Smartphones as Computing Platforms: An all-in-one Mobile Application

by

TRIVEDI JAINEEL UDAYAN
202111050

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

July, 2023

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

TRIVEDI JAINEEL UDAYAN

Certificate

This is to certify that the thesis work entitled Smartphones as Computing Platforms: An all-in-one Mobile Application has been carried out by TRIVEDI JAINEEL UDAYAN for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.

Prof. Kalyan Sasidhar
Thesis Supervisor

Acknowledgments

First and foremost, I would like to express my gratitude to my research supervisor and mentor, Prof PS. Kalyan Sasidhar, for his invaluable guidance and dedicated involvement in every aspect of my work. His detailed explanations and humble attitude profoundly impacted me, and his focused vision steered me in the right direction. I am thankful for his instruction on research methodology and for presenting research work on various platforms. Additionally, I am grateful for the workspace he provided me and for making me a part of the 'UbiSense Lab' in lab-003. I also appreciate the workstation he gave me to complete my research.

I sincerely thank Prof Bhaskar Chaudhary for his unwavering support, encouragement, and insightful feedback. His constructive criticism has challenged me to think critically and creatively, enabling me to reach my full potential. His expertise, dedication, and commitment have shaped my intellectual growth and success.

I would also like to thank my lab colleague, Vyom Shah, for assisting me throughout the process and providing me with theoretical knowledge about the project. I would be remiss not to mention Ph.D. scholars Pramod Tripathi and Maitri Vaghela for providing me with guidance for the research.

I am thankful to my friends and the entire MTech-2021 class for helping me throughout the two-year road to finish my MTech program and research dissertation, which required more than just academic assistance. I want to express my gratitude to my friends Vaishnavi, Vinay Sheth, Harsh Savaliya, Dhara Mehta, Kavan Hajare, Rohit Mishra, and many more for supporting me during the entire journey.

Lastly, I would like to thank my parents and family for their constant belief in me and their unwavering support throughout the process.

Contents

Abstract

Advancements in technology have led to more intelligent computing systems, but server-based computing presents itself with network delay and high energy consumption leading to significant carbon footprints. A lag in computing the result and rendering it on a user's end device, such as a smartphone could lead to poor quality of experience or user experience. Edge computing and its variants such as fog and cloudlet computing, are paradigms for processing data close to the source. However, the computing platform still remains to be a high-end server.

The pipeline of mobile (smartphone-based) sensing involves sensing, data preprocessing, feature extraction, model training, and testing. The majority of work implemented only the sensing phase whereas the remaining were offloaded to a server. However, smartphones have rich computing power in terms of octa-core processors, faster clock speeds that are under-utilized. Few attempts have been made to utilize this computing power. Benchmark studies have shown that the performance of smartphone processors equals an Intel i3 processor. Most importantly, these devices consumed 30 times lesser energy than traditional servers for the same computing task.

In this work, we design, develop and implement a mobile application where the entire pipeline is implemented on the phone. The application takes the collected sensor data on which we implement machine learning algorithms to classify the physical activity of a user. Performance results show that the app consumes 350mAh amount of battery, with a CPU utilization of 13%. We also received an average user rating of 4.5/5 for user experience on the impact of positive interventions that our app automatically provides. This thesis provides a framework for implementing applications on smartphones eliminating the need for offloading.

List of Principal Symbols and Acronyms

ANN	Artificial Network Network
ASIC	Application-specific Integrated Circuits
CPU	Cenral Processing Unit
CSV	Comma Seprated Values
DNN	Dense Neural Network
DNN	Dense Neural Network
GPS	Global Positioning System
HPC	High-Performance Computing
IHD	ischemic heart disease
IOT	Internet Of Things
ML	Machine Learning
OS	Operating System
PC	Personal Computer
RAM	Random-access Memory
ReLu	Rectified Linear Unit
SoC	System of Chips
tflite	TensorFlow Lite

List of Tables

List of Figures

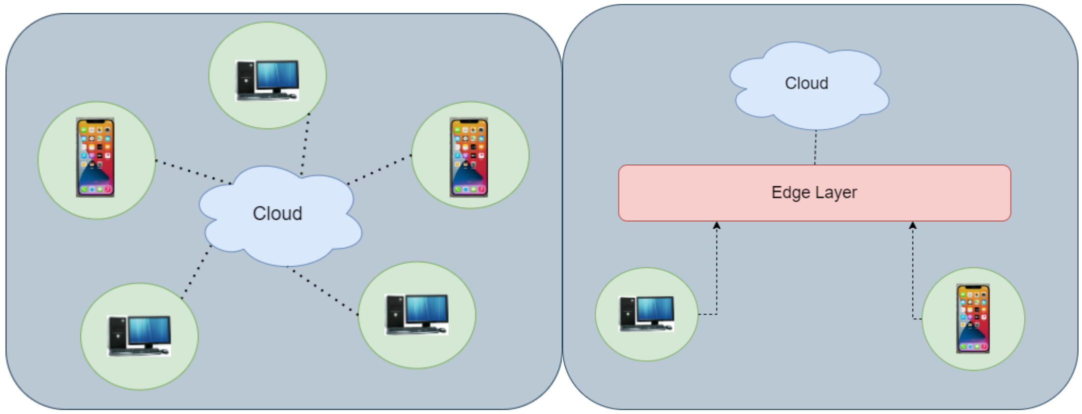
CHAPTER 1
Introduction

With the rapid development of the Internet of Things (IoT), the number of intelligent devices connected to the Internet is increasing. Cisco claims that there will be 29.3 billion devices by 2023, resulting in large-scale data [16]. Traditional cloud computing models incur bandwidth load, slow response speed, poor security, and poor privacy. To reduce the latency, edge and fog computing solutions have arisen. Edge computing facilitates the execution of computations at the network's edge - proximity to the user and proximity to the source of the data. It is lightweight at the network's edge for local, small-scale data storage and processing.

Edge computing [14] extends computational, network connectivity, and storage capabilities from the cloud to the network's edge. It permits business logic between the upstream data of the Internet of Things and the downstream data of the cloud service (IoT). Edge computing offers further advantages of agility, real-time processing, and autonomy in the area of Industrial IoT to provide value for intelligent manufacturing. In most IoT applications, sensor data is gathered from diverse sensors on a server for data analysis. For instance, the analysis could be as simple as calculating a region's average temperature or air quality. However, the computation may require good computing power considering smart city applications with high volume/variety of data from wearable sensors, deployed sensors, cameras, etc.

1.1 Overview of Edge Computing

Fig. 1.1(a) shows the simple architecture of Cloud Computing where the end devices are directly connected to the cloud. The raw data will be directly sent to the cloud. Fig. 1.1(b) shows the simple architecture of edge computing, where there is an extra edge layer between the end devices and the cloud, where the raw data



(a) (b)

Figure 1.1: (a) Cloud (b) Edge Layer

will be stored locally, and some computations can be done. Then the refined data will be sent to the cloud.

In edge computing, an edge serves as a gateway between the cloud and the data-generating devices. Any edge device, from smartphones to industrial equipment, can produce data that needs to be analyzed. As a mediator, the edge server gathers data from the devices and transmits it to the cloud for additional processing. This makes it possible to use less bandwidth and respond more quickly. The use of edge computing has the potential to change how we interact with the internet. Edge computing enables programs to be more responsive and effective by moving computation and data storage closer to the point of use.

Edge computing can drastically reduce latency by moving computation and storage closer to the data source [39]. This is especially crucial for real-time processing applications like self-driving cars and health applications. As most of the data can be processed at the edge, i.e., near the sensors, the bandwidth efficiency can also be improved. Another main advantage of using an edge server is that data will not be transferred to the cloud over the Internet. Instead, the raw data will be stored and processed locally near the sensors at an edge server. The use of edge computing has the potential to change how we interact with the internet. Edge computing can increase applications' responsiveness, efficiency, and security by moving computation and data storage closer to the point of use.

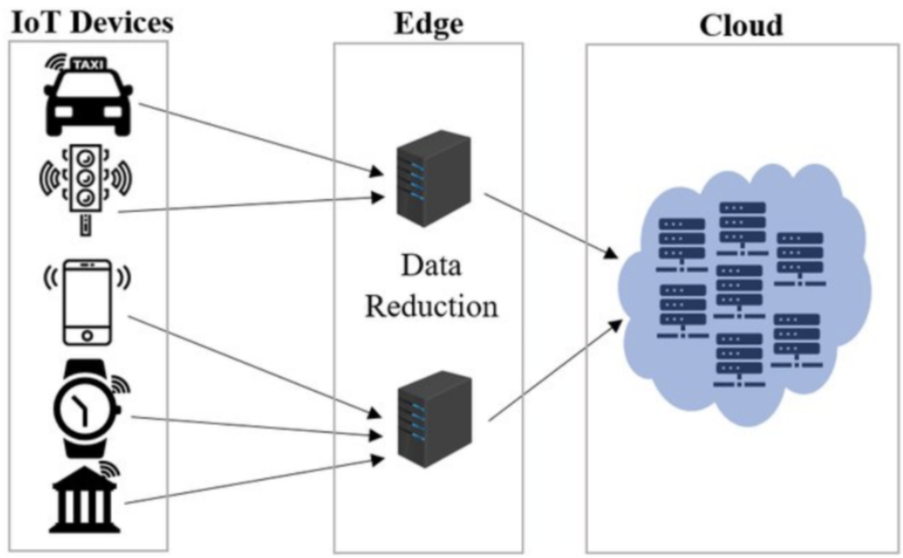


Figure 1.2: Edge Computing

Researchers are more focused on creating a local server environment where vital data can be stored or processed before sending it to the centralized data center or cloud storage repository. Fig. 1.2 illustrates the paradigm of edge computing, creating a server or network of servers which can act as a middleman between IOT devices and the cloud.

