Thesis for the Degree of Doctor of Philosophy

Human Commuting Activity Recognition based on Mobility Natural Vibration on a Mobile Device

Taeho Hur

Department of Computer Science and Engineering
Graduate School
Kyung Hee University
Seoul, Korea

February, 2020

Human Commuting Activity Recognition based on Mobility Natural Vibration on a Mobile Device

Taeho Hur

Department of Computer Science and Engineering
Graduate School
Kyung Hee University
Seoul, Korea

February, 2020

Human Commuting Activity Recognition based on Mobility Natural Vibration on a Mobile Device

by

Taeho Hur

Advised by

Professor. Sungyoung Lee

Submitted to the Department of Computer Science and Engineering
and the Faculty of the Graduate School of
Kyung Hee University in partial fulfillment
of the requirements for degree of
Doctor of Philosophy

Dissertation Committee:

Prof. Tae-Seong Kim ____________

Prof. LokWon Kim ____________

Prof. Sung-Ho Bae ____________

Prof. Hyon Woo Seung ____________

Prof. Sungyoung Lee ____________

Abstract

Traditional Human Activity Recognition attached wearable sensors on the body with fixed position and orientation. As the technology has developed, smartphone has emerged embedded with various kinds of sensors, substituting wearable sensors. Due to the nature of smartphone, people carry it on any place with different orientation. This lead to one of an important consideration, guaranteeing position and orientation independency.

Meanwhile, people nowadays show similar life pattern. This includes such as people go to work on week days, uses vehicles for movement, and carry a smartphone. This makes a perfect chance to recognize commuting activities. Commuting activity is an activity happening during commuting such as standing, sitting, walking, jogging, riding a car, bus, subway, train, motorcycle, bicycle, and many others. As most of the people carry smarpthone, we can extract inertial data from accelerometer and gyroscope, speed and coordinate information from GPS, sound from mic, video from camera and so on. From these data, we can extract features and classify them to know which activity the user is performing.

The advantage of automatic detection of commuting activity will be 1) Statistical data of transportation usage can be acquired automatically, reducing the cost and time spent via manual survey, 2) User customized services or advertisements can be provided based on transportation just-in-time, 3) Monitoring health status, safety issues, and consumed calories can be provided for health care, and 4) Carbon footprint can be estimated which is a unit of an object producing greenhouse gas.

Especially vehicles among commuting activities possess major issues. The problem is that user does not move inside the vehicle, only standing or sitting. These are static

activities having no big variation on sensor signal, and also showing no eigen pattern from time domain signal. To overcome this problem, some of the research extracted features from frequency domain signal. Frequency is the number of occurrences of a repeating event per unit of time. Mainly extracted features are statistical features such as mean and standard deviation, or peak frequency. But these features contains problems that statistical feature cannot be applied in frequency domain that each frequency is representing different dimension, and the peak frequency can be changed. Another consideration of commuting activity recognition is that due to unseen real world environmental factors, error may cause based on machine learning method. To correct this error, heuristic based post-processing is required after machine learning based classification.

Therefore, in this paper, two methods are proposed. The first is using natural vibration features to handle position independency and classifying vehicles. As every object has their own natural vibration, each vehicle will have their own too, and this does not affect the position of the sensor. The second is activity correction which adjusts the classified result taking into account real world scenario. In here, GPS coordinate is used to distinguish from road vehicles to subway when the subway runs ground level, state transition rule is adopted, and the activity is corrected referring current activity, previous activity and previously corrected activity. From our experimental result, the proposed method shows higher performance than existing method for commuting activity recognition.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Background

The goal of Human Activity Recognition (HAR) is to recognize common human activities in real-life settings such as locomotion, postures, and gestures [1]. Types of locomotions includes activity such as walking, jogging and running, and types of postures includes activity such as sitting, standing and lying, and types of gestures includes activities like shaking hands, waving hands, and so on. The general procedure of HAR is constituted of data acquisition, pre-processing, segmentation, feature extraction, and classification. Data acquisition is literally collecting the data to be used for recognition. Pre-processing includes the refinement of data, such as filtering, reducing, integrating, and transforming. Segmentation process is done where the data must be split to apply features and classify into groups. Feature extraction includes to extract prominent characteristics of data. Feature selection process is followed after extraction to only pick meaningful features. Finally, classification is conducted based on the features and produce the activity label in the end.

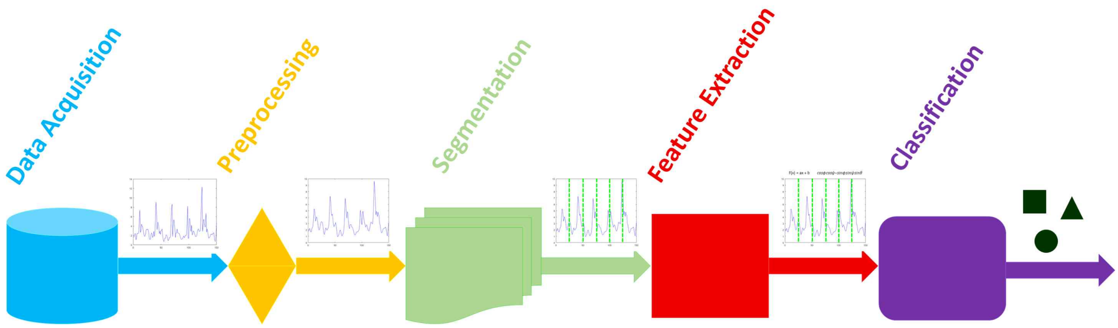


Figure 1-1. General activity recognition chain

Research on HAR began from the 1980s [2] and has become the basis of context awareness such as pervasive computing, wearable computing, human computer interaction, and in the support of health care, life care and wellness care. Most of these require information about users’ life patterns to provide personalized services or health promotion. Many studies have suggested methods to recognize numerous activities with high accuracy [3–7]. There are mainly three different methods of HAR, which are video-based, environmental sensor–based, and wearable sensor–based, while each method has its own pros and cons.

In video-based methods, video recorded from cameras such as CCTV are used as data [8-12]. At first, pre-processing method can be applied to remove noises such as blur, glitter, illumination and occlusion. Then video is segmented to frame by frame. Then the object and background are separated to only refer human being. Features are extracted afterwards to well detect which activity the human is performing. Lastly, the classification is made and activity is recognized on each frame. Compared to traditional RGB video based HAR, a skeleton based method is studied these days by using 3D depth cameras, where it can more easily separate human from background. But this method can only be applied when depth camera is installed, not usable from CCTV videos. The advantage of video-based method is that the subject doesn’t need to carry any kind of sensors, making it unobtrusiveness. On the other hand, the disadvantage comes from that the subject is hard to be traced. It can only be used where the camera is placed. Currently, even though there are lots of cameras installed, there are still many places having no cameras installed. It is even hard to detect specific subject while there are a lot of people inside on the scene. Therefore, it is still a challenging task to conclude whether the same subject has appeared from one video to another.

In environmental sensor–based methods, sensors such as infrared sensors, pressure sensors, or proximity sensors are installed in specific location such as smart home or a specific location with embedded sensors around [13-16]. Most of the environmental sensors has binary outputs which is much easier to infer user activity in combination with many sensors. And it is also good to trace the subject moving around indoor for whole day that specific location's sensor will response. But the problem comes from that it is hard to differentiate subjects if there are more than two subjects inside, having no particular factors to differentiate them such as tags. Another problem is that it can only detect abstract activities. For example, if the user is on the couch and reading a book, it may only conclude user activity as sitting, but not reading. Finally, recognition limitation comes from that this method only works in the place where sensors are installed.

In wearable sensor–based methods, devices containing inertial sensor units, such as an accelerometer (ACC) and gyroscope, and non-inertial sensor units such as magnetometer and GPS are attached to the body [17-20]. In particular, accelerometers are a popular way to recognize users' activity, while other sensor assists accelerometer. Activities are then classified into types, with each activity type showing a different pattern of sensor signal values. Normally, sensors attached on lower body part, such as thigh or ankle, are in charge to detect the leg movement, detecting locomotions or foot gestures such as kicking. Sensors attached on upper body part in torso such as waist or chest detects the overall body movement such as walking or jogging. Lastly, sensors attached on wrist or forearm will detect hand based gestures. As mentioned above, previous research tried to measure the change of motion precisely, so researchers attached many sensor devices on each part of the body and analyzed the resulting data. Table 1-1 shows the types of HAR and their used sensors.

