IMPROVING FEATURE RELEVANCE ANALYSIS SENSOR-BASED BEHAVIORAL CONTEXT RECOGNITION USING DEEP METRIC CONTEXT LEARNING

A PROJECT REPORT

Submitted by

VISHNUPRASAD V	**211419205184**
VIGNESH RAJ T	**211419205178**
SIDDHARTH K	**211419205156**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

PANIMALAR ENGINEERING COLLEGE,POONAMALLEE

ANNA UNIVERSITY :CHENNAI 600 025

APRIL 2023

BONAFIDE CERTIFICATE

Certified that this project report **IMPROVING FEATURE RELEVANCE ANALYSIS SENSOR-BASED BEHAVIORAL CONTEXT RECOGNITION USING DEEP METRIC CONTEXT LEARNING** is the bonafide work of **VISHNUPRASAD V(21141205184), VIGNESH RAJT(211419205178), SIDDHARTHK(211419205156)**who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
Dr. M. HELDA MERCY M.E., Ph.D.,	**Mr.M.DILLIBABU M.E., Ph.D.,**
PROFESSOR	**SUPERVISOR**
HEAD OF THE DEPARTMENT	**ASSOCIATE PROFESSOR**
Department of Information Technology	Department of Information Technology
Panimalar Engineering College	Panimalar Engineering College
Poonamallee, Chennai - 600 123	Poonamallee, Chennai - 600 123

Submitted for the project and viva-voce examination held on ______________

SIGNATURE	**SIGNATURE**
INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION

I hereby declare that the project report entitled **IMPROVING FEATURE RELEVANCE ANALYSIS SENSOR-BASED BEHAVIORAL CONTEXT RECOGNITION USING DEEP METRIC CONTEXT LEARNING** which is being submitted in partial fulfilment of the requirement of the course leading to the award of the 'Bachelor Of Technology in Information Technology ' in **Panimalar Engineering College, Autonomous institution Affiliated to Anna university-Chennai**is the result of the project carried out by me under the guidance **in the Department of Information Technology**. I further declared that I or any other person has not previously submitted this project report to any other institution/university for any other degree/ diploma or any other person.

VISHNUPRASAD V

VIGNESH RAJT

SIDDHARTH K

Date**:**

Place: Chennai

It is certified that this project has been prepared and submitted under my guidance.

Date: Mr.M.DILLIBABU M.E., Ph.D.,

Place: (ASSOCIATE PROFESSOR/ IT)

ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind co-operation and support from many, for successful completion . We wish to express our sincere thanks to all those who were involved in the completion of this project.

Our sincere thanks to **Our Honourable Secretary and Correspondent, Dr. P. CHINNADURAI, M.A., Ph.D.,** for his sincere endeavor in educating us in his premier institution.

We would like to express our deep gratitude to **Our Dynamic Directors , Mrs. C. VIJAYA RAJESHWARI and Dr. C. SAKTHI KUMAR, M.E.,M.B.A.,Ph.D.,** and **Dr.SARANYASREE SAKTHIKUMAR .,B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities for completion of this project.

We also express our appreciation and gratefulness to **Our Principal Dr. K. MANI, M.E., Ph.D.,** who helped us in the completion of the project. We wish to convey our thanks and gratitude to our head of the department, **Dr. M. HELDA MERCY,M.E., Ph.D.,** Department of Information Technology, for her support and by providing us ample time to complete our project.

We express our indebtedness and gratitude to our Project co-ordinator **Mr. M. DILLI BABU**,**M.E.,(Ph.D.,)** Associate Professor, Department of Information Technology for his guidance throughout the course of our project. We also express sincere thanks to our supervisor **Mr.M.DILLI BABU, ASSOCIATE PROFESSOR** for providing the support to carry out the project successfully. Last, we thank our parents and friends for providing their extensive moral support and encouragement during the course of the project.

TABLE OF CONTENTS

ABSTRACT

Smart devices of everyday use (such as smartphones and wearables) are increasingly integrated with sensors that provide immense amounts of information about a persons daily life such as behavior and context. The automatic and unobtrusive sensing of behavioral context can help develop solutions for assisted living, fitness tracking, sleep monitoring, and several other fields. Towards addressing this issue, we raise the question: can a machine learn to recognize a diverse set of contexts and activities in a real-life through joint learning from raw multi-modal signals (e.g. accelerometer, gyroscope and audio etc.). This work proposes the unified logical-based framework to recognize and analyze behavioral specifications understood as a formal logic language that avoids ambiguity typical for natural languages. Automatically discovering behaviors from sensory data streams as formal specifications is of fundamental importance to build seamless human-computer interactions. Thus, the knowledge about environment behaviors expressed in terms of temporal logic formulas constitutes a base for the reactive and precise reasoning processes to support trustworthy, unambiguous and pro-active decisions for applications that are smart and context-aware.

LIST OF FIGURES

LIST OF ABBREVIATION

LPMT - Location Privacy-Preserving mechanism based on Trajectory Obfuscation

HAR - Human Activity Recognition

FGA - Fine-Grained Activities

LSTM - Long Short-Term Memory

ANN - Artifical Neural Network

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THIS PROJECT:

Nowadays smart spaces are filled with different sensors and sensor-like equipments. A sensor is a device that detects events or changes from a physical environment, that is a devise which is sensitive to a physical stimulus. These sensors might constitute the IoT spaces (Internet of Things) in which objects with unique identifiers create their own scenarios and interactions. On the other hand, the decisive feature of smart spaces is context-awareness which stands for the capabilities to examine changes in the environment and to react to these changes adequately. Important aspects of context might be: where you are, who you are with, and what resources are nearby.

In other words, context is ...any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. Logic has simple, unambiguous syntax and semantics. It is thus ideally suited to the task of specifying information systems showing the form of an argument to be valid or invalid. Knowledge about arguments enable achieving clear thinking and relevant arguments. Sensing and context understanding are necessary and of critical importance to pro-active decisions which should be interpreted into domainrelevant concepts and situations.

1.2 NEED OF THIS PROJECT:

The problem of context recognition is centered on inferring persons environment, physical state, and activity performed at any particular time. Specifically, an understanding of the users current context requires determining where and with whom the person is? and in what type of activity the person is involved in? The behavioral and activity analysis is an important and challenging task mainly because it is crucial for several applications, including smart homes, assisted living, fitness tracking, sleep monitoring, user-adaptive services, social interaction and in industry. In particular, an accurate recognition of human context can greatly benefit healthcare and wellbeing through automatic monitoring and supervision of patients with chronic diseases such as hypertension, diabetes and dementia.

Furthermore, the gathered knowledge and extracted activity patterns can enable novel treatment design, adjustment of medications, better behavioral intervention and patient observation strategies In practice, for a context detection system to be effective in a real-life requires an unobtrusive monitoring. It is important to not distress a person in order to capture their realistic behaviors in a natural environment. The penetration of smart sensing devices (e.g. smartphones and wearables) that are integrated with sophisticated sensors in our daily lives provides a great opportunity to learn and infer about various aspects of a persons daily life. However, there is considerable variability in the human behavior in real-world situations that can cause the system to fail, if it is developed using data collected in a constrained environment.

1.3 OBJECTIVE OF THIS PROJECT:

The objective of this project is to improve feature relevance analysis and sensor-based behavioral context recognition using deep metric context learning. The project aims to develop an algorithm that can automatically and unobtrusively sense and recognize a diverse set of contexts and activities in a person's daily life using raw multi-modal signals (such as accelerometer, gyroscope, and audio data).

The algorithm is designed to learn from the raw signals and identify patterns of behavior, such as a person's normal routines, sleeping patterns, and mobility. It can also detect deviations from these patterns, such as prolonged inactivity or unusual movements, which can be potential indicators of health issues. The algorithm can be applied in various domains such as assisted living, fitness tracking, sleep monitoring, and healthcare.

The ultimate goal of this project is to develop a system that can improve the quality of life for individuals by providing insights into their daily routines and habits, detecting potential health issues, and alerting caregivers or healthcare providers when necessary. By recognizing these behaviors, the algorithm can provide insights into a person's daily routines and habits, which can be useful for a variety of applications such as assisted living, fitness tracking, sleep monitoring, and healthcare. For example, the algorithm can detect prolonged periods of inactivity or unusual movements that may indicate a potential health issue, and alert caregivers or healthcare providers.

1.4 SCOPE OF THE PROJECT:

The scope of the "Improving Feature Relevance Analysis Sensor-based Behavioral Context Recognition using Deep Metric Context Learning" project is focused on developing a deep learning model that can accurately recognize different behavioral contexts based on sensor data. Specifically, the project aims to improve the feature relevance analysis process by using a deep metric context learning approach.

The project's scope includes:

- Data collection and preprocessing: Collecting and preprocessing sensor data, such as accelerometer and gyroscope readings, to create a dataset suitable for training a deep learning model.
- Feature extraction and selection: Extracting and selecting relevant features from the sensor data that are most informative for recognizing different behavioral contexts.
- Deep metric context learning: Developing and training a deep metric context learning model that can effectively learn the feature space and accurately classify different behavioral contexts.
- Model evaluation: Evaluating the performance of the deep metric context learning model on the collected dataset and comparing it with other state-of-the-art methods.

CHAPTER 2

LITERATURE SURVEY

2.1 C2FHAR: Coarse-to-Fine Human Activity Recognition With Behavioral Context Modeling Using Smart Inertial Sensors:

AUTHOR: Muhammad Ehatisham-Ul-Haq , Muhammad Awais Azam , Yasar Amin and Usman Naeem

YEAR: 2020

In this paper, a novel multi-label classication scheme is presented for sensor-based human activity recognition, which models human contexts (such as users location, secondary activity, and phone position) with the daily living activities to obtain a coarse-to-ne representation of human activities in-the-wild. For this purpose, a set of 29 ne-grained human activities is obtained by adding context information with six primary activities (including lying, sitting, stand- ing, running, bicycling, and running). Decision Tree (DT), Random Forest (RF), and Neural Networks (NN) are utilized to recognize six context-independent activities as well as 29 context-dependent activities based on the inertial sensor data extracted from smartphone and smartwatch. A public domain ExtraSensory dataset is used for validating the pro- posed scheme, which shows that RF performs better than DT and NN classiers in recognizing both types of activities. The performance of context-dependent activity recognition is better, which state the efcacy of behavioral context modeling with the primary activities of daily living.

2.2 Context Awareness-Based Accident Prevention During Mobile Phone Use:

AUTHOR: Bowen Sun , Quanlong Li , Yonghui Guo and Guocheng Li.

