PREDICTING USER LOCATION AND INTENT USING SMARTPHONE WIRELESS SENSORS

A Thesis

by

SANJAY NAYAK

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Rabinarayan Mahapatra
Committee Members,	Vivek Sarin
	I-Hong Hou
Head of Department,	Scott Schaefer

May 2023

Major Subject: Computer Science

ABSTRACT

The widespread use of smartphones by most of the population has made smartphone data highly valuable in gaining insights into users' behavior by capturing their daily activities. This data can be leveraged to predict user behavior and intentions, forecast their location and suggest places and activities. Machine learning and deep learning techniques have been employed to extract hidden and valuable information from complex and diverse behavioral data to improve the ability to recognize and utilize human activities across different domains. The data is collected using various sensors, and learning algorithms generate valuable predictions in fields such as medicine, travel, and energy consumption. A robust framework is essential to perform these tasks, and in this thesis, we present a framework for collecting, processing, anonymizing, and predicting data.

Prior studies have employed different sensor data to capture various types of human activity. However, to capture and predict user behavioral patterns and intentions, we utilized commonly used sensors of smartphones, such as Bluetooth, Wi-Fi, and charging ports, which users frequently utilize. We developed an Android application to gather information from these wireless sensors and GPS. This research evaluated the accuracy and efficiency of three machine learning and four deep-learning architectures based on convolutional neural networks and LSTM variants. Additionally, we introduced a deep-learning architecture that employs ResNet and LSTM, which outperformed the traditional approach of combining convolutional networks with LSTM. The proposed model achieved high accuracy rates of **90.058%** and **90.261%** for predicting user locale and intent, respectively. Overall, this study highlights the potential of smartphone-based wireless sensor data and location services in predicting user activity and location. It emphasizes the importance of effectively utilizing deep learning techniques to process and analyze this data type and provides insight into leveraging smartphone data to predict human behavior in various application areas.

DEDICATION

To my mother, father, and family, with love.

ACKNOWLEDGMENTS

I am very grateful for the help and guidance my advisor, Dr. Rabinarayan Mahapatra, provided during my research work. This research would not have been possible without his support and insights. I would also like to thank Dr. Vivek Sarin and Dr. I-Hong Hou for being a part of my committee and providing insightful and constructive feedback. A special thanks to Dr. John Keyser and Karrie Bourquin, who provided necessary support and guidance as graduate advisors throughout my graduate program.

I would also like to thank Akash Sahoo, a Ph.D. student in the Department of Computer Science at Texas A&M University, for his valuable feedback, suggestions, and support during my thesis research. I am thankful to my family and friends for providing me with the necessary support and morale during my graduate studies. Also, I would like to express my sincere gratitude to all the individuals who assisted me in my research by granting permission to collect data from their smartphones.

Also, I would like to express my gratitude to Texas A&M University Graduate and Professional School and the Department of Computer Science for allowing me to conduct the research and complete my thesis.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee of Dr. Rabinarayan Mahapatra [advisor] and Dr. Vivek Sarin of the Department of Computer Science & Engineering, and Dr. I-Hong Hou of the Department of Electrical & Computer Engineering.

The data collection part using the android application provided in Chapter III, Data Gathering, is conducted with the help of Akash Sahoo, Ph.D. student of the Department of Computer Science.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

No other outside source of funding was provided.

NOMENCLATURE

HAR	Human Activity Recognition
GCP	Google Cloud Platform
LSTM	Long Short Term Memory
CNN	Convolutional Neural Network
ResNet	Residual Network
GPS	Global Positioning System
POI	Points of Interest
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
B/CS	Bryan and College Station
TAMU	Texas A&M University
OGAPS	Office of Graduate and Professional Studies at Texas A&M University

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Smartphones: Their Usage and Importance

The invention of the telephone was a significant milestone in communication, bringing about a revolutionary change in how people communicated over long distances. Before its creation, communication over long distances often took days, if not weeks, to complete. The telephone enabled people to connect with others within a matter of minutes or hours, thereby transforming the way the world communicated. With the rapid advancement of technology, the telephone underwent several changes and transformed into a hand-held device. Since its inception in 1992, the smartphone has brought about another wave of transformation in the field of communication, offering a wide range of features, including internet browsing, gaming, and social networking.

With further research and technological advancements, smartphones have evolved into sophisticated devices that can monitor and track various aspects of a user's life, including their health and fitness, daily activities, and location. The sensors integrated into smartphones can provide precise location information, reduce background noise during calls, record calls, and even integrate deep learning algorithms with camera hardware to enhance image and video quality.

The widespread availability and usage of GPS-enabled smartphones have also made collecting data for research in human activity recognition easier, providing valuable insights into people's behavior and movements. Smartphones have also emerged as essential tools for people with disabilities, helping them navigate and communicate easily.

Moreover, smartphones can serve as a lifeline during emergencies, enabling users to send SOS signals, quickly spread news, and monitor activities in real time. During the COVID-19 pandemic, for instance, smartphones played a crucial role in tracking the spread of the virus and facilitating contact tracing efforts. In short, smartphones have become an integral part of daily life, enabling people to stay connected, informed, and productive. With the constant evolution of technology, smartphones will likely continue transforming how we communicate and interact with the world.

1.2 Human Activity Recognition

Human Activity Recognition (HAR) research has expanded rapidly due to its wide application in various fields. In particular, it is being utilized in medical services such as pervasive health-care [1], elderly physical activity monitoring [2], physical activity detection [3][4], travel mode identification [5], and indoor localization [6]. Furthermore, HAR is also widely used in smart homes to provide recommendations on energy conservation and health using user data. The usage of smartphones and wearable sensors such as smartwatches in HAR helps to understand the behavior and habits of users, as well as provide insights related to their health using parameters like heart rate, sleep, and activity cycle.

Smartphones are increasingly used to personalize recommendations in shopping, entertainment, recreation, and other areas. These recommendations include groceries, movies, games, restaurants, and outdoor activities. Many studies focus on the smartphone data generated from GPS and sensors such as a gyroscope, magnetometer, barometer, and accelerometer, using which researchers can predict users' activities, such as walking, stillness, running, climbing stairs, and going up or down in an elevator. We can use smartphone data to predict the user's location, which can help in personalized recommendations and intent predictions beyond just activity predictions. The application of user locale and intent prediction extends to energy conservation at home, food and activity recommendations based on the user's lifestyle, and more. One of the challenges in studying smartphone-based data is the public availability of datasets owing to their sensitive nature. To overcome this challenge, we introduced an android application for data collection from users and mechanisms to anonymize the raw data.

1.3 Machine Learning

Machine learning is a field of computer science that enables machines to learn from data and make predictions or decisions without being explicitly programmed. It is a subset of artificial intelligence that uses algorithms and statistical models to analyze and identify patterns in large datasets. Machine learning has numerous applications in various industries, including healthcare, finance, retail, and marketing. It is used for image recognition, natural language processing, fraud detection, predictive maintenance, and personalized recommendations. With the increasing availability of data and computing power, machine learning is essential for businesses and organizations to gain insights, improve efficiency, and enhance customer experience.

1.4 Deep learning

Deep learning is a subfield of machine learning that has been rapidly growing and revolutionizing many industries in recent years. It is an artificial neural network that uses multiple layers of interconnected nodes to learn from large datasets and make predictions or decisions. Deep learning has been successful in tasks such as image and speech recognition, natural language processing, and autonomous vehicles. The key advantage of deep learning is its ability to automatically discover and extract complex features from raw data, which makes it suitable for tasks that involve large amounts of unstructured data. However, deep learning models are complex and require large amounts of data and computational resources to train. Despite these challenges, deep learning can transform many industries and solve some of society's most challenging problems.

1.5 Research Motivation & Goal

The widespread use of smartphones has enabled access to data that provides insights into a person's behavior and intentions. This has led to an opportunity to deliver personalized recommendations that cater to users' interests based on their smartphone usage data. Recommendations can cover a range of topics, such as content, restaurants, activities, and lifestyle, with a focus on energy consumption and savings becoming more critical. Additionally, predicting a user's next location or intention is vital for suggesting new places and activities. The increasing reliance on smartphones in everyday life highlights the importance of utilizing their data to deliver customized and meaningful recommendations to users. This research aims to motivate the use of smartphone data to provide users with personalized recommendations and enhance their overall experience.

This research study aims to showcase the potential of smartphone-based wireless sensor data and location services for predicting user activity and location. The study emphasizes the significance of employing deep learning techniques for efficient processing and analyzing smartphone-based data. By leveraging smartphone data, the study seeks to provide insights into predicting human behavior and demonstrate the usefulness of such predictions in various areas. The research findings enhance our understanding of how smartphone data can be harnessed for prediction, allowing for informed decision-making in diverse domains.

1.6 Research Contributions

This research has the following contributions.

1. A framework is introduced that can be used for the user's locale and intent prediction using GPS data and wireless sensors like Bluetooth and Wi-Fi data. It utilizes Google Cloud Platform (GCP) and Cloud Firestore to manage and store the data collected effectively and securely because GCP allows seamless integration with android applications and ensures secure data transmission from the app to the Cloud Firestore database.

2. An android application was developed and integrated with GCP for data collection and storage.

3. A deep learning architecture using Residual Network (ResNet) and Long Short-Term Memory (LSTM) is proposed, and the approach is benchmarked, which achieved greater accuracy with 90.058% and 90.261% for location and intent prediction, respectively.

4. User locale and intent prediction and evaluation of its accuracy.

For the proposed architecture benchmark, we selected LSTM, convolutional LSTM, Convolutional Neural Network (CNN) with LSTM, and machine learning models like Random Forest Classifier, XGBoost Classifier, and SVM that were used in previous research.