Table 1-1. Types of HAR and their used sensors

Types of HAR	Types of sensors	Description
Video-based HAR	RGB	A traditional video with moving image composed of RGB color.
	RGB Depth	A video showing the objects in different color of depths based on perspective. It is easier than RGB video to differentiate from object and background that object is usually former than background.
	Skeleton	A video showing the skeleton and joints of human. This method suits well for activity recognition by only capturing human body movement. Disadvantage of this is that it shows error on skeleton in high rate.
Environmental sensor-based HAR	Pressure	A sensor operated when pressure is detected. Normally used in furniture such as couch, chair, bed, etc.
	Proximity	A sensor operated when an object comes close to predefined distance. Normally used in doors or drawers.
	Infrared	A sensor which detects body heat. It is used to know whether the object exists.
	Microwave	A sensor detecting reflections on moving objects by sending microwave pulse.
	Ultrasonic	A sensor detecting reflections on moving objects by sending ultrasonic wave.
Wearable sensor-based HAR	Accelerometer	A sensor returning acceleration signal based on gravity. Normally used to detect moving direction.
	Gyroscope	A sensor returning angular velocity signal. Normally used to detect how much the object has rotated.
	Magnetometer	A sensor returning magnetic field signal. Normally used to find point of the compass.
	GPS	A sensor returns current coordinate latitude and longitude, and speed based on satellite signal.

HAR methods building upon multiple sensors became less frequent after the introduction of smartphones from early 2010s. The methods devised from previous research required users to buy sensors of different types and attach them to different parts of their bodies. Each sensor is expensive, and the resulting systems are cumbersome to use in real life. However, smartphones are equipped with multi-modal sensors, and many people worldwide (and more people every year) carry at least one of it [21]. Therefore, HAR research has significantly shifted direction in recent years, and now focuses more predominantly on the use of smartphones. Additionally, after the advent of smartphone, wrist type devices have come to the surface. A smartwatch, which is a small edition of smartphone is widely used nowadays. Also many kinds of wearable devices such as Fitbit, Misfit, and Fuelband were released. All of these contain inertial sensors and used for the purpose of human activity recognition, by itself or along with smartphone.

Although the unobtrusiveness has reduced than before and it is good to trace people that they carry a smartphone embedded with sensors whole time, it still contains weaknesses. A human being cannot perform the same activity in the same way. For example, if the same person tries to walk in the same route with the same speed, the sensor signal does not show exact same pattern that sensor is sensitive enough and respond to even minimum movements. For example, even the sensor is simply on the desk, the signal values does not show strict linearity but small sinusoidal curves. This differs much more as times goes by when the person gets older or having injuries. It also differs to every person that no one is identically same, even for twins, performing different activities.

1.2 Motivation

As the technology advances by time, people's life style is getting standardized. For example, people carry their smartphone and commute on weekdays, or go for shopping and traveling on weekends. In here, they do not walk but mostly ride vehicles. Contrary to modern technology development, deterioration of health caused by lack of exercise is becoming a social issue, and from this point of view, human movement is not required when riding vehicles and does not have a good effect in terms of health. If the vehicles are well recognized, their life pattern can be more well understood and could provide personalized health or convenient services. From here, detecting commuting activities such as taking a bus, subway, train, car, bicycle have following advantages [22]:

- Statistical data of transportation usage can be acquired automatically, reducing the cost and time spent via survey
- User customized services or advertisements can be provided based on transportation just-in-time
- Monitoring health status, safety issues, consumed calories, and impact on environment based on carbon footprint

Carbon footprint is a new concept. If a human moves from one location to another on foot, oneself will not produce any air pollutions. If a human rides a car for moving, it will produce one whole carbon one can make. And if a human takes public transportation, production of carbon will be distributed by the number of people taking it. Therefore, by automatically detecting type of transportation, it would be able to figure out how a human impacts to the environment.

There are three main factors that controls the activity recognition accuracy which are using quality of data, extracting discriminant features, and using appropriate classification methods. Using high quality data is the basic and fundamental factor. Because if the data contains too much noises or if the data labeling is wrong, it will never show good accuracy whatever you do on feature extraction and classification stage. Generally in most research, it is assumed that data is in good quality. Classification is also important to well distinguish the activity based on the features. But in general, well known statistical classifiers such as Support Vector Machine (SVM), Decision Tree (DT), K-Nearest Neighborhood (KNN), Random Forest (RF), Bayesian Network, and Multilayer Perceptron are used. Among three main factors, feature is considered to be the most important one. The features that can best represent the activity must be extracted and selected from the collected sensory data. This feature engineering process is the most significant problem for achieving accurate HAR [23].

Existing research on vehicle detection uses statistical features on time domain and frequency domain. Vehicles are static activities where there is no movement of human. Therefore, looking through the acceleration signal, no big fluctuation is made. And also, the signal pulse seems to be random without any particular pattern. This is applied to all kinds of vehicles. For example, subway has motor and runs on the rail. While inside subway, one cannot feel any big vibration but small ones. While in the car, one can feel bigger vibration than subway but still small ones. While in the bus, the vibration is more bigger. But when looking through the signal collected from accelerometer, it can be observed that it is hard to distinguish them intuitively. Even with extracting features and apply machine learning, the classification accuracy is poor. Therefore, frequency domain features are also used in the research.

Frequency is a unique signal to detect number of times an object vibrates over a period of time. The unit of this is hertz (Hz). 1Hz means an object vibrates one time for a second. For vehicle detection, frequency features are also extracted. But frequency changes if external factors are intervened, changing the vibration. For example, buildings or suspension bridges, exhibit a dynamic response that is severely shaken when seismic waves are received. As described above, all objects show dynamic fluctuations in the degree of disturbance, but are dynamic, and this phenomenon of vibration from the outside is called force vibration.

Meanwhile, even though the object is not disturbed, each object has its own vibration characteristics, and this unique vibration characteristic is called natural or eigen/vibration or frequency [24]. In reality, the object does not vibrate unless dynamic forces are applied from the outside. Thus, natural vibrations refer to the inherent dynamic nature of the object.

Natural vibration is a unique value that never changes once an object's shape, material, and constraints are determined. And the number of natural vibration exists as much as the degree of freedom of the object. For example, the clock of a wall clock has only one natural frequency because there is only one degree of freedom. For another example, a thin metal plate with one end fixed to the wall cannot vibrate without any dynamic force from the outside. But when pressing and releasing the other end, the plate vibrates up and down. The natural vibration representing the oscillating speed is the number of cycles oscillated per unit time, and one cycle is defined as starting and returning vibration to the initial position. In this case, it is an resilient continuum body which has infinite freedom because it has infinite degrees of freedom.

It is easy to understand the definition of a cycle when we consider the circular motion of the earth rotating around the sun. In this case, the rotation of the cycle corresponds to 2 * pi radians. Therefore, the natural frequency corresponds to the angle rotated per unit time when the vibration motion is converted into the circular motion. If we oscillate with one oscillation cycle per second, the natural frequency is 2 * pi radians / second, corresponding to 360 degrees. Natural vibration starts at the lowest value and begins with the order first, second, and so on. In particular, the first natural vibration is called the fundamental natural vibration. This refers to a frequency that vibrates in a shape that can be easily deformed when the object vibrates, and indicates a frequency that the object vibrates in a shape that is difficult to deform as the object becomes higher. The deformation shape of an object vibrating at each natural frequency is called a natural mode shape for that natural frequency. When the object vibrates forcibly, the resonance response increases infinitely when the frequency of disturbance coincides with or close to the natural frequency of the object. In addition, after a certain time after the disturbance is removed by forcibly vibrating the object, the object is freely oscillated to the first natural frequency. That is, the first mode shape refers to a vibration shape of an object vibrating at a first natural frequency, and as described above, it means a shape that an object can easily deform. And the dynamic response of the object due to forced vibration is expressed by the combination of all natural vibration modes of the object.

