the trials, we will discuss the outcomes and compare them to past study findings. In addition, we address the limitations of the study and recommend other research avenues.

1.6 THESIS STRUCTURE

The thesis is structured as follows: In Chapter 2, we will discuss some of the most important algorithms utilized for border control Recognition Systems in prior research. These algorithms have been used in previous studies. In Chapter 3 of this dissertation, both the proposed algorithms and their theoretical foundations are described. Chapter 4 discusses the implications that may be drawn from the outcomes of the tests. In the last chapter, we will discuss the significance of this study and the next measures that will be done.

.

2. LITERATURE REVIEW

2.1 CHAPTER OVERVIEW

Automating border control is a component of the smart borders' initiative. The program's objective is to protect the security of commodities passing through customs as well as other economic considerations. Referring to Adam and Others [1]. Countries from around the globe, including the United States of America, the European Union, the Republic of Rwanda, China, and Australia, are working to implement cutting-edge information technology that will allow for more precise control over the movement of people and goods across international borders. This will allow for improved international commerce regulation.

Since the turn of the century, there have been three generations of ABC systems that are notably distinct from one another. This was the conclusion reached by Gorodnichy and his coworkers. [2] First-generation ABC systems (such as Privium in The Netherlands, Global Entry in the United States, and Nexus across Canada and the United States) only process registered travelers, whereas second-generation ABC systems (e-Gates) serve travelers with an e-MRTD issued by the nation that installed the system, as well as foreign nationals via bilateral agreements. e-Gates serve passengers who possess an e-MRTD issued by the country that established the system. Listed below are some of the first ABC systems: (such as several states in the European Union and Australia).

The BGMS is responsible for ABC monitoring and control at the border, while the CSI is responsible for the interfaces between the various system components. In other words, the BGMS oversees and controls ABC at the border. Both of these duties come within the scope of the position of border patrol agent. By querying a variety of external systems, information required for an automated clearance request may be gathered from a vast array of sources. Authors Collaborators working under Mitra's direction The sharing of personal information such as biometrics necessitates a means of data transfer that is both safe and reliable across all participating platforms. In their study, Adam et al. achieved clearing with the application of technical methods. The justification for Based on the current condition of the biometric and document authentication systems, the Gate may be modified. Although multi-step procedures are increasingly common, the vast majority of activities require just one step. A single pass through the e-Gate is sufficient for a visitor to clear customs and cross the border

while simultaneously having their identity validated via this method. The clearing procedure consists of two phases: the first phase occurs away from the e-Gate, off-site, and the second phase occurs inside the e-Gate.

According to the results of Sun, Zhi, and their colleagues' research [5], traditional border patrol systems need a large number of human agents, while unmanned border patrol technologies necessitate the building of costly observation towers. You should be warned that these strategies have a high probability of activating false alarms and need a clear line of sight to be successful. In the form of a three-tiered system architecture termed BorderSense, academics have discovered a solution for problems of this kind. When information from depth sensors and information from multimedia sensors located at greater altitudes are combined, accuracy is enhanced. Robots and unmanned aerial vehicles (also known as UAVs) can cover more terrain from the air.

2.2 PREVIOUS WORKS

The ideal configuration for a border monitoring and defense system, as found by Suzuki, Takua, and other researchers. The proposed system can identify individuals who cross the border and communicate this information not just to a command center but also to border patrol officers. Along the border, a large number of sensor nodes have been erected to detect and identify anybody attempting to cross. Each sensor node has a perception sensor that interacts through wireless links. Due to the vastness of the area that must be monitored, extending cables from the power source to each individual sensor node in order to gather data from those nodes would be unfeasible. Individual sensor nodes would be connected to their own independent power source, such as a solar-electric generator, according to the suggested approach for border surveillance. Each node will be able to function autonomously as a direct result of this. The placement of the sensor nodes affects the effectiveness of the system when it is deployed in the field.

In [8], Succi et al. examine the use of kurtosis to the detection of footfalls. In the presence of background noise and at a variety of sensor distances, kurtosis measurements were collected from a large number of walkers. In addition, the recording parameters (soils) and the sensor's distance changed. The program was designed to reliably identify any security

violation occurring inside the sensors' area of vision. Despite its apparent simplicity, the technique performs its core functions well.

According to the conclusions of further research [9] conducted by Succi, Gervasio Prado, and others, waves are able to pass through and propagate through the earth, just as they do through other elastic media. The maximum distance a wave may travel is mostly determined by its characteristics and the rate at which it is weakened by the ground. Four fundamental types of seismic waves may be distinguished based on the different paths they travel through the Earth: shear waves, love waves, compression waves, and Rayleigh waves. Rayleigh waves are observable at a larger distance than compression and shear waves. 7 percent of the total energy is carried by compression waves, whereas 26 percent of the energy is carried by shear waves and 67 percent of the energy is carried by Rayleigh waves. Transverse waves include both shear waves and Rayleigh waves. To achieve the maximum degree of precision in footfall detection, it is necessary to use the more powerful Rayleigh wave.

Using a microphone-based system similar to the one described by Nakadai, Kazuhiro, and his colleagues [10] An array of microphones is utilized to record the background noise, followed by an upgraded peak-finding algorithm and a chronologically progressive approach to determine which sounds belong to which events. Support vector machines are used in the classification of data. These machines are used to generate a feature vector by combining many feature sets into one feature vector (Support Vector Machine). In this setup, eight microphones are distributed on a grid of 24 by 1.2 meters. This ensures that at least two of the microphones will capture the sound of the subject's footsteps. Individual sound occurrences are divided offline for the purpose of event identification. Using a feature extraction technique, each repeat of a sound may be assigned its own unique feature vector for characterizing it. Using classification algorithms, this research aims to determine the origin of a variety of auditory events, including regular footfall, quick footfall, a handclap, and a spoken phrase. Due to the regularity with which these audible phenomena occur in the real world, vigilance is required while looking for them. It is feasible to distinguish between the sound waves generated by a runner and those produced by a walker due to the greater frequency of the runner's sound waves. Using your hearing, you can discern between the footfalls of a runner and those of a pedestrian. There is a risk that some people may struggle to differentiate between the handclap and the footstep actions. The components of the system

architecture that were provided for the detection of footsteps operate as well when applied to the identification of other continuous events, such as speech.

[13] Sutin, Alexander, and their colleagues at the Stevens Institute of Technology created the Stevens Passive Acoustic Detection System, better known as SPADES, to detect, monitor, and categorize sounds occurring from both above and below the water's surface. The system's four hydrophone sensors allow for the collection of data in real time and the study of acoustic features. In this study, we integrate subjective digital filtering, spectrum analysis, and connectivity in order to analyze acoustic data simultaneously collected from multiple hydrophones, identify and isolate sources, and confirm bearing for multiple targets in relation to a single mooring point located in water. The purpose of this research is to improve the accuracy of acoustic data analysis, which is the main emphasis of this study. As a consequence of this design, researchers will have access to a considerable amount of the acoustic measurement data gathered. ITC 6050C hydrophones are included as a part of the design (International Transducer Corporation). Due to their underwater connections, hydrophones may be placed up to fifty meters distant from the principal underwater mooring without losing touch with it. This is made feasible by the fact that the connections are watertight. Hydrophones are utilized to record the acoustic data, which is subsequently sent to land-based computers and displayed in real time.

Jisha R.C, Maneesha V. Ramesh, and others [14] suggest integrating passive infrared (PIR) sensors with MICAz to recognize human invaders in border zones and communicate their position, direction of movement, and velocity to a central base. This would be done to prevent unauthorized entrance into a nation. * [This section should have references] (Below are a few references) The [footnotes and references] must include supporting references... * [You must provide a citation] [This section should have references] This article describes a new technique for developing a system that employs a passive infrared (PIR) sensor and MICAz to transmit the intruder's typical speed and approximate direction to a central base station. The network architecture determines the position of PIR sensors, with line topology being the most prevalent design. These sensors will only be effective as detectors if they have an unobstructed field of view in all directions. If the PIR sensor detects an unauthorized individual, a signal will be sent to the control room. From these data, we may deduce the typical speed and general direction of the intruder. Object identification and capture,

communication between nodes, data transmission between nodes and objects, and object collection comprise the four pillars of the system's proposed design.

Jin, Xin, and their coworkers [15] presented a method of feature extraction that employs symbolic dynamic modeling. This method transforms the sensor-generated wavelet coefficients into statistical patterns. After identifying these statistical patterns, the targets may next be categorized into the relevant categories. In order to correctly recover the original signal in the time domain after a wavelet transformation, noise reduction must be performed based on informed assumptions on the wavelet basis and range of scale. The symbolic representations of the wavelet coefficients, as opposed to the data in the real time domain, have the potential to provide a more precise reconstruction of the signal's characteristics. Despite not being assured, this potential does occur. It is possible to simulate low-dimensional statistical patterns obtained from symbolic images using probabilistic finite-state automata. The computation of lower-dimensional feature vectors is advantageous because it facilitates real-time transmission through a wireless sensor network, which often has a narrower bandwidth and fewer nodes than other kinds of networks.

Bhaskar, Harish, and coworkers [16] show that the integrated system takes into account a limited number of still images captured from real-time sequences at each phase of the monitoring procedure. Due to its emphasis on human targets, it employs both traditional detection and tracking techniques and face recognition technology. The difficult problem of recognizing and monitoring a large number of moving objects simultaneously starts with identifying moving targets. Despite the existence of common factors such as noise, camera motion, and lighting variances, this approach successfully distinguishes foreground components from background components. To accomplish this objective, we use a method called as blob detection, which tracks contours. This allows us to identify each component from the image's background noise and mark the portions of the image that correspond to each component. After targets have been discovered, the identification process may begin, and face recognition software is often utilized at this stage. If the target's face cannot be identified during the tracking phase, the technique should proceed without adding any identifying attributes (such as a signature) to the target. It is possible that the identity of the target might be determined using frontal facial recognition, making further identification for the tracking technique unnecessary. When a facial image of a target is unavailable, a

composite face representation of the target is created using normalization, alignment, and morphing algorithms. Due to this, we may decide to address the issue from the front, where it is more likely to stay stable, and give it our whole attention.

2.3 CONCLUSION

According to Harish, Palagati, and others [17], object tracking is used throughout the process of analyzing security camera data to identify potential attackers. This aids in obtaining the most precise findings possible. By extracting high-level class and event features from semantic video content, the proposed technique decreases the possibility of making an incorrect intruder-matching conclusion. Moreover, the research proposes that, depending on the conditions, semantic content object tracking approaches and ontologies should be implemented.

3. MATERIALS AND METHODS

3.1 CHAPTER OVERVIEW

The subject of computer science known as "Artificial Intelligence," or "AI" for short, focuses on the study of the development of hardware and software systems endowed with human-like abilities and capable of following a preset goal without human involvement. AI is the acronym for "Artificial Intelligence." Human capabilities include the ability to learn and reason, as well as the ability to organize and interact with people, as well as with technology and the environment. Other human abilities include the capacity for environmental interaction. Rather than predetermined instructions, the core of artificial intelligence is the ability to learn.

3.2 ARTIFICIAL INTELLIGENCE

When did the first attempts to train robots with artificial intelligence begin? In 1956, the first computer was constructed, marking the birth of AI as we know it today. The term "artificial intelligence" was not created until this year at a conference held in Hanover, in the United States. Many of the most significant individuals in the field of artificial intelligence, often known as the "Intelligent System," were present at the event. At this important event, several computer systems capable of logical reasoning, especially in the subject of mathematics, were shown. Allen Newell and Herbert Simon, who both worked in the area of computer science, created the Logic Theorist software. During World War II, the concept that computers might imitate human intellect gained ground. During this time period, the renowned computer scientist Alan Turing, who is sometimes referred to as the "father of modern computing," developed important ideas such as computability and computability. Among his other significant contributions are the Turing machine and the Turing test. In a 1950 article titled Computing Machinery and Intelligence for the magazine Mind, he designed a test called "The Imitation Game" to determine whether or not a computer is clever. This procedure, which is commonly known as the "Turing Test," consists of two phases. During the first phase of the procedure, a man, a woman, and an interviewer communicate by teleprinter while located in different rooms (Figure.3.1).

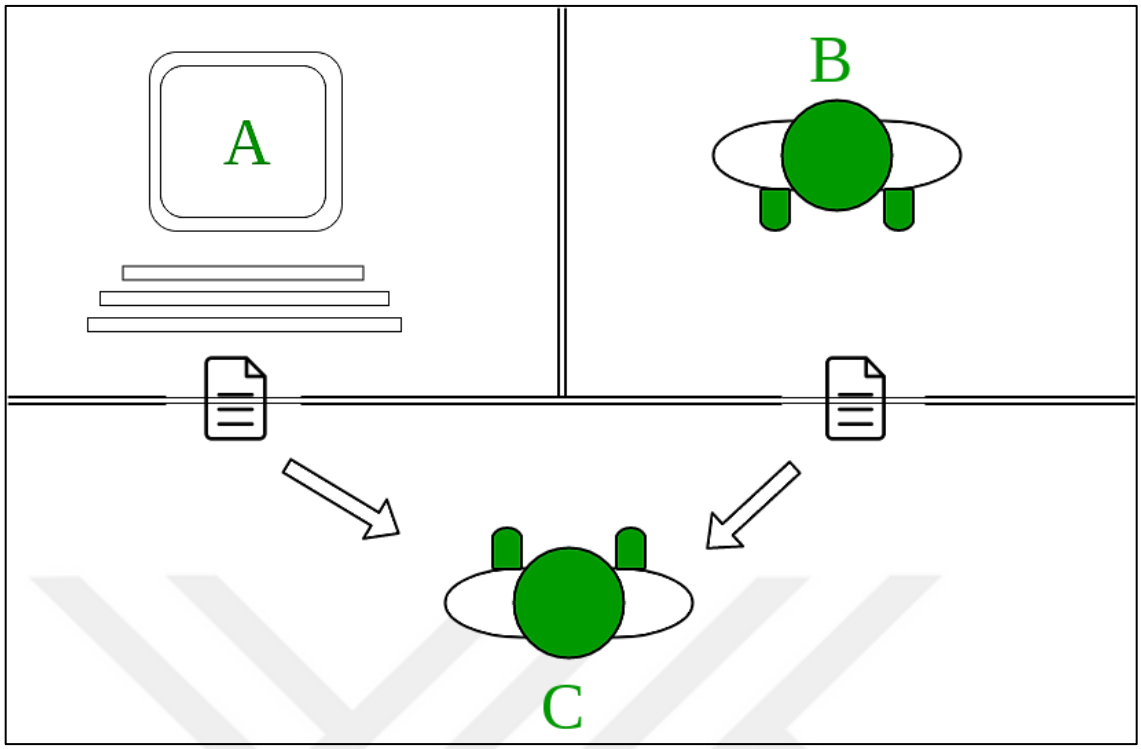


Figure 3.1: Representation of the First Phase of the Turing Test [8].

The examiner asks the two people questions and, not seeing them, does not know if he is addressing the question to the man or the woman. The questions are intended to be able to establish who the man is and who the woman is. Participants in the game receive the question and respond to the interviewer in writing, using the teleprinter.

So far, the game seems to be very simple, and the problem quickly solved. In reality this is not the case because the participants in the game can also lie. The two participants, in fact, have different purposes: one participant has the goal of facilitating identification by the interviewer, so he is sincere in his answers; the other participant, on the other hand, aims to misidentify the interviewer, so he provides untrue answers. Therefore, the interviewer not only does not know who the man is and who the woman, but he does not even know which of the two is lying. By repeating the game N times, the interviewer mistakes the sex of the participant's X times, so the error rate is equal to NX. In the second phase of the game, we replace one of the two participants with a computer (Figure. 3.2) and this time the interviewer has to figure out if the man or the computer is answering the process is always the same. The interviewer asks questions to participants via teleprinter and cannot see them. He doesn't know if he's talking to two men or if one of the m is a machine. At the end of the game, he

will have to identify the participants based solely on their written answers. The game repeats itself.