1.7 Thesis Organization

The rest of the thesis is organized as follows.

- Chapter 2 offers a summary of previous research and studies carried out on Human Activity Recognition using smartphone-based data.

- Chapter 3 depicts the initial stage of the research, which involves data collection, and provides a detailed description of the various techniques used during the process.
- Chapter 4 presents information about the data processing techniques employed in the research to prepare the data for training.
- Chapter 5 provides details about the hardware and software specifications, as well as information about the training of the model.
- Chapter 6 describes the three machine learning approaches employed in the research as part of the benchmarking process for evaluating the proposed model.
- Chapter 7 provides details regarding the architecture of the four deep-learning models employed in the thesis, including the proposed model.
- Chapter 8 displays the outcomes of the thesis, with a particular emphasis on the accuracy metric.
- Chapter 9 presents a summary of the findings and conclusions obtained from the thesis, outlining the challenges encountered throughout the research process. Additionally, it offers insights into potential areas of future research.

2. BACKGROUND & LITERATURE REVIEW

The research in reference [7] focuses on the individual's behavioral patterns and habits and the regularity of activities performed in a group. To capture these, the location data is obtained from mobile phones. This study uses the technique of clustering of location and their integration with the map and point of interest information to identify different regions of interest, which are further utilized in the user's location prediction that involves the user's current location, latest time, and prediction time. A framework that uses location-based recommendations using the geographic and semantic features extracted from the user's GPS trajectory is provided in the reference [8]. This research proposes a Bidirectional LSTM model for next activity predictions and uses a Similarity-based Markov Model to recommend the next locations.

Antonio and Chávez have also conducted similar research using Markov models, where they made future predictions using Hidden Markov Models [9]. They used Points of Interest (POI) to predict a user's next POI, considering their history of POIs. They also provided a hypothesis that the user's data follows the Markovian concept when the data points are grouped by weekday and daytime. They also supported this hypothesis through their research.

Another interesting approach to the Hidden Markov Model can be seen in the study conducted by Liu, Zhao, and Han. They proposed a spatio-temporal dynamic time warping algorithm, and Zhu [10]. They used activity data from Call Detail Records (CDR), which are obtained from mobile phones, and used their proposed algorithm to analyze the user's traffic and activity data. Through their study, they tried to reveal the user's behavior patterns and classify them into four categories as *dining*, *working*, *entertainment*, and *resting* modes. Also, they classified the transportation modes as *public transport* or *private car*. The study uses a Hidden Markov Model to find out the user's activities and also uses deep learning methods for human activity prediction.

Deep learning methods have also been used in other research to determine the activities of humans. One such research described in reference [11] utilizes the spatial location data, which is collected using Ubisense, and employed three deep learning algorithms: Convolutional Neural

Network (CNN), Artificial Neural Network (ANN), and Long Short-Term Memory (LSTM), and predicts the six different types of indoor activities, which can be categorized as textitwalking, *sitting*, *lying*, *standing*, *jumping*, and *jogging*. The study was mainly done to assist other researchers in effectively using neural networks with spatial location data.

The knowledge of spatial-temporal activity preference (STAP) has advanced with the increase in location-based social networks (LBSNs). LBSN has the potential in recommendations, advertisements, and other applications. Yang, Zhang, Zheng, and Yu proposed a STAP model [12], which models a user's activity preference. It uses users' historical data to determine their interest in an activity during a particular time and location.

Besides determining the user's activity, deep neural networks have also been used in applications such as indoor localization. The research in reference [6] uses a deep convolutional neural network to utilize data from accelerometers, magnetometers, gyroscopes, and barometers for recognizing and classifying nine different types of activities. Most previous works on smartphone-based data focused on making location recommendations using spatial data or determining human activities using machine learning, deep learning, or Markov models. Through this research, we aim to use the commonly used sensors of smartphones like Wi-Fi, Bluetooth, GPS, and charging ports and utilize their data to understand and predict human behavior and intention, including the next location type of the user. As the effectiveness of convolutional neural networks in activity and location predictions has been demonstrated in previous research, we have utilized residual networks, a variant of the deep convolutional neural network, in our proposed model.

3. DATA GATHERING

Data Gathering is the first step in any research. It describes the procedures for collecting data required for research and analysis. In our research, several steps were taken as part of the data-gathering procedure. The following sub-sections provide a detailed account of the various steps and actions taken throughout the research to gather and process the data, making it training-ready.

3.1 Android Application

For the study, we collected data from wireless sensors and location service of smartphones using an android application. The android application runs on android version 12 and higher and requests permission for Location, activity, and to run in the foreground from the user. It uses the WiFi, Bluetooth, and location services of android to connect with the smartphone's hardware and gather the data points. Some of the android services used in the application build include

- BluetoothAdapter
- BluetoothManager
- BluetoothDevice
- WifiManager
- GPS
- FirebseAuth
- FirebaseUser
- FirebaseFirestore

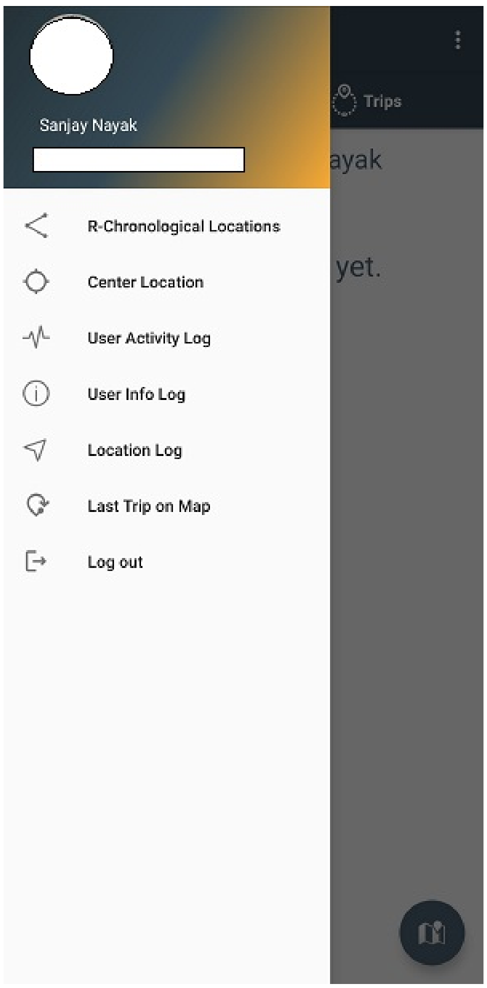
The *BluetoothAdapter*, *BluetoothManager*, and *BluetoothDevice* are used for Bluetooth sensor data, while *WifiManager* is used for WiFi sensor data. The location data are derived from GPS sensors of the smartphones, while *FirebseAuth* and *FirebaseUser* handle the user authentication and authorization in the application, and *FirebaseFirestore* deals with the Cloud Firestore database. The application can detect whether WiFi and Bluetooth are turned on or off and distinguish the different types of Bluetooth devices as per the device types defined in the android, which includes

- audio_video
- computer
- health
- networking
- peripheral
- phone
- wearable

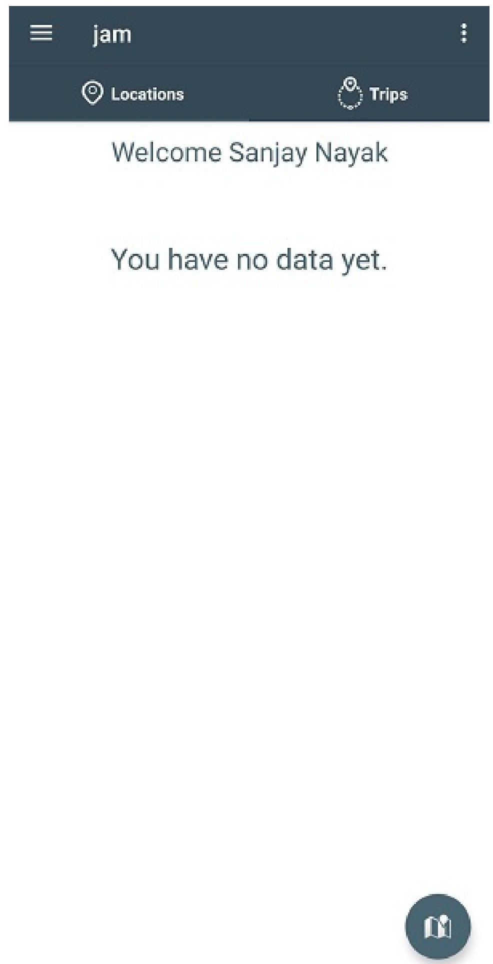
Fig. 3.1a and Fig. 3.1b show the application's main menu and home page, respectively. The user needs to authenticate using a Google account, and Google Firebase handles the authentication and authorization part. Twenty-three users used the application and provided consent before data collection. The app remains active throughout the day and sends data to the Cloud Firestore database every 60 seconds or when the user's location changes by 10 meters. $148,124$ data points are collected from 23 users over two to three weeks. Fig. 3.1 shows three views of the android app built for data collection, where Fig. 3.2b and Fig. 3.2d represent the activity log and location log that users can use to check their activity and location information, respectively. The following three permissions need to be enabled by the user while running the application for the application to collect data.

- Location
- Activity
- Foreground run

These permissions are an essential part of the application as they give the user control over the collected data. Fig. 3.2a and Fig. 3.2c show the activity and location permissions, respectively, being asked of the user when the application is installed and used for the first time after installation. Each time the user installs the application, the permission prompt is shown to the user for their approval.