1.2 Evolution of Smartphones

Nowadays, smartphones have quickly supplanted feature phones as the most common design for mobile devices. By the end of 2023, smartphones are expected to make up 90 percent of mobile devices used worldwide, according to [49], the rapid technological development of mobile handsets, from communication devices with fixed features to general-purpose devices with increased computational power and network connectivity, is reflected in the popularity of smartphones.

We will discuss about the rise of smartphones in the last decade in detail in the next chapter.

1.3 State of the art of Computational power (Server)

Advancements in the computational power of servers have improved in the last decades[36]. Servers comprise high computing power, where they can handle more than one computation request of the clients/users. These servers are used in client-server, i.e., the client will request the specific task, and the server will do the job and revert it back to the client. Using this architecture, even if the client has no high computational setup, it can also see the specific task's results or output. They are equipped with advanced networking capabilities, which give them the power to handle a significant amount of data traffic and give smooth communication between the client and server. Servers also play an essential role in tasks that needs high computational power. They are made to efficiently compute complex functions like machine learning, data analytics, etc. This architecture is efficient for cloud computing systems [40]. Traditionally, a conventional PC had been used to set up a small server that can facilitate the requests of the clients.

1.4 Issues and Challenges in Server

Although servers are undoubtedly the most powerful computers available today, there are a few drawbacks to be aware of. Firstly they can be costly to purchase and maintain and also require high technical expertise for smooth and efficient operations. A typical AWS server costs you around $52.57 for a month[10, 5], even if you are doing high computational tasks. As the server act as a central point of the architecture, failure in the server can disrupt the entire system, yes, there are solutions for this, like duplicating the data at different nodes/servers. Still, this may not be feasible or cost-effective for a small server setup. Delay in sending and receiving messages is also a significant disadvantage due to many reasons like network error, excessive traffic on the server, etc. To overcome these, we can use the edge layer architecture that instead of entirely relying on the cloud servers. Running a server also requires high energy input, i.e., it consumes more power. For example, let us take a PC acting as a server with an Intel Core i3, which will consume approximately 50*(1+2.5)*6 = 1050 WH(Watt-Hours) if we keep it running for 6 hours. Here 2.5 indicates that every Watt consumed it requires 2.5 Watts for cooling and power distribution [**?**]. Also, due to the extensive use of servers which leads to more power consumption, it emits more carbon dioxide to the environment, which leaves a carbon footprint[48]. The authors of this paper did an empirical study of using smartphones as a server/computing device for specific tasks and proposed an architecture for the same.

1.5 Phone as a Computing Platform (PaaP)

As discussed in the above chapters, the recent advancements in hardware and software of smartphones can also be used as a computational device. Also, the performance of SoC or the processor clock speed has increased significantly compared with conventional PCs. We will discuss this in detail in the next chapter.

Former studies show that we can use a smartphone as a computing platform [48, 9, 44], where the researchers have shown with different empirical studies that smartphones are capable of complex computations too. Edge computing applications like health care, smart city, and smart farming in these different applications we can use smartphones as local computational devices.

1.6 Why Smartphones are advantageous ?

Smartphones are the most used computing device in the world. According to [18], India will have around 1 billion smartphones by 2026. Also, as discussed in the above sections, the computing power of smartphones has increased significantly, and it is still improving. However, the irony is that such high computing power is under-utilized. Apart from gaming applications on phones, not many apps utilize CPU power. For instance, according to[32], we found that the most commonly used applications were Whatsapp, Instagram, and Snapchat, which despite being used for an entire day, utilized only a fraction of the CPU. Gaming apps were observed to be used for a mere 2-3 hours/per week. We can use smartphones' computing power to utilize it for complex specific tasks efficiently.
In the last decade, many works have been done by researchers who are trying to use the computational power of a smartphone efficiently and optimally. The work has been done in many different domains like smart cities, smart farming, health, transport, etc. We will discuss some of the works in the next chapter.

We have talked about edge computing and the capabilities of smartphones that how the CPUs of the smartphones can be utilized efficiently. Also one of the issues of edge and cloud computing is data privacy. Recent development in hacking techniques has raised flags of researchers on the privacy associated with edge and cloud computing. Privacy breach in edge and cloud computing has become a significant concern because of the potential vulnerabilities in their distributed framework of them. As there may be more than one node where the data may be stored, it becomes easy for the breacher to breach the confidential data. From the above sections it is clear that smartphones are significantly powerful for running machine learning algorithms like image processing, data mining[22, 53] etc.

1.7 Motivation

Many applications like smart farming, data analysis, smart home, and smart city use IoT sensors and edge computing devices for computation as a dedicated local server. Although not too much computation is required, high-end computing devices are kept as edge devices. Also As discussed, it is clear that smartphones comprise significant computational resources that it can process, compute and analyze the data on the device itself. Traditionally smartphones were used for collecting sensor data and as a device to visualize the analytics computed at the local or cloud server. To bring the computation power of smartphones from idle to some extent, we can use smartphones for on-device computation, replacing the conventional server at an edge or cloud. This can optimize resource utilization, reduce the network bandwidth, reduce the traffic of the edge/cloud server, and be cost-effective as well. It can also enhance the user's data privacy by not sending the raw data to any local or cloud server.

1.8 Problem statement

This thesis mainly talks about doing computation on a single device to predict the physical behavior of the human. Inbuilt sensors of smartphones can help to find a person's motion as well as physical activity performed using the inbuilt sensors of mobile phones such as accelerometer ,gyroscope and light sensor. This thesis also involves machine learning models to predict behavior from the sensor data collected to predict the particular activity of person such as walking, running and sleeping. This whole task is performed on a single device to eliminate the need for extra infrastructure.

1.9 Organization Of Thesis

A flow diagram for the organization of the thesis is shown in Figure 1.3.

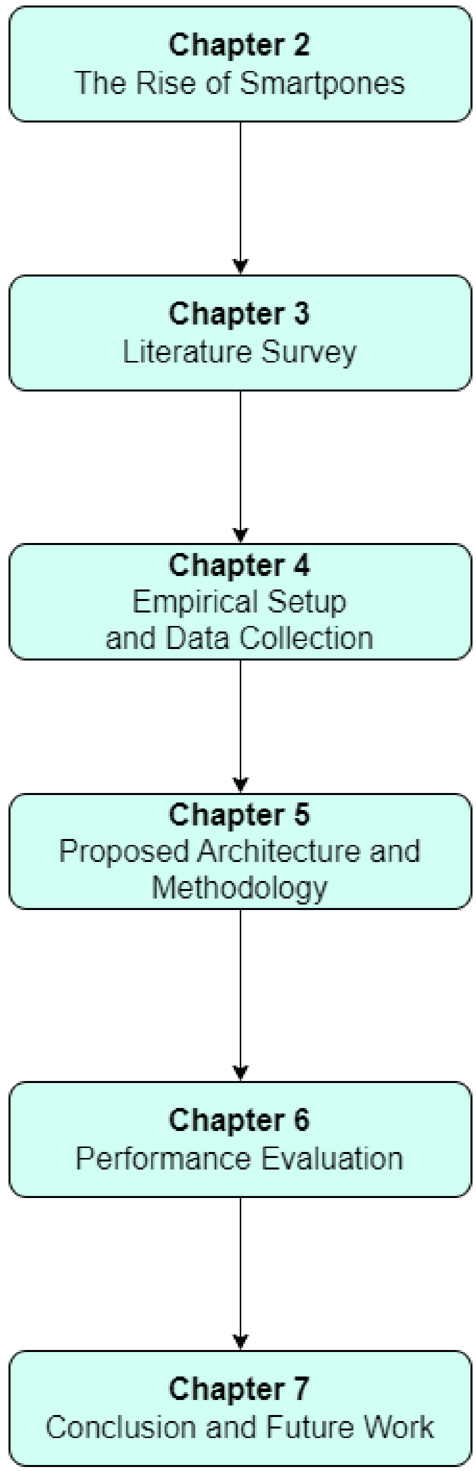


Figure 1.3: Organization of Thesis

Chapter 2 (Rise of smartphones): This chapter discusses about the evolution of smartphones, that how both hardware and software of the device has been improved in last few decades.

Chapter 3 (Literature Survey): This chapter discusses the existing work where smartphones are used as computational devices. And also some of the issues and challenges regarding the privacy aspect of the edge and cloud computing paradigm can be resolved with the use of smartphones as edge devices for specific workloads/tasks.

Chapter 4 (Proposed Architecture and Methodology): This chapter mainly discusses the system architecture of our methodology, which also includes the brief description of every different ML models used.

Chapter 5 (Experimental Setup and Data Collection): This chapter will discuss what type of hardware and software are used for the setup. Also, it includes a detailed idea of how and what data was collected.

Chapter 6 (Performance Evaluation): This chapter will give an overview of the behavior of our algorithm based on the three metrics CPU utilization, time, and scalability.

Chapter 7 (Conclusion and Future Work): This chapter concludes my work and discusses the future possibilities of work that could be done to improve the system.

CHAPTER 2
The Rise of Smartphones

Since their initial release, smartphones have undergone remarkable advancements in crucial hardware components, leading to rapid expansion in application capabilities. These improvements have also had significant implications for using a smartphone as a computational device, with the integration of Neural Processing Units (NPUs) in the latest smartphone models. These NPUs enable the implementation of machine learning techniques, such as deep learning, into standard phone functionalities. This integration has resulted in various enhancements, ranging from better computation power to efficient battery usage, which leads researchers to think that a smartphone can be used as a computing device. Along with this, smartphone manufacturers have also started leveraging dedicated deep-learning hardware to improve the overall utilization of smartphones. However, researchers have started working on using the capabilities of smartphones to revolutionize personalized health applications. In this area the smartphones are employed in many diverse diagnostic applications, including human activity recognition[41], human sleep analysis [32], stress analysis[52], and also heart attack risk prediction [35]. This chapter will discuss the main components that facilitate these technological advancements, which include smartphone processors/ System of Chip(SoC), Random-access Memory(RAM), and battery capacity.