Natural vibration characteristic is expressed by natural frequency, natural mode, and damping ratio. Natural frequency calculated without considering damping is called an undamped natural frequency, and the natural frequency reflecting the damping is called a damped natural frequency. Numerical analysis of natural frequencies and natural modes is called modal analysis. These natural vibration characteristics can be analyzed by experiments, theories and numerical methods, and in the case of experiments, they are measured using resonance phenomena with measuring devices such as oscilloscopes. Theoretical and numerical analysis techniques can be used to solve mathematical expressions that govern natural vibrations by hand or to obtain approximate solutions using numerical techniques such as the finite element method.

Therefore, we should take a look on natural vibration for vehicle. Vehicle's natural vibration comes from suspension such as spring stiffness, shock absorber damping force, and tire vertical stiffness. It is also affected by road condition, speed, loadage, and engine vibration [25].

Another important factor to consider for accurate HAR is that there is no such kind of statistical or probabilistic classification methods which can show complete accuracy. Most of the HAR research focuses on improving the accuracy in the stage of pre-processing, feature extraction and selection, and classification. To reduce errors at most, additional follow-up measures are required based on heuristic where machine learning method cannot solve the problems. A lot of works have proposed their own post-processing methods with complex mathematics showing good correction results. But in our case, it should be considered that our target is to run the HAR on smartphone, so a simple yet efficient method is required to be run in mobile environment in real-time.

1.3 Problem statement

Early stages of HAR for commuting activity did not differentiate vehicles in detail but regard as a single vehicle class. They did not just only unclassified with similar types such as road vehicles of car and bus or subway and train, but merge them all which has apparently different characteristics. But differentiating vehicles is necessary for higher accuracy. For instance, if a person commutes to company driving car, they will only walk few by walking from home to garage and parking lot to company. But when using public transportation, they should walk from home to station and station to company, which will include a lot more walking then driving a car. So using public transportation will increase the amount of walking exercise, which will make one more healthier than the person using a car. A person using a bicycle for commuting has even more amount of exercise while the bicycle is not automatic, requiring human labor. From these reason, in the perspective of health, recognizing each vehicle is preferred.

Most HAR research on smartphones mainly uses accelerometer and shows considerably high accuracy in recognizing ambulatory activities, such as staying, walking, and jogging [26-30]. However, it shows low accuracy in recognizing when a user is in vehicles such as in a bus, car or subway [31–33]. That discrepancy in accuracy occurs because different physical activities involve certain significant physical differences that are indicated in the magnitude and pattern of acceleration signals. When people are inside a vehicle like a bus or subway, they either stand or sit, which are static activities, relatively motionless.

One major consideration using smartphone is that the position and orientation is not fixed. Some people put it in trousers or jacket pocket. Some will put it inside a bag, where a bag is also divided such as backpack or handbag. Finally, some people will hold the phone, simply holding it or using it. Therefore, it is quite difficult to distinguish between standing/sitting on a static floor or being in a vehicle when the sensor is not fixed; no large difference appears in the patterns of raw acceleration signals while not knowing the orientation. Therefore, methods to offset position and orientation is necessary for smartphone based HAR.

As mentioned on section 1.2, we will have to take a look on vehicle's natural vibration. This can be done by taking a look on frequency signal. Inertial sensors on smartphone, accelerometer, mainly measures the acceleration force. This is a time domain signal that we can observe the pulse of the signal as time goes by. By using Fourier Transform, we can convert this into frequency domain signal. Many kinds of features are extracted on existing works. They extract mean which returns average of all frequency in segmented window, standard deviation of all frequency in segmented window, energy of all frequency, spectral entropy, energy proportion which returns proportion of energy in different frequency band or first and second peak frequency energy.

However, these features possess some problems. First, mean and deviation is a statistical feature. It cannot be applied on frequency domain. In time domain signal, the signal shows the variation in chronological sequence. Therefore, these statistical features can be applied. But in frequency domain, all the individual frequency possess different meaning, where the dimension is different. This cannot be treated as same as time where it only shows single dimension. Secondly, referring all frequency energy is inefficient where some of the frequency are meaningless to refer, and only increases the dimension of feature vector. This is applied same to dividing proportion of energy, where meaningless frequency will be included in the divided proportion.

Third, taking an look on first or second peak frequency energy is not appropriate that peak frequency will be changed due to environmental factors such as road conditions. Therefore, a method to extract the natural frequency of the vehicle itself is necessary.

Meanwhile for post-processing method, it is used to correct the result from the data driven machine learning based classification. Most of the work uses majority voting method by referring predefined set of windows. This will cause inefficiency by referring unnecessary data, and also has problem when voting shows equal.

In summary, for a practical mobile activity recognition for commuting activity, position and orientation problems must be solved, features which can well represent the nature of the vehicles should be extracted, and to overcome the confusion among static activities in smartphone environment after machine learning based classification, an efficient post-processing method is required which can be used in smartphone.

From the aspect of difficulties on distinguishing vehicles on different position of phone, some of the activities to be recognized in this thesis are static. These are the motorized vehicles such as car, bus, and subway, and non-motorized activities such as stay and bicycle. When people are inside a vehicle like a car, bus or subway, they either stand or sit, relatively motionless. By examining the raw sensor signals in these activities, no discriminative differences in pattern are shown. Not only vehicles but also stay is a static activity. Finally, when riding a bicycle, smartphone located in trousers pocket shows periodic signal pattern by cycling, but when the phone is located in jacket pocket, backpack and hand, this also shows similar patterns to vehicles and stay. This brings the confusion among five different activities. This discrepancy in accuracy occurs because different physical activities involve certain significant physical differences that are indicated in the magnitude and pattern of acceleration signals. To solve this problem, we must find the discriminative features that can best distinguish the vehicles by inertial sensor signals.

From the aspect of classification error on machine learning, even though using optimized features, no such classification method can show complete classification performance. And even though it showed perfect performance in off-line process, in on-line process, the data is different compared to the training data, and also the optimized feature will change, which eventually deteriorates the performance. A method is required to pull up the performance by changing which features to extract, changing the classification method, or applying post-processing to revising the final result in real-time.

1.4 Contribution

In this thesis, a commuting activity recognition using accelerometer, gyroscope and GPS data from a smartphone to recognize stay, walk, jog, in a bus, subway, car and riding a bicycle is proposed. These activities are chosen to be the basic activities in daily life commuting that occur frequently and can thus be used as source data for a life-log. Contributions of the thesis are represented as follows.

From the aspect of extracting natural vibration features, as the vehicles shows similar sensor signal pattern in time domain, using time domain signal features cannot well distinguish vehicles where it is hard to find periodic patterns. By assuming that different kind of vehicles may have different kind of natural vibrations, proposed method extracts natural vibration features based on frequency domain features. In here, each vehicle shows different dominant frequency amplitude, and also even in different positions, this does not change.

From the aspect of applying post-processing, an activity decision algorithm as a post-processing is proposed to revise the classification result from machine learning based classification. The purpose of this correction algorithm is to maintain the current activity result unless it confirms that it is truly changed. It refers to previous activity, corrected activity and current activity to determine the final result. It also uses GPS to aid the classification based on speed and uses coordinates to classify other vehicles from ground section subway. This method does not require complex computation, which can well fit for real-time mobile application.

1.5 Thesis Organization

The thesis is organized in five sections. Section 1 introduces the background of human activity recognition and different means of recognition such as video based, environmental sensor based and wearable sensor based. Then the motivation for this thesis is explained of the advantage of automatic commuting activity recognition, the difficulty of recognizing and classifying commuting activity such as public transportation using smartphone, and the necessity of post-processing. Finally, contribution is described. Section 2 introduces related works from traditional wearable sensor based HAR, HAR using smartphone, human commuting activity recognition using smartphone, and post-processing. Section 3 describes the proposed method of human commuting activity recognition using smartphone with activity correction. At first in pre-processing, projecting acceleration signal to earth coordinate, smooth filtering, and changing time signal to frequency signal is described. On feature extraction stage, sensor signal is analyzed to extract natural vibration features from frequency domain signal from vehicles. Finally, a correction method to correct the error from classification is introduced. Section 4 describes the experimental environment of how and which data is collected, subjects and the route. Then experiments are conducted by comparing different classifiers, comparing different research, and finally tested on differently collected data to check overfitting. And then regularization and optimization is performed for model generalization. And lastly, conclusion and future directions are described in section 5.