YEAR: 2020

This paper proposes a method and system for preventing acci- dental injuries during the use of mobile phones. The research is based on context-awareness, and we dene the dangerous situation that mobile phone holders may come into when using the mobile phone from three aspects: human behavior, spatial location, and the interaction with the mobile phone. The paper also proposes the research methods of dangerous situation recognition and the dangerous situation recognition model. Based on this model, this paper implements a dan- gerous situation recognition and prevention system (Android application). For the study of dangerous situation identica- tion and prevention, the follow-up work will be improved based on the following points. For the formal denition of dangerous situations, we will try to identify dangerous sit- uations from more angles to further increase the accuracy of the dangerous situation identication results. The interactive awareness subproblem refers to how the mobile phone holder interacts with the mobile phone to determine whether the mobile phone holder is in the process of using the mobile phone. The study of accident prevention methods during the use of mobile phones will be based on the output of the above subproblems. After designing and implementing the solutions toward the three subproblems, algorithms are integrated, and the application is designed.

2.3 On-Bird Sound Recordings: Automatic Acoustic Recognition of Activities and Contexts:

AUTHOR: Dan Stowell , Emmanouil Benetos and Lisa F. Gill

YEAR:2017

We introduce a novel approach to studying animal behavior and the context in which it occurs, through the use of microphone backpacks carried on the backs of individual free-flying birds. These sensors are increasingly used by animal behavior researchers to study individual vocalizations of freely behaving animals, even in the field. However, such devices may record more than an animals vocal behavior, and have the potential to be used for investigating specific activities (movement) and context (background) within which vocalizations occur. To facilitate this approach, we investigate the automatic annotation of such recordings through two different sound scene analysis paradigms: A scene-classification method using feature learning, and an event-detection method using probabilistic latent component analysis. We analyze recordings made with Eurasian jackdaws (Corvus monedula) in both captive and field settings. Results are comparable with the state of the art in sound scene analysis; we find that the current recognition quality level enables scalable automatic annotation of audio logger data, given partial annotation, but also find that individual differences between animals and/or their backpacks limit the generalization from one individual to another. we consider the interrelation of scenes and events in this particular task, and issues of temporal resolution.

2.4 Effects of feature selection on lane-change maneuver recognition: An analysis of naturalistic driving data

AUTHOR: Xiaohan Li , Wenshuo Wang , Zhang Zhang and Matthias Rtting

YEAR:2018

Purpose - Feature selection is crucial for machine learning to recognize lane-change (LC) maneuver as there exist a large number of feature candidates. Blindly using feature could take up large storage and excessive computation time, while insufficient feature selection would cause poor performance. Selecting high contributive features to classify LC and lane-keep behavior is effective for maneuver recognition. This paper aims to propose a feature selection method from a statistical view based on an analysis from naturalistic driving data. Design/methodology/approach - In total, 1,375 LC cases are analyzed. To comprehensively select features, the authors extract the feature candidates from both time and frequency domains with various LC scenarios segmented by an occupancy schedule grid. Then the effect size (Cohens d) and p-value of every feature are computed to assess their contribution for each scenario. Findings - It has been found that the common lateral features, e.g. yaw rate, lateral acceleration and timeto-lane crossing, are not strong features for recognition of LC maneuver as empirical knowledge. Finally, cross-validation tests are conducted to evaluate model performance using metrics of receiver operating characteristic. Experimental results show that the selected features can achieve better recognition performance than using all the features without purification.

2.5 A Two-Layer Risky Driver Recognition Model With Context Awareness:

AUTHOR: Ke Wang and Jian John Lu

YEAR:2021

This paper proposes two risky driver recognition structures: Two-layer Risky-driver-recognition with Context-awareness (TRC) and Probability-based Two-layer Risky-driver- recognition with Contextawareness (PTRC). Using vehicle trajectory data, we are able to demonstrate the impact of trafc state on driving behavior parameters and risky driving. Therefore, as an essential type of context information, trafc state should be considered in the risky driver recognition model. We conrm that TRC and PTRC are signicantly better than the single-layer structure by achieving higher AUPRC and F1 scores through the statistical test. PTRCs performance can be further increased after probability calibration. This research has a few limitations. The performance improvement of proposed structures is moderate. Future improvement could be achieved by considering more trafc states. However, the optimal number of trafc states and trafc states classication problems are not discussed in this paper. The performance evaluation of the proposed structure uses a single source dataset recorded on a German highway.

2.6 Improving Reliability: User Authentication on Smartphones Using Keystroke Biometrics:

AUTHOR: Yuhua Wang , Chunhua Wu , Kangfeng Zheng and Xiujuan Wang

YEAR:2019

Keystroke biometrics is a well-investigated dynamic behavioral methodology that utilizes the unique behavioral patterns of users to verify their identity when tapping keys. However, the performance of keystroke biometrics is unreliable due to its high error rate and low robustness. In this paper, we propose differential evolution and adversarial noise-based user authentication (DEANUA), which is a verification scheme for enhancing reliability by reducing the error rate and improving robustness. We investigate the current mainstream features and build a more comprehensive feature set that composed of 146 features. Then, we use a differential evolution method to select an optimized feature set. With the support vector regression method on this feature set, we achieve an equal error rate (EER) of 0.12660% and also a 31.25% energy consumption reduction rate. In this paper, the model is trained with the training samples collected from one situation, but the model is used in various situations. Thus, the robustness of the model is inadequate. We constructed the adversarial noise samples to simulate users behavioral characteristics in different situational contexts. We use the adversarial noise samples to test the models in a strict experimental environment, which raises the EER by 83.59%, to 10.9299%. Then, we enhance the model with adversarial noise samples to obtain an EER of 8.70932%, which is a reduction of 20.32%.

2.7 Empirical Evidence of Upward Social Comparison in a Prisoners Dilemma Game:

AUTHOR: Matthieu Nadini , Peerayos Pongsachai , Chiara Spinello

YEAR:2020

A large body of work has offered compelling evidence of the influence of social context on individual decision-making, but the reasons why individuals tend to cooperate with others remain elusive. The prisoners dilemma constitutes a powerful, yet elementary, social game to study the drivers underlying cooperation. Here, we empirically examined a prisoners dilemma game where small groups of participants played with controlled, virtual players over a series of rounds. Toward investigating how individual decisions on cooperation are influenced by others, the virtual players were engineered so that they would have a higher cumulative score than some participants and a lower cumulative score than others. Our results corroborate upward social comparison theory, whereby only participants who had a lower cumulative score than cooperating virtual players displayed an increased tendency to cooperate. Overall, our experimental findings indicate that the players cumulative score plays a critical role within the prisoners dilemma game, thereby offering a mean for increasing cooperation. For practitioners, this finding sheds light on how players cumulative score alone modulates decision-making processes toward choices that are suboptimal for the individual, but optimal for the entire group.

2.8 Motivation mechanism of gamification in crowdsourcing projects:

Author: Zhiyuan Zeng , Jian Tang and Tianmei Wang

Year:2017

The purpose of this paper is to study the participation behaviors in the context of crowdsourcing projects from the perspective of gamification. Design/methodology/approach This paper first proposed a model to depict the effect of four categories of game elements on three types of motivation based upon several motivation theories, which may, in turn, influence user participation. Then, 5×2 between-subject Web experiments were designed for collecting data and validating this model. Findings Game elements which provide participants with rewards and recognitions or remind participants of the completion progress of their tasks may positively influence the extrinsic motivation, whereas game elements which can help create a fantasy scene may strengthen intrinsic motivation. Besides, recognition-kind and progress-kind game elements may trigger the internalization of extrinsic motivation. In addition, when a task is of high complexity, the effects from game elements on extrinsic motivation and intrinsic motivation will be less prominent, whereas the internalization of extrinsic motivation may benefit from the increase of task complexity. Originality/value This study may uncover the motivation mechanism of several different kinds of game elements, which may help to find which game elements are more effective in enhancing engagement and participation in crowdsourcing projects.

2.9 Demographic Bias in Biometrics: A Survey on an Emerging Challenge

Author: Pawel Drozdowski , Christian Rathgeb , Antitza Dantcheva

Year:2020

Systems incorporating biometric technologies have become ubiquitous in personal, commercial, and governmental identity management applications. Both cooperative (e.g., access control) and noncooperative (e.g., surveillance and forensics) systems have benefited from biometrics. Such systems rely on the uniqueness of certain biological or behavioral characteristics of human beings, which enable for individuals to be reliably recognized using automated algorithms. Recently, however, there has been a wave of public and academic concerns regarding the existence of systemic bias in automated decision systems (including biometrics). Most prominently, face recognition algorithms have often been labeled as racist or biased by the media, nongovernmental organizations, and researchers alike. The main contributions of this article are: 1) an overview of the topic of algorithmic bias in the context of biometrics; 2) a comprehensive survey of the existing literature on biometric bias estimation and mitigation; 3) a discussion of the pertinent technical and social matters; and 4) an outline of the remaining challenges and future work items, both from technological and social points of view.

2.10 Targeted Aspect-Based Multimodal Sentiment Analysis: An Attention Capsule Extraction and MultiHead Fusion Network:

Author: Donghong Gu , Jiaqian Wang , Shaohua Cai

Year:2021

Multimodal sentiment analysis has currently identified its significance in a variety of domains. For the purpose of sentiment analysis, different aspects of distinguishing modalities, which correspond to one target, are processed and analyzed. In this work, the researchers propose the targeted aspect-based multimodal sentiment analysis (TABMSA) for the first time. Furthermore, an attention capsule extraction and multi-head fusion network (EF-Net) on the task of TABMSA is devised. The multi-head attention (MHA) based network and the ResNet-152 are employed to deal with texts and images, respectively. The integration of MHA and capsule network aims to capture the interaction among the multimodal inputs. In addition to the targeted aspect, the information from the context and the image is also incorporated for sentiment delivered. The researchers evaluate the proposed model on two manually annotated datasets. the experimental results demonstrate the effectiveness of our proposed model for this new task.