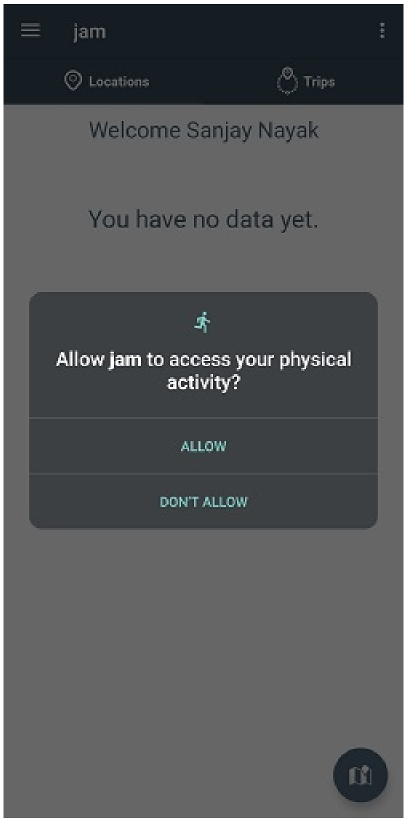


(a) Main menu

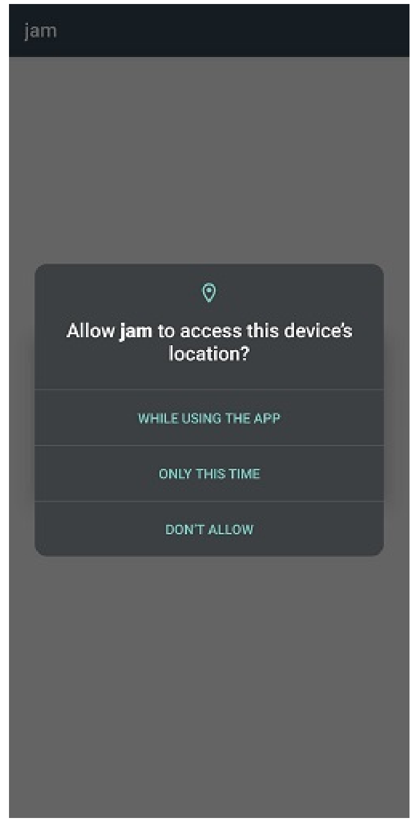


(b) Home page

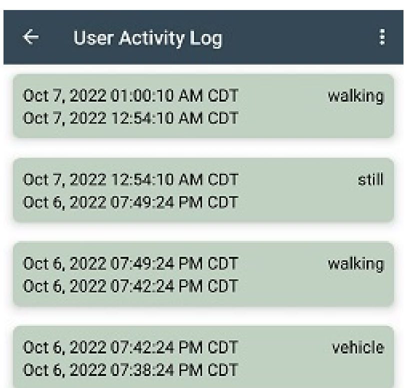
Figure 3.1: Android application screenshots built for data collection



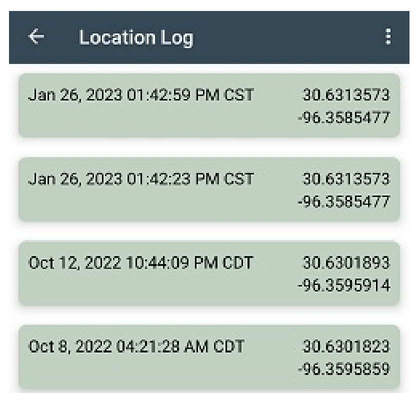
(a) Activity access permission



(c) GPS location access permission



(b) Activity logs



(d) Location logs

Figure 3.2: Android application permissions and logs screenshots

3.2 Data Storage

The raw data from each user's smartphone is saved in the Cloud Firestore database. It is a flexible, No-SQL, and scalable database solution provided by Google for client and server-side development and synchronization. We chose to use Cloud Firestore because of the following reasons that were very important for our use case.

- Cloud Firestore can provide offline support. If the device becomes offline, the data is cached in the smartphone, and data synchronization between the application and the server happens once the device returns online. This feature helps our use case because the device can get offline for unknown reasons, and the data collected during that period is preserved. This feature allowed us to remove the data loss due to network inconsistencies.

- The Cloud Firestore stores data as documents and uses collections to group the documents. Each document stores all data parameters for a single timestamp for one user in key-value format.

The *data* collection holds the data for each user as documents, where each user document contains a subcollection that includes the documents about that user for each timestamp. It forms a hierarchical data structure like */data/<user>/location/<documents>*. Fig. 3.3 displays the locations a user visits over 24 hours, where the *yellow* marker denotes the starting point, and the *orange* marker denotes the end point of the movements for that day. Fig. 3.4 shows the structure of data and the parameters stored in the Firestore database. The permissions provided in the database ensured that each user could view their location details when logged into the application.

Google Cloud Platform (GCP) is used to process the raw data into anonymized data, as security and anonymization are required when dealing with personalized data. Anonymity is one of the main reasons why personalized data is challenging to obtain. For the anonymization process, we used *Google Cloud Storage Buckets* and *Google BigQuery*, where Google BigQuery is the data warehouse solution of GCP.

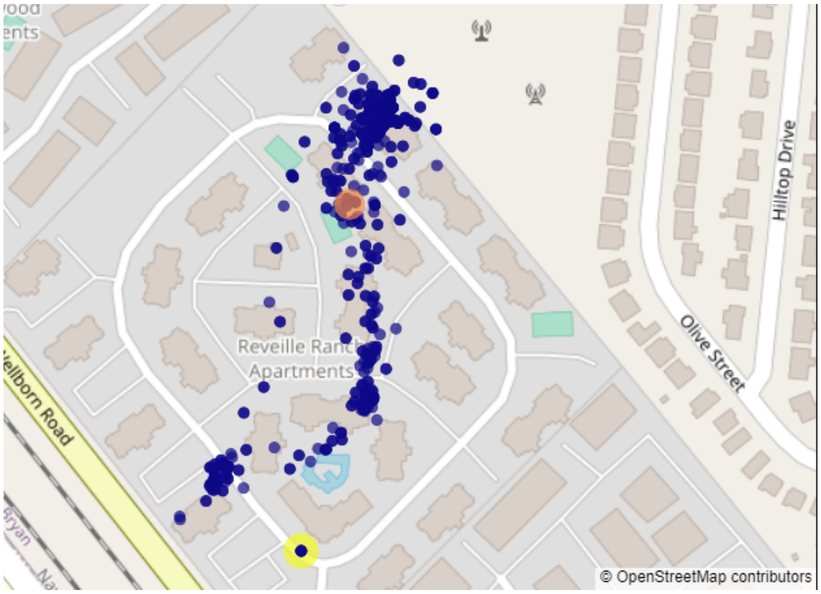


Figure 3.3: Single-day movement data points of a user

Apart from the required parameters, a few extra parameters are added to the raw data as part of the Firebase authorization process that includes *Username* and *User Email*. These parameters are removed from the final anonymized dataset as part of security measures. The required parameters collected using the android application include

- Timestamp
- Latitude
- Longitude
- Charging type
- Charging status
- WiFi status
- Bluetooth status
- Bluetooth device type

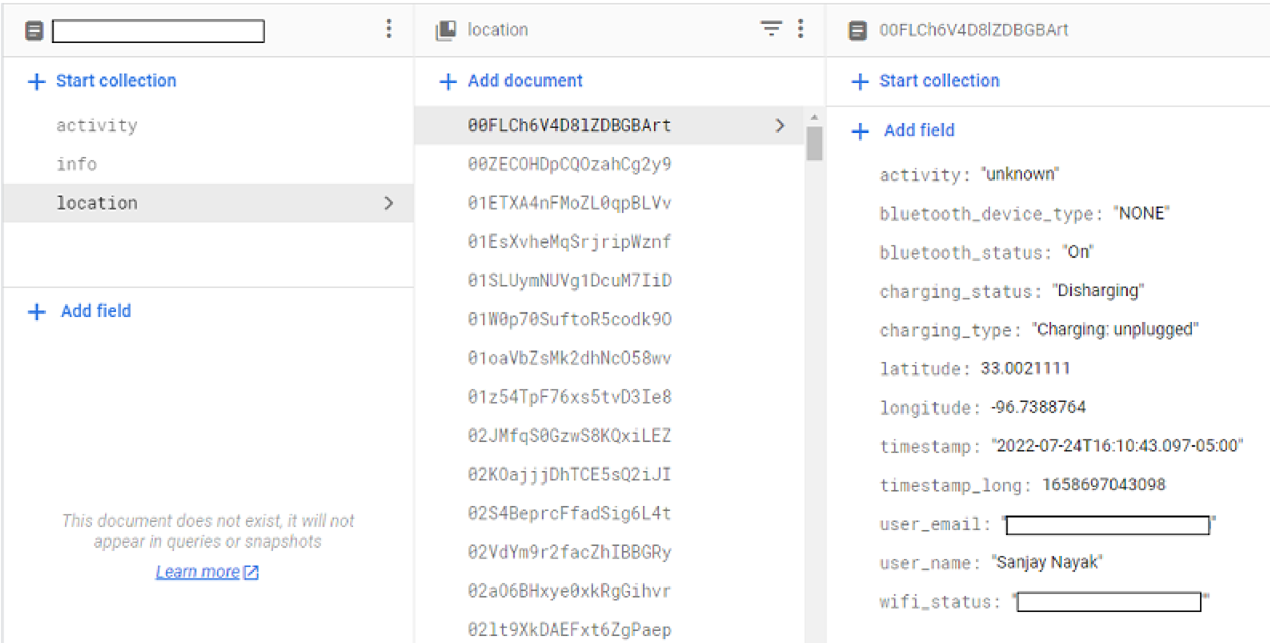


Figure 3.4: Google Cloud Firestore database

3.3 Data Movement

An intermediary step was employed to move data from Cloud Firestore to Google BigQuery, temporarily storing the data in a Google Cloud Storage Bucket. The raw data is exported in CSV format and saved to the Cloud Storage Bucket before being imported into Google BigQuery for the final anonymization process. This approach ensures that data is transferred securely and efficiently while maintaining the data's integrity throughout the process. By leveraging the capabilities of Google Cloud Storage and Google BigQuery, data can be effectively moved between systems, making it accessible for analysis and other purposes. Overall, this approach represents a valuable tool for managing and analyzing data across various platforms and systems. The overview of the data movement steps showcased in Fig. 3.5 is enumerated below.