2. Related Works

2.1 Wearable Sensor based HAR

Research on HAR using wearable sensors has long been undertaken, with various accomplishments. HAR was initially achieved by attaching individual sensor devices to various body parts, so called body sensor network. HAR can be categorized with different criteria such as what types of sensors to use, how many sensor devices to use, which position to attach the sensor device on the body, whether it is real time recognition or not, whether to recognize overall body movement, posture or gesture, what kind of activities to detect, handling the problems of sensor displacement, etc. In this section, research is classified based on the number of sensor devices.

2.1.1 Using single sensor device

Karantonis et al. [34] placed a single sensor on waist detecting walking and falling. The sensor device was mounted with wireless communication device where it showed the result in real-time. They have also detected the transitions between activities. Mathie et al. [35] also placed the sensor on the waist to detect falling, walking, sitting, standing and lying. They have adopted hierarchical classification where it first divides activity and rest, activity to fall, walk, postural orientation, others, and postural orientations to sitting, standing and lying. Yang et al. [36] placed the sensor on the waist to detect locomotions and also ADLs such as walking, running, scrubbing, standing, working at a pc, vacumming and brushing teeth. They have also adopted hierarchicl classification as dividing into static and dynamic activities. They have used neural network for classification. Bonomi et al. [37] has placed the sensor on lower back to detect lying, sitting, standing, working on a computer, walking, running and cycling. They have used decision tree as a classifier with different time segmentation which are 0.4, 1.6, 3.2, 6.4 and 12.8 second. Gupta et al. [38] has placed the sensor on waist to detect transitional activities such as walking, jumping, running, sit-to-stand and vice versa, stand-to-kneel and vice versa and still. They have compared feature selection methods such as Relief-F and sequential forward floating search (SFFS) and also compared Naive Bayes and KNN. Massé et al. [39] has placed the sensor on trunk to detect sitting, standing, walking and lying for mobility impaired stroke patients. They have additionaly used barometric pressure sensor. Frank et al. [40] has placed the sensor on the belt and recognized sitting, standing, walking, running, jumping, falling and lying using Bayes network. Garcia-Ceja et al. [41] has placed the sensor on wrist to detect long term activities such as shopping, showering, dinner, working, commuting and brushing teeth. They have compared the result with Hidden Markov Models and Conditional Random Fields. Khan et al. [42] have used accelerometer placed on chest recognizing lying, sitting, standing, walking and running, and transitional activities between lie, stand sit and walking. They have used hierarchical classifying as using Artificial Nerual Network for lower level activities and Autoregressive modeling the higher level activities. Casale et al. [43] have used accelerometer placed on the chest to recognize stair, walk, talk, stand and working on a pc. They have used random forest for classification. Long et al. [44] used accelerometer placed on waist to recognize walking, running, cycling, driving and sports. They have used principal component analysis to remove feature correlation and reduce feature vector dimension with Bayesian classification.

2.1.2 Using two sensor devices

Parkka et al. [45] has placed sensors on wrist and chest to detect lying, sitting, walking, rowing and cycling. They have compared using custom decision tree, automatically generated decision tree and artificial neural network. Lyons et al. [46] has placed the sensor on thigh and trunk to detect sitting, lying, standing and moving. They have detected the posture based on the inclination threshold. Yeoh et al. [47] has placed the sensor on wrist and two thighs to detect sitting, lying, standing and walking speed. They have divided the classifier for postural activity and movement activity. Salarian et al. [48] placed the sensor on trunk and two shanks to detect lie, sit, stand, walk and sit-to-stand and vice versa activities based on Parkinson's disease patients. Mannini et al. [49] used accelerometer placing wrist and ankle recognizing ambulation, cycling, sendentary and others. They have concluded that error from wrist based recognition can be corrected with ankle based recognition.

2.1.3 Using three sensor devices

Olguın et al. [50] has placed the sensor on wrist, hip and chest to detect sitting, running, walking, standing, lying and crawling. Chamroukhi et al. [51] has placed the sensor on chest, thigh and ankle to detect stairs, walking, sitting, standing up and sitting on the ground. They have adopted automatic temporal segmentation and expectation-maximazation algorithm. Moncada-Torres et al. [52] have placed the sensors on chest, thigh and ankle to detect 16 ADLs such as whole body movements and gross and dextrous upper-limb activities to compare the importanace of sensor placements. Attal et al. [53] have placed the sensors on chest, right thigh and left ankle and compared the result with different classification methods such as KNN, SVM, RF and HMM, GMM and K-Means. Pirttikangas et al. [54] has placed sensors on thigh, necklace and both wrists to detect typing, watching TV, drinking and stairs. They have also collected heart rate data. Based on the forward-backward sequential search based feature selection, sensor signal from necklace was shown to be important. Banos et al. [55] dealt with sensor displacement problems, which is also a classic problem in the area of wearable sensor based HAR. They have compared to put the sensor in ideal setting, placing the sensor wrongly by the user oneself who doesn't know the ideal position, and in the case when the sensor slips from the ideal position.

2.1.4 Using over four sensor devices

Gao et al. [56] has placed the sensors at chest, waist, thigh and side to detect standing, sitting, lying, walking and transition between activities. They have compared the affect of different sensor locations. Gjoreski et al. [57] has placed sensors on thigh, waist, chest and ankle to detect lying, sitting, standing and transitional actitivies. They have also conducted an experiment to check placement of sensor best detects the fall. The most remarkable research was conducted by Bao et al. [58]. They attached five 2D accelerometers on the forearm, wrist, pelvis, knee, and calf and recognized activities such as walking, running, standing, sitting, watching TV, cycling, eating, and reading. They showed an average of 84% accuracy independent of user. Atallah et al. [59] has used seven sensor devices to check which position is the best for HAR using accelerometer. They have placeed the sensors on ear, chest, arm, wrist, waist, knee and ankle. They have divided the activity into groups such as very low level activity, low level activity, meduim level activity, high level activity and transitional activity. Recognized activities were postures such as lie, locomotions such as walk and run, and others for ADLs. From their experiment, sensor on knee showed best for high level and transional activity. Cleland et al. [60] used six sensor devices to check which position is the best for HAR using accelerometer. They were placed on chest, wrist, lower back, hip, thigh and foot. Walking, running, treadmill, sitting, lying, standing and stairs were recognized. From the experiment, hip showed the best result when using alone, and using over two sensors did not show big difference on accuracy.

A lot of work has been proposed using wearable sensor based HAR. The number of sensors and their attachment place is one of the important aspect for highly accurate HAR. But the research trend has changed to use mobile phone after the emergence of smartphone. It has great advantage having multiple types sensors embedded on it, and most of the users carry them where obtrusiveness problem is solved. As the characteristic of smartphone is that position and orientation is free, sensor displacement research has faded away.

2.2 Smartphone based HAR

After smartphone has come to the surface, direction of the research has changed to use smartphone itself. The challenges for smartphone-based HAR are (1) deciding what kind of smartphone sensors to use, (2) handling the position and orientation problem, (3) saving energy to reduce battery consumption, and (4) using additional sensors along with smartphone. At the early stage of the research, the phone served merely as a data collection device. The data were either transmitted directly to the server through online communication or connected to the server after data collection was complete. In both cases, the server performed the whole recognition process. As smartphone performance increased, the recognition process moved to the smartphone itself.