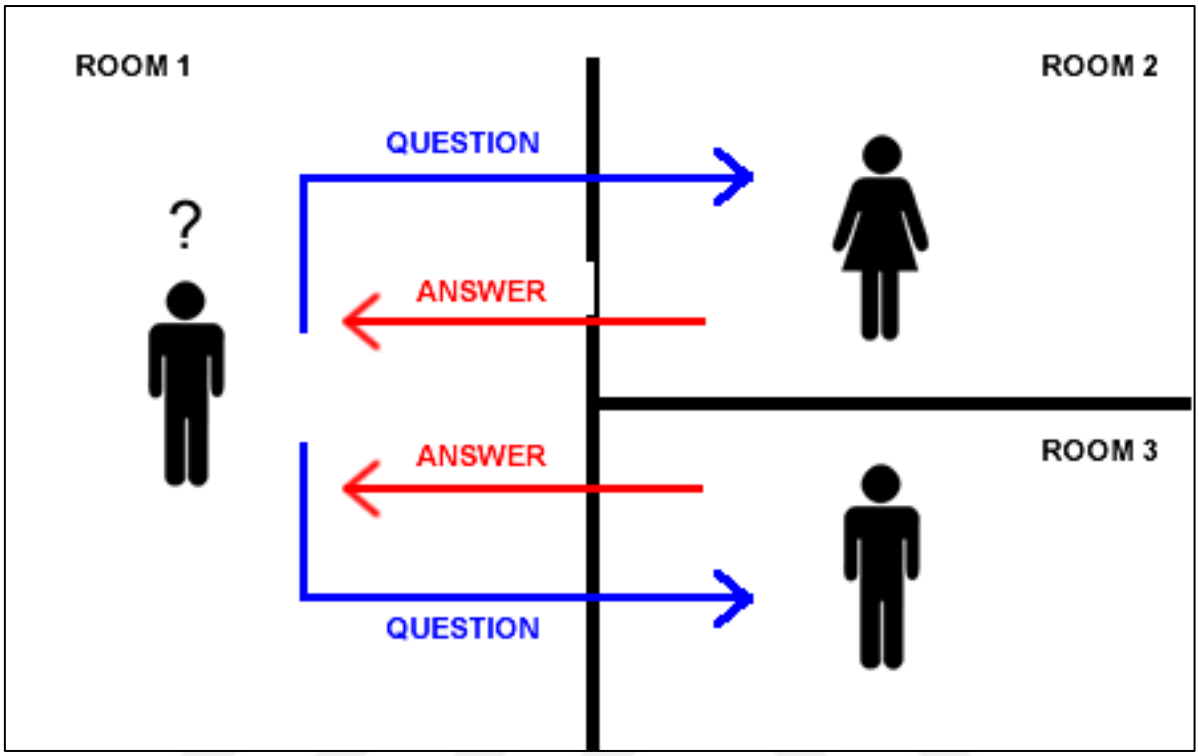


Figure 3.2: Representation of the Second Phase of the Turing Test [10].

No.times and the interviewer misidentifies the participant's Z times, resulting in a percentage error rate of NZ. The Turing Test is passed if the error rate in the game in which the computer participates is similar or less than that in which only humans participate, that is, if NX$\simeq$*NZ*. If the test is passed it can be said that the computer is intelligent as it is indistinguishable from the human being. The Turing Test is to be considered as the manifesto of Artificial Intelligence. In that historical period there were notable scientific discoveries. In the 1940s, biologists developed the first theories to explain how intelligence and learning were the result of signals transmitted between neurons in the human brain. Wiener developed the cybernetic theories of control and stability of electrical networks; Turing developed the theory of computation and Shannon the theory of information. In 1943 McCulloch and Walter Pitts published a work in which they showed how a system of artificial neurons could be able to perform basic logical functions. Starting from the theories of McCulloch and Pitts, Rosenblatt in 1956 conceived the first machine capable of simulating the functioning of neurons at a software and hardware level. The system he created is known as Perceptron.

The Perceptron was a system with only two layers of neurons: one for the input of data and the other for the output of the results. Rosenblatt then tried to create more complex networks by organizing them in a hierarchy of multiple layers: by passing data from one layer to another, it would have been possible to create pattern patterns and solve increasingly complex problems. published an essay in which they showed that neural networks were able to perform only a few elementary operations, while there was no hope that they would accomplish more complex tasks. This brought about the end of research on neural networks until deep learning was born. Rosenblatt then tried to create more complex networks by organizing them in a hierarchy of multiple layers: by passing data from one layer to another, it would have been possible to create pattern patterns and solve increasingly complex problems. published an essay in which they showed that neural networks were able to perform only a few elementary operations, while there was no hope that they would accomplish more complex tasks.

3.3 STRONG AND WEAK ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) is a relatively new science within the area of information technology (IT) that has gained attention for a variety of reasons. Several fields, including mathematics, economics, neurology, and cybernetics, have had a major impact on the development of artificial intelligence throughout its history.

AI, or artificial intelligence, is the study of theoretical foundations, methodologies, and techniques that enable the design of hardware systems and software program systems able to provide the electronic computer with performance that, to the untrained eye, appears to be the exclusive domain of human intelligence. AI was characterized by an Italian pioneer as the study of theoretical underpinnings, procedures, and techniques.

In reality, there is no universally accepted definition of artificial intelligence, and interpretations can vary depending on the focus: on the one hand, one can zero in on the internal processes of reasoning, and on the other hand, one can examine the external behavior of systems, with the similarity or proximity to human behavior serving as a sort of "effectiveness metric" in both cases. In other words, there is no commonly recognized definition of artificial intelligence. The scientific community has determined that there are two distinct varieties of artificial intelligence, which they refer to as "weak" and strong.

3.3.1 The Weak Artificial Intelligence

Weak artificial intelligence is comprised of problem-solving algorithms that can replicate certain sorts of logical human thought in order to solve issues and make decisions, but which fall short of the intellectual powers of humans. As a result, it does not strive to replicate the human brain as closely as possible. Weak artificial intelligence operates by studying comparable situations, comparing and contrasting them, constructing possible solutions, and selecting the most suitable one. In this setting, simulation of human behavior becomes vital.

3.3.2 Strong Artificial Intelligence

Those who subscribe to the Strong Theory of Artificial Intelligence believe that it is feasible for computers to attain consciousness. The technology is more than simply a tool; if it has been created properly, it has the capacity to think like a person. The basis of robust artificial intelligence consists of a collection of programs designed to imitate the abilities and body of knowledge of human specialists via the use of an automated system. These applications form the foundation of Strong AI. Since then, it has been plainly clear that robots are not just capable of reasoning and problem-solving.

3.4 TECHNIQUES AND LEARNING MODELS

By engaging in independent mental thought, he may build knowledge and self-awareness. Some researchers think that it will be feasible in the not-too-distant future to construct robots that are entirely autonomous and even more intelligent than humans. This imaginary phenomenon is known by its name, the singularity. Even before the development of artificial intelligence, robots were able to perform instructions correctly. Artificial Intelligence is distinguished from other types of intelligence from a technical and methodological standpoint by the learning process or model by which intelligence acquires task or activity competency. The key contrast between Machine Learning and Deep Learning is that their respective learning models are separate.

3.5 MACHINE LEARNING

The subject of Artificial Intelligence known as Machine Learning is primarily concerned with the development of data-driven learning systems (or just Machine Learning for short).

These approaches "teach" the software to "learn" from its mistakes and perform an activity or task without human intervention.

Machine learning may be divided into two separate sub-categories: supervised learning, in which the computer is given full examples to use as a guide to finish the work, and unsupervised learning, in which the computer is allowed to do the task on its own without human supervision (unsupervised learning).

The objective of supervised learning is to train a system to generalize from a set of input data to a set of output data by supplying input data and knowledge about the desired outputs. This is performed by giving the system with input data and information about the desired outcomes.

In order to facilitate unsupervised learning, the system merely gives data sets without indicating what the expected outcome would be. This second kind of learning aims to "reverse engineer" the data in order to discover previously concealed patterns and models or to identify an underlying coherence in the inputs without relying on labeling as its main strategy.

Other subcategories of machine learning, such as reinforcement learning and semi-supervised learning, allow for an even finer categorization of machine learning based on how it operates.

In reinforcement learning, the system interacts with a dynamic environment to gather input data, pursue a goal, and gain a reward upon success, all while identifying and correcting its own mistakes. When the system is effective, it gets rewarded. The system's activities are governed by a learning process based on reinforcement and correction. Using this method, the computer is able to learn how to play a game (or drive a vehicle) by focusing its efforts on completing a certain task effectively and increasing the amount of reward it gets. This suggests that the system will improve its performance appropriately based on prior results and will learn as it plays (or drives, in the case of a driving simulator).

Some inputs for supervised learning provide examples of related outputs, but others do not. This kind of hybrid learning is known as semi-supervised learning (as in unsupervised

learning). The ultimate objective is to identify effective problem-solving models, data structures, rules, and functions.

3.6 Deep Learning

The theoretical foundations of Deep Learning, often referred to as "deep learning," extend well beyond the standard multi-level machine learning. Deep Learning is a subfield of Machine Learning concerned with the study of artificial intelligence using artificial neural networks whose structure and function are modeled after that of the human brain.

Depending on the level of abstraction involved, computer models with several processing layers may learn to represent data in a number of ways within the topic of Deep Learning. It does this by explaining to a computer, using the backpropagation approach, how to alter the parameters required to construct the representation at each level depending on the representation at the previous level. This explains the intricate structure behind massive data sets. Deep learning is advancing rapidly in the area of artificial intelligence in terms of solving issues that, despite widespread attention, have remained unsolved for decades.

The current success of Deep Learning may be ascribed to a variety of variables that have helped fill the hole in some areas where achieving the desired outcomes was previously impossible. These variables include a growth in the quantity of accessible data and the evolution of extremely advanced parallel computing systems. Prior to this, it was impossible to obtain the desired outcomes. Utilizing GPGPU (General Purpose Graphics Processing Unit) and modifying neural network training algorithms to attain optimum results led to performance improvements.

Because "Deep Learning" and "neural networks" are often used interchangeably, their respective meanings may be misconstrued. In the context of deep learning, just the number of layers of a neural network is considered "deep." Considered to be an example of a Deep Learning algorithm is a neural network with more than three layers. Simple neural networks, such as those with just two or three layers, are not considered complicated.

3.7 Supervised Learning

The objective of supervised learning is to generate a function or mapping using a set of labeled training data. The training data consist of the X input vector and Y output vector of labels or tags. The label or tag included in vector Y describes the input instance in vector X. When together, they serve as an example for kids to emulate. In other words, examples for use in training are included within the data used for training purposes. The labels for each training sample in the training set are created as a vector called Y. The bulk of the time, machine learning models are constructed using a training set taken directly from the dataset. To evaluate the performance of a model in terms of both its output and its functionality, an additional subset of the dataset known as the test set is used. The samples that comprise the training set and the samples that comprise the test set are drawn from the same dataset.

The manager will determine which characteristics of the vector need labeling. Despite the fact that human subject matter experts are often engaged in supervisory positions, automated labeling techniques are becoming more prevalent. Data that has been manually categorized is very valuable for supervised learning owing to its intelligence and consistency. Under the umbrella of supervised learning, classification and regression are the two most used kinds of algorithms. The purpose of a classification technique is to develop a simple model of the frequency with which distinct class labels occur in response to the predictor parameters. When the class label is unknown but the values of the predictive attributes are known, the resultant classifier is used to assign class labels to test samples. This happens when the label for the class is unknown. a training set categorization job example The objective of supervised classification algorithms, which get their fundamental knowledge from training sets, is the determination of the class. Regression is a statistical strategy that aims to identify correlations between three or more factors in order to answer research questions. After entering the x value, the y value calculated by the regression model is returned. In contrast to classification, which is limited to grouping things into a preset number of categories (labels), linear regression uses data as its major input and yields meaningful results (unlike the classification that receives data as input and returns a label in output, a label to which the data belongs and in general therefore a discrete value). Following is a discussion of the most significant classification algorithms. Categorization techniques include decision trees, support vector machines, naive bayes, and artificial neural networks.

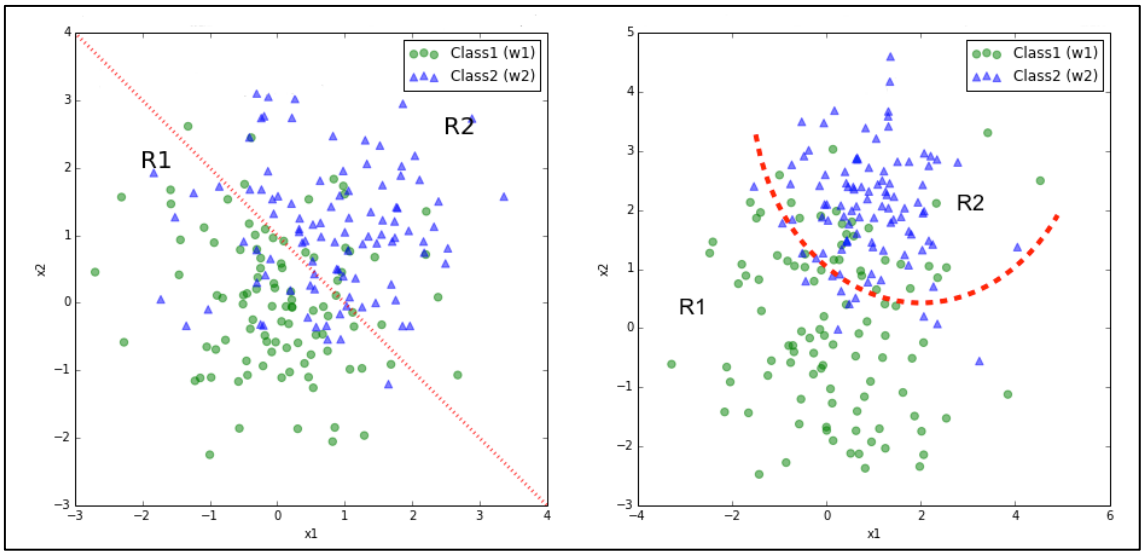


Figure 3.3: Example of Classification Data [11].

3.7.1 Decision Tree

Instances are categorized using decision trees that rank them based on the attribute values they hold. Each node in a decision tree represents a characteristic of an instance that must be classified, and each branch represents a possible value for that node's location on the tree. Classification begins at the root node, and the values of each instance's characteristics are used to rank the instances.

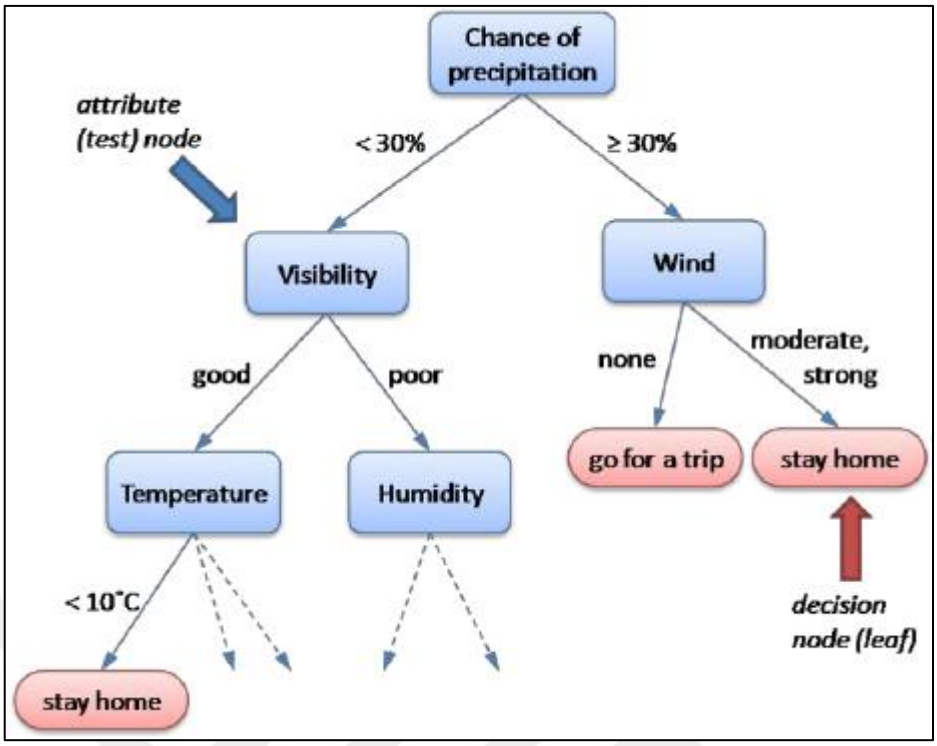


Figure 3.4: Example of a Decision Tree [24].