1. Export the raw data from Cloud Firestore to Google Cloud Bucket in flat file format (CSV format).

2. Import the CSV file from GCP Bucket to GCP BigQuery.

3. Anonymize the raw data in Google BigQuery.

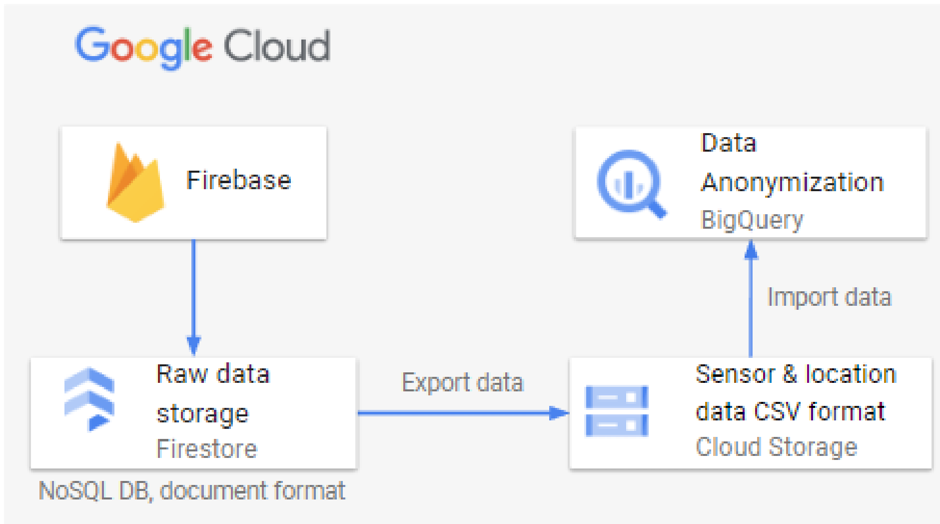


Figure 3.5: Part of architecture showing data movement inside Google Cloud

3.4 Data Anonymization

Ensuring data anonymity is critical as it obscures data points in a way that prevents user information identification. It not only masks the data but also protects user identity from the wrong usage. The below steps are taken for the data anonymization process in our research.

1. We removed the user's email and name, which were added to the metadata when users signed in using their Google accounts.

2. A random number is assigned to each user, and their corresponding information is marked with *user_i*, where *i* represents a randomly generated integer.

3. We replaced WiFi names with Boolean values to mask the user's smartphone connection to any WiFi network.

By taking these steps to anonymize the data, we can maintain user privacy while still providing valuable data for analysis and other purposes. This approach helps safeguard sensitive information and protects the data from unauthorized access or use. Fig. 3.6 shows part of the anonymized data in Google BigQuery, the data warehouse solution of GCP.

Row	user	latitude	longitude	charging_status	charging_type	bluetooth_status	bluetooth_device_type	timestamp	activity
1	user_14	33.0021153	-96.7389237	Charging	Charging:battery_plugged_ac	On	NONE	2022-07-11 00:41:42.267000 U...	still
2	user_13	30.6289718	-96.3599906	Disharging	Charging:unplugged	On	NONE	2022-08-04 06:36:59.515000 U...	still
3	user_16	30.6301166	-96.3592879	Disharging	Charging:unplugged	Off	NONE	2022-09-20 02:11:32.201000 U...	unknown
4	user_14	33.0022444	-96.7385361	Disharging	Charging:unplugged	On	NONE	2022-08-12 21:49:19.560000 U...	still
5	user_14	30.6302034	-96.3596079	Charging	Charging:battery_plugged_ac	On	NONE	2022-09-15 10:04:22.096000 U...	still
6	user_10	30.6100577	-96.3510597	Disharging	Charging:unplugged	On	UNCATEGORIZED	2022-09-16 19:45:24.196000 U...	unknown
7	user_7	39.1369214	-76.6246273	Disharging	Charging:unplugged	Off	NONE	2022-09-21 23:07:09.338000 U...	unknown
8	user_14	30.616071	-96.3386517	Disharging	Charging:unplugged	On	NONE	2022-09-06 22:07:57.443000 U...	walking
9	user_14	33.0021093	-96.73889	Disharging	Charging:unplugged	On	NONE	2022-07-07 07:05:56.548000 U...	still
10	user_18	30.631337	-96.3585389	Charging	Charging:battery_plugged_ac	On	NONE	2022-09-14 06:50:25.766000 U...	unknown
11	user_5	30.590443	-96.3424258	Charging	Charging:battery_plugged_ac	Off	NONE	2022-09-13 20:14:48.825000 U...	unknown
12	user_16	30.6337491	-96.35426	Disharging	Charging:unplugged	Off	NONE	2022-09-28 18:17:19.478000 U...	unknown
13	user_5	30.5904462	-96.3424163	Disharging	Charging:unplugged	Off	NONE	2022-09-15 06:36:41.319000 U...	unknown
14	user_12	30.6308222	-96.3590011	Charging	Charging:battery_plugged_ac	Off	NONE	2022-09-24 12:19:34.641000 U...	unknown
15	user_12	30.6119459	-96.3413759	Disharging	Charging:unplugged	Off	NONE	2022-09-22 15:00:52.585000 U...	unknown
16	user_14	30.6301595	-96.3597029	Charging	Charging:battery_plugged_ac	On	NONE	2022-09-02 13:03:44.122000 U...	still
17	user_13	30.6289889	-96.3599592	Disharging	Charging:unplugged	On	NONE	2022-08-02 15:26:26.943000 U...	still
18	user_16	30.6309195	-96.3587616	Disharging	Charging:unplugged	Off	NONE	2022-09-26 01:10:47.457000 U...	unknown
19	user_11	30.6301874	-96.3596084	Charging	Charging:battery_plugged_ac	On	NONE	2022-09-17 17:10:50.652000 U...	unknown
20	user_16	30.6308341	-96.3589691	Disharging	Charging:unplugged	Off	NONE	2022-09-29 03:53:56.104000 U...	unknown
21	user_12	30.6309108	-96.3589533	Disharging	Charging:unplugged	Off	NONE	2022-09-24 03:32:00.003000 U...	unknown
22	user_13	30.6311513	-96.3587286	Disharging	Charging:unplugged	On	NONE	2022-08-23 00:49:05.458000 U...	still
23	user_14	30.6336389	-96.3543309	Disharging	Charging:unplugged	On	NONE	2022-08-29 17:52:50.266000 U...	vehicle

Figure 3.6: Part of the anonymized data in Google BigQuery data warehouse

3.5 Summary

This chapter outlines the techniques utilized to ensure efficient and precise data-gathering for the research, highlighting its importance in obtaining accurate and reliable analysis. Several steps were considered while ensuring user anonymity during the data collection process. This is a vital aspect of the research as obtaining good quality data will enable effective and appropriate research outcomes.

4. DATA PROCESSING

After Data Gathering, Data Processing is the next step in our research. It provides a detailed account of processing the raw data collected in the previous step. This procedure ensured that the data format was in the proper format and followed the correct guidelines outlined in the research. This also helped generate initial data analysis and understand the nature and pattern of the data before using them as training data for various models used in the research. To prepare the dataset for model training, we processed the data by adding new features and ensuring it was ready for training. The following two steps are used as part of the initial data processing. By taking these steps to process and organize the data, we optimized its suitability for training and improved the accuracy of our resulting models.

1. We reviewed the data to identify any irregularities in the number of unique locations associated with each user. In cases where the number of unique locations for a user was found to be only one, we excluded that data as it would not be meaningful for training purposes.

2. To organize the data for training, we sorted it based on each user's timestamp, with the earliest timestamp representing the first data point and the latest timestamp representing the last data point for each user.

4.1 Places of Interests determination

The data's latitude and longitude are utilized to identify the places of interest. As the location data is in single-point format, the DBSCAN algorithm was employed to cluster the locations within a 200 meters radius for each user. The distance metric utilized for clustering was the *haversine* distance metric, which calculates the distance between two points on earth using the degree value of the points. The DBSCAN algorithm is sensitive to outliers and facilitates outlier detection. After employing the DBSCAN algorithm, the data points identified as outliers were removed. A total of 170 clusters were formed for all users by the algorithm.

We utilized the *geopandas* library of Python to compute the convex hull of each cluster and obtain the polygon of the cluster geometry. The geometry's compact polygon details contain all the points within it. We filtered out data points that did not form a polygon, which accounted for 43 clusters. After calculating the convex hull, we used the *geopy* Python library to reverse-geocode the centroid of each cluster, which provided us with the street address of each centroid point, indicating the locations of interest. We then categorized these locations into seven types, namely *Home (H)*, *Restaurant (RT)*, *University (U)*, *Office (O)*, *Recreational Centers (RC)*, *Airport (A)*, and *Supermarket (S)*. We selected these types based on the users' characteristics and the types of locations they visited the most. Fig. 4.1 represents the hourly categorization of data of four users.