2.1 Are smartphones nearing the processing speed capabilities of PCs?

In the past decade, there were seen improvements in the processing speed of smartphones, and think tanks were saying that smartphones may have a significant processing speed as compared to PCs. Comparing the processing power of both these computational devices was not simple. So many comparison frameworks were created.

To compare the processing speed of the smartphones with conventional PCs i.e. cross platform, cross architecture, it is necessary to test it with some standard benchmark tests. And they suggest the performance gap between smartphones and PCs is decreasing fast and much closer than thinked before. Geekbank5 [21] is a mainstream benchmark framework which was created by Primate Labs and it provides the comparison testing of device and its's CPU performance. It is a cross platform and cross-architecture, which makes it possible to compare the performances of different types of devices. For an instance lets discuss about how it compare the performance, Geekbench 5 uses Intel Core i3 as the base of different devices comparison and its test performance scored as 1000. In September 2021, phone manufacturer Apple releases the latest version of its A-series Bionic SoCs which they called as "A-15 Bionic" [7]. The preliminary Geekbench 5 results suggest that the processing speed of A15 approaches even more than that of the Intel's latest mobile CPUs.

The processing speed of smartphones is indeed approaching that of the most recent generation of PCs, as seen by the latest smartphones[8, 26], whether they run the iOS or Android operating systems. This is advantageous for using the smartphone for on-device processing and analytics rather than doing it elsewhere, in addition to using it for data collection.

2.2 Improvements in the hardware of smartphones(SoC)

In the past decade, there were seen improvements in the main smartphone's hardware i.e., SoC of the phone. An SoC is the main heart of the smartphone, where different chips are integrated. SoC typically has 3 major components, CPU/processor cores for computation, RAM for the local temporary storage of the computational data as well as dedicated input/output(I/O) interfaces for data accessibility by the other components [15].

Application-specific integrated circuits (ASICs) for accelerated video-encoding (camera record) and decoding (video playback) applications are now included in more modern SoCs. Additionally, SoCs provide wireless communication through

Bluetooth, Wi-Fi, and compatibility for older cellular networks, whereas future networks, such as fifth-generation "5G," are frequently initially provided on separate hardware[34]. Since the SoC in every smartphone device is the main component of the current application code, researchers are continuously improving and enhancing the three crucial aspects of SoC design that have brought performance levels up to par with those of laptop computers today.

2.3 Improvements in Processor Clock Speed

Clock speed, the number of processing cores, and the core microarchitecture are some of these aspects. [46]. The clock speed of a processor is the rate at which the CPU generates clock pulses to control all the functions including arithmetic, logic and data transfer for fetching and storing from RAM and external storage [45]. The advancements in the clock speed of the processors is showed in the fig 2.1, where the the processing speed has been increased significantly.

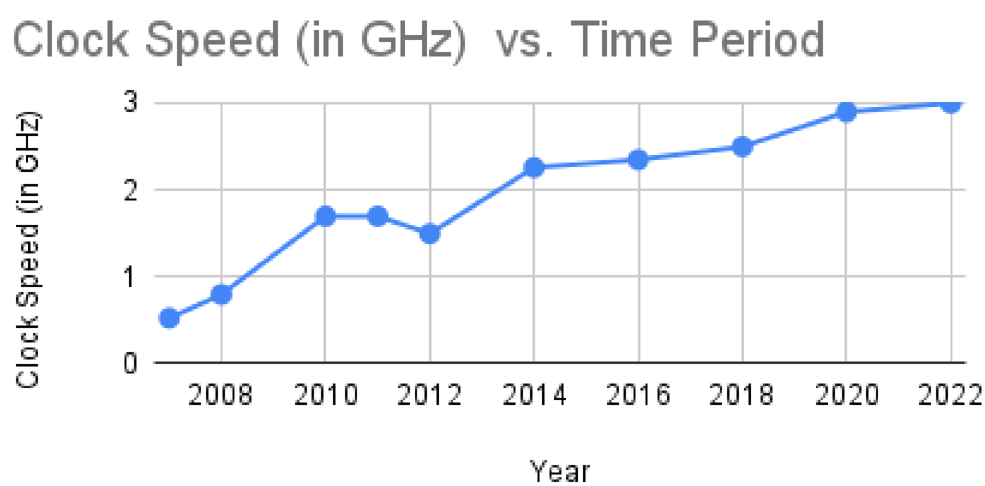


Figure 2.1: Evolution of Clock speed of mobiles

A CPU can execute software code more quickly and a mobile device appears to operate more quickly the higher the clock speed. In terms of smartphone SoCs, CPU clock speed is expressed in "Hertz".

2.3.1 Need of more number of cores

The more the number of cores the better computation power you can have. In the last decade the number of cores a processor of a smartphone has, has increased to 8, and research on multicore processors is still going on. A modern processor core consists of many millions of transistors.And electric signals are lept switching this transistors. The transistors' maximum switching frequency determines the maximum clock speed of a processor [4]. Hence, there is a limit to the clock speed of any processor.

To overcome this limitation of maximum clock speed, the solution was to increase the number of processor cores. A processor with more than one processor is enable to split a coded application to split the time consuming comples code into various threads. Each thread can be processed on a separate core, thus making the code parallely computing, will decrease the overall code execution time [51].

2.4 Improvements in Random-access Memory(RAM)

In smartphones, RAM is a high-speed volatile memory storage for temporarily storing the data related to the on-going processes or computations. More the RAM, larger and data intensive application can be executed on the phone.

The amount of RAM in smartphones has significantly increased over the last ten years. Devices quickly advanced from 256MB to 512MB of RAM, with limited memory capacity, reaching 1GB of RAM around 2010. Later, RAM capacities of 2GB and 3GB were the norm, enhancing multitasking efficiency and performance. 4GB of RAM became the industry standard by 2016–2017, greatly boosting smartphone performance. The RAM capacities of smartphones have increased in recent years, reaching 6GB, 8GB, or even 12GB [6]. The capacity to multitask and manage resource-intensive tasks has significantly increased thanks to these developments and software optimisations.

2.5 Chapter Summary

In this chapter, we have discussed the rise of smartphones in all the major aspects. The idea of this chapter was taken from [52], where the authors have explained the rise of smartphones.

CHAPTER 3
Literature Survey

Several works has already been done on using the power of smartphones efficiently.. As we have discussed earlier about the capabilities of smartphones and how efficiently we can use them.

The first question that arises is whether smartphones are capable of doing computationally intensive tasks. Researchers have empirically concluded that smartphones can do specific tasks and be utilized efficiently. In[44], the authors propose a system that was created to produce computational loads for desktop computers and cellphones, and it tracked how much time was used by each system under the same computational pressures. According to their study's findings, a desktop could do the assignment 10 to 30 times more quickly than a smartphone. On specific evaluations, however, they found that the PC used up to 2.26 times as much energy as the most energy-efficient phone. The research comes to the conclusion that using smartphones as computational engines in the cloud allows for more efficient use of the global computation infrastructure, although the total energy consequences of such an addition are not evident. Also, in[3], authors discuss the idea of using multi-core mobile devices for science study. On a quad-core mobile phone with a carefully constructed parallel programming environment, it suggests a real-time mobile lane departure warning system. The research also demonstrates ways to optimize CPU utilization to decrease system runtime. According to the authors, multi-core mobile phones are getting all the attention. Therefore, it would be advantageous to utilize complicated algorithms whenever possible because they typically produce more accurate results.

The past decade has seen an exponential rise in not only smartphone production but also in the advancement of hardware specifications. Smartphone CPUs, especially, are on par with desktop CPUs [37]. In the last five years, there have been smartphones with quad-core and octa-core processors. However, these multi-core processors may not be utilized to their full potential barring a few intensive mobile gaming or video streaming applications. For instance, in a recent study[50], we performed an experimental study through a custom-developed app [33] to quantify smartphone usage. We found that the average phone usage was 7.5 hours/per day. All phones were running with a quad-core CPU. The most commonly used applications were Whatsapp, Instagram, and Snapchat, which do not burden the CPU, even running for an entire day. Gaming apps were observed to be used for a mere 2-3 hours/per week. This study shows that the phones' CPU power is left underutilized. Garcia et al., [20] estimate the computation power of personal computers, smartphones, on-board car computers, WiFi routers, and other embedded computers across a city to be 19 $*10^{15}$ instructions per second. They propose the use of such computing power in place of high-end servers. The earliest attempts were made by authors in [13], who set up a smartphone cluster to execute an HPC application and found the performance nearly close to a high-end PC. Further, work in [9] proposed a distributed framework of smartphones and developed a scheduling algorithm to schedule and execute three compute-intensive tasks on smartphones. Recently, a similar study [43] evaluated the performance of Moto x4 and x5 in terms of their computation time and energy consumption while running matrix multiplication and other algorithms.