2.2.1 Types of sensors

Most HAR research uses only an accelerometer, but some studies have used other inertial sensors. Shoaib et al. [61] used an accelerometer, gyroscope, and magnetometer to evaluate recognition results using different combinations of sensors. The accelerometer and gyroscope showed reasonable results even when used independently, but the magnetometer was not sufficient to use alone because of the device orientation problem. Khan et al. [62] used a pressure sensor and microphone along with an accelerometer. They used the microphone based on other research [63]. The pressure sensor was used to track altitude—particularly the relative altitude between different points, which could be helpful in recognizing activities that result in an increase or decrease in altitude, such as climbing or descending stairs. Shoaib et al. [64] experimented on combination of different kind of inertial sensors which are accelerometer, gyroscope and magnetometer. As a result, magnetometer has only small contribution while accelerometer and gyroscope affects tremendously. Martin et al. [65] used all kinds of sensors such as accelerometer, gyroscope, magnetometer, gravity, Linear acceleration and orientation. They have compared the experiment with different combination such as features and classification methods. Ryder et al. [66] have used accelerometer and GPS to check ambulatory. The collected data are sent to server and processed. Kim et al. [67] has used accelerometer, gyroscope and magnetometer to recognize walking, stairs, running and motionless inside the building. They have used hierarchical SVM and measured enery expenditure of the user. Ouchi et al. [68] have split the activities into indoor and outdoor. Indoor activities includes walking and resting and some ADLs such as vacumming and brushing teeth. Outdoor activities include resting, walking, running and boarding. Two different classification methods are switched to recognize actitivties based on GPS. Zhao et al. [69] used accelerometer and magnetometer. They have divided the activity into two classes which are periodic and non-periodic, and then with the divided result, sencond classifier is applied.

2.2.2 Position and Orientation

Some researchers tried to achieve position or orientation independent recognition. Anjum et al. [70] collected data with the phone in different orientations. To solve the orientation problem, they rotated the three orthogonal reference axes to align with the three transformed axes with eigenvector corresponding to the axes in descending order of signal variation. Henpraserttae et al. [71] proposed a projection-based method for device coordinate estimation to handle the device orientation problem. They transformed all input signals into a single global reference coordinate system. Lu et al. [72] used orientation-independent features, one-time device calibration (which can be perceived by the user), and classification techniques in which activities are modeled by splitting them into several sub-classes, each of which is associated with particular body positions. Khan et al. [73] used features including autoregressive coefficients and signal magnitude area are calculated. Kernel Discriminant Analysis is then employed to extract the significant non-linear discriminating features which maximize the between-class variance and minimize the within-class variance to handle different positions of smartphone. Chen et al. [74] used diverse classifiers for both personalized and generalized model, and experimented in terms of various placement settings, sensitivity to user space, stability to combination of motion sensors, and impact of data imbalance. Thiemjarus et al. [75] placed the smartphone on chest, waist and both side of trousers pocket with different orientations. Based on the result, the highest accuracy showned one waist, even higher than combining all of the positions. Trousers pocket and combining all didn't show large difference while chest showed the lowest accuracy. Khan et al. [76] placed the sensor on either side of trouser's front and rear pocket, and jacket's inner pocket. Based on the result, the accuracy on any or the trouser's pocket was similar while jacket pocket slightly decreased when used with Kernel Discriminant Analysis (KDA). Otherwise, without using KDA, position in jacket showed lower than trouser's pocket with some difference. Yang et al. [77] have proposed position-independent method called PACP (Parameters Adjustment Corresponding to smartphone Position), where features were extracted from the raw accelerometer and gyroscope data to recognize the position of the smartphone first; then the accelerometer data were adjusted corresponding to the position; finally, the activities were recognized with the SVM (Support Vector Machine) model trained by the adjusted data. To avoid the interference of smartphone orientations, the coordinate system of the accelerometer was transformed to get more useful information during this process. Experimental results show that PACP can achieve an accuracy over 91%.

2.2.3 Energy saving methods

Some researchers tried to reduce the computational power. Anguita et al. [78] proposed multiclass classification which adapts the standard Support Vector Machine (SVM) and exploits fixed-point arithmetic for computational cost reduction. Lee et al. [79] used hierarchical hidden Markov models for recognition. To reduce the computational power, the recognition models are designed hierarchy as actions and activities, where actions are low-level context such as stand or walk and activities are high-level context such as shopping or taking bus. Reyes-Ortiz et al. [80] presents the Transition-Aware activity recognition where prediction technique deals with transitions either by directly learning them or by considering them as unknown activities. They have combined the probabilistic output of consecutive activity predictions of a SVM with a heuristic filtering approach. Lu et al. [81] proposed using an unsupervised classification method called MCODE to recognize activities including basketball activity.

2.2.4 Smartphone with additional sensors

Some ressearchers tried to use additional sensors along with smartphone. Keally et al. [82] have used both body sensor network and smartphone together with adaboost to improve the performance albeit unobtrusiveness. He et al. [83] used Fisher's discriminant ratio criterion and $J3$ criterion for feature selection to select features among 140 features. Then a hierarchical classifiers including fourteen classifiers were applied. Reiss et al. [84] introduced a new, confidence-based boosting algorithm called ConfAda-Boost.M1. which outperforms commonly used classifiers such as decision trees or AdaBoost.M1. Capela et al. [85] collected data from able-bodied, elderly, and stroke patients. Features were selected among 76 features using three filter-based, classifier-independent, feature selection methods (Relief-F, Correlation-based Feature Selection, Fast Correlation Based Filter). Ronao et al. [86] proposed a two-stage continuous hidden Markov model (CHMM) approach where first-level CHMMs is for coarse classification, which separates stationary and moving activities, and second-level CHMMs for fine classification, which classifies the data into their corresponding activity classes. Random Forests (RF) variable importance measures are exploited to determine the optimal feature subsets for both coarse and fine classification.

As the research shifted to smartphone based HAR, solving the orientation problem became the basic by getting signal magnitude or extracting vertical and horizontal component or projecting to earth coordinate. Also a lot of methods tries to save power consumption by using less type of sensors or reducing the sampling rate which was not shown phenomenon in wearable sensor based HAR.

2.3 Human Commuting Activity Recognition

Numerous works tried to identify vehicles, such as car, bus, subway, train, tram, motorbike and bicycle. They were to recognize more contexts including vehicles with a single mobile phone. Early works did not differentiated vehicles but categorized them as a single activity. Then many types of vehicles were introduced and tried to separate each of them.

Yu et al. [87] have used accelerometer, gyroscope and magnetometer to detect still, walking, running, biking and vehicles. They have extracted time domain features and frequency domain features of the peak value and ration between the first and second peak. And error correction is made based on the scores of series of activities. But they did not differentiated vehicles but classified as a single class. Hao et al. [88] used accelerometer and GPS to detect stay, walking, bicycling and motorized transport. Stay is divided into long term and short term. They have divided activity into moving and stationary for every activity but did not differentiated vehicles. Sankaran et al. [89] used only barometer to detect idle, walking and vehicle. As barometer consumes low power and shows efficiency, vehicle is not classified in detail yet the performance is not impressive. Yang et al. [90] used acclerometer to detect sitting, standing, walking, running, driving and bicycling. They have extracted vertical and horizontal components and extracted features rather than using magnitude. It has shown better result but still the vehicle is not separated. Reddy et al. [91] used accelerometer and GPS to detect stationary, walking, running, biking and motorized trasnport. They have used decision tree followed by first order discrete Hidden Markov Model. The result is good enough but the transport is not differentiated. Fang et al. [92] has used accelerometer, gyroscope, magnetometer to detect still, walk, run, bike and vehicle and adopted deep neural network. For the vehicle data, motorcycle, car, bus and metro data are collected. It showed better result than traditional statistical classifiers but didn’t differentiate vehicles.