3.7.2 Support Vector Machines

Margin, which refers to the distance between two sets of data on opposing sides of a hyperplane, is a crucial component of the SVM approach. It has been shown that the maximum generalization error that may be expected can be reduced by optimizing the margin and providing the greatest feasible distance between the dividing hyperplane and the instances on both sides of the split. If it is possible to differentiate two classes linearly, the ideal separation hyperplane may be obtained by minimizing the square norm of the separation hyperplane. The solution for data that are linearly separable is a linear combination of just the points on the ideal separation hyperplane. These positions, often referred to as support vector points, constitute the ideal separation hyperplane.

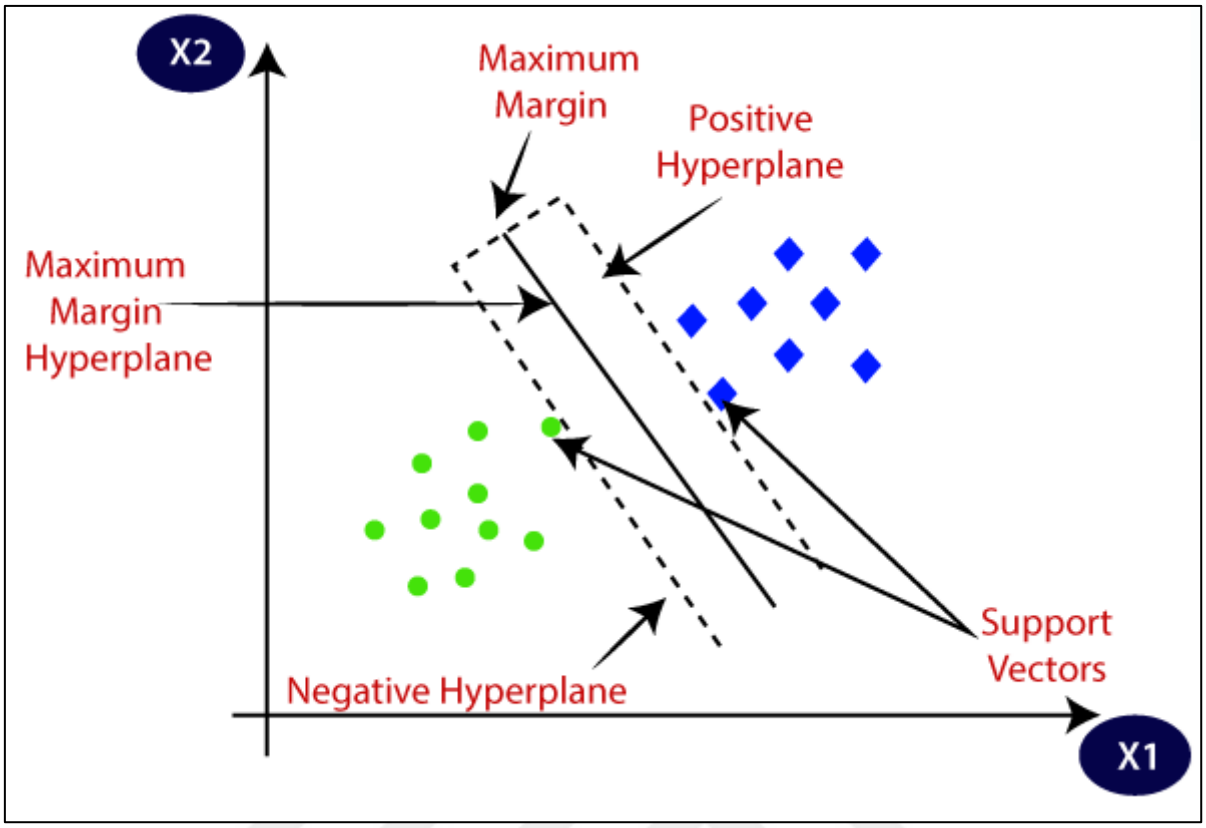


Figure 3.5: The SVM Algorithm [30].

3.7.3 Naive Bayes Stock Books

The Naive Bayes classifier has become one of the most well-known statistical learning algorithms in recent years. The simplest form of Bayesian network is the naïve Bayes (NB) network. It is an acyclic direct network with a single parent node (representing the unobserved node) and numerous child nodes (representing the observed nodes). In the context of their parents, there is a strong presumption of independence amongst the offspring of each node. The independence model, commonly referred to as the Naive Bayes model, is based on the following estimate:

$$R. = P.(i \mid X)P.(i)\ P\ (X \mid i)P.(i)\ QP\ (Xr \mid i)$$
$$P. =\ (j \mid X)P.(j)\ P\ (X \mid j)P.(j)\ QP\ (Xr \mid j) \quad (3.1)$$

Prediction I is made if R is larger than 1; else, prediction j is made. The class label value with the greater of these two probabilities is the one more likely to represent the actual label. Because the probabilities P (X, I are derived using multiplication in this classification

approach, they are especially sensitive to the emergence of a 0 when one of the variables in question is 0.

Because Bayesian classifiers assume that child nodes are independent, their accuracy is often inferior to that of other learning approaches. This is because it is nearly never correct to assume that child nodes are autonomous.

3.7.4 Artificial Neural Networks

An artificial neural network is a kind of computer model whose structure is derived from the topology of organic brain networks (ANN). Complex computations may be outsourced to artificial neural networks based on the premise that a large number of neurons can be interconnected in a network to produce a bigger entity. In a representation of a neural network that resembles a graph, neurons represent "nodes" and (direct) arcs linking neurons represent the connections formed between the outputs of the neurons. In order to provide a high-level explanation of artificial neural networks (ANNs), we will utilize a feed-forward network as an example. This network may be represented as an acyclic direct graph, and its vertex and edge weights can be expressed as G = (V, E) and w: E R, respectively. Individual nodes in this network represent neuronal components. R R is a scalar function used to represent each neuron and to describe the state and activity of each neuron. The sign function, a uniform approximation of the threshold function, and the function itself will be our major emphasis. These capabilities are known as "firing" functions of neurons. The output of a neuron is linked to its matching input along each arc of the graph. When computing the input of a single neuron, the outputs of all neurons to which it is linked are first added together, and then each output is assigned a weight [10]. Each "layer" in a typical network has about the same number of nodes. An input layer, a configurable number of intermediate or hidden layers, and an output layer make up the basic structure.

This is the reason why we speak about "deep learning.

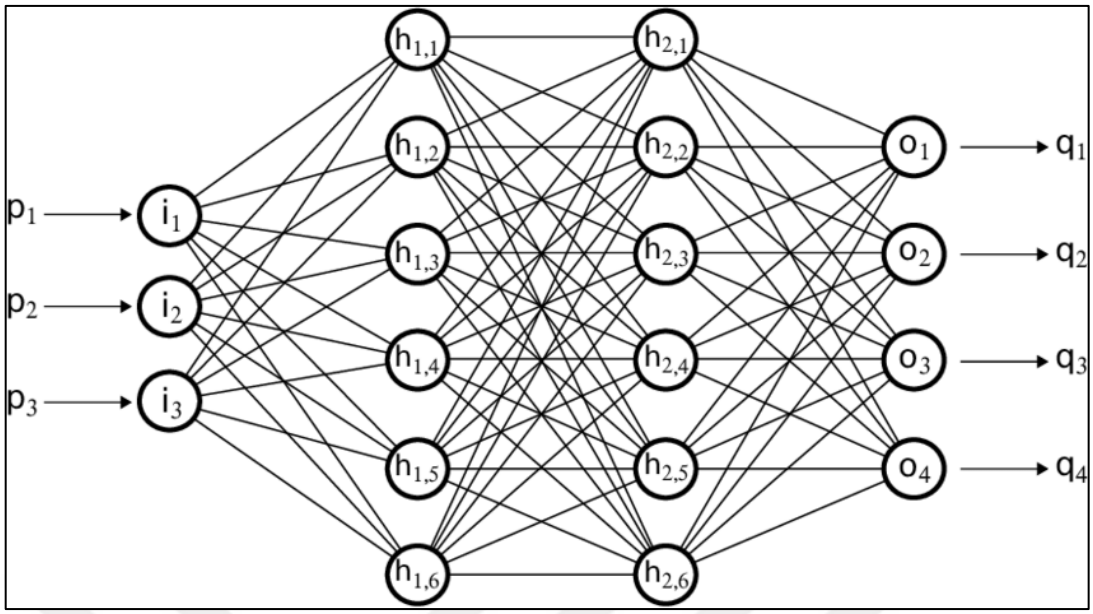


Figure 3.6: Example of an ANN Feed-Forward [33].

The two components of an artificial neural networks (ANN) learning process are training and inference. During the training phase, a set of data is supplied to the network in order to determine its input-output mapping. The weights of the connections created between neurons are then calibrated before to training the network using a fresh data set. During classification, the activation levels of each output unit are computed depending on the signal present on the input units. This signal is then sent across the whole network. As a result, there are three crucial factors that influence the performance of an ANN: the kind of inputs, the design of the network, and the relative importance of the weight applied to each input link. Since the first two parameters cannot be modified, the behavior of the ANN is dictated by the values presently allocated to the weights. The network's weights are first initialized at random, and it is then repeatedly presented with examples from the training set. The output of the network is then assessed. The actual output of the network is compared to the predicted output using an error function, and the network's output values are assigned to the network's input nodes. Then, a minute modification is performed to each weight in the network in the direction that yields output values that are the closest feasible match to the target values. It is feasible to train a network using any number of distinct algorithms. On the other hand, the Back Propagation (BP) technique is the most well-known and often used method for obtaining weight values. The BP algorithm may be reduced to its simplest form, which consists of the six stages shown below.

a. Determine the degree to which the actual output of the network matches the expected output based on the available information. Determine the extent to which the output of each neuron is diminished.
b. Determine the output that should be anticipated for each neuron as well as a scaling factor, which is the amount by which the actual output should be decreased or increased. This is the exact area where the mistake occurred.
c. Adjusting the weights of individual neurons may aid in reducing the model's local errors.
d. Imposing a bigger "duty" on neurons linked with larger weights by making them liable for a larger proportion of the entire mistake.
e. Consider the "responsibility" of all players on the previous level to be an error and epeat steps 1 through 3 on the neurons on their level.

The BP algorithm will need to make a few adjustments to the weights in order for everything to be properly positioned and operating. During the inference phase, however, a fresh set of data is sent to the trained network so that it may make classifications based on those it has previously made. After the training period has concluded, something occurs.
The designer is responsible for determining all network parameters, including the network's size, architecture, error function, batch size, and activation function. In addition to the feed forward ANNs mentioned in this page, there are a number of other accessible ANNs. Despite this, they all use the same strategy. There are several significant ANN networks, including the ones listed below:

a. Convolutional neural networks, which are a subtype of ANNs built specifically for handling high-dimensional data, are a subset of artificial neural networks (grid data). In at least one of its layers, the standard multiplicative matrix is replaced with convolution using a specialized matrix called the kernel. The name for this matrix is the kernel.
b. Recurrent neural networks, a kind of artificial neural network in which the output of one layer serves as the input for the layer below it in the hierarchical structure. Due to the interconnected nature of the layers, it is possible to employ a single layer as a state memory and to show dynamic temporal behavior based on data received at earlier time instants. Using a single layer makes these two possibilities attainable. Autoencoders are an artificial neural network used for unsupervised learning. We will discuss autoencoders

in more detail in the future, since they are a crucial component of the technique utilized here.

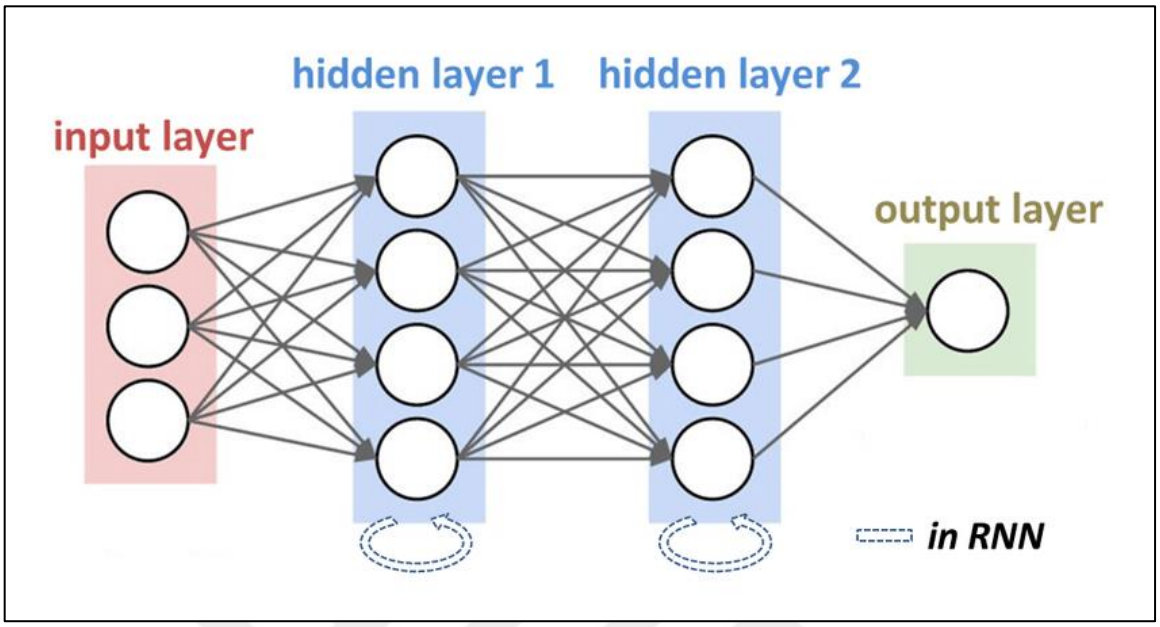


Figure 3.7: Architecture of a Recurrent Neural Network [42].

3.8 SUPERVISED LEARNING FOR HUMAN SHAPE DETECTION

Utilizing supervised learning in human form recognition enables the development of trustworthy and effective automated systems. However, sightings of humans are quite uncommon. Finding and categorizing anomalous behavior is challenging for a variety of reasons, including the unpredictability of the human form and the necessity for data sets to contain a label that distinguishes normal points from outliers. A fair and balanced training data collection with about the same number of examples for each class is necessary for an effective learning process. As a consequence of these restrictions, the training process is far more difficult than it should be. Since the conditions around an issue are not always documented in logs or databases, our HPC's system administrators are solely responsible for categorizing datasets. The accurate labeled datasets required for supervised algorithms are thus not always available.

In contrast, as part of our study, we have access to data sets that have been labeled as being in an aberrant state, since the MARCONI HPCs are equipped with very sophisticated data acquisition tools. As a result, we may choose for the method of supervised learning.