Traditionally smartphones were used only for collecting data and sending to the edge or the server for analysis and computational purposes, this concept was used in may IOT based applications like, Smart farming [12] is related to real-time data gathering, processing, and analysis, as well as automating the technologies used in farming. Also, this concept can improve the overall operations of farming and management. Moreover, it can also help the farmers in decision-making regarding farming procedures. By taking into account consumer transparency regarding the products they purchase, mobile applications that interact

The table 3.1 shows the bifurcation of the different research work basically into 3 different models, the first is Data Collection where the data is only collected and sent to the local or cloud server. Data Collection and execution where a pretrained model will be deployed in the phones rather than sending the raw data

Research	Citation	Data Collection	Data Collection and execution	Data Collection, model and execution
Smart Farming	[12]	✓		
Herd Management	[25]	✓		
MooMonitor+	[17]	✓		
Tech in agriculture	[31]	✓		
m-health	[33]	✓		
Human Stress Detection	[42]	✓		
Human Fall Detection	[23]	✓		
Cough Detection	[28]	✓		
Personality Prediction	[47]	✓		
Fitness Activity Recognition	[19]	✓		
FoodLoop	[1]		✓	
Heart attack Risk Prediction	[35]		✓	
Covid-19 detection	[29]		✓	
DeepRec	[24]		✓	
Data Analytics on phones	[38]		✓	
MindFull App	[52]			✓
Human Fall Detection	[11]			✓

Table 3.1: Classification of research papers featuring smartphones

to the server.with the agricultural environment can facilitate the management of cultivation and/or livestock, as well as take into account the entire value chain from farm to fork. The authors have done a similar kind of study in[25], where they discuss using a smartphone or tablet, a herd management system that enables cow ranchers to manage their dairy or beef herds.. MooMonitor+ [17] uses wearable collars to track the health and fertility of each individual cow, allowing farmers to keep an eye on their entire herd from their phones. Map your meal [2]is a smartphone app focused on agriculture that explores the world's food system with the goal of raising public awareness of global interconnectedness. Farmers and customers may use this smartphone app to scan products to determine how fair and environmentally friendly they are. Finally, FoodLoop[1] provides real-time deals on "nearly expired" products and connects the grocer inventory system and smart tagging to consumer-facing mobile applications to help the reduction of food waste.In this [31], the authors propose developing a system optimally watering agricultural crops based on a wireless sensor network. The system consists of three components: hardware, web application, and mobile application. Soil moisture sensors are used to monitor the field, connecting to the control box. The web-based application manipulates the details of crop data and field information. It applies data mining to analyze the data for predicting suitable temperature, humidity, and soil moisture for optimal future management of crop growth. The mobile application is mainly used to control crop watering through a smartphone.

Authors in [53] have surveyed how data mining and smartphones interact. It focuses on the improvement of smartphone performance, surveys smartphone usage patterns from earlier studies, and examines recent advances in locally implemented smartphone data mining. The authors also look into the various data applications that cellphones are utilized in, primarily how they utilize data given their fast-improving processing speed and expanding onboard sensor cache. Also, in this paper, the author has explained three models of utilizing smartphones, "collect", "collect and execute," and "collect, model, and execute." As discussed earlier, traditionally, smartphones were used only in the "Collect" model only in the current time, there are similar applications that are running on the "collect and execute" model. And from the survey they have conducted, they have concluded that the implementation of smartphones is majorly in the data collection only, with data sent to an online server and few works where actual on-device computation is implemented.

On-device computation has become a hot topic for researchers working in the field of data privacy, and edge computing as utilizing the power of a smartphone device can be helpful in many aspects, firstly, the dependency on the cloud reduces, and also, it makes the process hazel free. This architecture has also been used in many health-related sectors where one can analyze and compute the data on the device itself. In[35], The authors describe the layout of a smartphone app to address ischemic heart disease (IHD) (heart attack). Clinical data collected from patients with ischemic heart disease (IHD) have been coordinated using an Android-based mobile application. The clinical data from a number of individuals were examined and linked to risk variables such as smoking, diabetes, hypertension, dyslipidemia (abnormal cholesterol), family history, obesity, stress, and any current clinical side effects that may suggest a fundamental, unidentified IHD. Information mining innovation was used to mine the data, and a score was generated. The study seeks to inspire simple methods for dealing with IHD risk recognition and cautions the population to get themselves evaluated by a cardiologist to keep a safe distance from unexpected death. In the past few years, we all know how covid-19 has affected the human race. A leading cause behind the uncertain deaths and high death rates was not able to detect the diseases quickly, in[29] Authors propose a new system for exploiting the smartphone's built-in sensors to find the coronavirus COVID-19. Since most radiologists already use their smartphones for many daily tasks, the framework offers a low-cost option. The created framework, which is AI-enabled, examines the signal readings from the smartphone sensors to forecast both the grade of pneumonia severity and the outcome of the illness. The difficult COVID-19 clinical presentation may include fever, coughing, and excruciating headaches.

As discussed in the earlier chapter, due to the advancements in the hacking techniques, data privacy has became a very important task to be looked at. Researchers are working on the on device computation aspect where rather than using any edge or cloud layer, the computation will be done on the device itself[54]. In [38], the authors discusses that the applications that are asking to upload data to the cloud storage for analysis, rather than doing that, we can compute the analysis of the data on the device as many if the devices coming now a days comes with a high computational power. They recommend to focus on on-device analysis rather than on uploading to any edge or cloud. Authors in [24], the authors have proposed on-device deep learning framework namely DeepRec for privacy enhancement in mobile commerce. They have created a model to mine the data if the user for the online shopping recommendation system. Here instead of sending the raw data of the user to the cloud or edge, running the on-device model and sending the output to the cloud. Doin this the raw data or the confidential information of the user will not be compromised. The authors have also claimed that the DeepRec has achieved significant accuracy to the existing centralized models and can rescue the network overhead 10x to the existing model. [27], proposed a model of multimedia context identification for smartphones to enhance user data privacy.

In [52], the authors have discussed about the practical application of smartphones' computing power with the help of in-build sensors, to develop a pesonalized machine-learning framework for monitoring the mental health of the user. Initially they ask a set of quetionneries, by the answers of that it will train a personalized model for the user. The model will be trained on the smartphone itself.

The authors concluded by discussing that with the use of similar frameworks i.e. processing the data locally and executing machine learning algorithms too.

3.1 Chapter Summary

Summarizing this chapter, we learned about various methods of how smartphones can be used as computing devices. We discussed the capabilities of smartphones and some real-world applications where smartphones are doing on-device computation rather than sending the raw data of the user to any local or cloud server. I've discussed some of the well-known works related to smartphone computing and the benefits of on-device computation.

CHAPTER 4

Proposed Architecture and Methodology

We have implemented an application on the mobile phone which implements the algorithm for predicting the physical activity of a person with the help of machine learning models. We have chosen this application for predicting physical activity with the assumption that a smartphone is with the user most of the time, and using the data collected with different sensors of the smartphone and computing it locally can be a better and optimized approach.

4.1 System Architecture

Fig 4.1 shows the whole system architecture along with the working of the system. The sensors inbuilt into the phone were used to collect sensor data. Data collected includes data from accelerometers of X, Y, and Z axes to basically find the movement of a person(mobile), gyroscope data of X, Y, and Z axes, and light sensors to find whether a person is sleeping(idle) or not . Bluetooth and Proximity sensors to know the distance of the phone from the person using it or not. Data was stored in a CSV format which was used to predict the person's usual behavior using the machine learning model, which is directly integrated with the mobile application and categorized the behavior into three labels namely idle, non-idle, and par-idle. If a person's behavior is active/non-idle then the same data was passed to different ML models, which again predict, using the same type of ML model but trained on a different dataset, the physical activity as cycling, walking and running.

4.2 Methodology

We have developed an android application that collects data of inbuilt accelerometer X,Y, Z axis data and stores it in the CSV format. Android application fetch the CSV file from the location on which the Usage Tracker app has stored the CSV file

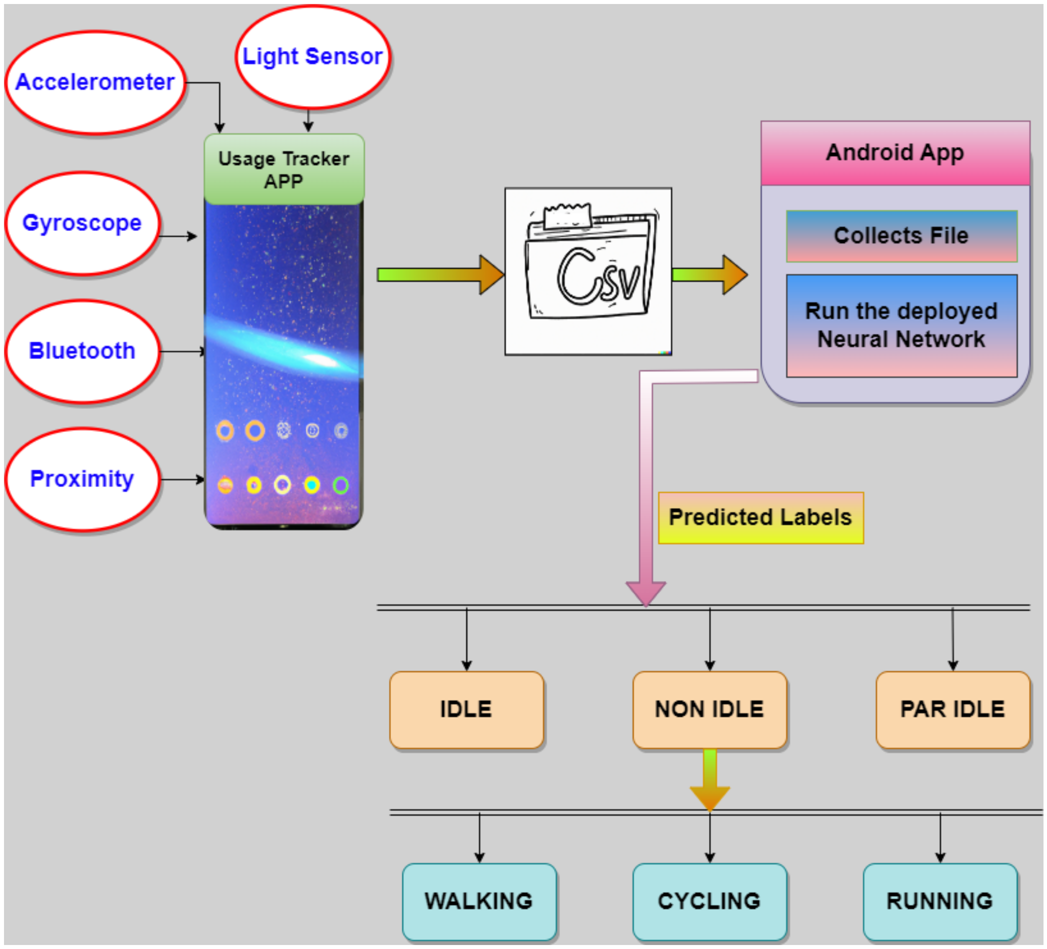


Figure 4.1: Architecture of proposed model

containing data of sensors. The application will linearly traverse the CSV file, and each line containing X, Y, and Z axis data of the accelerometer was given input to the pre-trained model, which classify the person's physical activity into categories like idle, non-idle, and partial idle. The main challenge is that we cannot directly integrate the ML model with the mobile application. First, it must be converted into the compatible version named tflite. It has to be put into the assets folder of the Android, so whenever anyone installs the application, pre-trained-model will also be installed as a part of the application and also fetch the CSV file to execute and predict the physical activity of humans.