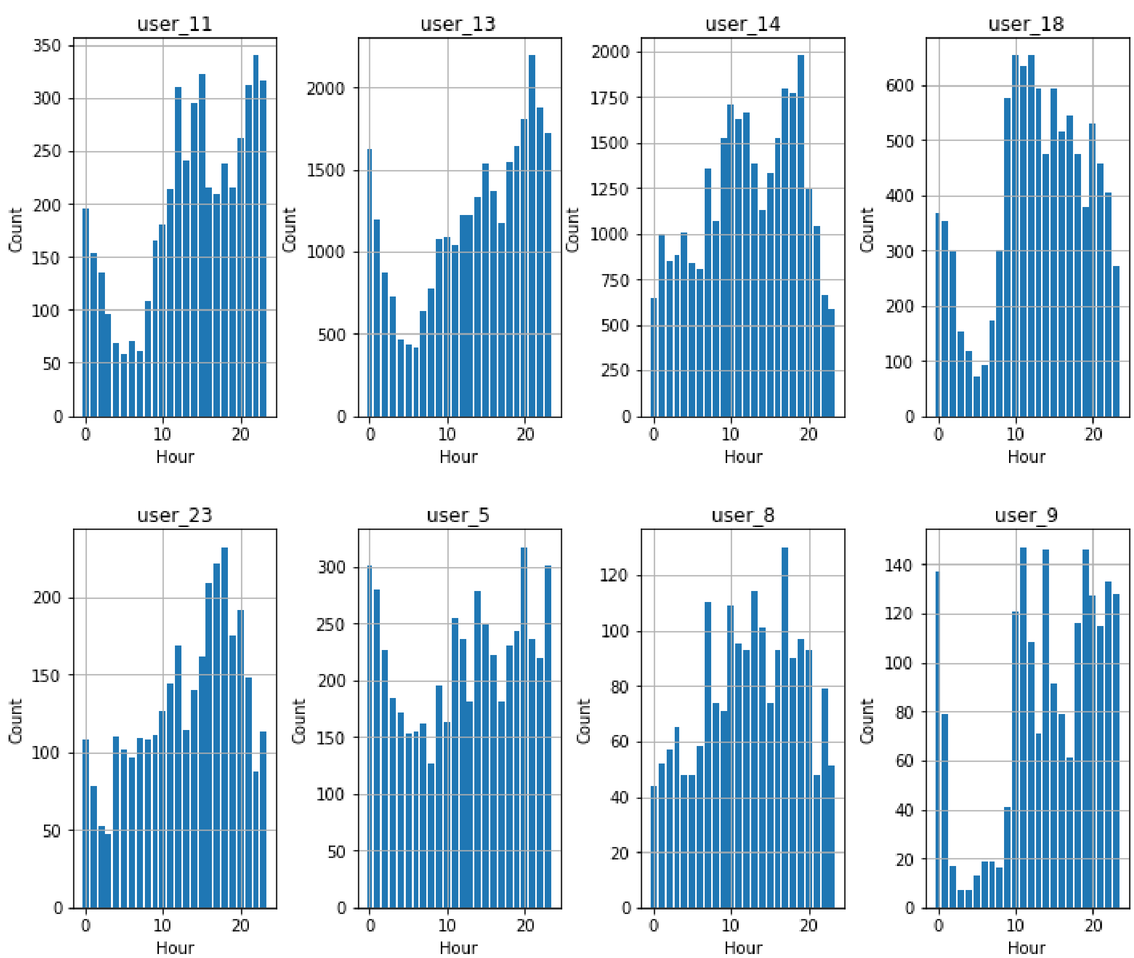


Figure 4.1: Hourly categorization of data per user

4.2 Attributes Addition and Categorization

It is crucial to translate attribute values into corresponding categorical values because the architecture of the models does not support using attributes in decimal format for latitudes and longitudes and text format for Bluetooth device categorization. The categorical values for the three different charging types, *Charging:unplugged*, *Charging:battery_plugged_ac*, and *Charging:battery_plugged_usb*, are $1, 2,$ and 3. Similarly, the Bluetooth device type and status are combined to create five distinct categories, each with a value between 1 and 5. Based on whether WiFi was turned on or off and whether the battery was charging or discharging, we converted the WiFi and charging status data to binary values of 0 and 1, respectively.

The dataset now includes a new attribute called *hour* that specifies the time of day in 24-hour format. After extracting the weekday information from the timestamp data, we added two attributes, *weekday* and *daytype*, which stand for the types of weekday and day, respectively. The *weekday* attribute denotes the day of the week with Monday through Sunday mapped as 0 - 6, and the *daytype* attribute contains the binary value of 0 and 1. The seven locations, which include H, RT, U, O, RC, A, and S, are also mapped to values from 1 - 7 in the same order. Additionally, to predict user intent, we mapped the seven locations to five different intents consisting of *Shopping*, *Recreation*, *Work & Travel*, *Studying*, and *Chilling*. We also mapped these five intents to integer values between 1 and 5. Table 4.1 provides five randomly sampled parameter data, and Table 4.2 contains the labels of the same five randomly sampled users from the final processed dataset. The final dataset contained the following attributes used during model training.

- user
- hour
- weekday
- daytype
- charging_type

- bluetooth
- wifi
- charging_status

Table 4.1: Sample parameter dataset used for model training and testing

Parameters							
user	***hour***	***weekday***	***daytype***	***charging_type***	***bluetooth***	***wifi***	***charging_status***
5	11	4	0	1	5	1	1
11	17	0	0	1	1	1	1
13	1	4	0	2	4	1	2
18	2	6	1	1	3	2	1
18	13	4	0	1	3	1	1

Table 4.2: Sample label dataset used for model training and testing

user	**Location label**	**Intent Label**
5	3	4
11	1	5
13	1	5
18	1	5
18	1	5

4.3 Summary

This chapter describes the techniques utilized to process the anonymized data for the research. The data processing procedure involves two significant steps, namely identifying Points of Interest (POI) and mapping them to seven predefined location categories, and adding further attributes as features during the model training phase. Additionally, the feature data is assigned integer values based on their respective categories to facilitate the model's comprehension and training. For this research, the location types are categorized into seven different types based on the initial data analysis of each user's data. Also, the intent types are kept to five in this research and are based on the different location types of users.

5. REQUIREMENT SPECIFICATIONS & MODEL TRAINING

5.1 Software/Hardware Specifications

To manage and train our data models, we utilized the Google Colab platform and wrote our code in Python version $3.8.10$. We chose Python as our programming language due to its extensive libraries, which streamlined our development process, especially when working with deep learning models. Additionally, we used two important libraries, namely *geopandas* and *hdbscan*, which helped us analyze the location data and extract relevant cluster information. Before diving into the data processing phases, we performed initial data visualization and analysis using the *streamlit* package. With the help of the visualization tools, we gained useful insights into the data and adjusted our models accordingly. We utilized the *Tensorflow* $2.9.2$ library to create our deep learning models, readily available on Google Colab. The platform's GPU hardware accelerator enabled us to train our models more efficiently and reduced the overall training time.

5.2 Model Training

Fig. 5.1 provides the correlation matrix of the final attributes used in the study. Once all data processing methods were implemented, the resulting dataset contained $90,641$ data rows, sorted in ascending order based on each user's timestamp to maintain the sequence of events. To train the models, the final dataset was split into training and testing sets at a ratio of $7:3$. The ResNet + LSTM model was trained with batch sizes of 128, while the remaining three deep learning models were trained with a batch size of 64, each for a total of 80 epochs. We trained each of the models with varying hyperparameters and evaluated their accuracy to identify the model with the highest accuracy as the most suitable one.

5.3 Summary

This chapter provides an overview of the hardware and software specifications used in the research. The development of models and data processing used Python as the main language, and multiple Python libraries are utilized to achieve various parts of the development. JAVA is

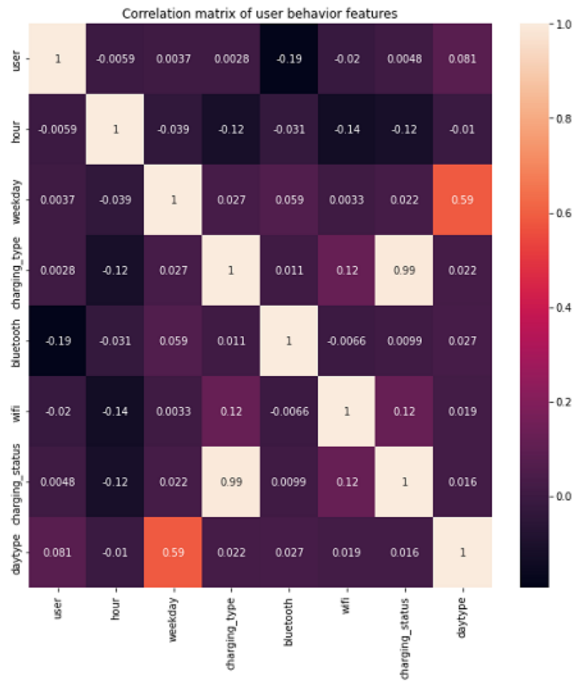


Figure 5.1: Correlation matrix of final dataset

used along with android libraries provided as part of the development kit for the development of the android application. The chapter also details the model training, specifically the training and testing ratio and batch size.