Su et al. [93] have used accelerometer to detect bus, subway, car, bicycle, walking and jogging. They have proposed hierarchical classification where wheeled and unwheeled activitiy is divided and used SVM based classification. Siirtola et al. [94] used accelerometer to detect walking running, cycling, driving a car, sitting and standing. Unlike other research, from pre-processing, feature extraction and classification are all made on the phone with active and inactive hierarchical classification. Sonderen et al. [95] used accelerometer, gyroscope and magnetometer to detect running, bike, and car. They have compared with different classifiers to find the best sampling rate to reduce the power consumption, but only has car class in vehicle. Feng et al. [96] compared the usage of accelerometer only, GPS only and using both sensors to detect walking, running, bicycle, motorcycle, bus, car, tram and metro. They have used their own heuristic based classification and concluded that using both sensors shows the best result. Lu et al. [97] used accelerometer, GPS and microphone to detect stationary, walking, running, cycling, and vehicle. Shafique et al. [98] used accelerometer and GPS to detect train, walk, bicycle and car. They have compared with different classifiers such as SVM, AdaBoost, Decision Tree and Random Forest. Shin et al. [99] used accelerometer and GPS to detect walk, tram, train, bus and car. They have proposed hierarchical classification with walk-stop and walk-start mode with vehicle and non-vehicle mode. Nick et al. [100] used accelerometer to detect car, train and pedestrian. The size of the class is too small to say that the result is reliable. Jahangiri et al. [101] used accelerometer, gyroscope and rotation vector sensor to detect bike, car, walk, run and bus. All of the values from each frequency is extracted compared with SVM, KNN and DT.

Han et al. [102] used additional sensors in the smartphone with a hierarchical structure. They recognized staying, walking, running, and shaking using the accelerometer. Shaking indicates either a bus or a subway, and they used the microphone to distinguish between them. Ashqar et al. [103] used accelerometer and GPS to detect bike, car, run and bus. They have extracted time and frequency domain features. From the frequency domain, they have extracted all of the values from frequency. Random Forest based classification was done first and then with the first and second highest probability, SVM is performed. Lari et al. [104] used accelerometer and GPS to detect car, bus and walk. They have used Random Forest and included the speed as feature. But only three class is not enough for practical usage. Balli et al. [105] have used accelerometer, gyroscope and GPS to detect walking, running, biking, bus and car. Speed is used for feature acquiring max and average speed. Experiments using different combination of sensors was performed where the result showed accelerometer is mandatory. Lee et al. [106] has used accelerometer and microphone to detect bus, subway and taxi. They have used the audio data when the accelerometer based result is concluded as transportation. Then the final activity is concluded based on sound. Chen et al. [107] has used accelerometer, gyroscope, magnetometer and pressure sensor to detect bus, car, metro and train. They have proposed to use multiple SVM on each pair combination of activities. Yanyun et al. [108] used accelerometer, gyroscope, magnetometer and pressure sensor and input to Convolutional Neural Network. Time and frequency domain are extracted first and input to CNN. The result is much higher compared to using traditional methods.

Byon et al. [109, 110] has used only GPS and classified the activity using artificial neural network. Bus, car, bike and walk is classified. The data was clearly collected within the same route which may posses the possibility of overfitting.

Dabiri et al. [111] also used only GPS and classified based on CNN to detect walk, bike, bus, driving and train. It showed better result than traditional statistical classifiers. Hedemalm et al. [112] used accelerometer and gyroscope and segmented window into 5 seconds. The purpose is to reduce energy, so they have included sleep mode between activities and consider the unrecognized activities as previously recognized activities. Manzoni et al. [113] collected accelerometer and GPS to recognize walking, bike, metro, bus and car. From the frequency domain signal, they have extracted coefficients with high frequency magnitude. Zheng et al. [114] has used the GPS trajectory data to detect walking, taking a bus and bicycle, and riding a car. Using only the GPS has weakness that it shows lower results than using accelerometer. Xiao et al. [115] used GPS to recognize walk, bus, taxi, bike, car, subway and train. They have extracted global and local features from trajectory data and applied ensamble classification using Random Forest, Gradient Boosting Decision Tree and XGBoost. But they did not classify road vehicles. Endo et al. [116] has used GPS and DNN to recognize bus, train and walking. The GPS is not used as raw data but trajectory image is input for DNN. But the accuracy was not good representing that using trajectroy image data to DNN is not good. Stenneth et el. [117] used GPS and GIS to detect car, bus, ground level train, walking, bike and stationary. When transportation network features are not used, the accuracy is poor. But when using them, it goes higher and reasonable. Semanjski et al. [118] used GPS data from 8,000 subjects. With this tremendous amount of data, the accuracy is very high compared to other research which only uses the GPS data. But it did not test on new area where the route is different. Bolbol et al. [119] also only used GPS data to detect car, walk, cycle, underground, train and bus. They have applied moving window SVM classification but still the accuracy is not good. Gong et al. [120] used GPS data to detect walk, car, bus, subway and commuter rail. They have classified hierarchically as train like, road vehicles and others. Gonzalez et al. [121]

used GPS data and neural network to detect car, bus and walk. Result is good enough but has limitaion on having only three activities. Nitsche et al. [122] used accelerometer and GPS to detect walk, bike, motorcycle, car, bus, tram, metro, train and wait. But the train-like activities showed poor performance, and when the GPS signal was lost, car and motorcycle showed very poor accuracy.

The trend of studies to detect commuting activities has changed from using traditional statistical classifiers to use deep neural networks. The usage of sensor also divided by only using inertial sensors, inertial sensors with GPS and only using GPS. In most cases, using only GPS shows poor accuracy than others and accelerometer is the must have sensor for accurate detection.

2.4 Post-processing

Post-processing is to alleviate the error of machine learning based classification. Research can be divided into three ways as using only machine learning methods, adopting heuristic methods, and using both methods

Wenand et al. [123] adopted Adaboost, to retrain the model by updating the weights and conclude the final activity based on posterior probability. Wang et al. [124] have claimed WOODY, where activity is corrected based on the observation probability and weight of each state in observation vector. San-Segundo et al. [125] has used Moving Mean (MM) based low-pass filter for feature. In this case, it will reduce the noisy variations between consecutive frames.

Zheng et al. [126] proposed to refer previous activities to choose the current activity. From a given activity subset (2n+1), the most frequent activity is chosen and set to final activity. Zhong, et al. [127, 128] also proposed to use previous activity. Based on the minimum confidence value set as threshold, the current activity is changed if it exceeds the threshold. Otherwise, Markov smoother backtracks previous activity. Parkka et al. [45] has used median filter and referred 31 seconds of window to rather change the activity or not. This is because recognized activities in this research are continuous activity, and it will prevent from short transitional activities. Wang et al. [129] has updated activity by majority voting looking 1 minute duration due to continuous activity. Because of this long period, the comparison experiment started after 1 minute. Performance is good referring lots of data but is inefficient. Grzeszick et al. [130] sets five past activities to refer. The problem comes when majority voting shows equality. In this case, they haven't updated current activity. Han et al. [131] has first calculated likelihood and than make the transitions of the previous activity to new activity only when the likelihood is maximum for at least eight consecutive windows.

Machine learning based methods are more complex than heuristic based methods. These method usually refers probability based on the base classifier. This is a good approach that mostly first and second highest class has majority of the probability. But the problem comes when it comes to real world scenario with unseesn circumstances that it cannot handle. Meanwhile, heuristic approach is simple. Although the performance will not be better than machine learning based approach, it can reflect unseen circumstances which is special characteristic of human thinking.

3. Proposed Method

3.1 Overview

In this section, we introduce our methodology to recognize stay, walk, jog, and riding a bus, car, subway, and bicycle in real-life situations by using a single smartphone without considering any specific placement or orientation. For analysis of the problems, we have used the data collected by ourselves. Figure 3-1. shows the overall architecture of our proposed method.

First acceleration signal, angular velocity signal, speed, latitude and longitude data is collected from smartphone. Then inertial data are sent for pre-processing such as projecting acceleration signal into earth coordinate, combining different axis into one to refer only the magnitude variation, and smoothing by order 3 moving average filter. Then features are extracted from time domain and frequency domain. In time domain, general statistical features were selected and extracted. In frequency domain, natural vibration features were extracted. Before classfication based on statistical classifier, GPS speed based classification is undergone. If the speed is low, it could be one of all activities where whole activity based classification is made. If the speed is high, it would be one of car, bus, subway and bike. Stay is also included in here due to the possibility of misclassification. When the speed is high, coordinate data is used to distinguish subway and other vehicles when running ground section. After classifying activity, to revise errors from pre-defined classification, post-processing is undergone, namely activity correction. Then with proposed correction rule, current activity, previous activity and previously corrected activity result is referred for final revision.