Once the physical activity is predicted then, it was stored in the text file as per the student. From that text file, the application traverse over the file, and in the user interface, the end user was able to see the movement of dots where each dot represents each student. The GUI for our developed application is shown fig 4.2

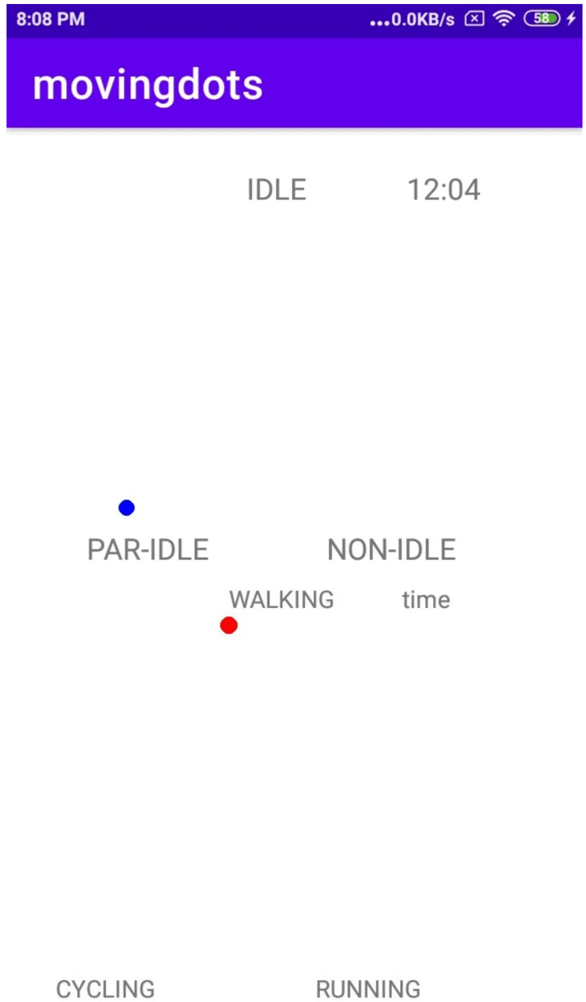


Figure 4.2: GUI of our application

4.2.1 Converting into Android Compatible model

Above all, models were developed using the Keras library of TensorFlow. Keras is the open-source Application Programming Interface that is developed using Python language, and also it is integrated with the TensorFlow library.

To make it compatible with the Android device model, it must be converted into the lighter version called the TensorFlow lite model. While converting from the tensorflow model to the tensorflow lite model, many types of optimizations is handled by the class called Converter itself, which makes the model run on resource-constrained. Different optimizations that take place are:-

- Quantization- This technique is used to reduce the precisions of the floating point weights and activations into integers, reducing the model's memory. Also, TensorFlow lite supports quantization, including post-training quantization and quantization-aware training.

- Weight Activation Fusion- this type of quantization works like pipelining method in which it can fuse one layer with a normalization layer, which does not require storing the intermediate results.
- Operator Kernel Optimization- The model can select the effective kernel implementation for each operation required to perform the instructions and also take advantage of using hardware acceleration of the platform.
- Pruning and Sparsity- Pruning is the method that removes or deletes the weights that are not needed in the neural network also even reduces the parameters by encoding it into integers which can increase the efficiency of the model as well as it can get fitted on the resource constraint device. Also, Tensorflow provides sparse tensors in which some weights will be set to zero, and only non-zero weights will be considered to predict the output.
- Model Slicing- Sometimes, it happens that tensorflow model has many outputs. Using this technique, we can reduce the model size by involving only necessary parts based on output and inputs.

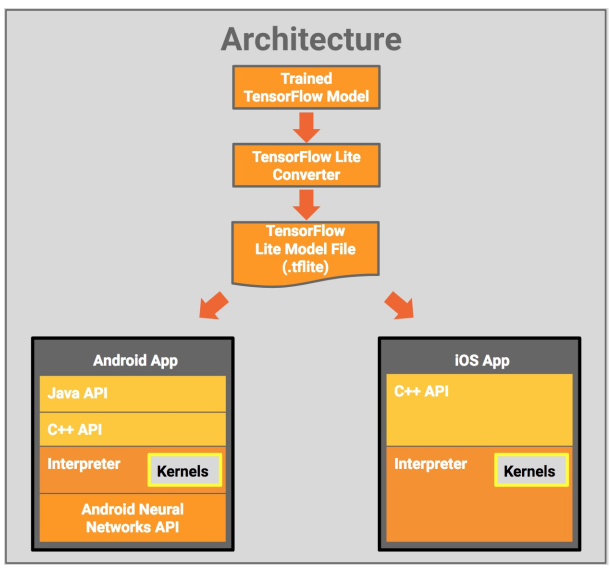


Figure 4.3: Conversion from tensorflow to tflite [30]

Fig 4.3 we can see a converter class is used to convert from tensorflow to tflite model and to again integrate with the android its own dependency has to be included $compile'org.tensorflow : tensorflow - lite : +'$ which in return we can directly call Interpreter class in java to predict with help of model for feeding input and getting outputs.

4.2.2 Artificial Neural Network

A neural network is termed from the human brain, and the working of that is similar to the human brain in which every neuron learns from the activity performed. So we try to implement the first model as ANN, whose architecture is shown in Fig 4.4. This model contains four layers: the input layer, which takes input as parameters, and two hidden layers, which have neurons and make the prediction and onre output layer which predicts the physical activity of human. Here the layers are stacked sequentially with hidden layer of three neurons with L2 regularization parameter of 0.001. The output layer has three neurons representing the three labels. So every neuron will have a probability value after predicting between 0 and 1. So out of all three neurons, whichever neurons have the higher value will be active and assigned the corresponding physical activity label. Again if the non-idle label gets activated by high probability same data of the three-axis accelerometer was passed to the next neural network model, which work same as the first one but will give the specific name of the physical activity performed. The activation function used for this model is Sigmoid Activation Function. The equation of it is,

$$S(x) = \frac{1}{1 + e^{-}x} \tag{4.1}$$

where e = 2.71.

The sigmoid function is also called as S-curve property function in which the value will be taken x, and the function will return the value between 0 and 1. The loss function used is Categorical Cross-entropy because we have three classes/labels of physical activity for the output, which uses one-hot encoding, due to which output label will be given in the encoded part so that for training also labels have to be encoded into to numbers and Optimistic Stochastic Gradient Decent is used as an optimization algorithm.

4.2.3 Deep Neural Network (DNN)

DNN is the neural network which is another variant of ANN consisting of multiple layers of neurons. In this neural network multiple hidden layers are been setup between input layers and output layers. As an input we have given the three features as the three axis of accelerometer sensor based on which physical activity of the human will be predicted. The total data samples we collected was around 500000 which were manually labelled around 80% of total data which is equivalent to the 400000 that were used to train data whereas 100000 samples

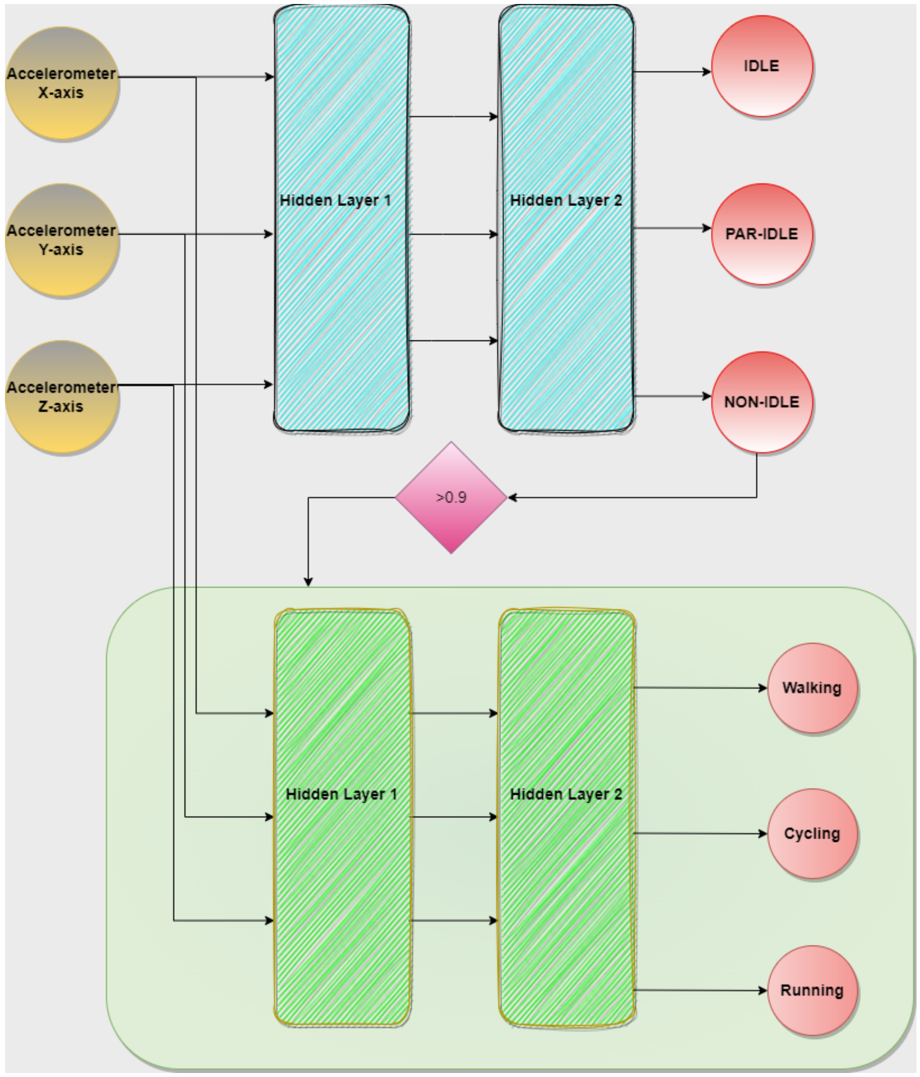


Figure 4.4: Architecture of ANN model

were kept for testing of the data. In this deep neural network, Rectified Linear Unit(ReLu) is used which is non-linear to the layers, and it is defined as

$$f(x) = max(0, x) \tag{4.2}$$

So in this, if the input value is >0, then f(x)= input value; otherwise, it will be 0. The saturation problem that occurs in the sigmoid hyperbolic function can be eliminated in the ReLu function by setting negative values to 0. Three layers are used. The first layer takes the ReLu activation and shape of 32 while the second layer, which is the hidden layer, again works on ReLu activation with the shape of 16 and at last shape of 1, which is kept as the output layer. by running for 100 epochs we gain an accuracy of 98.67%. Like in fig 4.4 as ANN is used, we have also used two DNN models in which accelerometer data is fed, and we get three labels as idle,non-idle, and par-idle again if it is non-idle, then again the second model predicts which exact physical activity is performed like running, walking and cycling.