6. MACHINE LEARNING MODELS

The below architecture diagram provides an overview of the whole process involved in the research using the machine learning models described below. The *Machine Learning Model* in point 7 in Fig. 6.1 refers to the different machine learning models used.

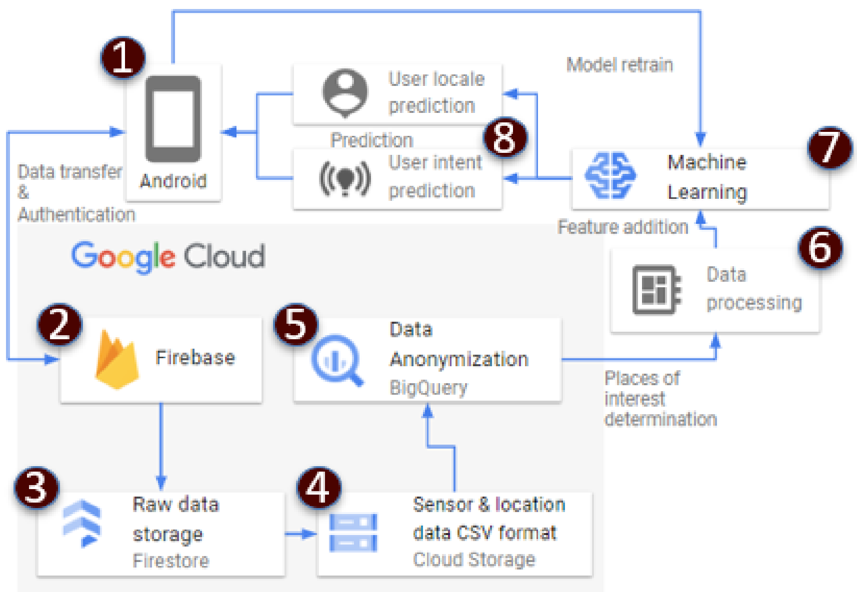


Figure 6.1: Machine Learning Architecture diagram

6.1 Random Forest Classifier

The random forest classifier is a meta-estimator that employs several decision trees, with each tree classifier being fitted on a random subset of the input dataset. This classifier uses the averaging method to arrive at the final prediction, leading to enhanced accuracy while avoiding over-fitting. By combining multiple decision trees and utilizing random subsets of data, the random forest classifier offers a more robust and reliable method for classification. This approach can be particularly useful when working with large datasets with complex structures, as it allows for more efficient and accurate predictions. Overall, the random forest classifier is a valuable tool for classification tasks and offers a powerful approach to data analysis.

6.2 Extreme Gradient Boost Classifier

Extreme Gradient Boosting (XGBoost) is a supervised learning technique known for its efficiency, flexibility, and portability. It utilizes the Gradient Boosting framework as the foundation for its machine learning algorithms. The boosting algorithm adds weak learners to reduce loss, allowing XGBoost to create a strong learner from a collection of weak learners. This approach effectively improves the accuracy and predictive power of the model, making it well-suited for use in various applications. Additionally, XGBoost is highly adaptable and can be applied to diverse datasets, making it a versatile tool in machine learning. Overall, XGBoost is a valuable approach to supervised learning and is highly regarded for its ability to generate accurate predictions efficiently.

6.3 Support Vector Machine(s)

Support Vector Machines (SVM) is a group of supervised learning techniques that utilize a hyperplane to split D-dimensional data, where D is the number of features with the greatest margin. Using SVC kernels, SVMs can transform non-linearly separable data into a linearly separable feature space. SVMs are adaptable and can be used for various tasks, including classification, regression, and outlier detection. This makes them a valuable tool in data analysis and machine learning applications. SVMs can generate highly accurate predictions and models by optimizing the margin between data points. Overall, SVMs are a powerful approach to supervised learning that offers various applications and benefits.

6.4 Summary

The chapter outlines the various machine-learning algorithms that were employed in the research. It highlights the three specific algorithms, namely Random Forest Classifier, XGBoost Classifier, and SVMs, used in previous studies related to smartphone-based data and human activity recognition. The primary objective of utilizing these algorithms was to evaluate and compare their performance with the proposed architecture in this study.

7. DEEP LEARNING MODELS

The below architecture diagram provides an overview of the whole process involved in the research using the machine learning models described below. The *Machine Learning Model* in point 7 in Fig. 6.1 refers to the different machine learning models used.

The architecture diagram showcases the research procedures involved, including deep learning models. Both the machine-learning and deep-learning architectures are similar, with the difference being the type of model utilized in the model training. Also, the models are re-trained only when the accuracy deteriorates or if the business model requires model re-training after certain intervals. Fig. 7.1 shows the research framework involving deep learning models, where point 7 indicates the different deep learning models utilized in the research.

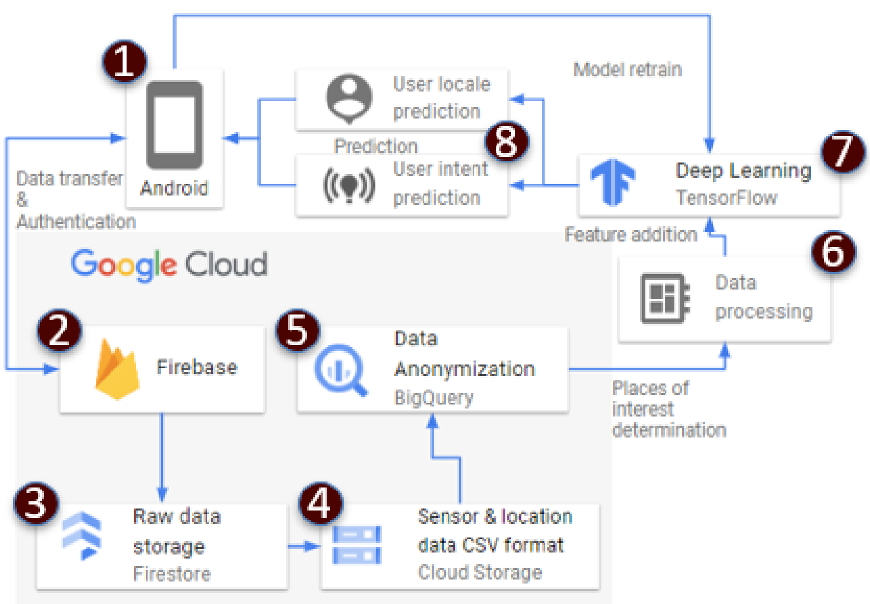


Figure 7.1: Machine Learning Architecture diagram

7.1 Long Short-Term Memory

Long Short-Term Memory, also known as LSTM, is a sophisticated Recurrent Neural Network (RNN) variant created to address the vanishing gradient in RNN. Since time is a crucial component of the research's data, LSTM is employed since it can forecast future activity based on a collection of sequential activity data from the past. It uses the gate concept that includes the read gate, forget gate, and write gate, which provides an output of binary values based on the input. Activation functions, like *tanh* and *sigmoid*, are used within each LSTM gate.

7.1.1 Architecture

Fig. 7.2 represents the architecture used in our research. This study used three layers of LSTM in the LSTM architecture. The goal is to make the model deeper by doing this. A dropout rate of 40% of units in the first two layers and 50% of units in the final LSTM layer is applied to the layers. Each LSTM layer has 100 units, employs bias, and has a hyperbolic tangent, *tanh*, as its activation function. Two densely connected neural network layers follow it with *relu* and *softmax* activation, respectively. This design combines an Adam optimizer with a categorical cross-entropy loss function.

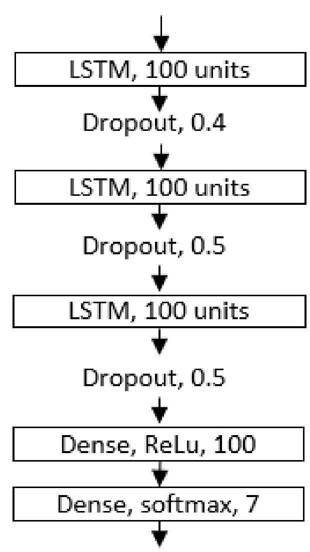


Figure 7.2: LSTM architecture

7.2 Convolutional Neural Network + LSTM

Convolution neural Network (CNN) is mostly used in temporal or spatial datasets. Because of the ability to detect and extract features using the shared weights in the filters, CNN is a good choice for finding the features in time series data. It uses the concept of convolution, a mathematical operation that merges two sets of information to generate a feature map. A set of feature maps are generated from one convolution layer acting as input to the next layer in the network. These feature maps help in the feature extraction process from the data.

7.2.1 Architecture

The research uses a combination of CNN layers followed by LSTM layers similar to the architecture presented in the Reference[13]. Three 1-D convolution layers of filters 64, 128, and 64 are used to create the CNN layers. The activation function for the convolution layers is *relu*, and its kernel size is 1. A dropout layer with a dropout rate of 50% is added to reduce the over-fitting of the model, followed by a pooling layer. The pooling layer decreases over-fitting and calculations by lowering the parameters in the data. We employ max pooling in our architecture. The output is then flattened before being supplied as input to the LSTM layers.

Two LSTM layers follow the CNN layer, with 100 units in each layer. The dropout layer is followed after each LSTM layer with a dropout rate of 40% and 50%, respectively. Each LSTM layer uses the hyperbolic tangent function as its activation function, followed by two densely connected layers with *relu* and *softmax* as their activation functions. Fig. 7.3 presents the CNN + LSTM architecture designed for our research.