2.11 FEASIBILITY

- **Social feasibility:** The project is highly socially feasible as it aims to provide assistance to activities of humans, which is a prevalent and concerning health issue. The application's features, such as Sensor-based behavioral context, are designed to enhance the safety and security of patients, and the humans recognizesand help to improve their daily lives. As a result, the project can have a positive impact on the human activities.
- **Technical feasibility:** The project appears technically feasible as it utilizes common and well-established technologies such as Python, Jupyter Notebook, Android. These technologies have been used extensively in the development of mobile applications and deep learning. The use of TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural, and the development team has experience in implementing them. However, some challenges may arise when it comes to integrating different components, such as the sensor-based behavioral contextmodules, and ensuring that they work together seamlessly.
- **Economic feasibility:** The project is economically feasible as it utilizes mostly free and open-source technologies, such as Jupyter Notebook and Android Studio, which are readily available for development. However, these are freeand can be managed effectively by using efficient coding practices and optimizing applications. Furthermore, the potential benefits of the application, such as improving human activity recognition and use to monitoring the human activityto monitor the sleep, health care tracking, etc.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 EXISTING SYSTEM:

In mobile crowdsensing activities, it is usually necessary for participants to upload sensing data and related locations. The existing location privacy-preserving mechanisms cannot well protect a users trajectory privacy because attackers can mine the users trajectory features through data analysis techniques.Aiming at the trajectory privacy protection problem, this article proposes a differential location privacy-preserving mechanism based on trajectory obfuscation (LPMT).In the trajectory feature extraction phase, LPMT extracts the stay points as the feature of the users trajectory and then generates obfuscation regions and subregions. A stay point is a spatiotemporal point that may leak the users privacy information.A characteristic point of trajectories where a user stays on a specific location for a timespan greater than or equal to a time threshold.LPMT first extracts the stay points as the features of a trajectory based on the sliding window algorithm, and then obfuscates each stay point to a target obfuscation subregion through the exponential mechanism, and finally performs the Laplace sampling in the target obfuscation subregion to obtain the obfuscated GPS points.Compared with the baseline mechanisms, LPMT can reduce data quality loss bymore than 20% while providing the same level of obfuscation quality, which indicates that LPMT has the advantages of strong security and high quality of service. In the stay points obfuscation phase, the location context similarity data is pre-downloaded from the data center, The existing system can focus on SPOA. SPOA traverses each stay point, and selects a target obfuscation subregion for each stay point through the exponential mechanism.

Drawbacks:

- They are not precise when applied to high-frequency decomposition.
- Method requires a lot of time to extract characteristics.
- Likely to lose coupling information, causing low accuracy and weak generalization.
- May not solve the inherent problem that leads to bias of the model toward the majority class.
- Will degrade because of overfitting.

3.2 PROPOSED SYSTEM :

This study focuses on human activity recognition based on inertial sensor data collected using smartphone sensors. There are already a lot of wellness wearables in the market which rely on sensor data. Most of them are using static pre-build algorithms, and thus, they are not able to adapt to unseen situations. In fact, the most common way to build a machine learning prediction model is to rely on data that are given prior to training the model. The problem of this approach is that it assumes that structure of the data remains static. However, this is not the case in the real-world problems as the world around us constantly changes.In this aspect, we performed a systematic analysis of the ExtraSensory dataset to realize the co-occurrences of the selected ADLs with different behavioral contexts (such as users secondary activity/ location/body state) and phone positions. Following this, for each users data, we calculated the frequencies of different behavioral contexts and phone positions that occur in pairs with each primary activity.However, this is not the case in the real-world problems as the world around us constantly changes.

Moreover, while pre-build algorithms provide high recognition rates on most of the people, but not for all. Due to this, the recognition should be based on adaptive personalized models and not on static user-independent models.The effect of behavioral context modeling on HAR performance is investigated and discussed in detail based on context-independent and context-dependent HAR experiments.These context labels provide auxiliary information about human behavioral contexts when executing a particular primary activity inthe-wild. Owing to an in-the-wild collection of the dataset, the contexts labels relating to each primary activity are not consistent for all the users. Consequently, the selection and annotation of fine-grained activities, i.e., primary ADLs incorporated with behavioral context information, is not a straightforward task.Following this, for each users data, we calculated the frequencies of different behavioral contexts and phone positions that occur in pairs with each primary activity.

Advantages :

- Extract non-linear features and has good generalizing abilities.
- Does not require complex algorithms to process data.
- Adds complexity and increases the inference overhead as input sequences grow longer.
- It can reduce the computation time, improve model prediction performance
- It helps to get a better understanding of the dataset.

3.3 ARCHITECTURE DIAGRAM:

In this architecture, the system first collects sensor data from various sources, such as accelerometers, gyroscopes, or other environmental sensors. This data is then preprocessed to clean, filter, and transform it into a format that can be used by the machine learning model. Next, the system uses feature selection techniques such as statistical analysis or machine learning algorithms to identify a set of relevant features that can be used to predict different behavioral contexts accurately.The system then uses deep metric context learning to train a machine learning model that can learn a metric space where similar instances are closer together and dissimilar instances are further apart. The model is trained using the relevant features identified in the previous step.

Finally, the trained machine learning model is used to predict the behavioral context based on the sensor data. The predicted behavioral context can then be used for a variety of applications, such as improving health monitoring, enhancing security systems, or optimizing energy consumption in smart homes. Overall, this architecture diagram illustrates a high-level overview of the system components involved in using deep metric context learning for sensor-based behavioral context recognition.

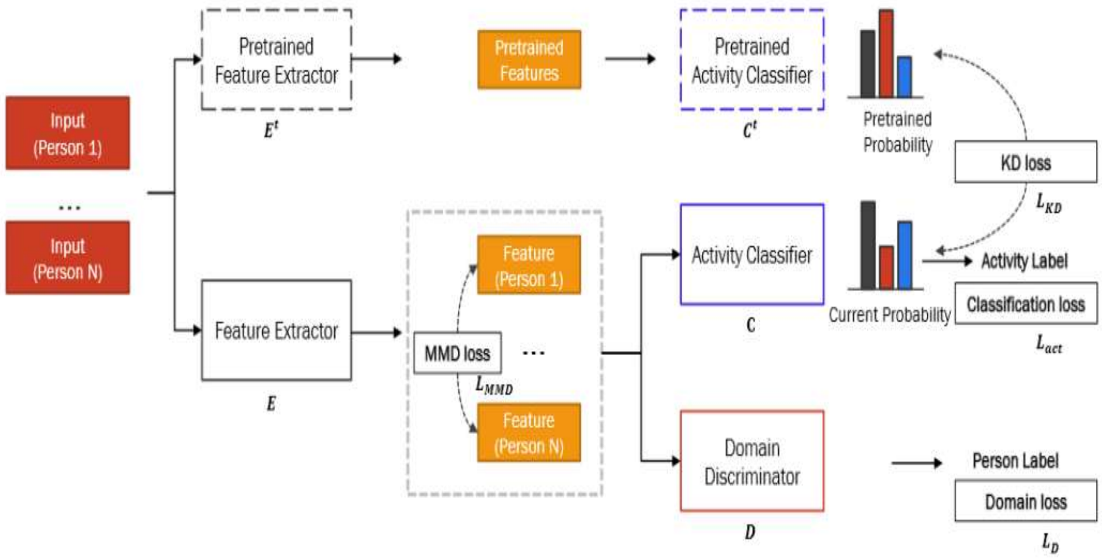
Input (Person 1)
...
Input (Person N)
Pretrained Feature Extractor
E^t
Pretrained Features
Pretrained Activity Classifier
C^t
Pretrained Probability
KD loss
L_{KD}
Feature Extractor
E
Feature (Person 1)
MMD loss
L_{MMD}
...
Feature (Person N)
Activity Classifier
C
Current Probability
Activity Label
Classification loss
L_{act}
Domain Discriminator
D
Person Label
Domain loss
L_D

Fig no 3.1

CHAPTER 4

REQUIREMENTS AND ITS SPECIFICATION

4 .HARWARE AND SOFTWARE REQUIREMENTS:

Hardware Requirements :

- Processor :Minimum i3 Dual Core
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 100 GB; Recommended 200 GB or more
- Memory (RAM): Minimum 8 GB; Recommended 32 GB or above

Software Requirements :

- Python
- Anaconda
- Jupyter Notebook
- TensorFlow
- Keras

4.1 HARDWARE REQUIREMENTS DESCRIPTION

4.1.1 Hard Drive:

Android provides a number of methods for data storage depending on the needs of the user, developer, and application. For example, some apps use data storage **to keep track of user settings or user-provided data**. At least 250GB of free disk space to check out the code and an extra 150 GB to build it. he hard disk is a secondary storage device, which is designed to store data permanently. The secondary storage devices include a large storage capacity as compared to the primary storage devices. The data stored in a hard disk is retained when our computer system shuts down. The data stored in the hard disk can be of many types such as the operating system, installed software, documents, and other files of computer.

Hard disk was introduced in the year 1956 by IBM. The first personal computer contains a hard drive of less than 1 megabyte, while modern computers contain a hard drive of 1 terabyte.

Fig no 4.1

4.1.2 Ram Space:

Ram are temporary storage so they are used when an app needs memory to execute it when you run it… Ram pass the location of the file to CPU processor, so that it can be solved and you get the display of the calculation. Calculation can be anything from graphics addition, multiple etc. It is dependent upon the type of application that how much it consumes the app. A laptop with 4GB of RAM should suffice. However, application or software developers who need to run virtual machines, emulators and IDEs to compile massive projects will need more RAM. A laptop with **at least 8GB of RAM is ideal**. The requirement goes even higher for game developers.

RAM stands for Random Access Memory. It's a short-term storage medium that holds programs and processes currently running on your computer.The more RAM that's in your machine, the more programs you can run at once without negatively affecting performance. When your computer runs low on RAM, it uses a part of the storage drive called the page file, which acts as pretend RAM. This is much slower than actual RAM, which is why you notice slowdowns when Windows has to use it.

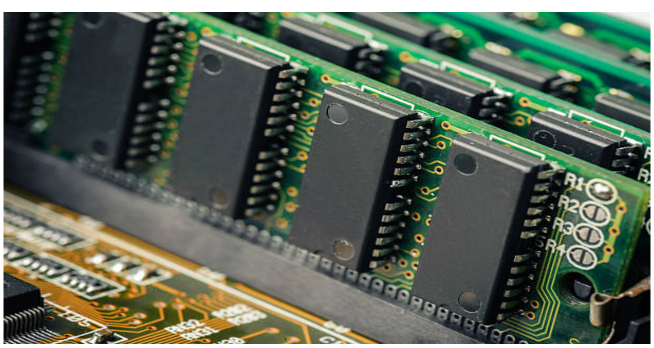

Fig no 4.2

4.1.3 Wireless Adapter:

A WiFi adapter **allows your wired device to pick up WiFi signals**. So if you have a desktop computer, you can connect it to a network wirelessly. A WiFi adapter can also serve as a hardware upgrade to your laptop, strengthening its wireless reception or updating it to the latest WiFi generation. A wireless adapter is a hardware device that is attached to a computer or laptop and allows it to connect to a wireless network. Typically, they come in the form of a USB dongle device that you input into your computer. There are two main types of wireless adapters, based on the network type they help you connect to:

Wi-Fi adapters: they help you connect to Wi-Fi networks nearby.