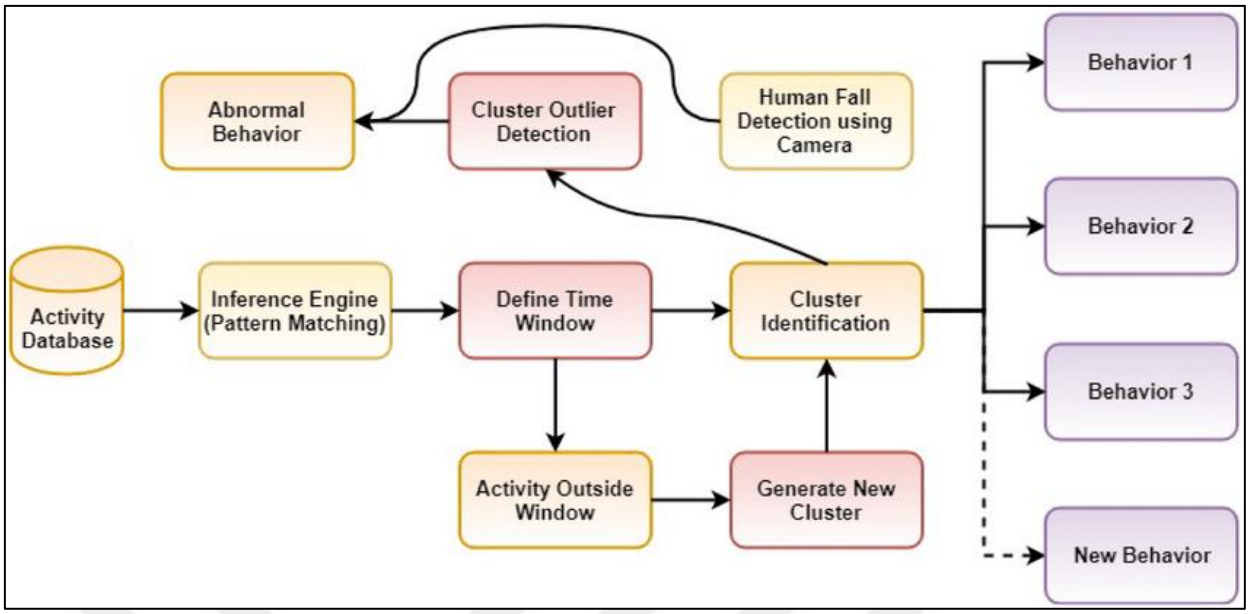


Figure 3.8: Example of our Dataset the State Column Tells us if There is a Human Shape or Normal Behavior [32].

3.9 UNSUPERVISED LEARNING

Unsupervised learning is a field of study that examines how computers might discover how to represent data in a manner that accurately represents the data's underlying statistical structure. In contrast to learning that is monitored or rewarded, no outcomes or evaluations have been set for each input when it comes to learning without supervision. Unsupervised algorithms consist of clustering algorithms, statistical methods, and artificial neural networks, among others (ANNs).

3.9.1 Clustering Techniques: K-Mean's, Hierarchical and DBSCAN

The act of quickly dividing a dataset into a partition containing k separate clusters, indicated by the notation C = C1, Ck, is fundamental to clustering approaches. The purpose of these approaches is to provide a local approximation of a global objective function by repeatedly upgrading an existing answer. K-means is the preferred technique for partition clustering. This technique employs an iterative transfer methodology to develop k-way clustering that reduces locally the distortion between individual data items and a set of k cluster representatives. Each centroid, which defines a cluster, is created by averaging the vector coordinates of all the components that comprise it. In the first method, distortion is assessed

in terms of the Euclidean distance, but the operation itself is separated into two stages. To begin, we relocate all of the objects until they are positioned in the cluster with the nearest centroid. After all of the items have been processed, the centroid vectors are changed to reflect the new groupings. The process of iterative refinement continues until a previously established halting condition is met. This often occurs when there is no longer any variation between iterations in the assignment of items to clusters.

Instead of generating a flat data split, hierarchical algorithms may generate a series of nested clusters that can be combined to form a tree structure. This enables the establishment of an idea hierarchy, which is the reverse of what a flat split of data would produce. There are fundamentally two distinct types of classification methods used to classify them:

a. In the agglomerative method, each item is initially positioned into a distinct cluster. At each level, the two clusters that are geographically closest to one another are connected, establishing an ascending hierarchy that begins at the bottom.
b. Separative: All n items are grouped into a single group in the first phase. Utilize the hierarchical descending technique, which includes separating a cluster into two new clusters at each level of the hierarchy.

In any case, the resulting hierarchy may be represented graphically using a dendrogram, which is a kind of tree structure. A dendrogram is composed of nodes that represent each cluster that the algorithm constructs. Cluster-relationships illustrate the merging and splitting activities that occur throughout the clustering procedure. illustrates the foundations of agglomerative clustering using a sample dataset of five items and their relative cluster locations as an illustration. When two clusters combine, their similarities often become less prominent. In contrast to the overwhelming majority of data partitioning techniques, which require the user to supply an initial value for the number of clusters k, this approach allows the number of clusters k to be determined dynamically.

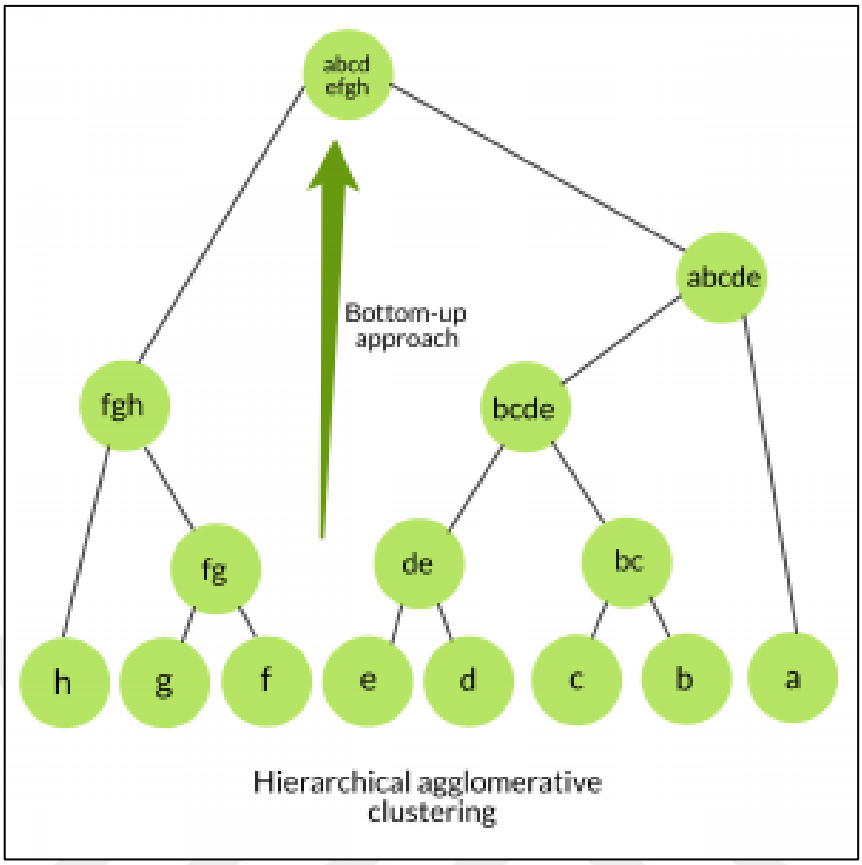


Figure 3.9: Example of an Agglomerate Hierarchical Clustering Algorithm [39].

The purpose of density-based clustering algorithms, or DBSCAN for short, is to identify clusters of any size or form. Cluster refers to a region of high density in the data space that is unique from other clusters owing to the existence of low density regions. DBSCAN is sensitive to input settings, particularly when data density is not uniform, despite its ability to distinguish clusters of any shape. A traditional DBSCAN method is characterized by two factors: the number of points that need to be grouped and their distance from one another. The trip begins at a location chosen at random and never visited before. If this point is not linked to enough other nearby points to create a cluster, it is referred to as noise. Remember that this point may be clustered with another point in the future if the second point's surrounds are large enough to permit the clustering. If it is determined that a certain point is an important component of a cluster, then all points in close proximity to that point will be deemed to be part of the cluster. When there is intense rivalry in a particular region, the points directly next to them and those around them are added to the total. This method is repeated several times in order to find the whole density-connected cluster. When a

previously unexplored region is recovered and researched, it is common to find yet another cluster of ambient noise. When there are a large number of points in one location, those points and the surrounding area become intertwined. This occurs everywhere there are a large number of points. This strategy will be used to continue the search for the density-linked cluster until its whole structure is determined. The subsequent step is the acquisition and examination of a new location, which often leads to the discovery of yet another cluster or background noise. The undiscovered place is then obtained and analyzed. In addition to their own dense neighborhood, all locations in surrounding dense areas are considered. This strategy will be used to continue the search for the density-linked cluster until its whole structure is determined. When a previously unexplored region is recovered and researched, it is common to find yet another cluster of ambient noise.

The following is a list of some of DBSCAN's benefits:

a. DBSCAN may be used to any dataset without requiring a priori knowledge of the number of existing clusters. In contrast to the k-means method, which requires an initial cluster size specification, this technique does not need an initial cluster size definition.
b. DBSCAN is capable of identifying clusters of various sizes and forms.
c. It can handle a certain amount of naturally occurring noise in the data.
d. DBSCAN utilizes just two parameters and is not too concerned with the presentation order of the data items.

One of the shortcomings of DBSCAN is that it is sensitive to the distance measure used for the Query (P,) operator. The most frequent kind of distance measurement is known as the Euclidean distance. Due to something known as "dimensionality care," it may be difficult to find an adequate value for this measure when working with high-dimensional data, which might render it almost unusable. This impact is nonetheless constant across all approaches derived from the Euclidean distance. In addition, the approach is incapable of clustering data sets with significant density variations since an optimum combination of minimum points (within a cluster) cannot be determined for all clusters. This prohibits the algorithm from clustering certain data sets.

3.9.2 Support Vector Machine SVM

a. Statistical Techniques

In the majority of situations, it is prudent to presume that the data was generated by a statistical procedure. Following this, it is feasible to describe the data by identifying the statistical model that corresponds to the data the closest. The model is explained in terms of the distribution it generates as well as the distribution's parameters. This procedure begins with the selection of an appropriate statistical model, followed by the inference of the model's parameters from the data. Mixture models, which employ many distributions to simulate data, serve as the foundation for these tactics. A distribution is used to represent a cluster, and the distribution's parameters provide insight into some feature of the cluster, most often its center and spread. There are several statistical approaches to consider.

In mixtures models, the data is presented as a collection of observations obtained from a number of distinct probability distributions. Normal multivariate distributions are typically assumed despite the absence of a hard-and-fast rule specifying the sort of probability distribution that should be used. This is because normal multivariate distributions are simple to compute and have been shown to yield adequate outcomes in a number of situations. Models of mixtures may be considered as conceptualizations of the data generation process outlined in the next paragraph. Randomly choose a distribution from a group of distributions of the same type but with different parameter values. This will lead to the creation of an item. Yes, iterates through the required operations to generate m brand-new objects.

Using statistical approaches, it is possible to both characterize and estimate the model parameters for these distributions (clusters). Additionally, it is feasible to instruct them on which things belong to certain categories. In contrast, mixed modeling does not provide a definite assignment of objects to clusters; rather, it provides the chance that an object belongs to a certain cluster. This is in contrast to conventional modeling, which does assign objects to clusters definitively.

If a statistical model for the data already exists, the next step is to estimate the model's parameters. In order to achieve this objective, maximum probability esimation is often used (MLE).

where and are the mixture model parameters, which are based on a Gaussian distribution (Univariate Gaussian Mixture). Because the logarithm is a monotonically growing function,

it is crucial to remember that the parameter values that maximize the probability of the loglikelihood function simultaneously increase the probability. This is because logarithms are rising functions. Using the MLE method, it is also possible to estimate the mixture model's parameters. If we know the distributions from which the data originated, we may reduce the job to predicting the parameters of a single distribution given the data. In an environment that is more inclusive and representative of real life, it is unclear which distribution was used to create which points. Since there is no direct method for calculating the likelihood of each data point, it would seem that the MLE cannot be utilized to arrive at accurate parameter estimates. The EM algorithm is the answer to this current issue.

The EM method starts with an informed estimate of the values of the parameters, then uses this information to compute the probability that a given data point is generated from a certain distribution. (These factors should be used in order to get the largest potential improvement in likelihood.) This technique will be repeated until the projected parameters experience little or negligible change, or almost no change. Therefore, we use an iterative strategy that optimizes the estimated probability. Consider the following as an example of the EM algorithm's application:

a. Determine substitute values for the model's configuration choices. Similarly, to k-means, this may be accomplished in a variety of methods or randomly.
b. During the step of expectation, you will calculate the value of the function prob (distribution | xi,) to determine the likelihood that a certain item belongs to each distribution.
c. The Maximization Step entails leveraging the learned probabilities from the Expectation Step to update parameter estimations that maximize the projected probability. This is accomplished by using the probabilities discovered in the Expectation Step.
d. The settings provide no wriggle room whatsoever. (Another option is to terminate the procedure if the parameter's change is less than a threshold value.)

There are a lot of benefits and drawbacks associated with both techniques when it comes to modeling data using mixed models and utilizing the EM methodology to estimate the parameters of these models in order to identify clusters. The most significant potential drawbacks of the EM method include its potential for slowness, inapplicability to models with a large number of components, inefficiency when clusters contain only a limited number of data points, and difficulty to efficiently deal with data that is nearly linear. These

models have the advantage of being more generalizable than k-means and other similar clustering techniques. Using these methods, clusters of elliptical forms and sizes ranging from very tiny to extremely enormous may be identified (based on Gaussian distributions).

3.9.3 Neural Networks

b. Autoencoders

Autoencoders are a crucial component of unsupervised learning, and human shape identification in particular notably benefits from their use. Autoencoders are a kind of neural network that may be taught to create outputs that are an identical replica of their inputs. The encoder layer, which is located on the internal side, contains a description of the encoding of the input. This paradigm divides the network into an encoder function represented by h = f (x) and a decoder that creates a reconstruction represented by r = g (h) (Figure 3.9 shows an example of a typical architecture of an autoencoder). Therefore, autoencoders are not meant to learn the abilities required to recreate accurate data. In most instances, they are configured to simply deliver an approximation and to only recreate inputs that are comparable to the training data.

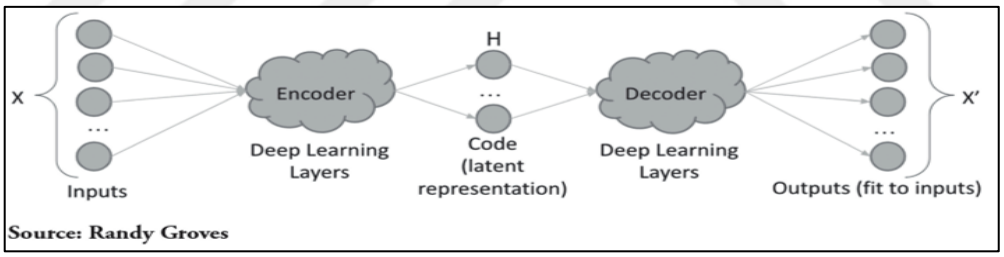


Figure 3.10: The General Structure of an Autoencoder [50].

There are many various classifications for autoencoders, including the following:

a. It is argued that an autoencoder is "incomplete" when the size of its code layer is smaller than the quantity of input data. The objective of this exercise is to determine the essential characteristics of the training data samples.
b. A orcomplete autoencoder, on the other hand, has a coding layer that is more comprehensive than the input.
c. An autoencoder is said to have sparse training criteria if, in addition to the reconstruction error, the layer code additionally includes the sparsity penalty (h).

$$L(x, g\ (f\ (x)))\ +\ \Omega\ (h) \qquad (3.2)$$

h = f (x) represents the input decoder, whereas g (h) represents the output decoder. They are often used to comprehend how another activity, such as categorization, operates.

a. One may use a Denoising Autoencoder to recover the original data from one that has been corrupted by noise. This is accomplished by minimizing the loss function L (x, g (f (x0)) where x0 has the same shape as x prior to being influenced by noise.
b. A Variational Autoencoder, or VAE, is a two-part autoencoder that learns to reduce the reconstruction error differential between encoded and decoded data and the original data. This is achieved by a comparison of the encoded and decoded data to the original data. We make a minor adjustment to the encoding-decoding technique to incorporate regularization into the latent space. Instead of storing each input as a single point in the latent space, we encode it as a distribution. This permits us to introduce some regularity into the latent space.
c. The information is shown as a distribution in latent space.
d. The distribution of the latent space is examined, and a point is selected at random.
e. Once the sampled point is decoded, the reconstruction error may be calculated.
f. The effort to postpone the network-wide rebuild was unsuccessful.