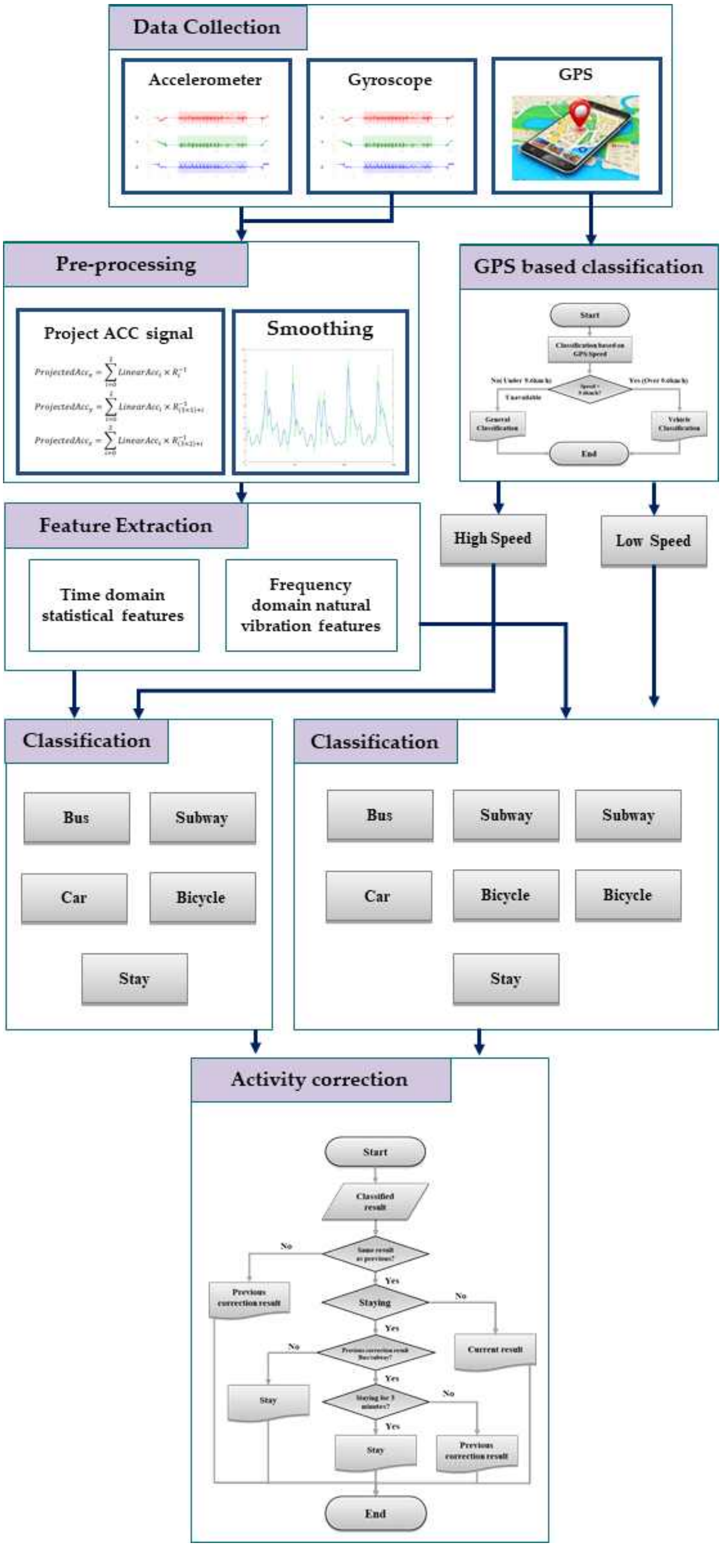


Figure 3-1. Proposed overall architecture

3.2 Pre-processing

Before the emergence of smartphones, HAR researchers attached individual sensor devices on the body such as torso or limbs and extracted features based on the axis with the pattern most characteristic of an activity. However, this method is not applicable for smartphones because people do not carry their smartphones in a fixed position and do not consider the orientation. Even in the same activity with same axis, the signal differs due to unfixed orientation. Thus, a method to offset the directional information is required. Total Acceleration (TA) [132], which is a vector sum of three axis signal into one is used to solve this problem. It powers each axis signal and apply square root on the sum of powered axis. In this way, only the magnitude variation of the signal can be referred, where it can easily distinguish walk, and jog which shows large differences in pattern and magnitude. TA is both applied for acceleration signal and angular velocity signal. Equation 3-1 is TA equation which shows how to get overall signal magnitude.

$$Total\,Acceleration = \sqrt{signal_x^2 + signal_y^2 + signal_z^2} \tag{3-1}$$

If we take a look at the signal drawn from both acceleration and angular velocity signal, we can notice that these two are resembled. It is clear that both signals are measuring different values for different purpose. But we cannot say that these two are not related at all. This is because human being cannot move straightly without any slight rotation. When walking or jogging, hands and feet will move back and forth, and the body will keep the balance as the center, which causes minimum rotation. Therefore, acceleration signal represents the variation of magnitude of moving. For example, when walking, when one of the leg goes lifts from the ground and goes forth, the magnitude of acceleration signal increases. And when the lifted leg stops going forth and comes down to the ground, the magnitude of acceleration signal decreases. This variation is recorded in accelerometer. In the same time, when lifting the leg to forth, there will be a rotation of lower body part, and vice versa when leg coming down. Figure 3-2. illustrates the acceleration signal magnitude while Figure 3-3. illustrates the angular velocity signal magnitude of all the activities using our collected data in TA.

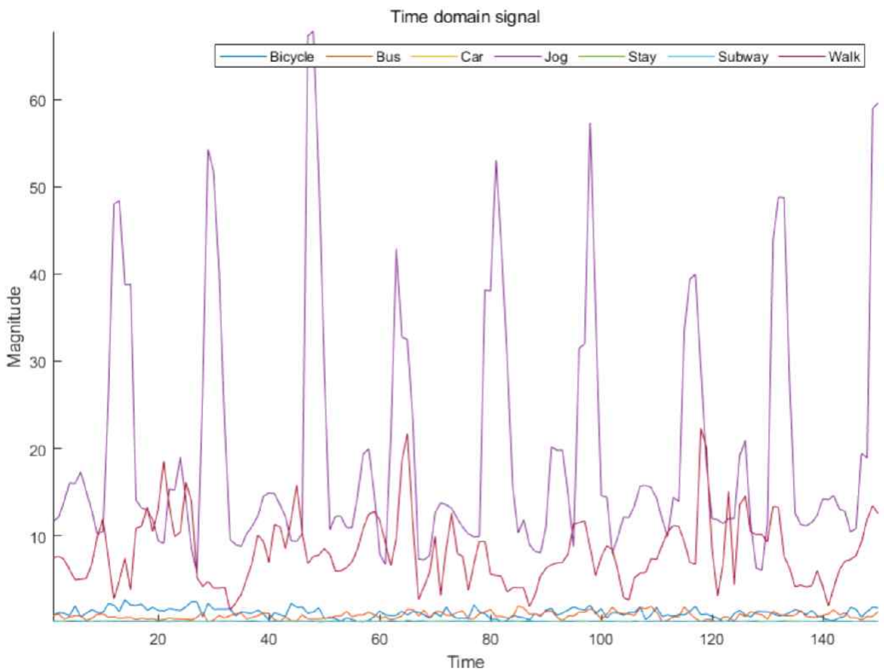


Figure 3-2. Comparison of all activities using acceleration magnitude signals based on TA