Cellular / mobile broadband adapters: they connect to 3G or 4G / LTE cellular networks.

A wireless adapter is a hardware device that is attached to a computer or laptop and allows it to connect to a wireless network. Typically, they come in the form of a USB dongle device that you input into your computer.

Fig no 4.3

4.1.4 Processor:

Intel sells its best processors under the Core brand. Core processors come in three categories of performance: i3, i5, and i7. The Core i3 processors round out the low end of the Core series. They offer robust performance but have limited support for high-end applications and technologies. The processor of a computer, or CPU, acts as its brain and allows it to perform calculations and other functions associated with any programming on the computer. The processor turns the information entered into a binary code consisting of zeros and ones. Once converted, this information goes to the CPU, which uses its Arithmetical Logical Unit, or ALU, to perform any mathematical or logical operations.

In addition, a CPU needs other systems within the computer housing to help it remain functional. One of the biggest enemies of a CPU is the heat generated from performing all the functions associated with running a computer. Most computers use a fan to cool the processor, though some high-end gaming computers use cooling systems to help remove excess heat from the CPU.

Fig no 4.4

4.2 SOFTWARE REQUIREMENTS DESCRIPTION:

4.2.1 Anaconda:

Anaconda is an open-source distribution of the Python and R programming languages for data science that aims to simplify package management and deployment. Package versions in Anaconda are managed by the package management system, conda, which analyzes the current environment before executing an installation to avoid disrupting other frameworks and packages.The Anaconda distribution comes with over 250 packages automatically installed. Over 7500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI (graphical user interface), Anaconda Navigator, as a graphical alternative to the command line interface. Anaconda Navigator is included in the Anaconda distribution, and allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages, install them in an environment, run the packages and update them.

Fig no 4.5

4.2.2Juypter notebook and Python:

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Its uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents.

The “notebook” term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. According to the official website of Jupyter, Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

Fig no 4.6

4.3.3 Tensorflow and keras:

TensorFlow is an open-sourced end-to-end platform, a library for multiple machine learning tasks, while Keras is a high-level neural network library that runs on top of TensorFlow. Both provide high-level APIs used for easily building and training models, but Keras is more user-friendly because it's built-in Python.

Keras is a high-level API to build and train neural network models. Keras does not depend on TensorFlow, and vice versa . Keras can use TensorFlow as its backend. TensorFlow is an open-source library developed by Google primarily for deep learning applications. It also supports traditional machine learning. TensorFlow was originally developed for large numerical computations without keeping deep learning in mind.

Fig no 4.7

CHAPTER 5

IMPLEMENTATION:

5.1 FLOW CHART:

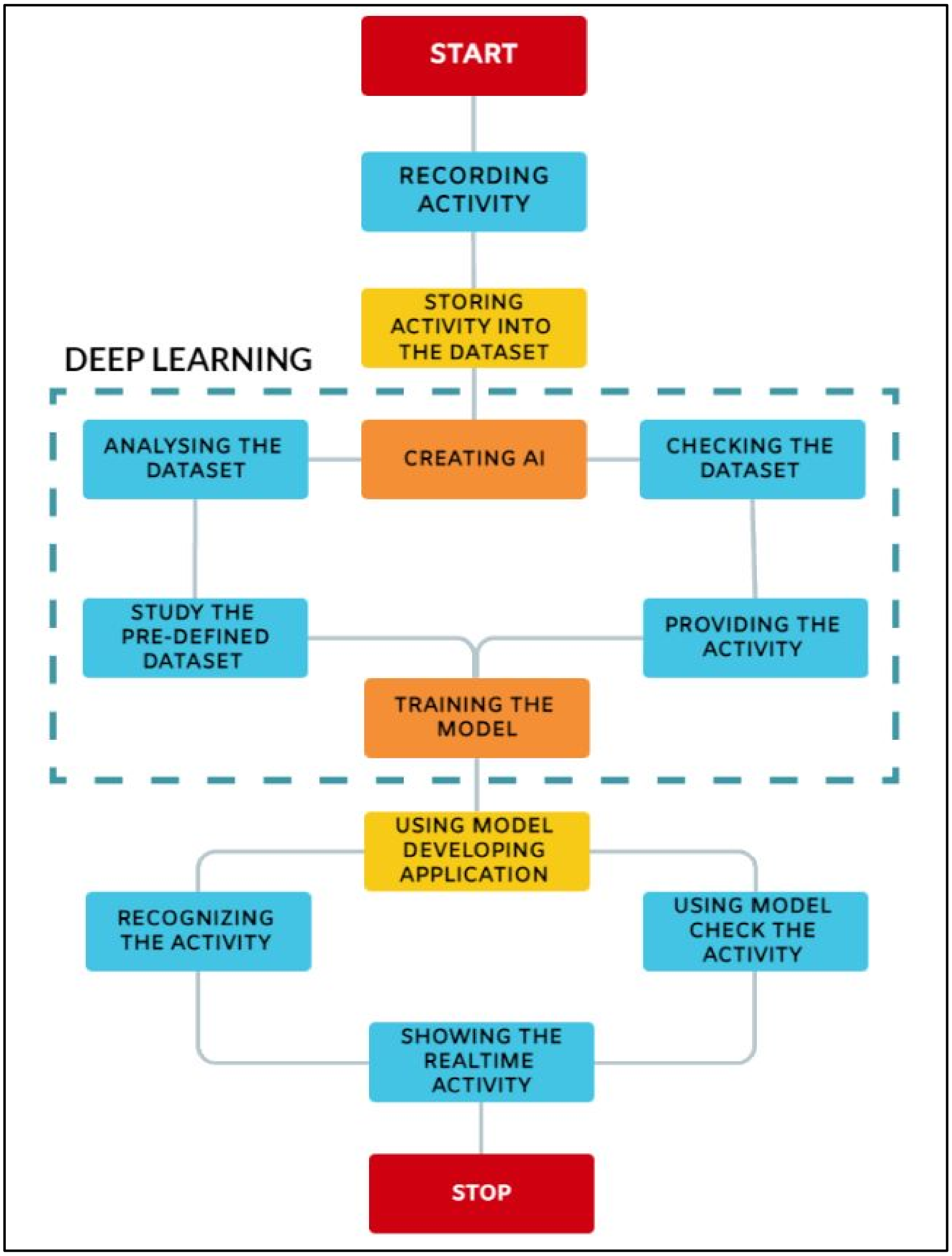

Fig no 5.1

5.2 PROJECT MODULES:

Module 1 : Data Preprocessing

Initially, data is analyzed in two steps: data acquisition and data integration.Firstly, data is acquired from the real world. The data may be incomplete, noisy or redundant. Therefore, data cleansing is used for removing redundancy in data to make it consistent.Missing values are filled and smoothing of data is done using clustering, binning and regression. Secondly, data taken from multiple sources such as files, databases, etc., is initially integrated into a single source.Afterwards, data is reduced and transformed in such a way that it does not alter its identity and becomes useful for further processing. Inconsistent and noisy data is removed in preprocessing, which plays a very important role in data analysis.

Module 2 : Featuring processing

Feature extraction generally computes important signal attributes from the preprocessed data, which are helpful for resolving the classification problem in hand. However, the selection and choice of features to be extracted is crucial as it directly influences the classification results.For this purpose, we conducted an empirical analysis of different sets of hand-crafted features used in the existing HAR studies using a supervised correlation-based feature subset selection (CfsSubetSel) approach . We initially extracted a set of twenty-five (25) time-domain features from the raw sensor data.These features include maximum and minimum amplitudes, mean, variance, standard deviation, kurtosis, skewness, peak-to-peak amplitude and time, peak-to-peak slope, minimum and maximum latencies, latency-amplitude ratio, signal percentiles (i.e., 25th, 50th, and 75th), energy and normalized energy, mean and normalized mean of first and second difference of the signal, and signal entropy.

Module 3: Model Training and Behavioral Context Recognition:

Deep Metric Context Learning Neural Network models usually have three layers: input, hidden, and output . The layers comprise interconnected neurons with nonlinear switching activation functions to enhance nonlinear capacity. First, the input layer gets the data, then passes it to a hidden layer for analysis and returns the results to the output layer. Results shows are now displayed through the output layer. However, given the constraints, training an Deep Metric Context Learning Neural Network will likely require long informal chains of computing processes.

There are three dense layers and two dropout levels in the ANN structure used in this study. The Deep Metric Context Learning Neural Network , on the other hand, is made up of five dense layers and three dropout layers.

5.3 SAMPLE CODING

CODE:

```
import warnings
warnings.filterwarnings('ignore')
from IPython.display import display, HTML
display(HTML("""
<style>
.messagebox{
border-radius: 2px;
padding: 1.25em 1.5em;
border: 1px solid;
}
.messagelightgreen{
border-color: hsl(164deg 95% 38%);
color: rgb(5 139 102);
background-color: rgb(236 255 250);
}
.messagelightgreen b{
color:rgb(139 77 5);
}
.messagebrown{
border-color: hsl(35deg 96% 62%);
color: rgb(143 84 4);
background-color: rgb(255 245 234);
```

```
}
.messagebrown b{
color: rgb(5 139 102);
}
</style>"""))
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
plt.rcParams['font.size'] = 14
plt.rcParams['axes.labelsize'] = 12
plt.rcParams['axes.titlesize'] = 12
plt.rcParams['xtick.labelsize'] = 12
plt.rcParams['ytick.labelsize'] = 12
plt.rcParams['legend.fontsize'] = 12
plt.rcParams['figure.titlesize'] = 14
plt.rcParams['figure.figsize'] = (20,8)
plt.rcParams["ps.useafm"] = True
plt.rcParams['font.family'] = 'Dejavu Sans'
import seaborn as sns
import plotly.express as px
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.linear_model import LinearRegression
```

```
from sklearn.multiclass import OneVsRestClassifier
from skmultilearn.problem_transform import BinaryRelevance
from sklearn.naive_bayes import GaussianNB
from skmultilearn.problem_transform import ClassifierChain
from sklearn.metrics import accuracy_score, mean_absolute_error, ␣
↪mean_squared_error, confusion_matrix
from sklearn.preprocessing import MinMaxScaler
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
df = pd.read_csv("Dataset/train.csv")
display(HTML("
```

No of Rows Available in ␣ ↪Dataset {0}

```
".format(df1.shape[0]))) display(HTML("
```