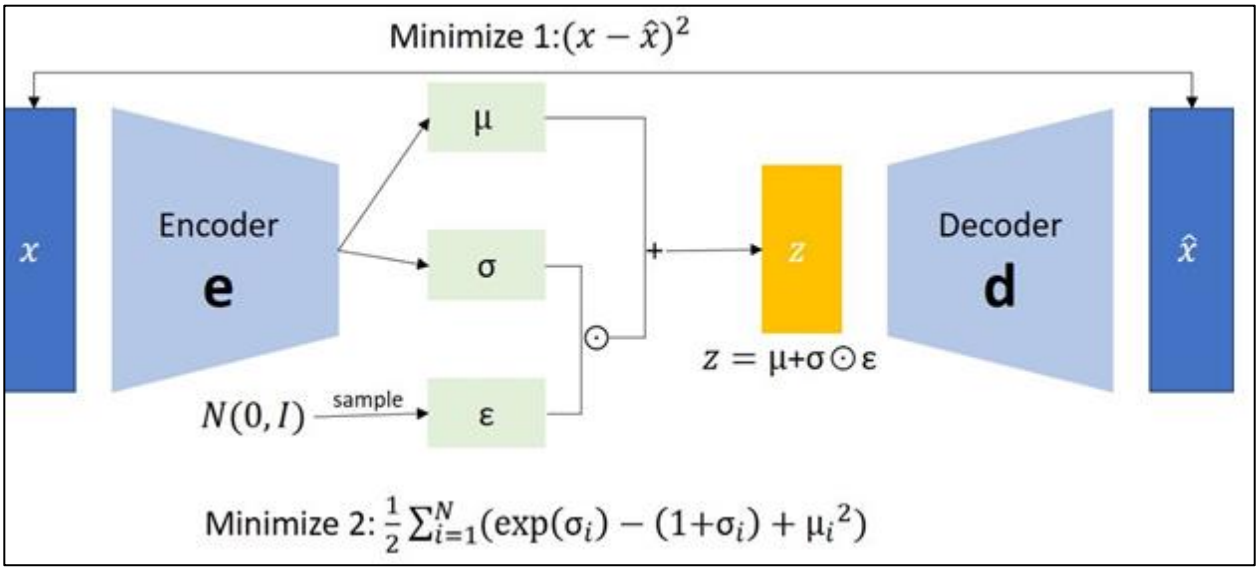


Figure 3.11: Differences Between a Normal Autoencoder and a VAE [27].

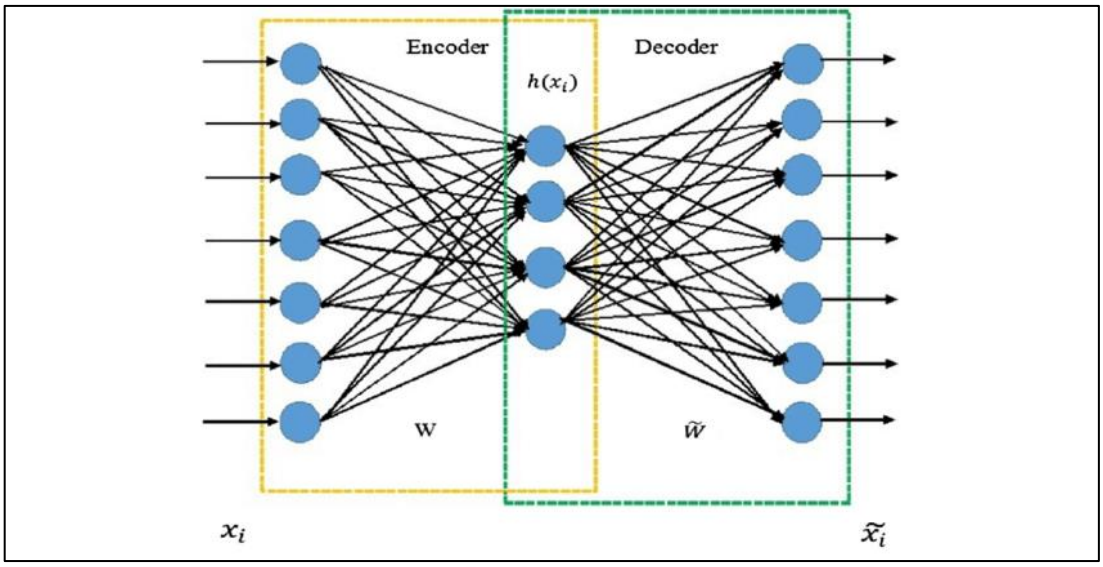


Figure 3.12: Architecture of an Autoencoder [29].

3.10 METHODOLOGIES AND TECHNIQUES USED

This study used a considerable proportion of the previously discussed approaches. Specifically, we used a synthetic neural network in combination with supervised learning. Due to the tremendous complexity of the data, we additionally used autoencoders to extract the most important characteristics from our data set. In practice, the autoencoder is taught to reconstruct accurately the input data it receives. Following the training of the encoder layers, a classifier was constructed by adding an extra hidden layer and an output-only neuron. Figure 3.15 illustrates the paradigm stated in the preceding paragraph.

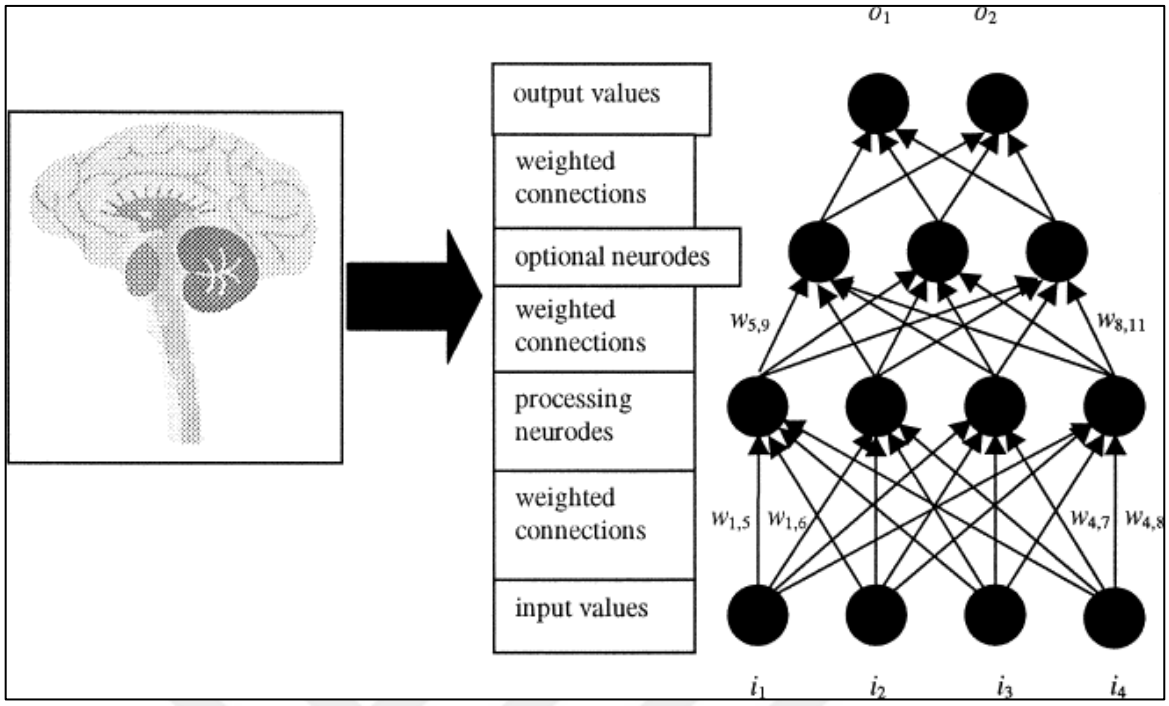


Figure 3.13: The Model Used in This Work [31].

A threshold determines the categorization due to the fact that the output value of a neuron is a real integer that might vary from 0 to 1. If the threshold is set to 0.5, any inputs that create outputs more than 0.5 are considered anomalous, but inputs that produce outputs less than 0.5 are considered typical. Figure 3.16 depicts how adjusting the threshold might provide the desired effect.

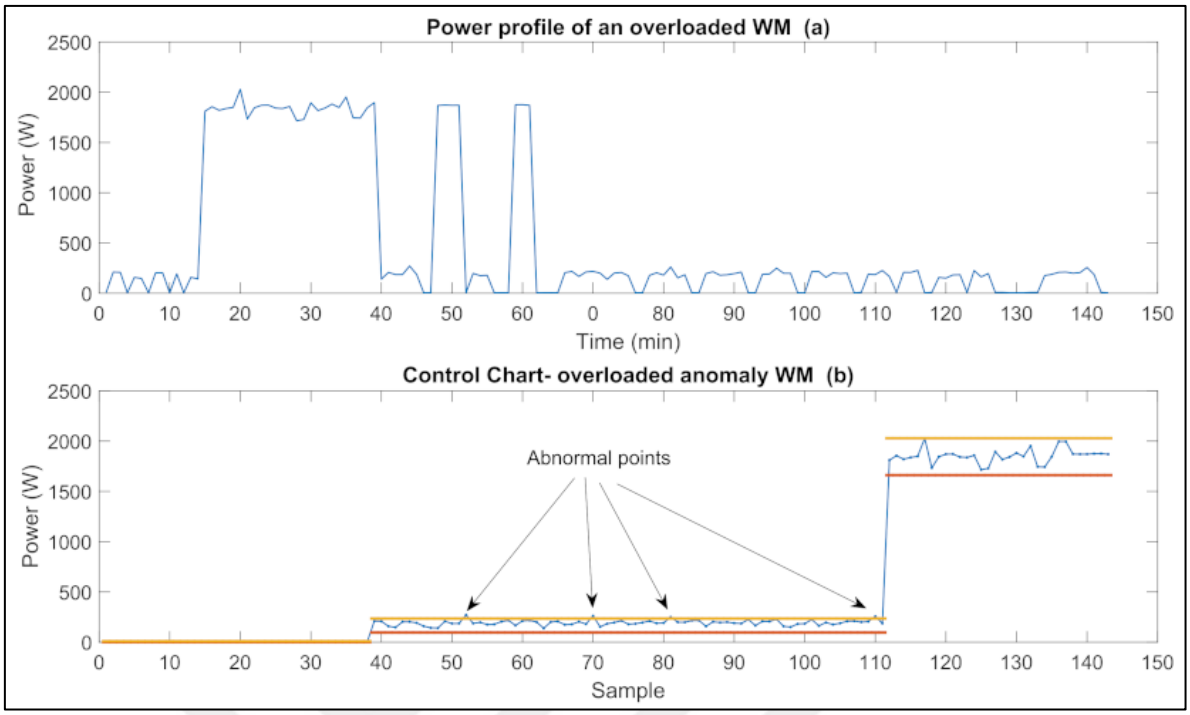


Figure 3.14: The Threshold Discriminates Anomalous Points and Normal Points [31].

The threshold's value is customizable, making it appropriate for a variety of applications. Eventually, modifications were made to the model in order to allow multi-label categorization in the completed job. The model is identical to the first scenario with the exception of the neurons responsible for output in the second scenario. There is a direct correlation between the number of recognized classes and the number of output neurons required for this activity. In the chapters that correspond to the different model settings, we will examine how these parameters were updated.

4. PROPOSED METHOD

The work aimed to assemble a public database, with inertial and video data (RGB and deep), collected from humans performing various daily activities, in different environments. In addition, it is also intended to study models to classify data from human activities, based on inertial data. With this, the OPPORTUNITY database will be explored, in addition to the basis created in this work, to compare different models according to their performances.

4.1 DATA COLLECT

The base built in this work (MATLAB) is composed of RGB video data, depth video and two wearable inertial units. The activities carried out by the humans are daily, carried out in a natural way, to facilitate the generalization of a model extracted from this base. Data were collected and stored in csv format, with the following format: for an activity X, a human Y and a recording Z, each file generated by each sensor is saved in a folder whose name indicates the sensor and the name of the file is constituted as actXseqYptZ.csv. Since each line of inertial files is constructed as {timestamp, sensor1.X, sensor1.Y, sensor1.Z, ...}, where timestamp indicates the instant of time in seconds since January 1, 1970 (UTC), used to synchronize the different sensors. For videos, the lines of each file are constructed by a timestamp and a frame (image collected at that instant).

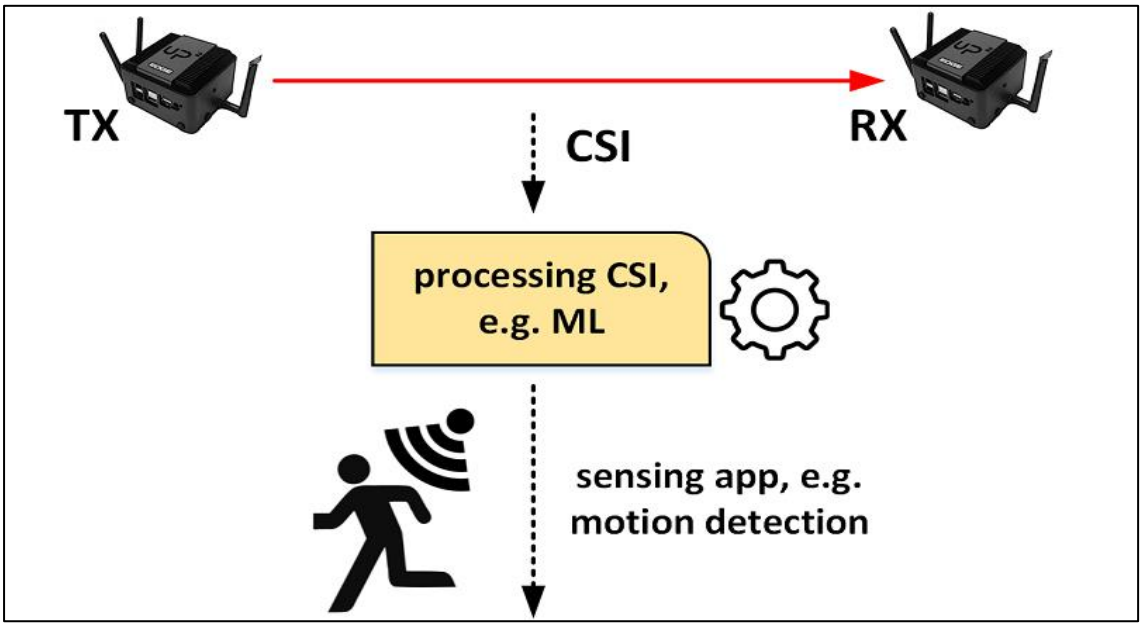


Figure 4.1: Activity Capture Device Scheme.

4.2 MATERIALS

To collect the videos, he used a Activity capture device ACD v1 (shown in Figure 4.1), which has an RGB and depth camera, and for inertial data, a Motorola X4 smartphone (collecting accelerometer, magnetometer and gyroscope data at a rate of approximately 120 Hz) and a MetaMotionR wearable sensor (MMR) were used.) (collecting accelerometer, magnetometer and gyroscope data, at a rate of approximately 100 Hz), both shown, respectively, in Figures 4.2.

The chosen inertial units have the advantage of collecting information in a non-invasive way, given that objects such as cell phones and watches are commonly carried in everyday life. In addition, Kinect v1 also captures data in a natural way, as it does not interfere with the user's movement. Therefore, with the materials used, it is possible to capture the movements performed by the humans naturally, with little interference in their movements.