4.2.4 Logistic Regression

Logistic regression is a widely used supervised statistical model are often used, especially for classification and predictive analytic tasks. It is intended primarily to deal with situations in which the dependent variable has a binary or categorical nature. In this case, we have trained a model to forecast a single outcome using three independent factors. The model incorporates three independent variables to predict a single output, which is typically expressed as the probability of an event occurring or a particular class being assigned. These independent variables are selected based on their potential influence on the outcome of interest and are often obtained through data collection or experimentation. The direction and size of each independent variable's influence on the dependent variable are indicated by the logistic regression model's estimated coefficients for each independent variable.

4.3 Chapter Summary

In this chapter, we discussed the system architecture as well as we have discussed about all three models used for the prediction of physical activity. We also tried using the random forest classification model but unfortunately, it was not supported by tflite module.

CHAPTER 5

Experimental Setup and Data Collection

This chapter discusses the setup needed for the experiment and the data collection techniques required for our proposed methods.

5.1 Setup

5.1.1 Software Setup

We developed an Android-based application using Android Studio which is compatible with almost all mobile phones having any Android version. The benefit of using this software is that we can easily observe the CPU utilization and the memory used by the application we developed. Apart from this, Android Studio also shows the energy consumption by our application only for those mobile phones whose API version is greater than 26. There is the inbuilt interface of Profiler, which easily shows the CPU utilization.

Apart from this also Google colab was used to train the Machine Learning model, which allocates RAM and CPU for single notebook running. Also it provides direct connectivity to google drive and has the facility to upload the files needed to run the program. It also attaches the GPU to the notebook if needed for the high computation power. Many python libraries come inbuilt installed in when we start a new notebook.

5.1.2 Smartphone Setup

Initially, we collected unused but active mobile phones from our family and friends. After gathering the mobile phones of different specifications, we rooted the phone to get access to the terminal emulator of Android OS, which is the flavor of Linux only. However, we were not able to get access to the kernel level. We have used around four different smartphones with diverse specifications regarding the number of cores and CPU speed. Varying RAM storage and different I/O read-write speed can lead to experimenting with various accuracies. Also, phones vary from the older Android version of 5 to the newer version of 10. Table 5.1 displays a detailed description of the smartphones used and their specifications.

Name of Mobiles	**Processor**	**Processor Speed(in GHz)**	**Number of cores**	**RAM (in GB)**	**Storage (in GB)**	**Read /Write Speed(in GHz)**	**Android Version**
Lenovo K5	ARM Cortex A53 415	1.3	4	2	12	0.677	5.1
Moto G7	Kryo 250 636	1.7	8	4	64	0.933	10
Redmi 5	Cortex A53	1.8	8	3	32	0.933	8.1
Redmi 5A	Cortex A53	1.4	4	3	32	0.667	8.1

Table 5.1: Mobile Specifications

5.2 Data Collection

Many sensors are already in-built in the mobile phone itself. So to collect data, we have implemented an Android application that only collects the data from the mobile phone's sensors.

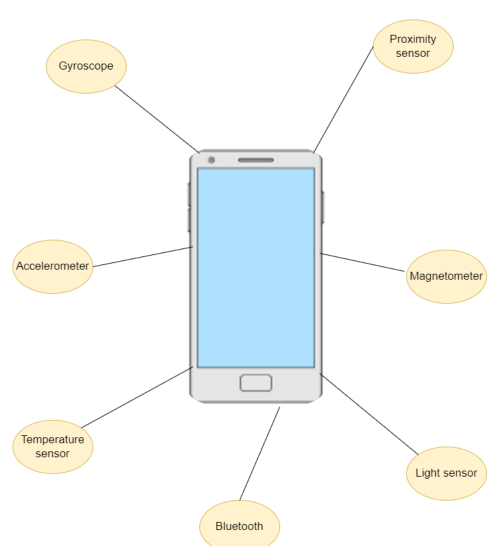


Figure 5.1: Sensors inbuilt of mobile

This mobile application was installed on the mobile phones of three students to collect data at the frequency of 0.2 seconds. Data was stored in CSV format, which contains data from sensors such as accelerometer, gyroscope, proximity, light sensor. Around 600000 samples were collected from each mobile phone at the end of one month. Out of data from all the sensors, only data from an accelerometer of all three axis, X,Y, and Z, were used for further computation. Out

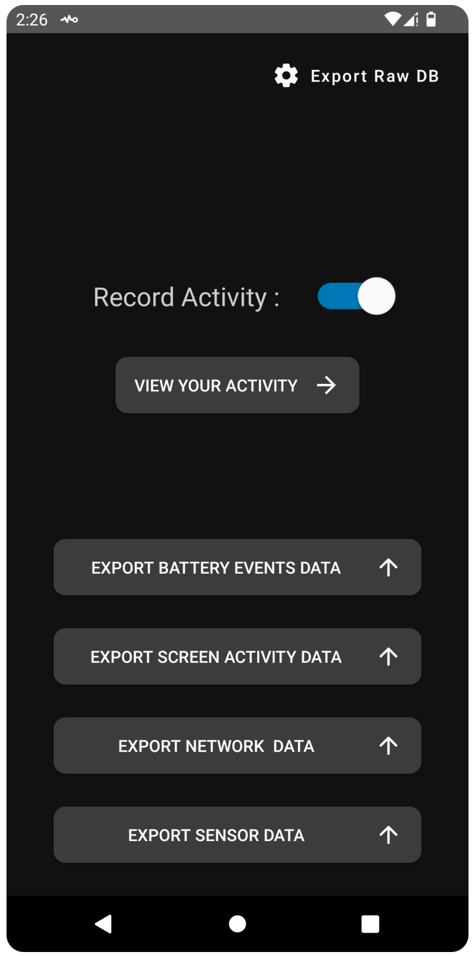


Figure 5.2: User Interface of Usage Tracker

of the total collected data 200000 data samples were manually verified with the students for the physical activity, After verification, data was manually labeled into three categories of physical activity:-

- Idle
- Partial Idle
- Non Idle

After labeling the data, the Machine Learning model was trained to predict the physical activity based on the accelerometer sensor data manually labelled

200000 data samples only from the phone itself. Remaining 400000 data samples were classified with the help of the machine learning models and also they were manually verified with the students .Fig 5.2 shows the user interface used to collect sensor data.

5.3 Chapter Summary

This chapter has discussed the overall setup of mobile phones' architecture and how the data was collected from the application. Also, the overall idea for the software used was discussed in this chapter. The next chapter will discuss the proposed methods applicable to the implementation of android application.

Chapter 6

Results

In, this chapter we will discuss the results of the classification accuracy, Computation time, and also the CPU utilization of the application.

6.1 Comparison with different models

To predict the person's different behavior or physical activity, different ML models were used and compared, so Fig 6.1 shows the accuracy of all three models. The prediction model used were ANN, DNN, and LR. Out of which DNN works with the highest accuracy. Here we have used Neural Networks, which has dense multi-layer architecture, which helped to classify the data more efficiently and accurately. I also created the classification function using the Random Forest algorithm, and it gave the highest accuracy, but unfortunately, it is not compatible with tflite module converter, hence I have not considered it for the analysis.

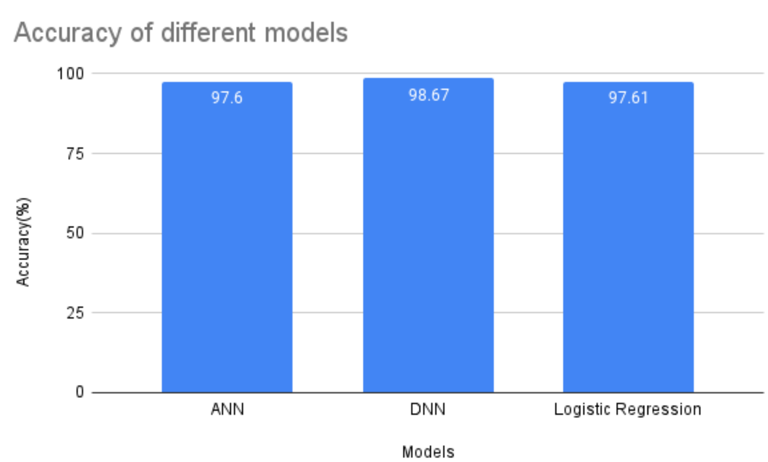


Figure 6.1: Accuracy of Different Models

Models	Accuracy(%)	Loss
Artificial Neural Network	97.60	0.136
Deep Neural Network	98.51	0.0164
Logistic Regression	97.61	0.135

Table 6.1: Accuracy and Loss function of different models

The fig6.2 is the confusion matrix which shows the overall summary or report of the model in a matrix form.