No of Columns Available ␣ ↪in Dataset {0}

```
".format(df1.shape[1])))
df.info()
test_df = pd.read_csv("Dataset/test.csv")
test_df.sample(10).style.background_gradient(cmap='inferno')
duplicate_count_in_train = df.duplicated().sum()
duplicate_count_in_test = test_df.duplicated().sum()
null_count_in_train = df.isnull().values.sum()
null_count_in_train
null_count_in_test = test_df.isnull().values.sum()
plt.figure(figsize=(16,8))
plt.title('Data provided by each user', fontsize=20)
```

```
sns.countplot(x='subject',hue='Activity', data = df)
plt.show()plt.figure(figsize=(16,8))
plt.title('Data provided by each user', fontsize=20)
sns.countplot(x='subject',hue='Activity', data = df)
plt.show()
plt.title('No of Datapoints per Activity', fontsize=15)
sns.countplot(df.Activity)
 plt.xticks(rotation=90)
plt.show()
columns = df.columns
columns = columns.str.replace('[()]','')
columns = columns.str.replace('[-]', '')
columns = columns.str.replace('[,]','') 5
df.columns = columns test_
df.columns = columns
 test_df.columns
sns.set_palette("Set1", desat=0.80)
facetgrid = sns.FacetGrid(df, hue='Activity', size=6,aspect=2)
facetgrid.map(sns.distplot,'tBodyAccMagmean', hist=False)\ .add_legend()
plt.annotate("Stationary Activities", xy=(-0.956,17),
xytext=(-0.9, 23),␣ ↪size=20,\ va='center', ha='left',\
arrowprops=dict(arrowstyle="simple",connectionstyle="arc3,rad=0.1"))
plt.annotate("Moving Activities", xy=(0,3), xytext=(0.2, 9), size=20,\ va='center', ha='left',\
arrowprops=dict(arrowstyle="simple",connectionstyle="arc3,rad=0.1"))
```

```
plt.show()
df1 = df[df['Activity']==1]
df2 = df[df['Activity']==2]
df3 = df[df['Activity']==3]
df4 = df[df['Activity']==4]
df5 = df[df['Activity']==5]
 df6 = df[df['Activity']==6]
plt.figure(figsize=(7,7))
sns.boxplot(x='Activity', y='tBodyAccMagmean',data=df, showfliers=False, ␣
↪saturation=1) plt.ylabel('Acceleration Magnitude mean')
plt.axhline(y=-0.7, xmin=0.1, xmax=0.9,dashes=(5,5), c='g')
plt.axhline(y=-0.05, xmin=0.4, dashes=(5,5), c='m')
plt.xticks(rotation=90)
plt.show()
sns.boxplot(x='Activity', y='angleXgravityMean', data=df)
plt.axhline(y=0.08, xmin=0.1, xmax=0.9,c='m',dashes=(5,3))
plt.title('Angle between X-axis and Gravity_mean', fontsize=15)
plt.xticks(rotation = 40)
plt.show()
X_pre_tsne = df.drop(['subject', 'Activity'], axis=1)
 y_pre_tsne = df['Activity']
perform_tsne(X_data = X_pre_tsne,y_data=y_pre_tsne, perplexities =[5,10,20])
y_train = df.Activity
X_train = df.drop(['subject','Activity'], axis = 1)
X_test = test_df.drop(['subject','Activity'], axis = 1)
```

```
y_train = le.fit_transform(y_train)

y_test = le.fit_transform(y_test)

scaling_data = MinMaxScaler()

rf_classifier = RandomForestClassifier()

rf_classifier.fit(X_train, y_train)

y_pred = rf_classifier.predict(X_test)

def print_confusionMatrix(Y_TestLabels, PredictedLabels):

confusionMatx = confusion_matrix(Y_TestLabels, PredictedLabels)

precision = confusionMatx/confusionMatx.sum(axis = 0)

recall = (confusionMatx.T/confusionMatx.sum(axis = 1)).T

sns.set(font_scale=1.5)

labels = ["WALKING", "WALKING_UPSTAIRS",
"WALKING_DOWNSTAIRS", "SITTING","STANDING", "LYING"]

plt.figure(figsize=(18,14))

sns.heatmap(confusionMatx, cmap = "Blues", annot = True, fmt =
".1f",xticklabels=labels, yticklabels=labels) plt.title("Confusion Matrix", fontsize =
30)

plt.xlabel('Predicted Class', fontsize = 20)

plt.ylabel('Original Class', fontsize = 20)

plt.tick_params(labelsize = 15)

plt.show()

print_confusionMatrix(y_test, y_pred)
```

FOR MOBILE APPLICATION:

CODE:

```
package com.developershutt.har;

import androidx.appcompat.app.AppCompatActivity;

import android.hardware.Sensor;

import android.hardware.SensorEvent;

import android.hardware.SensorEventListener;

import android.hardware.SensorManager;

import android.os.Bundle;

import android.util.Log;

import android.widget.TextView;

import java.math.BigDecimal;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.Timer;


public class MainActivity extends AppCompatActivity implements
SensorEventListener {
```

```
    private static final int TIME_STAMP = 100;
    private static final String TAG = "MainActivity";

    private static List<Float> ax,ay,az;
    private static List<Float> gx,gy,gz;
    private static List<Float> lx,ly,lz;
    private SensorManager mSensorManager;
    private Sensor mAccelerometer, mGyroscope, mLinearAcceleration;
    private float[] results;
    private ActivityClassifier classifier;
    private TextView runningTextView, downstairsTextView, joggingTextView,
sittingTextView, standingTextView, upstairsTextView, walkingTextView;
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initLayoutItems();
        ax=new ArrayList<>(); ay=new ArrayList<>(); az=new ArrayList<>();
```

```
        gx=new ArrayList<>(); gy=new ArrayList<>(); gz=new ArrayList<>();

        lx=new ArrayList<>(); ly=new ArrayList<>(); lz=new ArrayList<>();

         mSensorManager=(SensorManager)
getSystemService(SENSOR_SERVICE);


mAccelerometer=mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELERO
METER);

        mGyroscope                                                    =
mSensorManager.getDefaultSensor(Sensor.TYPE_GYROSCOPE);

        mLinearAcceleration                                           =
mSensorManager.getDefaultSensor(Sensor.TYPE_LINEAR_ACCELERATION);

classifier=new ActivityClassifier(getApplicationContext());

mSensorManager.registerListener(this,mAccelerometer,
SensorManager.SENSOR_DELAY_FASTEST);

        mSensorManager.registerListener(this,mGyroscope,
SensorManager.SENSOR_DELAY_FASTEST);

        mSensorManager.registerListener(this,mLinearAcceleration,
SensorManager.SENSOR_DELAY_FASTEST);

    }

  private void initLayoutItems() {

        runningTextView = findViewById(R.id.running_TextView);
```

```
    downstairsTextView = findViewById(R.id.downstairs_TextView);
    joggingTextView = findViewById(R.id.jogging_TextView);
    sittingTextView  = findViewById(R.id.sitting_TextView);
    standingTextView = findViewById(R.id.standing_TextView);
    upstairsTextView = findViewById(R.id.upstairs_TextView);
    walkingTextView = findViewById(R.id.walking_TextView);
}


@Override
public void onSensorChanged(SensorEvent event) {
    Sensor sensor = event.sensor;
    if (sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        ax.add(event.values[0]);
        ay.add(event.values[1]);
        az.add(event.values[2]);
    } else if(sensor.getType() == Sensor.TYPE_GYROSCOPE) {
        gx.add(event.values[0]);
        gy.add(event.values[1]);
```

```
            gz.add(event.values[2]);
        } else {
            lx.add(event.values[0]);
            ly.add(event.values[1]);
            lz.add(event.values[2]);
        }
           predictActivity();
    }
@Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {
}
private void predictActivity() {
        List<Float> data=new ArrayList<>();
        if (ax.size() >= TIME_STAMP && ay.size() >= TIME_STAMP && az.size()
>= TIME_STAMP
        && gx.size() >= TIME_STAMP && gy.size() >= TIME_STAMP &&
gz.size() >= TIME_STAMP
        && lx.size() >= TIME_STAMP && ly.size() >= TIME_STAMP && lz.size()
>= TIME_STAMP) {
            data.addAll(ax.subList(0,TIME_STAMP));
```

```
data.addAll(ay.subList(0,TIME_STAMP));
data.addAll(az.subList(0,TIME_STAMP));


data.addAll(gx.subList(0,TIME_STAMP));
data.addAll(gy.subList(0,TIME_STAMP));
data.addAll(gz.subList(0,TIME_STAMP));
data.addAll(lx.subList(0,TIME_STAMP));
data.addAll(ly.subList(0,TIME_STAMP));
data.addAll(lz.subList(0,TIME_STAMP));
results = classifier.predictProbabilities(toFloatArray(data));
Log.i(TAG, "predictActivity: "+ Arrays.toString(results));
runningTextView.setText("Running: \t" + round(results[0],2));
downstairsTextView.setText("DownStairs: \t" + round(results[1],2));
joggingTextView.setText("Jogging: \t" + round(results[2],2));
sittingTextView.setText("Sitting: \t" + round(results[3],2));
standingTextView.setText("Standing: \t" + round(results[4],2));
upstairsTextView.setText("Upstairs: \t" + round(results[5],2));;
walkingTextView.setText("Walking: \t" + round(results[6],2));
```

```
            data.clear();
            ax.clear(); ay.clear(); az.clear();
            gx.clear(); gy.clear(); gz.clear();
            lx.clear();ly.clear(); lz.clear();
        }
    }

    private float round(float value, int decimal_places) {
        BigDecimal bigDecimal=new BigDecimal(Float.toString(value));
        bigDecimal=bigDecimal.setScale(decimal_places,
BigDecimal.ROUND_HALF_UP);
        return bigDecimal.floatValue();
    }

    private float[] toFloatArray(List<Float> data) {
        int i=0;
        float[] array=new float[data.size()];
        for (Float f:data) {
```

```
            array[i++] = (f != null ? f: Float.NaN);
        }
        return array;
    }


    @Override
    protected void onResume() {
        super.onResume();
        mSensorManager.registerListener(this,mAccelerometer,
SensorManager.SENSOR_DELAY_FASTEST);
        mSensorManager.registerListener(this,mGyroscope,
SensorManager.SENSOR_DELAY_FASTEST);
        mSensorManager.registerListener(this,mLinearAcceleration,
SensorManager.SENSOR_DELAY_FASTEST);
    }
@Override
    protected void onDestroy() {
        super.onDestroy();
        mSensorManager.unregisterListener(this);
    }
```

5.4 SAMPLE SCREENSHOT

OUTPUT:

GyroMean,gravityMean)	angle(tBodyGyroJerkMean,gravityMean)	angle(X,gravityMean)	angle(Y,gravityMean)	angle(Z,gravityMean)	subject	Activity
-0.765982	-0.503815	-0.821846	0.169574	0.122720	24	WALKING
0.140347	-0.431206	0.606948	-0.839474	-0.153953	20	LAYING
-0.147103	-0.089153	-0.623939	-0.126464	-0.119953	4	SITTING
0.701660	0.556372	-0.737415	0.226713	0.150138	13	WALKING_DOWNSTAIRS
-0.468525	0.930765	-0.801583	0.222411	-0.042257	13	STANDING
-0.926184	0.143887	-0.743653	0.220097	0.150778	13	WALKING_DOWNSTAIRS
0.438147	-0.727073	0.353466	-0.783418	-0.133095	13	LAYING
-0.298127	0.159888	-0.521553	-0.182923	-0.159336	18	SITTING
0.092784	-0.528653	-0.898684	0.024580	0.011453	24	SITTING
0.158806	0.196950	-0.942696	0.075130	-0.019381	4	SITTING

Fig no 5.2

angle(tBodyGyroMean,gravityMean)	angle(tBodyGyroJerkMean,gravityMean)	angle(X,gravityMean)	angle(Y,gravityMean)	angle(Z,gravityMean)	subject	Activity
-0.464761	-0.018446	-0.841247	0.179941	-0.058627	1	STANDING
-0.732626	0.703511	-0.844788	0.180289	-0.054317	1	STANDING
0.100699	0.808529	-0.848933	0.180637	-0.049118	1	STANDING
0.640011	-0.485366	-0.848649	0.181935	-0.047663	1	STANDING
0.693578	-0.615971	-0.847865	0.185151	-0.043892	1	STANDING

Fig no 5.2

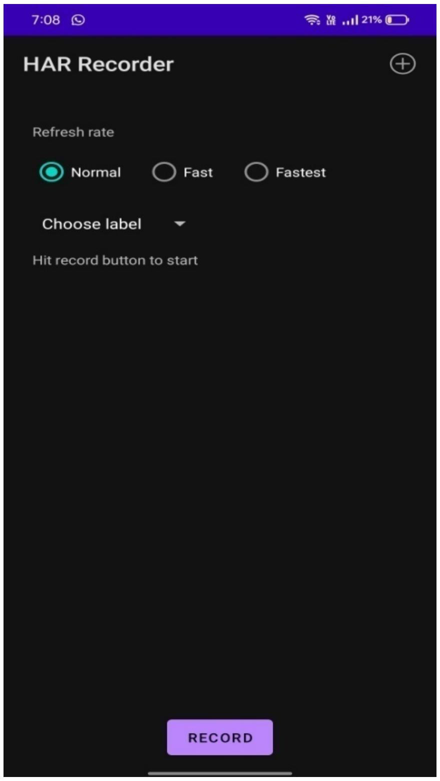

Fig no 5.2

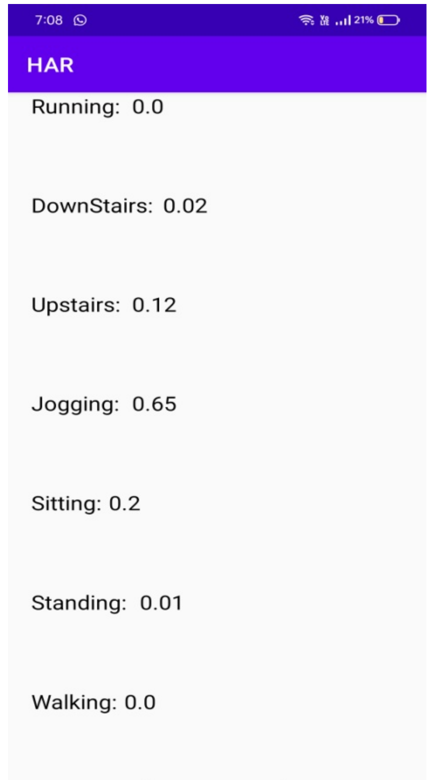

Fig no 5.2

BAR DIAGRAM:

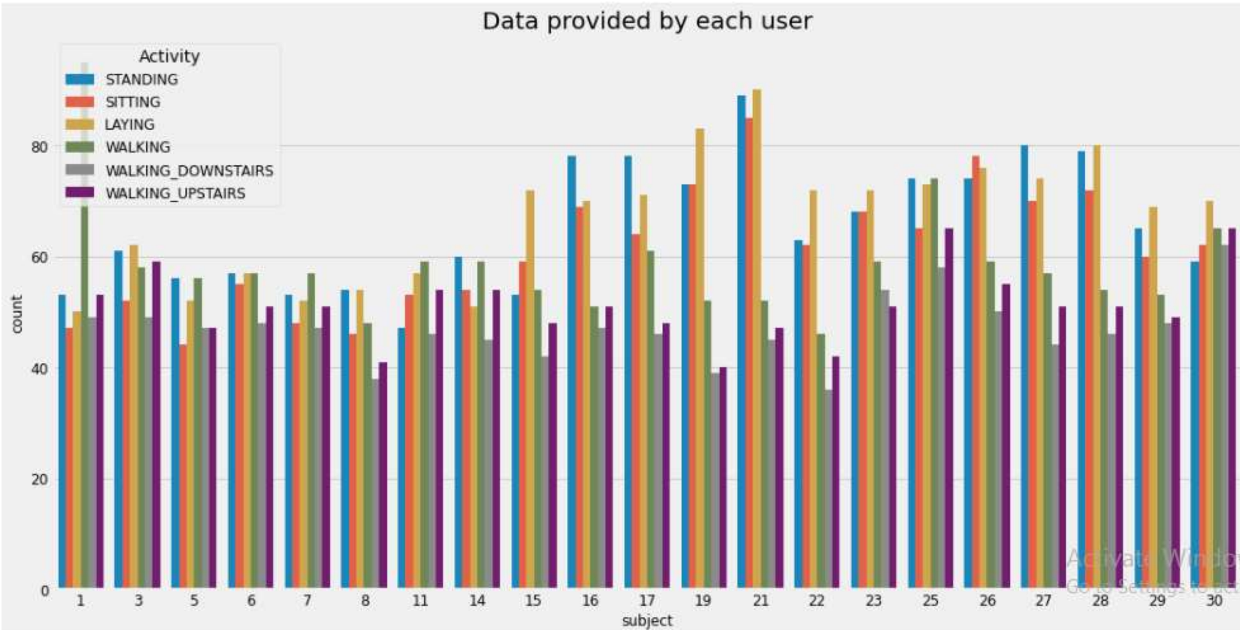

Fig no 5.3

GRAPH DIAGRAM:

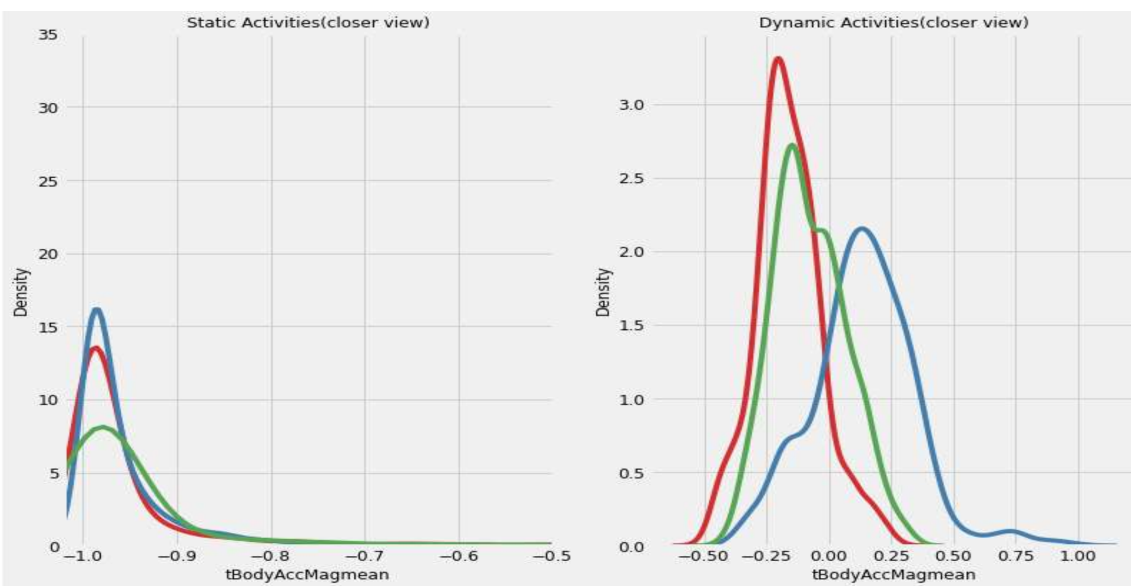

Fig no 5.4

CONFUSION MATRIX:

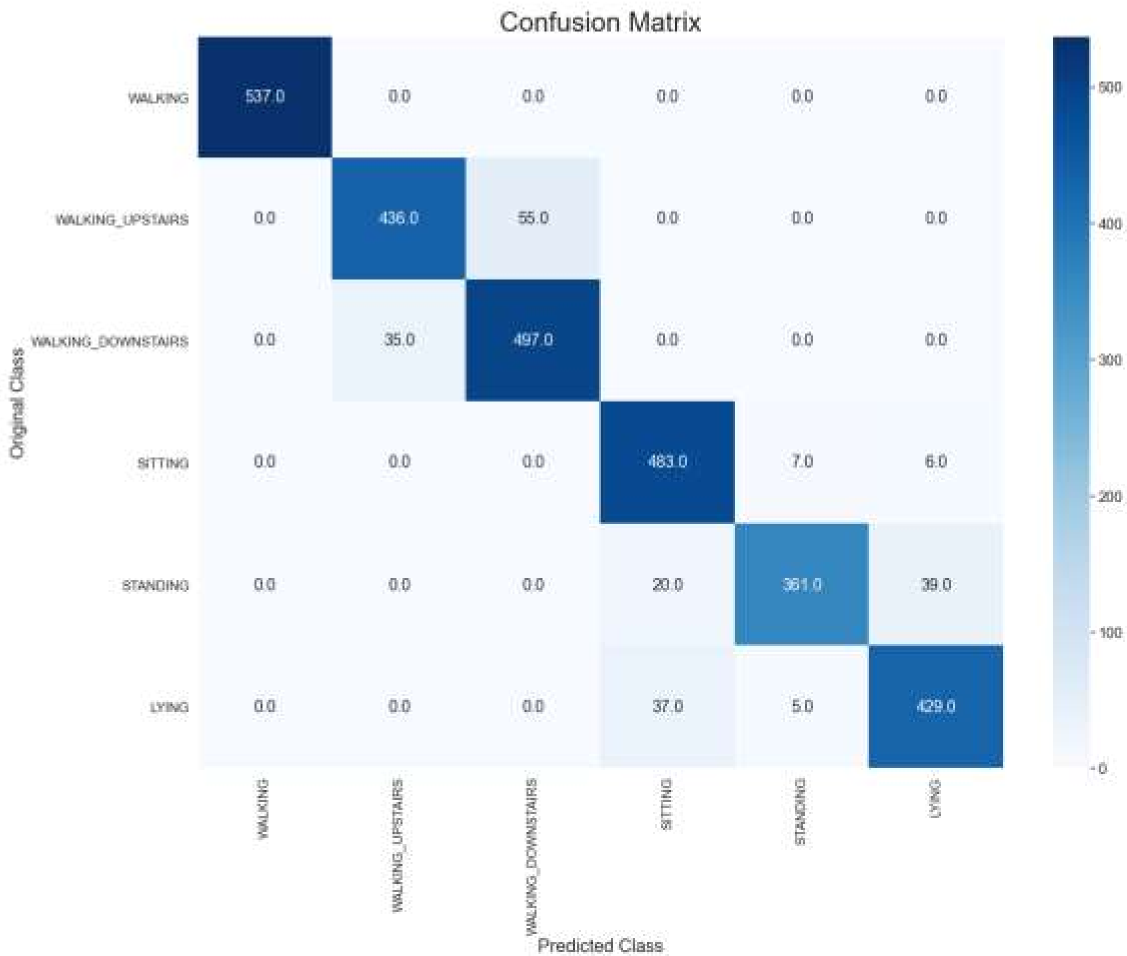

Fig no 5.5

CHAPTER 6

TESTING AND MAINTENANCE

6.1 SOFTWARE TESTING

Testing documentation is the documentation of artifacts that are created during or before the testing of a software application. Documentation reflects the importance of processes for the customer, individual and organization. Works which contain all documents have a high level of maturity. Careful documentation can save the time, efforts and wealth of the organization.

If the testing or development team gets software that is not working correctly and developed by someone else, so to find the error, the team will first need a document. Now, if the documents are available then the team will quickly find out the cause of the error by examining documentation. But, if the documents are not available then the tester need to do black box and white box testing again, which will waste the time and money of the organization. More than that, Lack of documentation becomes a problem for acceptance.

Benefits of using Documentation:

Documentation clarifies the quality of methods and objectives.

It ensures internal coordination when a customer uses software application.

It ensures clarity about the stability of tasks and performance.

It provides feedback on preventive tasks. It provides feedback for your planning cycle.

It creates objective evidence for the performance of the quality management system the test scenario is a detailed document of test cases that cover end to end functionality of a software application in liner statements. The liner statement is considered as a scenario. The test scenario is a high-level classification of testable requirements. These requirements are grouped on the basis of the functionality of a module and obtained from the use cases.

In the test scenario, there is a detailed testing process due to many associated test cases. Before performing the test scenario, the tester has to consider the test cases for each scenario.

In the test scenario, testers need to put themselves in the place of the user because they test the software application under the user's point of view. Preparation of scenarios is the most critical part, and it is necessary to seek advice or help from customers, stakeholders or developers to prepare the scenario.

As per the IEEE Documentation describing plans for, or results of, the testing of a system or component, Types include test case specification, test incident work, test log, test plan, test procedure, test work. Hence the testing of all the above mentioned documents is known as documentation testing. This is one of the most cost effective approaches to testing.

This is one of the most cost effective approaches to testing. If the documentation is not right: there will be major and costly problems. The documentation can be tested in a number of different ways to many different degrees of complexity. These range from running the documents through a spelling and grammar checking device, to manually reviewing the documentation to remove any ambiguity or inconsistency.

Documentation testing can start at the very beginning of the software process and hence save large amounts of money, since the earlier a defect is found the less it will cost to be fixed.

The most popular testing documentation files are test works, plans, and checklists. These documents are used to outline the team's workload and keep track of the process. Let's take a look at the key requirements for these files and see how they contribute to the process

Test strategy:

An outline of the full approach to product testing. As the Work moves along, developers, designers, product owners can come back to the document and see if the actual performance corresponds to the planned activities.

Test data:

The data that testers enter into the software to verify certain features and their outputs. Examples of such data can be fake user profiles, statistics, media content, similar to files that would be uploaded by an end user in a ready solution.

Test plans:

A file that describes the strategy, resources, environment, limitations, and schedule of the testing process. It's the fullest testing document, essential for informed planning. Such a document is distributed between team members and shared with all stakeholders.

Test scenarios:

In scenarios, testers break down the product functionality and interface by modules and provide real-time status updates at all testing stages. A module can be described by a single statement, or require hundreds of statuses, depending on its size and scope.

Test cases:

If the test scenario describes the object of testing (what), a scenario describes a procedure (how). These files cover step-by-step guidance, detailed conditions, and current inputs of a testing task. Test cases have their own kinds that depend on the type of testing, functional, UI, physical, logical cases, etc. Test cases compare available resources and current conditions with desired outcomes and determine if the functionality can be released or not.

Traceability Matrix:

This software testing documentation maps test cases and their requirements. All entries have their custom IDs team members and stakeholders can track the progress of any tasks by simply entering its ID to the search.

The combination of internal and external documentation is the key to a deep understanding of all testing processes. Although stakeholders typically have access to the majority of documentation, they mostly work with external files, since they are more concise and tackle tangible issues and results. Internal files, on the other hand, are used by team members to optimize the testing process.

Unit testing is not a new concept. It's been there since the early days of programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test drove development TDD or test-first development).

6.2 TEST CASE SPECIFICATION:

TEST CASE ID	MODULE	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC1	Analysis the pre-defined datasets	Getting a online datasets to analysis the activity	Analysis the datasets to get the activity of the human	Getting the reference model	Pass
TC2	Record the real-time activity	Using the mobile application recording the real-time activity	Real-time recorded datasets for accuracy	Real-time recorded datasets for accuracy	Pass
TC3	Train the datasets	With the recorded activity train into the model	Trained into model to develop into application	Trained model	Pass
TC4	Deploy into the mobile application	Showing the real-time activity	Reading the actual activity	Reading the actual activity	Pass

6.3 TEST DRIVEN DEVELOPMENT:

Test Driven Development, or TDD, is a code design technique where the programmer writes a test before any production code, and then writes the code that will make that test pass. The idea is that with a tiny bit of assurance from that initial test, the programmer can feel free to refactor and refactor some more to get the cleanest code they know how to write. The idea is simple, but like most simple things, the execution is hard. TDD requires a completely different mindset from what most people are used to and the tenacity to deal with a learning curve that may slow you down at first.

Functional Testing types include:

Unit Testing

Integration Testing

System Testing

Sanity Testing

Smoke Testing

Interface Testing

Regression Testing

Beta/Acceptance Testing

Non-functional Testing types include:

Load Testing

Stress Testing

Volume Testing

Security Testing

Compatibility Testing

Install Testing

Recovery Testing

Reliability Testing

Usability Testing

Compliance Testing

Localization Testing

6.3.1UNIT TESTING:

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

A unit can be almost anything you want it to be -- a line of code, a method, or a class. Generally though, smaller is better. Smaller tests give you a much more granular view of how your code is performing. There is also the practical aspect that when you test very small units, your tests can be run fast; like a thousand tests in a second fast.

Black Box testers don't care about Unit Testing. Their main goal is to validate the application against the requirements without going into the implementation details.

Unit testing is not a new concept. It's been there since the early days of programming. Usually, developers and sometimes White box testers write Unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements (aka test drove development TDD or test-first development). Most of us might know the classic definition of Unit Testing.

Unit testing is the method of verifying the smallest piece of testable code against its purpose

If the purpose or requirement failed then the unit test has failed. In simple words, Unit Testing means - writing a piece of code (unit test) to verify the code (unit) written for implementing requirements.

6.3.2 BLACKBOX TESTING:

During functional testing, testers verify the app features against the user specifications. This is completely different from testing done by developers which is unit testing. It checks whether the code works as expected. Because unit testing focuses on the internal structure of the code, it is called the white box testing. On the other hand, functional testing checks functionalities without looking at the internal structure of the code, hence it is called black box testing. Despite how flawless the various individual code components may be, it is essential to check that the app is functioning as expected, when all components are combined. Here you can find a detailed comparison between functional testing vs unit testing.

6.3.3 INTEGRATION TESTING:

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

Integration testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration.

Component integration testing Testing performed to expose defects in the interfaces and interaction between integrated components. System integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

Integration tests determine if independently developed units of software work correctly when they are connected to each other. The term has become blurred even by the diffuse standards of the software industry, so I've been wary of using it in my writing. In particular, many people assume integration tests are necessarily broad in scope, while they can be more effectively done with a narrower scope.

As often with these things, it's best to start with a bit of history. When I first learned about integration testing, it was in the 1980's and the waterfall was the dominant influence of software development thinking. In a larger Work, we would have a design phase that would specify the interface and behavior of the various modules in the system. Modules would then be assigned to developers to program. It was not unusual for one programmer to be responsible for a single module, but this would be big enough that it could take months to build it. All this work was done in isolation, and when the programmer believed it was finished they would hand it over to QA for testing.

Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems. Integration testing is a key aspect of software testing.

6.3.4 SYSTEM TESTING:

System testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the systems compliance with the specified requirements. System Testing means testing the system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not.

System Testing is done after Integration Testing. This plays an important role in delivering a high-quality product. System testing is a method of monitoring and assessing the behavior of the complete and fully-integrated software product or system, on the basis of pre-decided specifications and functional requirements. It is a solution to the question "whether the complete system functions in accordance to its pre-defined requirements?"

It's comes under black box testing i.e. only external working features of the software are evaluated during this testing. It does not requires any internal knowledge of the coding, programming, design, etc., and is completely based on usersperspective.

A black box testing type, system testing is the first testing technique that carries out the task of testing a software product as a whole. This System testing tests the integrated system and validates whether it meets the specified requirements of the client.

System testing is a process of testing the entire system that is fully functional, in order to ensure the system is bound to all the requirements provided by the client in the form of the functional specification or system specification documentation. In most cases, it is done next to the Integration testing, as this testing should be covering the end-to-end systems actual routine.