4.3 METHODOLOGY

From the dataset we chose 6 humans participated, being 5 men and 1 woman with 22.5±4.72 years, chosen with only the minimum age of 18 years as a requirement, performing each task twice, for five seconds for each recording. The MMR was positioned on the corner

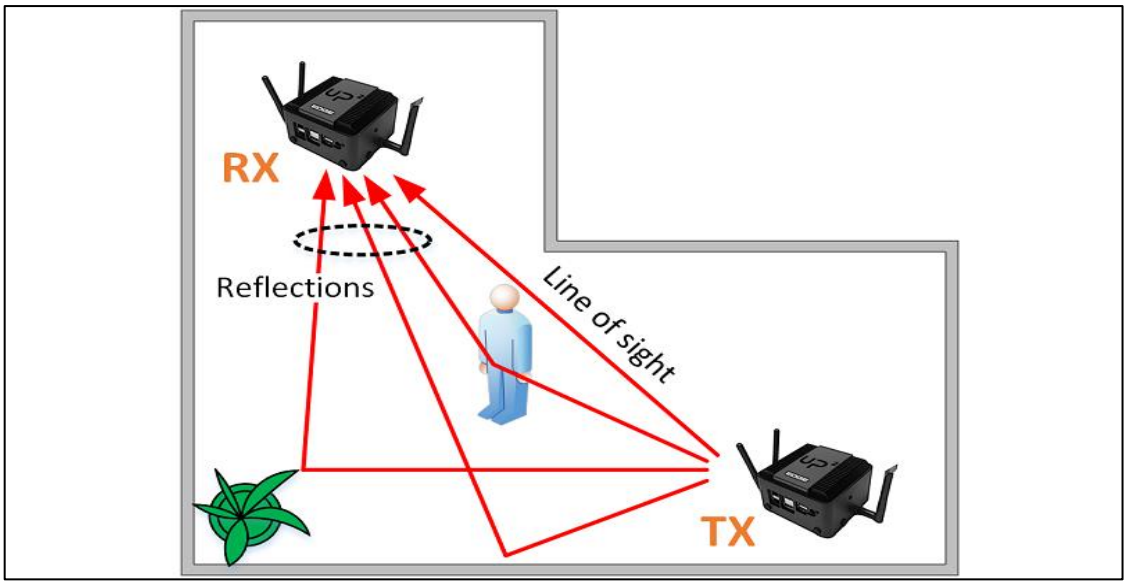


Figure 4.2: MMR Device Positioned at the Border.

human's dominant position and the smartphone in the human's right front pants pocket. Kinect v1 was positioned to capture movements from favorable angles, in general, filming

the front part of the humans' bodies. The positioning of the inertial sensors can be seen in Figure8. The recordings were carried out in four different environments, two kitchens and two living rooms, with different levels of lighting, and each human participated in the recording in a maximum of two environments. In total, the base added up to 10 minutes of data, with a varied number of tuples for each type of data collected (because of the frequency of collection of each sensor).

On this basis, the following daily tasks were recorded, where the numbering indicates the activity ID in the file:

a. To walk
b. run
c. use laptop
d. Eat
e. pick up objects
f. jump
g. crawl

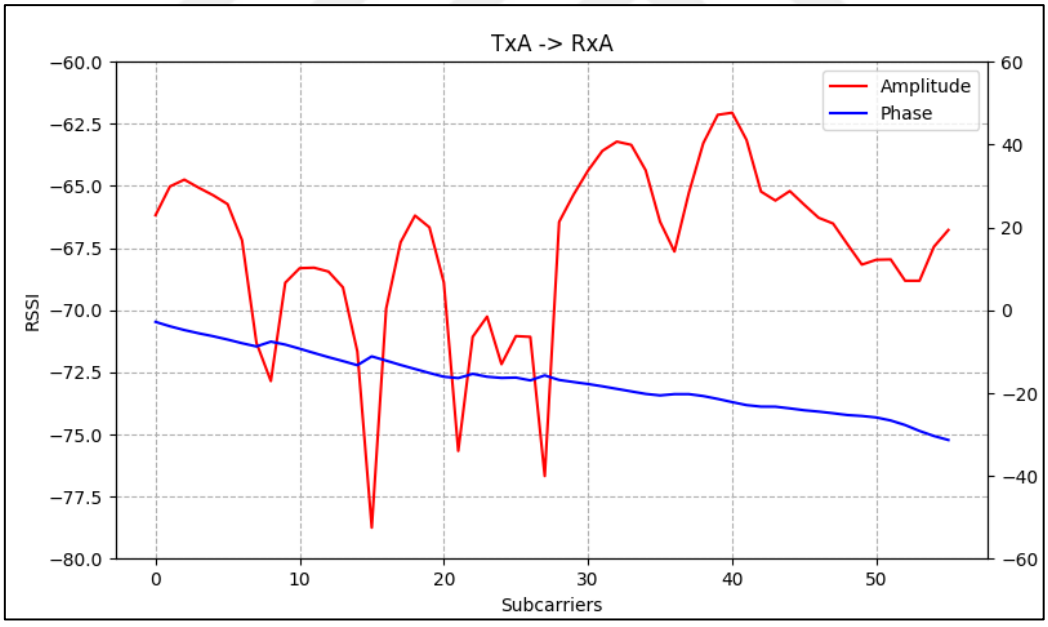


Figure 4.3: MMR Capture of Human Activity.

Source code 1: Pseudocode of the collection script.

a. start_connection_with_sensors()
b. collection_time = 0 // in seconds
c. start_time = now_time() 4 While collection_time < 5:

d. collect_data_from_sensors()
e. past_time = now_time() - start_time
f. collection_time = collection_time + elapsed_time
g. end while
h. finish_connection_with_sensors()
i. save_data_to_files()

For each task, the interpretation of the performance was left to the human's discretion, so that they could be performed as naturally as possible and not be biased. In addition

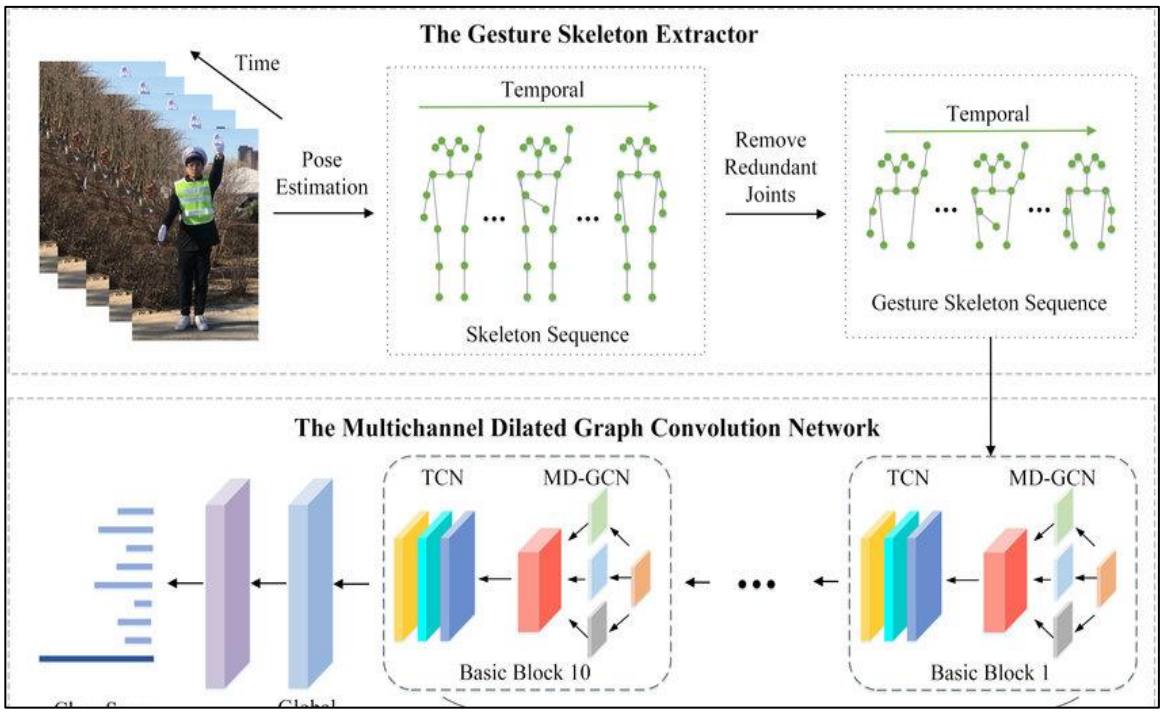


Figure 4.4: TCN-FCN Network.

In addition, we chose to collect for five seconds due to the range of the sensors in tasks that involved displacement, such as walking.

The script that starts collecting and synchronizing the video data acquisition was developed in MATLAB r2022a, with the help of the Machine learning library, to handle Kinect v1, with the MetaWear library, to handle the MMR and the communication with the smartphone was performed via Socket TCP. the source code1displays a pseudocode from the collection script.

4.4 BASE CLASSIFICATION

Recognition of human activity involves temporal data and therefore temporal dependence is an important factor in this task. Thus, models that can model temporal dependencies, such as RNN and TCN, are good starting points. In particular, TCN, which may have a longer memory and shorter training time Thus, in this work, different architectures, composed of TCN or RNN with LSTM, were used to classify the OPPORTUNITY and MATLAB bases, comparing the performance of each model in order to determine the best approach for HAR.

Thus, the Temporal Convolutional Network-Fully Convolutional Network (TCN-FCN) and ConvTCN architectures, proposed in in addition to the LSTM-FCN and ConvLSTM, proposed, respectively). In this way, the performance of the TCN and LSTM layers can be compared when modeling temporal dependencies. Furthermore, as it is intended to work with raw data, extracting relevant features can generate significant improvements in the results, thus, the CNN layers will be used to extract such features, which can be classified by TCN or LSTM. With such models, it will be possible to classify the OPPORTUNITY and

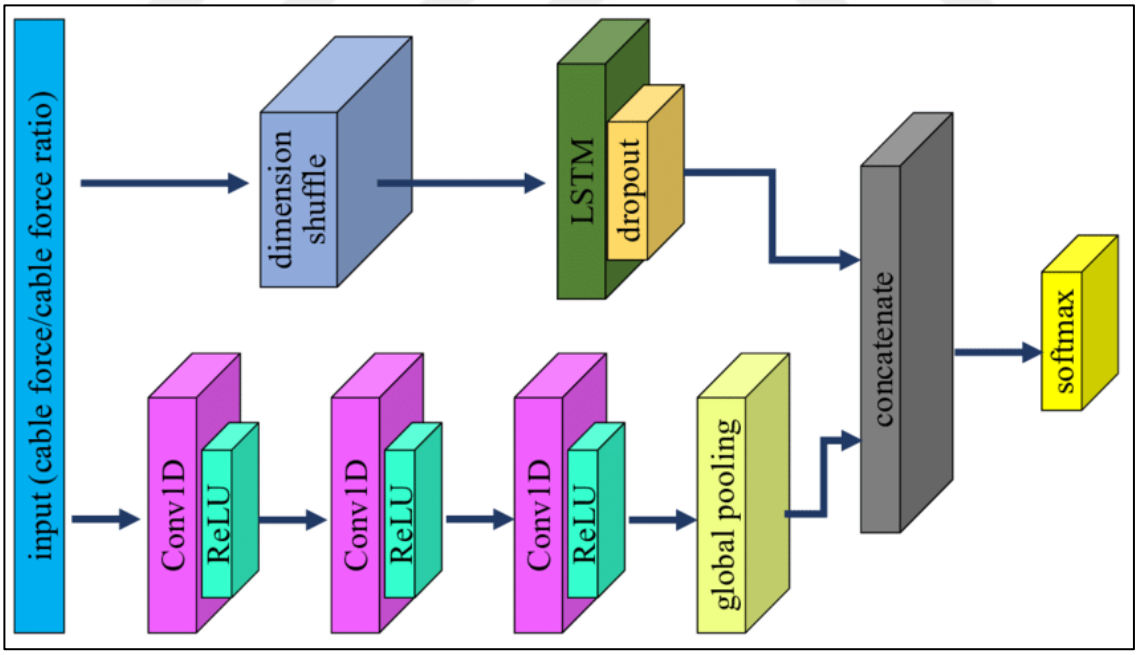


Figure 4.5: LSTM-FCN Network.

MATLAB, which will be preprocessed with L2 normalization and time windowing. All development was carried out in MATLAB r2022a, with the help of the ML Library library for building the models.

The TCN-FCN network is composed of two parallel blocks, the FCN and the TCN, where both receive the same input simultaneously and their outputs are concatenated, feeding a SoftMax layer. The FCN block has three chained 1D convolutions, with filters of size 128, 256 and 128, respectively, where each convolution is followed by a batch normalization and a Rectified Linear Unit (ReLU) activation function. After such convolutional blocks, a Global Pooling is applied. In the TCN block, a Dimension Shuffle is applied to the input, which basically transposes the time dimension with the features dimension. Then there is a TCN layer, with filter size 128, dilations d= {1,two,4,8,16,32}and a kernel with size 2. After the TCN layer, a dropout of 80% is applied. The figure9displays a representation of this network. The LSTM-FCN network has the same structure as the TCN-FCN, with the only difference being to replace the TCN layer with an LSTM layer with 128 cells, as shown in Figure10. The parameters for such architectures were chosen based on the literature.

On the other hand, the ConvTCN network is constructed by a 1D convolution at the input, followed by a Max Pooling and a ReLU function, preceding another convolutional layer constituted by the same components, both convolutions have size filters

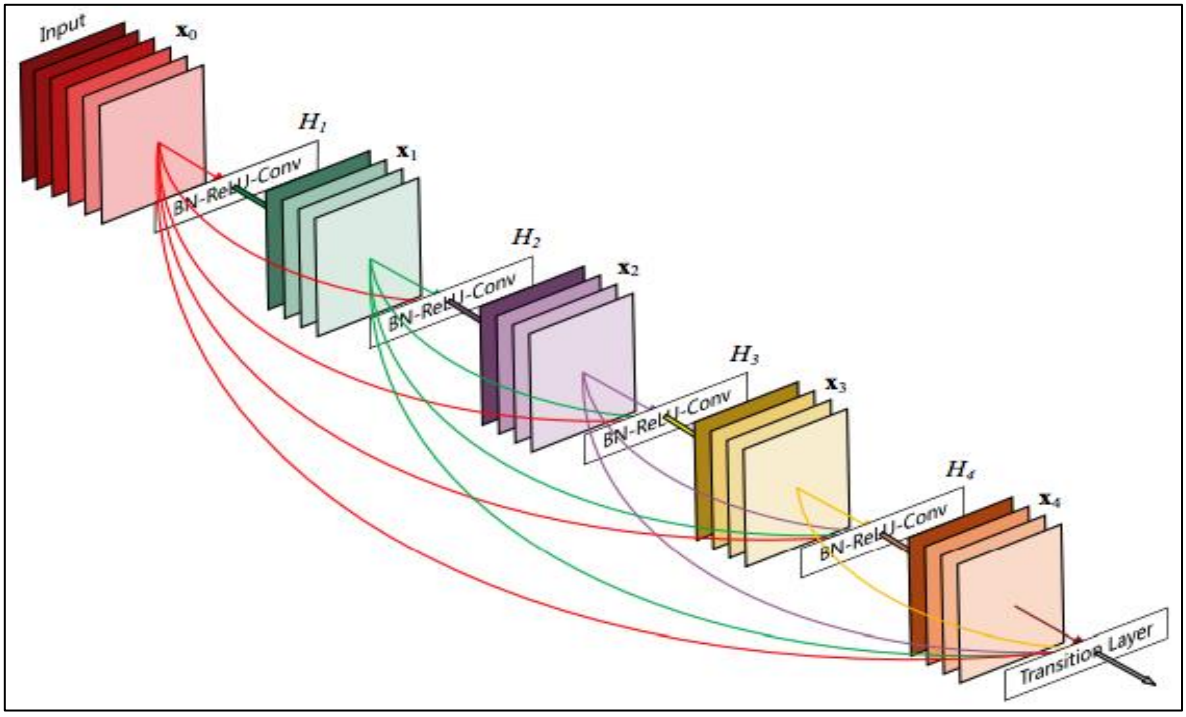


Figure 4.6: ConvTCN Network.