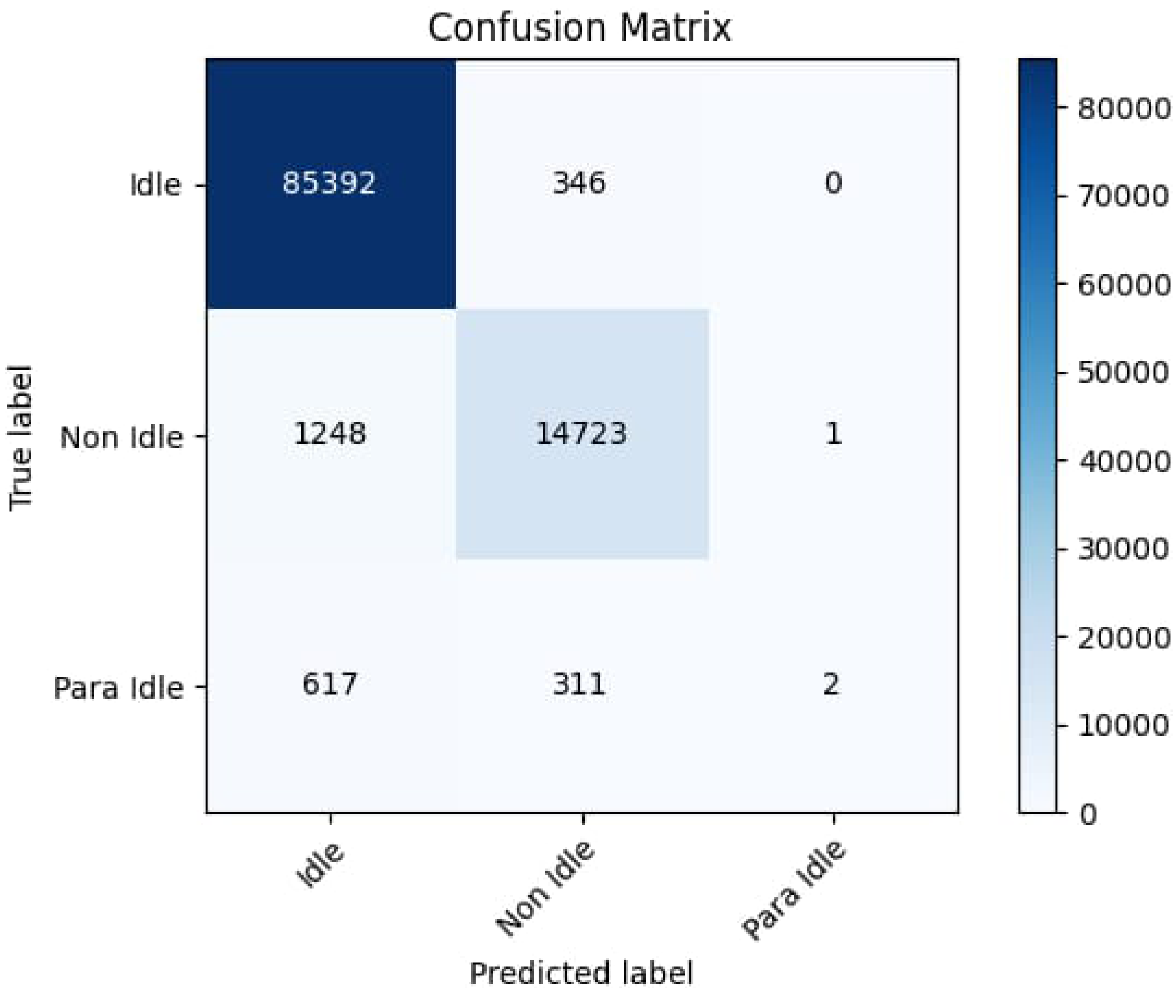


Figure 6.2: Confusion Matrix

6.2 CPU Utilization

One of the main metrics to know the usage of cores is the CPU utilization, and in Android, there is an interface called Profiler which helps us to know how much CPU is utilized for a particular application on the mobile device.

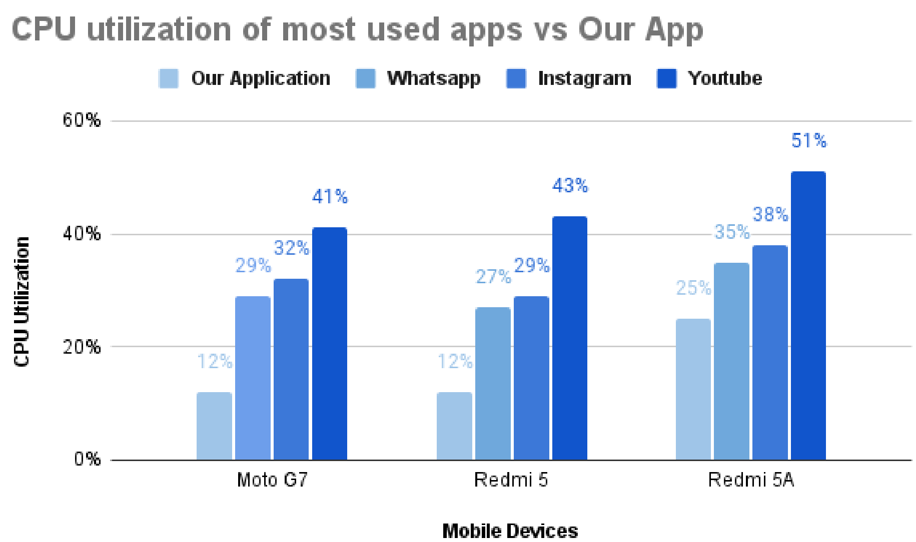


Figure 6.3: CPU usage

In Fig 6.3 we have tried to compare the CPU utilization of our Android application with the most commonly used apps [33] and we found out our app uses half times the CPU that is used by WhatsApp and Instagram and 1/4 times CPU used by Youtube. From 6.3 we can observe that redmi5 is using almost double the cpu utilization than other two mobiles. Actually the fact is all phones are creating single thread to run the algorithm and as redmi5 is quadcore so single core is used for single thread= 1/4=25% while others are octacores so it will be 1/8= 12.5%.

6.3 Computation Time

Apart from this, we have also implemented our application on different mobile phones covering octa-cores and quad-cores.

Fig 6.4 shows the computation time taken by our application to execute on the octa-core and the quadcore. Almost octa cores are 12% faster than quadcore to execute the application completely. The algorithm was also run for different number of samples from 10k to 60k to observe the behavior of algorithm. As the

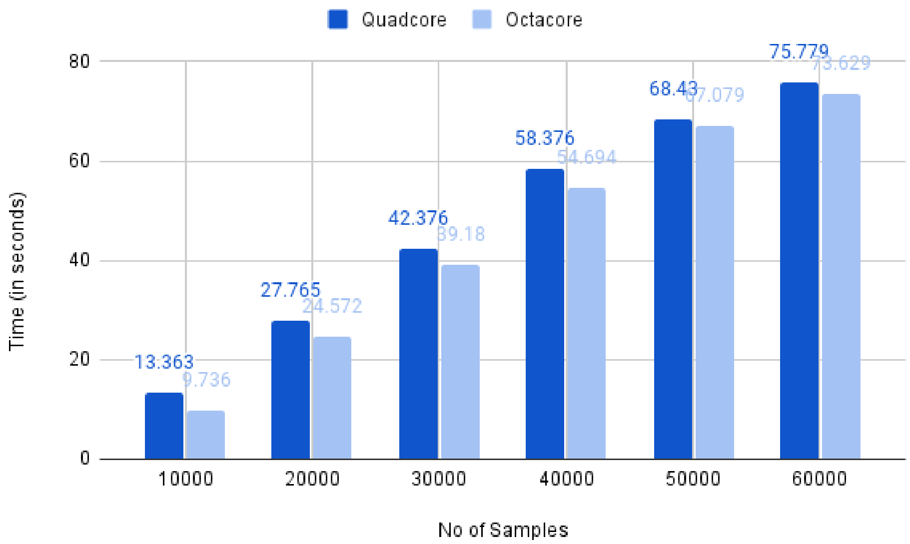


Figure 6.4: Computation Time

samples increases from 10k to 60k which is 6 times, also the time was increased by almost 6 times which can easily be observed from the fig 6.4. Octacores with a high number of cores also come high number of memory and many such parameters which somewhat increase the speed of the algorithm than on quad cores. Here the time was measured using the inbuilt java function and it gives the precision up to the nanoseconds. Also, we tried measuring with milliseconds but it was not accurate and precise such as the nanoseconds clock of the JVM machine.

6.4 Chapter Summary

This chapter has discussed the analysis behavior of the algorithm on different mobile phones and also compared the CPU usage with trending mobile applications.

Chapter 7

Consclusion and Future Work

7.1 Conclusion

In this thesis, we have proposed architecture based which uses deep learning models and machine learning models to predict human behavior based on the inbuilt sensor called an accelerometer in the mobile phone, which especially measures the movement or acceleration of mobile. A detailed description is provided for different models used for the prediction of physical activity. This can help to know person that how much of time out of 24 hours, he was active. The main advantage of this system is that it can be computed on the same device from which sensor data was collected without any need for external infrastructure. Only pre-trained model was integrated with application so it will be installed whenever the application gets installed.

7.2 Future Work

More data can be collected to increase the accuracy of the model. Also, currently model is able to predict only three specific physical activities- cycling, running, and walking. We can expand this work to predict further more activities of the person based on the mobile sensor data only. Also, some interactive User interface can be developed which is user-friendly to display the activity hours taken.

References

[1] Foodloop gmbh. https://www.foodloop.net/en/, 2016. [Online; accessed-2016].

[2] Map your meal. http://www.mapyourmeal.org/, 2016. [Online; accessed-2016].