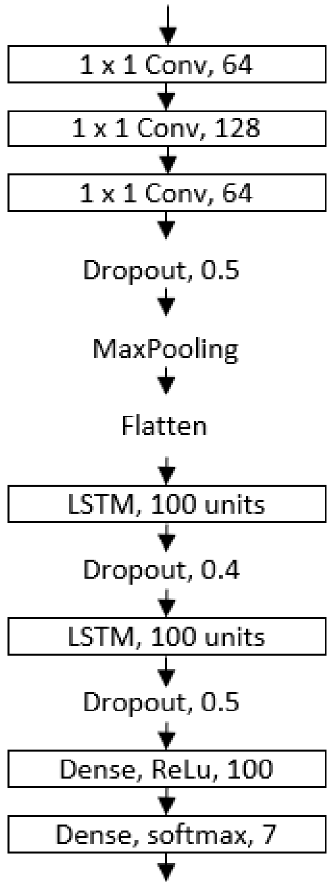


Figure 7.3: CNN + LSTM architecture

7.3 Convolutional LSTM

Convolutional LSTM is a variant of LSTM with differences in the way the internal calculations are being handled within a cell. The main difference between convolutional LSTM and CNN paired with LSTM is how the calculations are done within the model. For Convolutional LSTM [14], the convolution operations replace the internal matrix multiplications of the LSTM cell. For the architecture involving the pairing of CNN and LSTM, the CNN layers are stacked above the LSTM layers such that the output from the CNN layers is passed as input to the LSTM layers after flattening.

7.3.1 Architecture

For the convolutional LSTM architecture, a three-layer architecture is followed. Each of the three layers has filters 128, 64, and 128, respectively, with a kernel size of 1×1 and *relu* as their activation function. A dropout layer follows each convolutional LSTM layer with a dropout rate of 40% in the first two layers and 50% in the final layer. The output is then flattened and passed onto two densely connected neural networks with *relu* and *softmax* as their activation functions. The model uses *adam* algorithm for its optimization *categorical cross entropy* for its loss calculations. Fig. 7.4 shows the convolutional LSTM architecture.

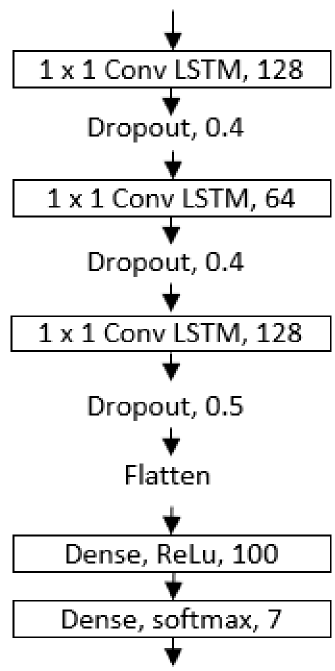


Figure 7.4: Convolutional LSTM architecture

7.4 Residual Network + LSTM

The Deep Convolutional Neural network has a major problem of vanishing gradient obstructing convergence. This issue was addressed by using normalized initialization and intermediate normalization layers. But when the depth of the networks was increased, a degradation problem was noticed that made the training stagnate by saturating the accuracy. This degradation made the training stagnate and reduced the accuracy of the models. Simply put, a deeper network should not have an accuracy higher than its shallower network. The Deep Residual Learning framework [15] solved this problem. Residual Neural Network, or ResNet, solves this problem by adding a shortcut connection that skips one or more layers. Our research includes the concept of ResNet and adds it to the LSTM layers by stacking them up. The input is first passed through the ResNet layers, flattened, and onto LSTM layers for final predictions.

7.4.1 Architecture

The residual block of ResNet is formed by using two convolution layers having kernel size 1×1. Each convolution layer is followed by *relu* activation function. The convolution layers' input and output are added together in the final step before passing the final output to the next layer as input. The addition of the original input with the output of the convolution layers denotes the shortcut connection, which gave ResNet an advantage over CNN.

The main architecture uses three residual network blocks with a filter size of 128. A dropout layer with a dropout rate of 50% is added after the three residual blocks. The final output is flattened and passed onto the subsequent layers of LSTM. Three LSTM layers are added to the architecture stack, each having 100 units, before passing them onto the two densely connected layers for the final prediction. The first dense layer has 100 units, and the second densely connected network contains 7 units for the final classification based on the number of prediction labels. Each of the three LSTM layers is followed by dropout layers with dropout rates of 40%, 40%, and 50%, respectively. The optimization algorithm and the loss function used in this model are *adam* algorithm and *categorical cross entropy*, respectively. Fig. 7.5 provides an overview of the residual block employed in the ResNet architecture, and Fig. 7.6 illustrates the proposed ResNet + lSTM architecture.

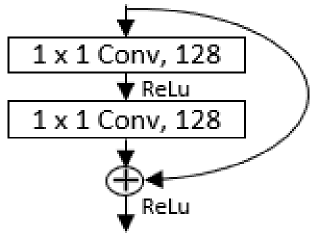


Figure 7.5: ResNet block

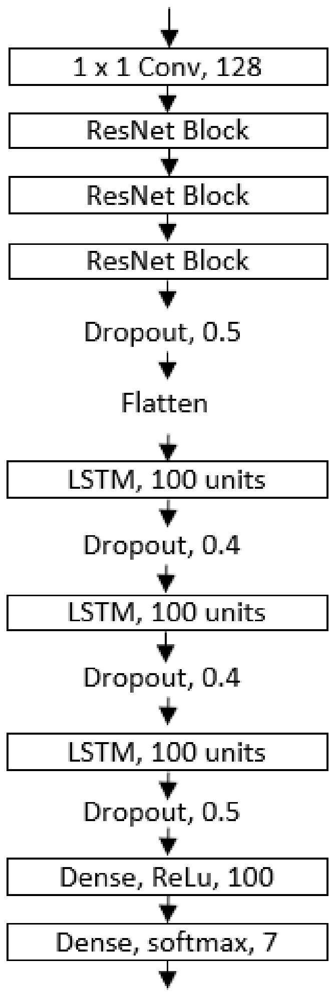


Figure 7.6: ResNet + LSTM architecture

7.5 Summary

In this chapter of the thesis, the focus is on the four deep-learning models that were utilized in the research. The architecture of each model is presented, featuring two densely connected layers with 100 and 7 units in the first and second dense layers, respectively, for user locale prediction. Additionally, the final dense layer was modified to predict user intent based on the number of intents defined. To prevent overfitting, some of the models included dropout and pooling layers. For the CNN or ResNet models stacked on LSTM, the output from the CNN and ResNet layers was flattened to make them suitable for ingestion into the LSTM layers. A novel architecture that combines ResNet and LSTM was proposed as a part of this research.

8. RESULTS

In the thesis, multiple experiments using various machine learning and deep learning models are conducted to benchmark their accuracy, and the accuracy outcomes of the machine learning and deep learning models used are provided in Table 8.1. LSTM, convolutional LSTM, and Convolutional Neural Network (CNN) with LSTM and machine learning models such as Random Forest Classifier, XGBoost Classifier, and SVM models are used for benchmarking the proposed model. Multiple runs were conducted with different hyperparameters, and the optimal model was chosen based on the best accuracy. The deep learning models took around $15 - 25$ minutes to complete 80 epochs. Fig. 8.1 represents the loss versus epoch plot showcasing the progress of the loss values with different epochs during the training for user locale prediction. Fig. 8.2 shows the loss versus epoch plot during user intent prediction training.

Among the different models used, the Random Forest Classifier (RFC) has the highest accuracy among the machine learning models and outperforms the LSTM and Convolutional LSTM models by a margin of $0.4 - 0.5$ in accuracy. The LSTM and Convolutional LSTM architectures exhibit similar accuracies, while the LSTM architecture accuracy exceeds the accuracy of Convolutional LSTM architecture in both user locale and intent predictions. Both LSTM and Convolutional LSTM models perform better than SVM and XGBoost Classifiers. The CNN + LSTM architecture also outperforms LSTM, Convolutional LSTM, and machine learning models. The proposed model, which combines ResNet and LSTM, gives the highest accuracy for user locale and intent prediction, achieving an accuracy of 90.058% and 90.261%, respectively. Both variants of CNN gave similar accuracy results, with the proposed architecture consisting of ResNet performing better. All findings and conclusions are accurate and based on the dataset used in this research, with no tampering with any values.