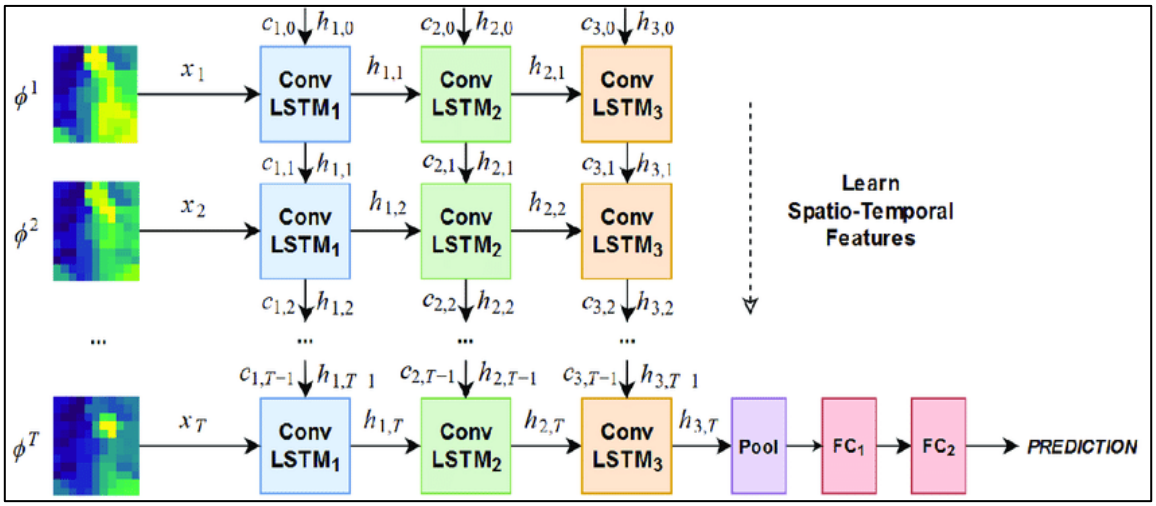


Figure 4.7: ConvLSTM Network.

64 and kernel size 2. After the convolutions, there is a TCN layer, with the same parameters as the TCN layer of the TCN-FCN network. ConvLSTM also has the same convolution structure, differentiating itself by exchanging the TCN layer for an LSTM layer with 128 cells. The ConvTCN network can be seen in Figure11, as well as ConvLSTM in Figure12. In the same way, the parameters of the literature were used for the construction of the networks.

With such models, the OPPORTUNITY database was classified, using 250 training epochs and validating the approach with 10-Fold Cross Validation, leaving 80% of the data for training and 20% for testing. In addition, these architectures were also used to classify the MATLAB base, training each model for 50 epochs, due to the smaller volume of data compared to OPPORTUNITY, and validating with 6-Fold Cross Validation (remembering that the base was built with 6 humans) and Leave-One-Out, due to the low number of samples available. Thus, in MATLAB, each model was trained with data from 5 humans and validated with data from the remaining human, varying, in each Fold, the human used for validation. For both validations, the accuracy and mean F1-Score between the Folds were measured.

5. SIMULATION AND RESULTS

5.1 SYSTEM OUTLINE

The work proposed a public multimodal database, with data from RGB and depth video and from inertial sensors, collected in daily tasks performed by humans. In addition, the OPPORTUNITY database, and the created database, called MATLAB, were classified with four different architectures based on the literature, comparing the performance of TCN and LSTM layers when classifying temporal data.

5.2 PERFORMED ACTIVITIES

To solve the problem, inertial and video data were collected from 6 humans, who performed 10 activities for five seconds each, with each activity recorded twice. While these humans wore an MMR bracelet on the wrist of the dominant corner and a Motorola X4 cell phone, positioned in the right front pocket of the user's thigh. Additionally, RGB video and depth data were collected using a Kinect v1, focusing, in general, on the front part of the human's body, in order to capture the greatest amount of movement information. In total, 10 minutes of data were collected.

In the classification, neural network architectures proposed in the literature were used: TCNFCN, LSTM-FCN, ConvTCN and ConvLSTM. With such models, it was possible to carry out training and validation in the OPPORTUNITY and MATLAB databases, focusing, in the second case, only on inertial data. Furthermore, with such networks, it is possible to compare the performance of the TCN layers with the LSTM, verifying the suitability of each approach for the task of classifying human activity.

5.3 RESULTS

In the classification of the OPPORTUNITY base, the results shown in the Table were obtained1, where the TCN-FCN network surpassed the others in accuracy and F1-Score, but with an F1-Score very similar to that of the LSTM-FCN network. Showing that, for this case, TCN improved the model, but not so significantly, given that both obtained very similar performances. Among the ConvLSTM and ConvTCN networks, the ConvLSTM network achieved the best results.

Table 5.1: Results of Accuracy and F1_Score on the Database.

Architecture	Accuracy (%)	F1 Score (%)
LSTM-FCN	86.11	78.09
TCN-FCN	89.15	83.16
ConvLSTM	91.19	94.27
ConvTCN	95.12	93.29

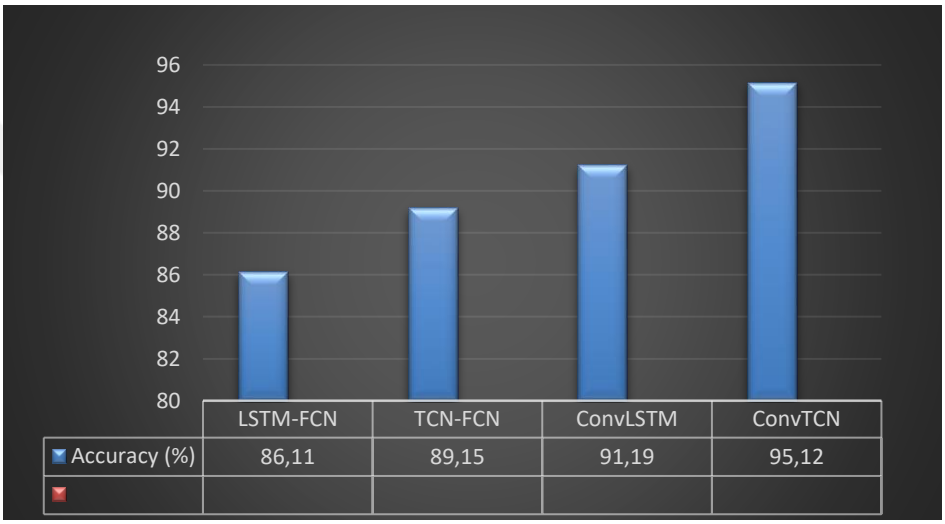


Figure 5.1: Results of Accuracy and F1_Score on the Database.

Opportunity database, with the best shown in bold.

However, the accuracy of both was also close. Thus, it is noted that TCN networks are also capable of modeling temporal dependencies, obtaining similar or even superior results in relation to LSTM.

However, in the MATLAB database, the results were presented in the Table two, where the classification by sensors used was separated. In this way, the models were validated using only MMR or Smartphone data as input or using the combination of both sensors. Thus, the TCN-FCN network surpassed the others when considering only the MMR, and the ConvTCN achieved the second best result, with values similar to those of the LSTM-FCN and the network

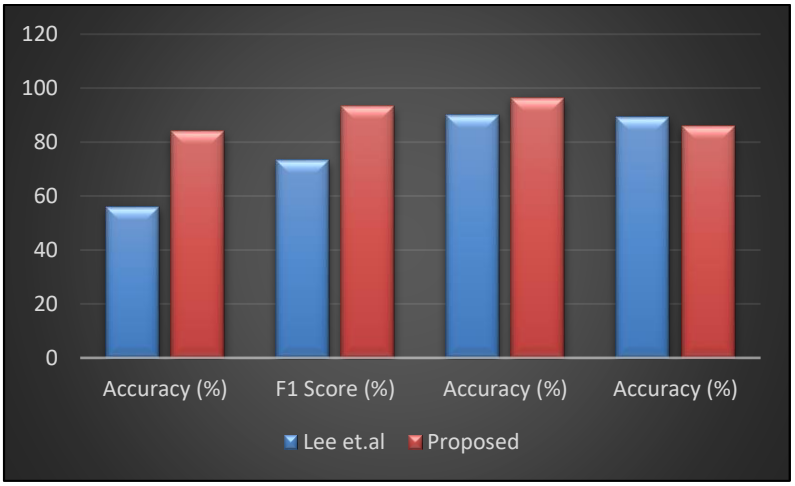
ConvLSTM obtained the lowest values. However, using only the Smartphone, the LSTMFCN network achieved the highest accuracy and F1-Score, with, again, ConvLSTM with the worst performance and LSTM-FCN and ConvTCN with similar values, in second

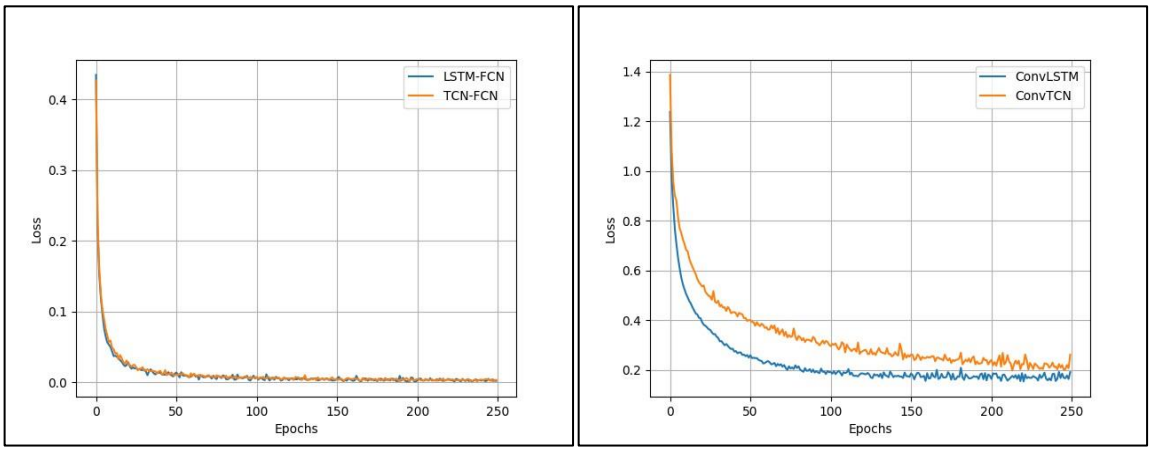
place. Finally, by combining the sensors, the TCN-FCN network obtained the best results, even surpassing the values achieved using the inertial units individually. Furthermore, the LSTM-FCN network obtained the second best performance, also showing an improvement in relation to the results of the same network without combining the sensors. On the other hand, the ConvLSTM architecture obtained similar results when using only the MMR, with the worst results, as well as the ConvTCN, which, despite surpassing the ConvLSTM, also did not show significant improvement with the combination of sensors.

Therefore, we observed that combining the inertial sensors can considerably improve the results, although some architectures benefit from adding information more intensively than others. Additionally, the TCN layer achieved better results than the LSTM in most approaches, showing the potential of such an approach. Thus, when analyzing temporal data, the use of TCN can be a good starting point, especially when considering it's possible shorter training time.

Table 5.2: Results for the MATLAB Database, with the Best Shown in Bold.

Architecture	Accuracy (%)	F1 Score (%)	Accuracy (%)	Accuracy (%)
Lee et.al	55.83	73.17	90.16	89.28
Proposed	83.88	93.21	96.41	85.94





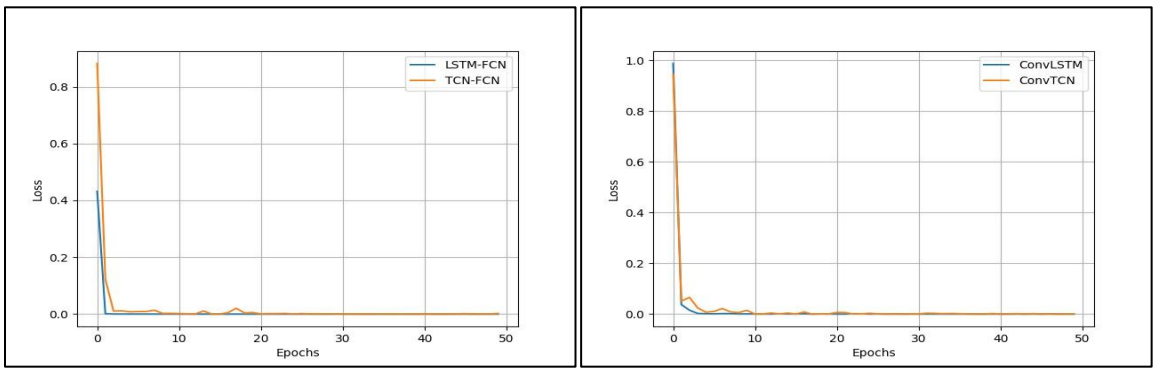
(A) TCN-FCN and LSTM-FCN (B) ConvTCN and ConvLSTM

Figure 5.2: Loss Per Season for the First Fold in Opportunity Base Training.

in the bases used, it can be seen in Figure13ththat, in OPPORTUNITY, the TCN-FCN and LSTM-FCN networks reached practically the same loss at the end of the training, with very similar convergence curves, however, in Figure13b, it is noted that ConvTCN took more epochs to converge, reaching a greater loss at the end of training, compared to ConvLSTM. In the MATLAB database, it is possible to see that, by the Figures14thand14b, that networks with TCN converged a little slower, with a loss similar to that of LSTM at the end of training.

5.4 DIFFICULTIES AND LIMITATIONS

Among the problems that arose in this work, we highlight the difficulty in recording and synchronizing data from four different sources, simultaneously. This problem was solved by starting the recording of all sensors at the same time, recording the instant of time in



(a) TCN-FCN and LSTM-FCN (B)ConvTCN and ConvLSTM

Figure 5.3: Loss Per Season for The First Fold in The HIVAD Base Training.

Figure 5.3 – Loss per season for the first Fold in the HIVAD base training, with the combined sensors.

seconds for each saved point and stopping recording simultaneously. Thus, the files had approximately the same duration and could be synchronized using the instant of time saved. However, the positioning of the sensors during the activities was also a difficulty encountered, resolved with a literature review and planning for non-invasive collection. Additionally, proposing activities for the built base was also a difficulty, solved based on activities commonly cited in articles in the bibliography. Finally, another problem encountered in the construction of the database was to gather humans for the collection, which could have been more interesting with more participants.

In addition, in the classification, there was not enough time to compose the models with multimodal data, also using the camera as input and validating its contribution to the performance of the model, as well as creating a model for each sensor, validating their individual and combined contribution. .

REFERENCES

[1] Raj, Aswathy B., Maneesha V. Ramesh, Raghavendra V. Kulkarni, and T. Hemalatha. "Security enhancement in wireless sensor networks using machine learning." In High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on, pp. 1264-1269. IEEE, 2012.

[2] Ramesh, Maneesha V. "Real-time wireless sensor network for landslide detection." In Sensor Technologies and Applications, 2009. SENSORCOMM'09. Third International Conference on, pp. 405-409. IEEE, 2009.

[3] Ramesh, Maneesha V., and Nirmala Vasudevan. "The deployment of deep-earth sensor probes for landslide detection." Landslides 9, no. 4 (2012): 457-474.

[4] Gajjar, Sachin, Nilav Choksi, Mohanchur Sarkar, and Kankar Dasgupta. "Comparative analysis of wireless sensor network motes." In Signal Processing and Integrated Networks (SPIN), 2014 International Conference on, pp. 426-431. IEEE, 2014.

[5] Sun, Zhi, Pu Wang, Mehmet C. Vuran, Mznah A. Al-Rodhaan, Abdullah M. Al-Dhelaan, and Ian F. Akyildiz. "BorderSense: Border patrol through advanced wireless sensor networks." Ad Hoc Networks 9, no. 3 (2011): 468-477.