[3] D. B. Abdullah and M. M. Al-Hafidh. Developing parallel application on multi-core mobile phone. *International Journal of Advanced Computer Science and Applications*, 4(11), 2013.

[4] V. Agarwal, M. S. Hrishikesh, S. W. Keckler, and D. Burger. Clock rate versus ipc: The end of the road for conventional microarchitectures. 28(2):248–259, may 2000.

[5] O. Alzakholi, H. Shukur, R. Zebari, S. Abas, M. Sadeeq, et al. Comparison among cloud technologies and cloud performance. *Journal of Applied Science and Technology Trends*, 1(2):40–47, 2020.

[6] Apple. ike elgan. 2017. with smartphones like these, why do we need laptops? https://www.computerworld.com/article/3241233/with-smartphones-like-these-why-do-we-need-laptops.html. [Online].

[7] Apple. Samuel axon. 2021. iphone 13 and 13 pro review: If you could have three wishes. https://arstechnica.com/gadgets/2021/09/iphone-13-and-13-pro-review-if-you-could-have-three-wishes/. [Online].

[8] Apple. A15 bio. https://nanoreview.net/en/soc/apple-a15-bionic/, 2016. [Online; accessed-2016].

[9] M. Y. Arslan, I. Singh, S. Singh, H. V. Madhyastha, K. Sundaresan, and S. V. Krishnamurthy. Cwc: A distributed computing infrastructure using smartphones. *IEEE Transactions on Mobile Computing*, 14(8):1587–1600, 2015.

[10] AWS. Amazon web services. https://aws.amazon.com/getting-started/projects/host-net-web-app/services-costs/. [Online].

[11] S. Ballı, E. A. Sağbaş, and M. Peker. A mobile solution based on soft computing for fall detection. *Mobile Solutions and Their Usefulness in Everyday Life*, pages 275–294, 2019.

[12] R. Bongiovanni and J. Lowenberg-DeBoer. *Precision agriculture*, 5:359–387, 2004.

[13] F. Büsching, S. Schildt, and L. Wolf. Droidcluster: Towards smartphone cluster computing - the streets are paved with potential computer clusters. 06 2012.

[14] K. Cao, Y. Liu, G. Meng, and Q. Sun. An overview on edge computing research. *IEEE access*, 8:85714–85728, 2020.

[15] W. O. Cesário, D. Lyonnard, G. Nicolescu, Y. Paviot, S. Yoo, A. A. Jerraya, L. Gauthier, and M. Diaz-Nava. Multiprocessor soc platforms: a component-based design approach. *IEEE Design & Test of Computers*, 19(6):52–63, 2002.

[16] CISCO. Cisco white paper. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, 2018.

[17] M. O. DairyMaster. http://www.dairymaster.com/heat-detection/. 2016.

[18] Delloite. Mobile device statistics. https://www.business-standard.com/article/current-affairs/india, 2017. [Online; accessed 19-July-2017].

[19] B. Fu, F. Kirchbuchner, A. Kuijper, A. Braun, and D. V. Gangatharan. Fitness activity recognition on smartphones using doppler measurements. *Informatics*, 5:24, 2018.

[20] J. Garcia Almiñana, E. Simo, X. Masip, E. Marin-Tordera, and S. Sanchez-Lopez. Do we really need cloud? estimating the fog computing capacities in the city of barcelona. pages 290–295, 12 2018.

[21] Geekbench. John Poole. 2021. Introducing Geekbench 5. . https://www.geekbench.com/. [Online].

[22] S. George, T. Eiszler, R. Iyengar, H. Turki, Z. Feng, J. Wang, P. Pillai, and M. Satyanarayanan. Openrtist: End-to-end benchmarking for edge computing. 19(4), 2020.

[23] A. Hakim, M. Huq, S. Shanta, and B. Ibrahim. Smartphone based data mining for fall detection: Analysis and design. *Procedia Computer Science*, 105:46–51, Jan. 2017.

[24] J. Han, Y. Ma, Q. Mei, and X. Liu. Deeprec: On-device deep learning for privacy-preserving sequential recommendation in mobile commerce. In *Proceedings of the Web Conference 2021*, WWW '21, page 900–911, New York, NY, USA, 2021. Association for Computing Machinery.

[25] Herdwatch. https://www.herdwatch.ie/. [Online].

[26] Intel. https://ark.intel.com/content/www/us/en/ark/products/series/88393/6th-generation-intel-core-i5-processors.html. .

[27] A. Kumar. *Towards Trustworthy Ubiquitous Computing: Bringing Privacy and Robustness to Computing in Context*. PhD thesis, University of Helsinki, Finland, 2022.

[28] L. Kvapilova, V. Boa, P. Dubec, M. Majerník, J. Bogar, J. Jamison, J. C. Goldsack, D. Kimmel, and D. R. Karlin. Continuous sound collection using smartphones and machine learning to measure cough. *Digital biomarkers*, 3 3:166–175, 2019.

[29] H. S. Maghded, K. Ghafoor, A. Sadiq, K. Curran, D. B. Rawat, and K. Rabie. A novel ai-enabled framework to diagnose coronavirus covid-19 using smartphone embedded sensors: Design study. In *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, pages 180–187, Los Alamitos, CA, USA, aug 2020. IEEE Computer Society.

[30] L. Moroney. Using tensorflow lite on android: [online], https://blog.tensorflow.org/2018/03/using-tensorflow-lite-on-android.html. , 2018.

[31] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, and P. Nillaor. *Computers and electronics in agriculture*, 156:467–474, 2019.

[32] S. Patel, K. Sasidhar, M. Vaghela, P. Katrodia, and R. Wagani. An m-health app that goes beyond screen time applications. In *2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC)*, pages 94–99, 2022.

[33] S. Patel, K. Sasidhar, M. Vaghela, P. Katrodia, and R. Wagani. An m-health app that goes beyond screen time applications. In *2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC)*, pages 94–99, 2022.

[34] QUALCOMM. https://arstechnica.com/gadgets/2019/02/qualcomm-is-already-announcing-next-years-5g-chips. , 2019.

[35] M. Raihan, S. Mondal, A. More, M. O. F. Sagor, G. Sikder, M. A. Majumder, M. A. Al Manjur, and K. Ghosh. Smartphone based ischemic heart disease (heart attack) risk prediction using clinical data and data mining approaches, a prototype design. In *2016 19th International Conference on Computer and Information Technology (ICCIT)*, pages 299–303, 2016.

[36] A. P. Rajan et al. Evolution of cloud storage as cloud computing infrastructure service. *arXiv preprint arXiv:1308.1303*, 2013.

[37] N. Rajovic, P. M. Carpenter, I. Gelado, N. Puzovic, A. Ramirez, and M. Valero. Supercomputing with commodity cpus: Are mobile socs ready for hpc? In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, New York, NY, USA, 2013. Association for Computing Machinery.

[38] R. Rawassizadeh, T. J. Pierson, R. Peterson, and D. Kotz. Nocloud: Exploring network disconnection through on-device data analysis. *IEEE Pervasive Computing*, 17(1):64–74, 2018.

[39] J. Ren, G. Yu, Y. He, and G. Y. Li. Collaborative cloud and edge computing for latency minimization. *IEEE Transactions on Vehicular Technology*, 68(5):5031–5044, 2019.

[40] B. P. Rimal, E. Choi, and I. Lumb. A taxonomy and survey of cloud computing systems. In *2009 Fifth international joint conference on INC, IMS and IDC*, pages 44–51. Ieee, 2009.

[41] C. A. Ronao and S.-B. Cho. Human activity recognition with smartphone sensors using deep learning neural networks. 59(C):235–244, oct 2016.

[42] E. A. Sağbaş, S. Korukoglu, and S. Balli. Stress detection via keyboard typing behaviors by using smartphone sensors and machine learning techniques. *J. Med. Syst.*, 44(4), feb 2020.

[43] B. Schaffner, J. Sawin, and J. Myre. Smartphones as alternative cloud computing engines: Benefits and trade-offs. pages 244–250, 08 2018.

[44] B. Schaffner, J. Sawin, and J. M. Myre. Smartphones as alternative cloud computing engines: Benefits and trade-offs. In *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 244–250, 2018.

[45] C. L. Seitz et al. System timing. *Introduction to VLSI systems*, pages 218–262, 1980.

[46] M. P. Singh and M. K. Jain. Evolution of processor architecture in mobile phones. *International Journal of Computer Applications*, 90(4):34–39, 2014.

[47] C. Stachl, Q. Au, R. Schoedel, S. D. Gosling, G. M. Harari, D. Buschek, S. T. Völkel, T. Schuwerk, M. Oldemeier, T. Ullmann, et al. Predicting personality from patterns of behavior collected with smartphones. *Proceedings of the National Academy of Sciences*, 117(30):17680–17687, 2020.

[48] J. Switzer, G. Marcano, R. Kastner, and P. Pannuto. Junkyard computing: Repurposing discarded smartphones to minimize carbon. New York, NY, USA, 2023. Association for Computing Machinery.

[49] E. topics blog. https://explodingtopics.com/blog/smartphone-stats. , 2019.

[50] M. Vaghela, K. Sasidhar, A. Parikh, and R. Wagani. Assessing mobile usage, physical activity and sleep through smartphone sensing: A digital phenotype study. *SN Computer Science*, 3, 06 2022.

[51] M. D. Ventra and Y. V. Pershin. The parallel approach. *Nature Physics*, 9:200 – 202, 2013.

[52] D. Yates and M. Islam. Readiness of smartphones for data collection and data mining with an example application in mental health. 10 2019.

[53] D. Yates and M. Z. Islam. Data mining on smartphones: An introduction and survey. 55(5), dec 2022.

[54] S. Yi, Z. Qin, and Q. Li. Security and privacy issues of fog computing: A survey. In K. Xu and H. Zhu, editors, *Wireless Algorithms, Systems, and Applications*, pages 685–695, Cham, 2015. Springer International Publishing.