This type of testing requires a dedicated Test Plan and other test documentation derived from the system specification document that should cover both software and hardware requirements. By this test, we uncover the errors. It ensures that all the system works as expected. We check System performance and functionality to get a quality product. System testing is nothing but testing the system as a whole. This testing checks complete end-to-end scenario as per the point of view. Functional and Non-Functional tests also done by System testing. All things are done to maintain trust within the development that the system is defect-free and bug-free. System testing is also intended to test hardware/software requirements specifications. System testing is more of a limited type of testing;

6.3.5 SANITY TESTING:

Sanity testing is done when as a QA we do not have sufficient time to run all the test cases, be it Functional Testing, UI, OS or Browser Testing. Sanity testing is a subset of regression testing. After receiving the software build, sanity testing is performed to ensure that the code changes introduced are working as expected. This testing is a checkpoint to determine if testing for the build can proceed or not. The main purpose of this testing is to determine that the changes or the proposed functionality are working as expected. If the sanity test fails, the build is rejected by the testing team to save time and money. It is performed only after the build has cleared the smoke test and been accepted by the Quality Assurance team for further testing. The focus of the team during this testing process is to validate the functionality of the application and not detailed testing.

Smoke Testing is done to make sure if the build we received from the development team is testable or not. It is done at the build level.

It helps not to waste the testing time to simply testing the whole application when the key features work or the key bugs have not been fixed yet. Here our focus will be on primary and core application work flow.

To conduct smoke testing, we do not write test cases. We just pick the necessary test cases from already written test cases. As mentioned earlier, here in Smoke Testing, our main focus will be on core application work flow. So we pick the test cases from our test suite which cover major functionality of the application. In general, we pick minimal number of test cases that won't take more than half an hour to execute.

The main aim of Sanity testing to check the planned functionality is working as expected. Instead of doing whole regression testing the Sanity testing is perform. Sanity tests helps to avoid wasting time and cost involved in testing if the build is failed. Tester should reject the build upon build failure. After completion of regression testing the Sanity testing is started to check the defect fixes & changes done in the software application is not breaking the core functionality of the software. Typically this is done nearing end of SDLC i.e. while releasing the software. You can say that sanity testing is a subset of acceptance testing. We can also say Tester Acceptance Testing for Sanity testing.

6.3.6 REGRESSION TESTING:

Regression testing is a type of testing that is done to verify that a code change in the software does not impact the existing functionality of the product. This is to make sure the product works fine with new functionality, bug fixes or any change in the existing feature. Previously executed test cases are re-executed in order to verify the impact of change.

Regression Testing is a Software Testing type in which test cases are reexecuted in order to check whether the previous functionality of the application is working fine and the new changes have not introduced any new bugs.

This test can be performed on a new build when there is a significant change in the original functionality that too even in a single bug fix. For regression testing to be effective, it needs to be seen as one part of a comprehensive testing methodology that is cost-effective and efficient while still incorporating enough as well-designed frontend UI automated tests alongside targeted unit testing, based on smart risk prevent any aspects of your software applications from going unchecked. These days, many Agile work environments employing workflow practices such as XP (Extreme Programming), RUP (Rational Unified Process), or Scrum appreciate regression testing as an essential aspect of a dynamic, iterative development and deployment schedule. But no matter what software development and quality-assurance process your organization uses, if you take the time to put in enough careful planning up front, crafting a clear and diverse testing strategy with automated regression testing at its core, you can help prevent Works from going over budget, keep your team on track, and, most importantly, prevent unexpected bugs from damaging your products and your company’s bottom line.

6.3.7 PERFORMANCE TESTING:

Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size. Performance Testing (Taken from Google.com)

Fig6.1 PERFORMANCE TESTING

Performance testing gathers all the tests that verify an applications speed, robustness, reliability, and correct sizing. It examines several indicators such as a browser, page and network response times, server query processing time, number of acceptable concurrent users architected, CPU memory consumption, and number/type of errors which may be encountered when using an application. Performance testing is the testing that is performed to ascertain how the components of a system are performing under a certain given situation. Resource usage, scalability, and reliability of the product are also validated under this testing. This testing is the subset of performance engineering, which is focused on addressing performance issues in the design and architecture of a software product.

Software Performance testing is type of testing perform to determine the performance of system to major the measure, validate or verify quality attributes of the system like responsiveness, Speed, Scalability, Stability under variety of load conditions. Software performance testing involves the testing of application under test to ensure that application is working as expected under variety of load conditions. The goal of performance testing is not only find the bugs in the system but also eliminate the performance bottlenecks from the system.

Load Testing is type of performance testing to check system with constantly increasing the load on the system until the time load is reaches to its threshold value. Here Increasing load means increasing number of concurrent users, transactions & check the behavior of application under test. It is normally carried out underneath controlled environment in order to distinguish between two different systems. It is also called as Endurance testing Volume the main purpose of load testing is to monitor the response time and staying power of application when system is performing well under heavy load.

CHAPTER 7

CONCLUSION AND FUTURE WORK:

7.1 CONCLUSION:

This paper presents a method for behavior discovery as well as the logical satisfiability-oriented reactive analysis for smart and sensor-based environments to support context-aware and pro-active decisions.This approach constructs the process for building logical specifications that fulfill the recognition process providing behavioral specification in terms of logic formulas. The proposed unified logical framework is focused on sensor based activity recognition.In this work, we tackled the problem of multi-class behavioral context recognition with deep multi-modal convolutional neural networks. We propose to train an end-to-end model for jointly-learning from low-level sensory data (accelerometer, gyroscope, audio and phone state) of smart devices collected in-the-wild.We also showed that instance-weighted cross-entropy loss (as also leveraged in regularization schemes enable the model to generalize well on highly imbalanced (sparsely labeled) dataset.Furthermore, we present a slight modification in the proposed networks architecture to handle missing sensors; potentially taking advantage of Deep Metric Context Learning.

7.2 FUTURE WORK:

There are several avenues for future work that could build upon the "Improving Feature Relevance Analysis Sensor-based Behavioral Context Recognition using Deep Metric Context Learning" project. Here are some potential directions for future research:

Multi-modal sensor fusion: This project uses data from a single type of sensor to recognize behavioral context. However, using multiple sensors, such as accelerometers, gyroscopes, and magnetometers, can improve the accuracy of behavioral context recognition. Future work could explore how to combine data from multiple sensors to improve the performance of the deep metric context learning model.Multi-modal sensor fusion: This project uses data from a single type of sensor to recognize behavioral context. However, using multiple sensors, such as accelerometers, gyroscopes, and magnetometers, can improve the accuracy of behavioral context recognition. Future work could explore how to combine data from multiple sensors to improve the performance of the deep metric context learning model.

CHAPTER 8

REFERENCES

8.REFERENCES:

[1] “R. Mojarad, F. Attal, A. Chibani, S. R. Fiorini, and Y. Amirat, Hybrid approach for human activity recognition by ubiquitous robots, in Proc.

[2] “ D. M. Hilty and S. Chan, Human behavior with mobile health: Smart- , L. Cao, Y. Wang, B. Zhang, Q. Jin, and A. V. Vasilakos, GCHAR: An efcient group-based contextAware human activity recognition Aug. .

[3] “Y. Xu, Z. Shen, X. Zhang, Y. Gao, S. Deng, Y. Wang, Y. Fan, and E. I.-C. Chang, Learning multi-level features for sensor-based human Sep.

[4] “A. Sucerquia, J. Lpez, and J. Vargas-Bonilla, Real-life/real-time elderly p. , Apr. , .

[5] “D. Coskun, O. D. Incel, and A. Ozgovde, Phone position/placement detection using accelerometer: Impact on activity recognition, in Proc. Apr.

[6] “L. Xu, W. Yang, Y. Cao, and Q. Li, Human activity recognition based on random forests, in Proc. , th Int. Conf. Natural Comput., Fuzzy Syst. CA, USA: Morgan Kaufmann,

[7] “T. H. Itkonen, E. Lehtonen, and Selpi, Characterisation of motorway driv- ing style using naturalistic driving data, Transp. Res. F: Trafc Psychol.

[8] “A. K. Jain, P. Flynn, and A. Ross, Handbook of Biometrics. New York, , S. Z. Li and A. K. Jain, Handbook of Face Recognition. London, U.K.: USA: Springer, , . NY, USA: Springer, , . Springer, , .

[9] “Y. Tang, F. Gao, J. Feng, and Y. Liu, "FingerNet: An unied deep network for ngerprint minutiae extraction, " in Proc. Int. Joint Conf.

[10] “L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, and M. Pietikinen, "From BoW to CNN: Two decades of texture representation for tex- Jan. , .

[11] “K. L. Mosier, L. J. Skitka, S. Heers, and M. Burdick, "Automation bias: Decision making and performance in high-tech cockpits, " Int. J.

[12] “S. Desmarais and J. Singh, Risk Assessment Instruments Validated and Implemented in Correctional Settings in the United States. New York, NY, USA: Council State Govt. Justice Center, Mar. ,

[13] “C. Garvie, The Perpetual Line-Up: Unregulated Police Face Recognition in America, Center Privacy Technol., Georgetown Law, Washington, DC, USA, Oct.

[14] “A. Bhutani and P. Bhardwaj, "Biometrics market size by application, " Global Market Insights, Pune, India, Rep. GMI, Aug. , .

[15] “F. Pasquale, The Black Box Society: The Secret Algorithms That Control Money and Information. Cambridge, MA, USA: Harvard Univ. Press, , .

[16] “R. H. Smith, Assimilative and contrastive emotional reactions to upward and downward social comparisons, in Handbook of Social Comparison.

[17] “S. Nakayama, E. Krasner, L. Zino, and M. Porri, Social information and spontaneous emergence of leaders in human groups, J. Roy. Soc.

[18] “C. Stramann, A. M. Rosenthal-von der Ptten, and N. C. Krmer, With or against each other, The inuence of a virtual agents (non) cooperative behavior on users cooperation behavior in the prisoners dilemma, Adv.

[19] “E. A. Vogel, J. P. Rose, L. R. Roberts, and K. Eckles, Social comparison, no. , p. ,

[20] “J. Reiman and P. Leighton, Rich Get Richer and the Poor Get Prison: Ideology, Class, and Criminal Justice. Evanston, IL, USA: Routledge, , .

[21] “C. Gracia-Lazaro, A. Ferrer, G. Ruiz, A. Tarancon, J. A. Cuesta, A. Sanchez, and Y. Moreno, Heterogeneous networks do not promote cooperation when humans play a prisoners dilemma, Proc. Nat. Acad.

[22] “, P. M. Kappeler and C. P. Van Schaik, Cooperation in Primates and Humans. Berlin, Germany: Springer, ,