[6] Suzuki, Takuya, Koji Yamamoto, Hiroaki Koyamashita, Tomotaka Wada, Kouichi Mutsuura, and Hiromi Okada. "Wave-type barrier coverage for border security in wireless sensor networks." In Computing and Convergence Technology (ICCCT), 2012 7th International Conference on, pp. 78-83. IEEE, 2012.

[7] Ekimov, Alexander, and James M. Sabatier. "Human motion characterization." In Human Light Vehicle and Tunnel Detection Workshop. 2009.

[8] Succi, George P., Daniel Clapp, Robert Gampert, and Gervasio Prado. "Footstep detection and tracking." In Aerospace/Defense Sensing, Simulation, and Controls, pp. 22-29. International Society for Optics and Photonics, 2001.

[9] Succi, George P., Gervasio Prado, Robert Gampert, Torstein K. Pedersen, and Hardave Dhaliwal. "Problems in seismic detection and tracking." In AeroSense 2000, pp. 165-173. International Society for Optics and Photonics, 2000.

[10] Nakadai, Kazuhiro, Yuta Fujii, and Shigeki Sugano. "Footstep detection and classification using distributed microphones." In Image Analysis for Multimedia

Interactive Services (WIAMIS), 2013 14th International Workshop on, pp. 1-4. IEEE, 2013.

[11] Ghiurcau, Marius Vasile, Corneliu Rusu, and Radu Ciprian Bilcu. "Wildlife intruder detection using sounds captured by acoustic sensors." In Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pp. 297-300. IEEE, 2010.

[12] King, R. A., and Mr TC Phipps. "Shannon, TESPAR and approximation strategies." Computers & Security 18, no. 5 (1999): 445-453.

[13] Sutin, Alexander, Barry Bunin, Alexander Sedunov, Nikolay Sedunov, Laurent Fillinger, Mikhail Tsionskiy, and Michael Bruno. "Stevens passive acoustic system for underwater surveillance." In Waterside Security Conference (WSS), 2010 International, pp. 1-6. IEEE, 2010.

[14] Jisha, R. C., Maneesha V. Ramesh, and G. S. Lekshmi. "Intruder tracking using wireless sensor network." In Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on, pp. 1-5. IEEE, 2010.

[15] Jin, Xin, Soumalya Sarkar, Asok Ray, Shalabh Gupta, and Thyagaraju Damarla. "Target detection and classification using seismic and PIR sensors." IEEE Sensors Journal 12, no. 6 (2012): 1709-1718.

[16] Rajasekaran and V. Nagarajan, "Adaptive intelligent hybrid MAC protocol for wireless sensor network," 2016 International Conference on Communication and Signal Processing (ICCSP), India, 2016, pp. 2284-2289.

[17] Harish, Palagati, R. Subhashini, and K. Priya. "Intruder detection by extracting semantic content from surveillance videos." In Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on, pp. 1-5. IEEE, 2014.

[18] D. Gorodnichy, S. Yanushkevich, and V. Shmerko. 2014. Automated border control: Problem formalization. In Proceedings of the IEEE Symposium on Computational Intelligence in Biometrics and Identity Management. 118–125.

[19] P. Grother, E. Tabassi, G. W. Quinn, and W. Salamon. 2009. IREX I Performance of iris Recognition Algorithms on Standard Images. Technical Report NISTIR 7629.

[20] G. Guo, S. Z. Li, and K. Chan. 2000. Face recognition by support vector machines. In Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition. 196–201.

[21] F. Hao, J. Daugman, and P. Zielinski. 2008. A fast search algorithm for a large fuzzy database. IEEE Transactions on Information Forensics and Security 3, 2 (2008), 203–212.

[22] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. 2005. Face recognition using Laplacianfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 3 (2005), 328–340.

[23] L. Hong, Y. Wan, and A. Jain. 1998. Fingerprint image enhancement: Algorithm and performance evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 8 (1998), 777–789.

[24] T. Huang, Z. Xiong, and Z. Zhang. 2011. Face recognition applications. In Handbook of Face Recognition, Z. S. Li and A. K. Jain (Eds.). Springer, 617–638.

[25] T. Bourlai, A. Ross, and A. K. Jain. 2011. Restoring degraded face images: A case study in matching faxed, printed, and scanned photos. IEEE Transactions on Information Forensics and Security 6, 2 (2011), 371–384.

[26] K. W. Bowyer, K. Hollingsworth, and P. J. Flynn. 2008. Image understanding for iris biometrics: A survey. Computer Vision and Image Understanding 110, 2 (2008), 281–307.

[27] K. W. Bowyer, K. Hollingsworth, and P. J. Flynn. 2013. A survey of iris biometrics research: 2008-2010. In Handbook of Iris Recognition, M. J. Burge and K. W. Bowyer (Eds.). Springer, 15–54.

[28] M. Brauckmann and M. Werner. 2006. Proceedings of NIST Biometric Quality Workshop. Technical Report. NIST

[29] van Kasteren, T.; Krose, B. Bayesian activity recognition in residence for elders. In Proceedings of the 2007 3rd IET International Conference on Intelligent Environments, Ulm, Germany, 24–25 September 2007; pp. 209–212.

[30] Cook, D.J. Learning setting-generalized activity models for smart spaces. *IEEE Intell. Syst.* 2010, *2010*, 1. [CrossRef]

[31] Sedky, M.; Howard, C.; Alshammari, T.; Alshammari, N. Evaluating machine learning techniques for activity classification in smart home environments. *Int. J. Inf. Syst. Comput. Sci.* 2018, *12*, 48–54.

[32] Chinellato, E.; Hogg, D.C.; Cohn, A.G. Feature space analysis for human activity recognition in smart environments. In Proceedings of the 2016 12th International Conference on Intelligent Environments (IE), London, UK, 14–16 September 2016; pp. 194–197.

[33] Cook, D.J.; Krishnan, N.C.; Rashidi, P. Activity discovery and activity recognition: A new partnership. *IEEE Trans. Cybern.* 2013, *43*, 820–828. [CrossRef] [PubMed]

[34] Yala, N.; Fergani, B.; Fleury, A. Feature extraction for human activity recognition on streaming data. In Proceedings of the 2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA), Madrid, Spain, 2–4 September 2015; pp. 1–6.

[35] Aminikhanghahi, S.; Cook, D.J. Enhancing activity recognition using CPD-based activity segmentation. *Pervasive Mob. Comput.* 2019, *53*, 75–89. [CrossRef]

[36] Pouyanfar, S.; Sadiq, S.; Yan, Y.; Tian, H.; Tao, Y.; Reyes, M.P.; Shyu, M.L.; Chen, S.C.; Iyengar, S. A survey on deep learning: Algorithms, techniques, and applications. *ACM Comput. Surv. (CSUR)* 2018, *51*, 1–36. [CrossRef]

[37] Fang, H.; He, L.; Si, H.; Liu, P.; Xie, X. Human activity recognition based on feature selection in smart home using back-propagation algorithm. *ISA Trans.* **2014**, *53*, 1629–1638. [CrossRef] [PubMed]

[38] Irvine, N.; Nugent, C.; Zhang, S.; Wang, H.; Ng, W.W. Neural network ensembles for sensor-based human activity recognition within smart environments. *Sensors* **2020**, *20*, 216. [CrossRef]

[39] Tan, T.H.; Gochoo, M.; Huang, S.C.; Liu, Y.H.; Liu, S.H.; Huang, Y.F. Multi-resident activity recognition in a smart home using RGB activity image and DCNN. *IEEE Sens. J.* 2018, *18*, 9718–9727. [CrossRef]

[40] Mohmed, G.; Lotfi, A.; Pourabdollah, A. Employing a deep convolutional neural network for human activity recognition based on binary ambient sensor data. In Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments, Corfu, Greece, 30 June–3 July 2020; pp. 1–7.

[41] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 2012, *25*, 1097–1105. [CrossRef]

[42] Singh, D.; Merdivan, E.; Hanke, S.; Kropf, J.; Geist, M.; Holzinger, A. Convolutional and recurrent neural networks for activity recognition in smart environment. In *Towards Integrative Machine Learning and Knowledge Extraction*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 194–205.

[43] Wang, A.; Chen, G.; Shang, C.; Zhang, M.; Liu, L. Human activity recognition in a smart home environment with stacked denoising autoencoders. In Proceedings of the International Conference on Web-Age Information Management, Nanchang, China, 3–5 June 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 29–40.

[44] van Kasteren, T.L.; Englebienne, G.; Kröse, B.J. Human activity recognition from wireless sensor network data: Benchmark and software. In *Activity Recognition in Pervasive Intelligent Environments*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 165–186.

[45] Ghods, A.; Cook, D.J. Activity2vec: Learning adl embeddings from sensor data with a sequence-to-sequence model. *arXiv* 2019, arXiv:1907.05597.

[46] Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2014 ; pp. 3104–3112.

[47] Bouchabou, D.; Nguyen, S.M.; Lohr, C.; Kanellos, I.; Leduc, B. Fully Convolutional Network Bootstrapped by Word Encoding and Embedding for Activity Recognition in Smart Homes. In Proceedings of the IJCAI 2020 Workshop on Deep Learning for Human Activity Recognition, Yokohama, Japan, 8 January 2021.

[48] Quigley, B.; Donnelly, M.; Moore, G.; Galway, L. A Comparative Analysis of Windowing Approaches in Dense Sensing Environments. *Proceedings* 2018, *2*, 1245. [CrossRef]

[49] van Kasteren, T.L.M. Activity Recognition for Health Monitoring Elderly Using Temporal Probabilistic Models. Ph.D. Thesis, Universiteit van Amsterdam, Amsterdam, The Netherlands, 2011.

[50] Medina-Quero, J.; Zhang, S.; Nugent, C.; Espinilla, M. Ensemble classifier of long short-term memory with fuzzy temporal windows on binary sensors for activity recognition. *Expert Syst. Appl.* 2018, *114*, 441–453. [CrossRef]

[51] Hamad, R.A.; Hidalgo, A.S.; Bouguelia, M.R.; Estevez, M.E.; Quero, J.M. Efficient activity recognition in smart homes using delayed fuzzy temporal windows on binary sensors. *IEEE J. Biomed. Health Inform.* 2019, *24*, 387–395. [CrossRef] [PubMed]

[52] Hamad, R.A.; Yang, L.; Woo, W.L.; Wei, B. Joint learning of temporal models to handle imbalanced data for human activity recognition. *Appl. Sci.* 2020, *10*, 5293. [CrossRef]

[53] Hamad, R.A.; Kimura, M.; Yang, L.; Woo, W.L.; Wei, B. Dilated causal convolution with multi-head self attention for sensor human activity recognition. *Neural Comput. Appl.* 2021, 1–18.

[54] Krishnan, N.C.; Cook, D.J. Activity recognition on streaming sensor data. *Pervasive Mob. Comput.* 2014, *10*, 138–154. [CrossRef] [PubMed]

[55] Al Machot, F.; Mayr, H.C.; Ranasinghe, S. A windowing approach for activity recognition in sensor data streams. In Proceedings of the 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Austria, 5–8 July 2016; pp. 951–953.

[56] Cook, D.J.; Schmitter-Edgecombe, M. Assessing the quality of activities in a smart environment. *Methods Inf. Med.* 2009, *48*, 480.

[57] Philipose, M.; Fishkin, K.P.; Perkowitz, M.; Patterson, D.J.; Fox, D.; Kautz, H.; Hahnel, D. Inferring activities from interactions with objects. *IEEE Pervasive Comput.* 2004, *3*, 50–57. [CrossRef]

[58] Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* 1994, *5*, 157–166. [CrossRef]

[59] Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* 1997, *9*, 1735–1780. [CrossRef]

[60] Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* 2014, arXiv:1406.1078.

[61] Singh, D.; Merdivan, E.; Psychoula, I.; Kropf, J.; Hanke, S.; Geist, M.; Holzinger, A. Human activity recognition using recurrent neural networks. In Proceedings of the

International Cross-Domain Conference for Machine Learning and Knowledge Extraction, Reggio, Italy, 29 August–1 September 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 267–274.

[62] Park, J.; Jang, K.; Yang, S.B. Deep neural networks for activity recognition with multi-sensor data in a smart home. In Proceedings of the 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), Singapore, 5–8 February 2018; pp. 155–160.

[63] Hong, X.; Nugent, C.; Mulvenna, M.; McClean, S.; Scotney, B.; Devlin, S. Evidential fusion of sensor data for activity recognition in smart homes. *Pervasive Mob. Comput.* 2009, *5*, 236–252. doi: 10.1016/j.pmcj.2008.05.002. [CrossRef]

[64] Asghari, P.; Soelimani, E.; Nazerfard, E. Online Human Activity Recognition Employing Hierarchical Hidden Markov Models.
arXiv 2019, arXiv:1903.04820.

[65] Devanne, M.; Papadakis, P.; Nguyen, S.M. Recognition of Activities of Daily Living via Hierarchical Long-Short Term Memory Networks. In Proceedings of the International Conference on Systems Man and Cybernetics, Bari, Italy, 6–9 October 2019; pp. 3318–3324. [CrossRef]

[66] Wang, L.; Liu, R. Human Activity Recognition Based on Wearable Sensor Using Hierarchical Deep LSTM Networks. *Circuits Syst. Signal Process.* 2020, *39*, 837–856. [CrossRef]

[67] Tayyub, J.; Hawasly, M.; Hogg, D.C.; Cohn, A.G. Learning Hierarchical Models of Complex Daily Activities from Annotated Videos. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1633–1641.

[68] Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations.
arXiv 2018, arXiv:1802.05365.

[69] Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding.
arXiv 2018, arXiv:1810.04805.

[70] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need.
arXiv 2017, arXiv:1706.03762.

[71] Safyan, M.; Qayyum, Z.U.; Sarwar, S.; García-Castro, R.; Ahmed, M. Ontology-driven semantic unified modelling for concurrent activity recognition (OSCAR). *Multimed. Tools Appl.* 2019, *78*, 2073–2104. [CrossRef]

[72] Li, X.; Zhang, Y.; Zhang, J.; Chen, S.; Marsic, I.; Farneth, R.A.; Burd, R.S. Concurrent activity recognition with multimodal CNN-LSTM structure. *arXiv* 2017, arXiv:1702.01638.

[73] Alhamoud, A.; Muradi, V.; Böhnstedt, D.; Steinmetz, R. Activity recognition in multi-user environments using techniques of multi-label classification. In Proceedings of the 6th International Conference on the Internet of Things, Stuttgart, Germany, 7–9 November 2016; pp. 15–23.

[74] Tran, S.N.; Zhang, Q.; Smallbon, V.; Karunanithi, M. Multi-resident activity monitoring in smart homes: A case study. In

Proceedings of the 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Athens, Greece, 19–23 March 2018; pp. 698–703.

[75] Natani, A.; Sharma, A.; Perumal, T. Sequential neural networks for multi-resident activity recognition in ambient sensing smart homes. *Appl. Intell.* 2021, *51*, 6014–6028. [CrossRef]

[76] Cook, D.J.; Crandall, A.S.; Thomas, B.L.; Krishnan, N.C. CASAS: A smart home in a box. *Computer* 2012, *46*, 62–69. [CrossRef] [PubMed]

[77] Tapia, E.M.; Intille, S.S.; Larson, K. Activity recognition in the home using simple and ubiquitous sensors. In Proceedings of the International Conference on Pervasive Computing, Linz and Vienna, Austria, 21–23 April 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 158–175.

[78] Ordóñez, F.; De Toledo, P.; Sanchis, A. Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. *Sensors* 2013, *13*, 5460–5477. [CrossRef] [PubMed]

[79] Wang, A.; Zhao, S.; Zheng, C.; Yang, J.; Chen, G.; Chang, C.Y. Activities of Daily Living Recognition With Binary Environment Sensors Using Deep Learning: A Comparative Study. *IEEE Sens. J.* 2020, *21*, 5423–5433. [CrossRef]

[80] Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.

[81] Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.