Table 8.1: Best accuracy of different algorithms used in the research

Model	Accuracy	
	Location Prediction	***Intent Prediction***
Deep Learning models		
LSTM	89.574	89.699
CNN + LSTM	90.044	90.179
Convolutional LSTM	89.563	89.653
ResNet + LSTM	**90.058**	**90.261**
Machine Learning models		
SVM	89.431	89.391
XGBoost Classifier	81.745	81.583
Random Forest Classifier	89.913	90.012

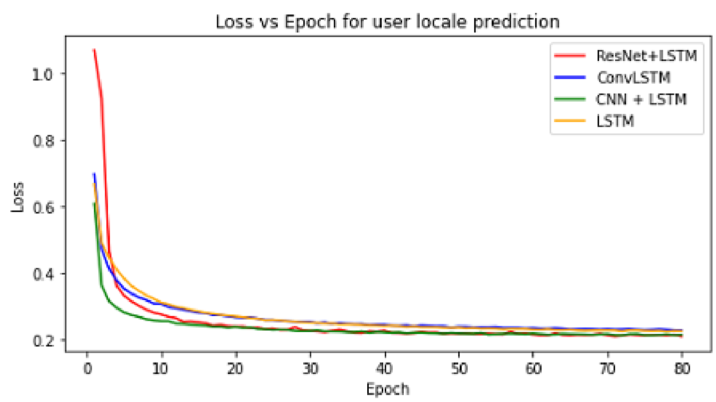


Figure 8.1: Loss versus Epoch depicting the progression of loss values with different values of the epoch during Location prediction

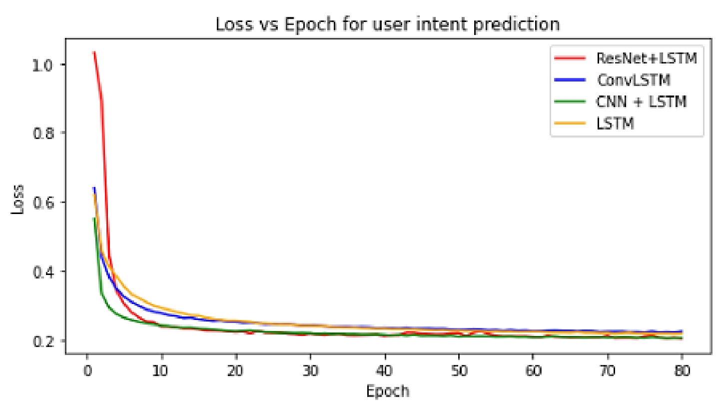


Figure 8.2: Loss versus Epoch depicting the progression of loss values with different values of the epoch during Intent prediction

9. SUMMARY AND CONCLUSIONS

9.1 Conclusion

In this thesis, we provided a framework that uses GPS and wireless sensor data, android applications, Google Cloud Platform (GCP), and Cloud Firestore. The framework is described in detail in Chapter 3. This framework provides a way to collect sensor data from smartphones using an android application. It takes the help of Cloud Firestore to store the raw data and uses GCP services like Bucket and BigQuery to transport and anonymize the data. This chapter also describes the different steps taken for data collection, data storage, data movement, and data anonymization. To collect the data, we developed an android application and maintained user privacy using the data anonymization techniques described. The security during the data transfer between android application and the Firebase database is ensured by Cloud Firestore. In Chapter 4, we provided the necessary steps needed for data processing, which include Places of Interest Determination using the DBSCAN algorithm and reverse geocoding of centroid points of generated clusters.

In Chapter 7, we presented the proposed deep learning model that combines the Residual Network and LSTM, enhancing the user location and intent prediction. The proposed model was compared with existing machine learning and deep learning models utilized in previous research for smartphone-based data, and our results and analysis indicate that the ResNet-LSTM model outperforms its counterparts for the dataset used in the research. It achieved an accuracy of 90.058% for user locale prediction and 90.261% for user intent prediction. ResNet's convolutional operations can extract hidden features from the data, effectively addressing the problem of training accuracy degradation of CNN giving it an edge over the CNN model. Stacking ResNet and LSTM further improves the extraction of hidden features from sequential data, resulting in superior prediction accuracy.

The deep learning models use a shallow network because of the smaller training dataset. Increasing the depth of the models gave rise to the issue of overfitting by the training data, performing worse than their shallow networks. It was also noticed that machine learning models' accuracy had high fluctuations even on re-training with the same hyperparameters, which was not the case in the deep learning models. Also, deep learning is able to extract hidden features efficiently as opposed to machine learning even though it uses less computing power. These findings suggest that our framework and the proposed deep learning model of ResNet-LSTM are promising avenues for future location and intention prediction research.

The proposed framework offers more than just location prediction for user recommendation. It also predicts user intent, providing a deeper understanding of user behavior. This additional information can be used to generate personalized recommendations, enhancing the user experience.

9.2 Challenges

A few challenges occurred during the progress of the research, out of which the major challenges are listed below.

- **Data Availability**: HAR datasets used in earlier research in multiple research papers consisted mainly of location and sensory data such as barometer, gyroscope, magnetometer, and altimeter. Some location datasets used were not available freely. To overcome this challenge, an android application was built that was distributed to multiple random persons for data collection. The data collection happened for two weeks, and the android application was developed in four months.

- **Data Collection**: Another major challenge that arose during the thesis was data collection. Around 90 people were approached for data collection, but only 23 users agreed to participate in the research and allowed us to collect the data. Since the data collection consisted of location and smartphone sensor data, proper permissions were taken from the users. Devices issues related to GPS, Bluetooth, Wi-Fi, and charging ports led to around $50,000$ in data points getting discarded.

- **Data Anonymization**: Anonymization of the collected data was another major challenge as the data collected has sensitive data like username, user email, Wi-Fi name, and location data. To address this, proper data anonymization steps were taken to remove unused features and mask the user data before using it as training data.

- **Data Security**: Apart from anonymization, data security was also a challenge during the data collection and storage. To overcome this, Firebase and GCP are used. These services from Google have the in-built feature of securing the data transmission and encryption in data storage. The Firebase also helped in user authorization and authentication, which restricted the user to view their own data points only. Also, proper steps were taken not to publish the raw data to any kind of public platform.

9.3 Further Study

As part of our future endeavors, we plan to make certain updates to the application. One such update would be to decrease battery usage, as many users have provided feedback regarding this issue. Additionally, we aim to expand our data collection process to incorporate more users, as this will help in obtaining a higher volume of quality datasets, which in turn will enable us to train our models better.

Another potential update we plan to implement is to enable users to create custom intents, which can be used to re-train our models based on those intents. Moving forward, we can also include other smartphone sensor data in our studies to determine their effectiveness in predicting user behavior.

We also plan to use the concept of the Hidden Markov Model and Transformers to process the sequential data and provide predictions based on past events. Finally, we plan to extend our study by providing personalized recommendations to the users based on their location and intent through the application.

REFERENCES

[1] V. Osmani, S. Balasubramaniam, and D. Botvich, "Human activity recognition in pervasive health-care: Supporting efficient remote collaboration," Journal of Network & Computer Applications, Nov. 2008, vol. 31, no. 4, pp. 628–655, doi: 10.1016/j.jnca.2007.11.002.

[2] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. J. Bula and P. Robert, "Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly," in IEEE Transactions on Biomedical Engineering, vol. 50, no. 6, pp. 711-723, June 2003, doi: 10.1109/TBME.2003.812189.

[3] Y. -L. Hsu, S. -C. Yang, H. -C. Chang and H. -C. Lai, "Human Daily and Sport Activity Recognition Using a Wearable Inertial Sensor Network," in IEEE Access, vol. 6, pp. 31715-31728, 2018, doi: 10.1109/ACCESS.2018.2839766.

[4] M. Ermes, J. PÄrkkÄ, J. MÄntyjÄrvi and I. Korhonen, "Detection of Daily Activities and Sports With Wearable Sensors in Controlled and Uncontrolled Conditions," in IEEE Transactions on Information Technology in Biomedicine, vol. 12, no. 1, pp. 20-26, Jan. 2008, doi: 10.1109/TITB.2007.899496.

[5] X. Su, H. Caceres, H. Tong and Q. He, "Online Travel Mode Identification Using Smartphones With Battery Saving Considerations," in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 10, pp. 2921-2934, Oct. 2016, doi: 10.1109/TITS.2016.2530999.

[6] J. Yang, K. Cheng, J. Chen, B. Zhou and Q. Li, "Smartphones based Online Activity Recognition for Indoor Localization using Deep Convolutional Neural Network," 2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS), Wuhan, China, 2018, pp. 1-7, doi: 10.1109/UPINLBS.2018.8559719.

[7] R. Xie, "Exploring Mobile Phone Data for Urban Activity Analysis: Applications from Individual Activity Pattern to Group Activity Regularity," 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communi-

cations, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 2019, pp. 1213-1218, doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00226.

[8] M. Xu and J. Han, "Next Location Recommendation Based on Semantic-Behavior Prediction," In Proceedings of the 5th International Conference on Big Data and Computing (ICBDC '20), Association for Computing Machinery, New York, NY, USA, 2020, pp. 65–73. doi: 10.1145/3404687.3404699.

[9] J. Alvarez-Lozano, J. A. García-Macías, and E. Chávez, "User location forecasting at points of interest," In Proceedings of the 2012 RecSys workshop on Personalizing the local mobile experience (LocalPeMA '12), Association for Computing Machinery, New York, NY, USA, 2012, pp. 7–12. https://doi.org/10.1145/2365946.2365949.

[10] Y. Liu, N. Zhao, W. Han and Q. Zhu, "User Behavior Patterns and Activity Recognition Based on Mobile Big Data," 2021 International Conference on Electronic Information Technology and Smart Agriculture (ICEITSA), Huaihua, China, 2021, pp. 203-209, doi: 10.1109/ICEITSA54226.2021.00048.

[11] J. Li, Y. Guo and Y. Qi, "Using Neural Networks For Indoor Human Activity Recognition with Spatial Location Information," 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 2019, pp. 146-149, doi: 10.1109/IHMSC.2019.10130.

[12] D. Yang, D. Zhang, V. W. Zheng and Z. Yu, "Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 45, no. 1, pp. 129-142, Jan. 2015, doi: 10.1109/TSMC.2014.2327053.

[13] T. N. Sainath, O. Vinyals, A. Senior and H. Sak, "Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks," 2015 IEEE International Conference on Acoustics,

Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 2015, pp. 4580-4584, doi: 10.1109/ICASSP.2015.7178838.

[14] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," Advances in Neural Information Processing Systems, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama and R. Garnett, Eds. 2015, vol. 28, pp. 802-810.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015 ArXiv, doi: 10.48550/arXiv.1512.03385.