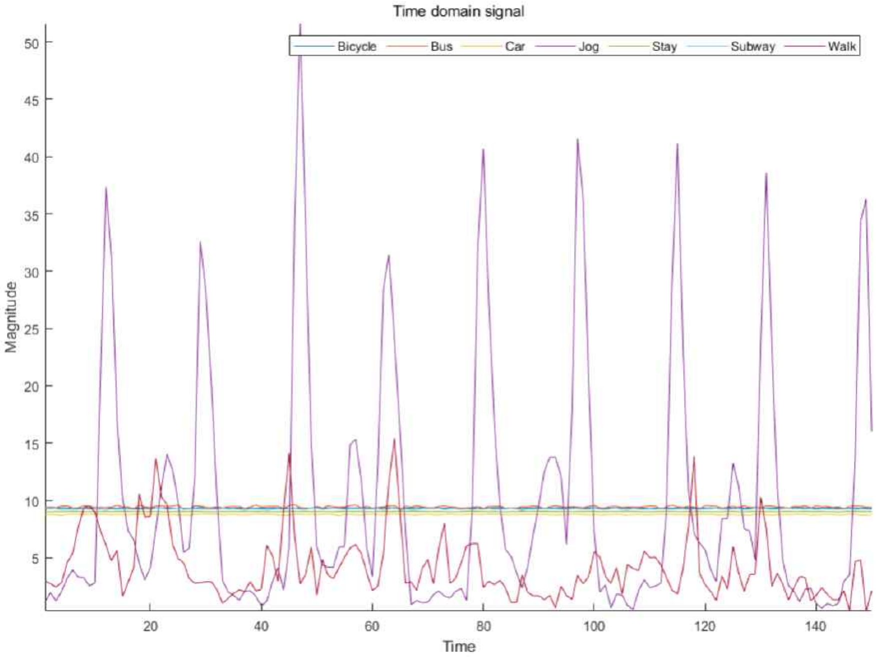


Figure 3-3. Comparison of all activities using angular velocity magnitude signals based on TA

Although using TA can distinguish walk and jog well, it is clear that the rest of the five activities doesn’t show big difference. This is because only walk and jog activities are dynamic while the others are static. For example, while staying inside a car, bus, or subway, one do not move but stand or sit still inside the vehicle. A little bit different but similarly, when riding a bicycle, legs are moved for pedaling, but the body itself is fixed on the saddle. So the signal difference comes from the vibration of the vehicle itself, not the users movement. Looking into the time domain signal, we can see that static activities are in the range of magnitude from about zero to two for the acceleration signal, and about eight to ten for angular velocity signal. Figure 3-4. and Figure 3-5. illustrates signal of five static activities of acceleration and angular velocity signal based on TA respectively.

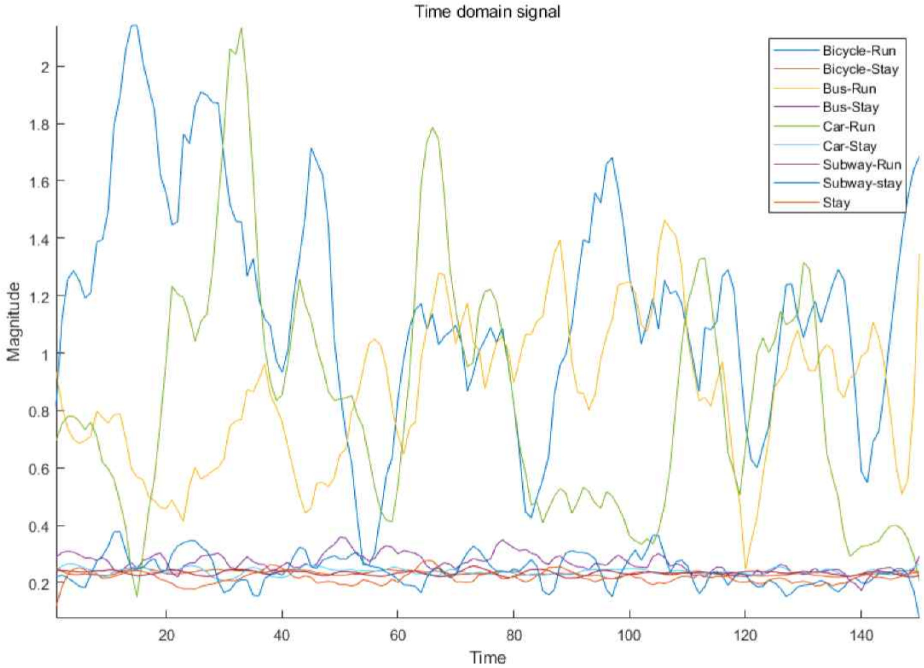


Figure 3-4. Comparison of five static activities using accelerometer magnitude signals based on TA

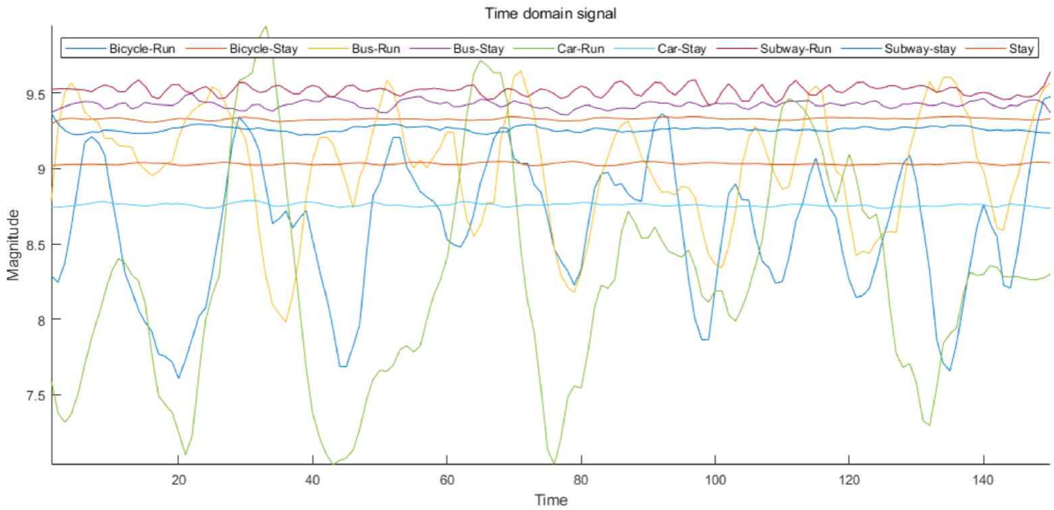


Figure 3-5. Comparison of five static activities using angular velocity magnitude signals based on TA

As shown in the graph, only the bike shows difference while the other four activities show minimum to low difference. It is clear that it will be hard to differentiate vehicles with TA signal. It is noticeable that running bicycle also shows similar acceleration magnitude to running bus and car that they can certainly be divided as vehicle with engine or not. There exists quite large vibration when in the running bus and car. For the bicycle case, this can be considered as pedaling. On the contrary, while car and bus are running by fuel and having an engine, subway is run by electricity, which uses a motor. We can figure out from the electric car launched these days that they do not make noises or vibration, so that speaker makes artificial engine sound to warn people that a car is passing by. Motor gives the electric power to the coil, where it rotates the drive shaft based on the magnetic difference. On the other hand, engine ignites the fuel and uses the explosion power to rotate the cylinder, which rotates the drive shaft, and finally makes the vibration.

We have made the hypothesis that each vehicle will have a natural vibration that can be used to accurately determine the corresponding activity taking place. For this, we need to take a look into the gravity axis. Because user doesn’t move inside the vechile, acceleration signal for forth and back and left to right will not highly affect but up and down due to vibration. Existing research used accelerometer magnitude for cases in which the direction of a smartphone was not fixed, because accelerometer magnitude offsets the effect of the axis. However, this method cannot reflect the natural features that occur in each axis. A method was proposed to solve that problem by extracting vertical and horizontal features from an accelerometer magnitude signal. However, that method could not perfectly restore the original characteristics when reversing, which degraded the accuracy of vehicle recognition. Thanks to the smartphone environment we are using, android API provides the rotation vector based on magnetic field. We can obtain the 3x3 rotation matrix to represent the angles relative to magnetic North as show in Figure 3-6.

$$\begin{bmatrix} \cos\theta\cos\psi - \sin\Phi\sin\psi\sin\theta & \sin\Phi\cos\theta & \cos\Phi\sin\psi + \sin\Phi\cos\psi\sin\theta \\ -(\sin\Phi\cos\psi + \cos\Phi\sin\psi\sin\theta) & \cos\theta\cos\Phi & -\sin\Phi\sin\psi + \cos\Phi\cos\psi\sin\theta \\ -\sin\psi\cos\theta & -\sin\theta & \cos\Phi\cos\theta \end{bmatrix}$$

Figure 3-6. Rotation matrix

where $\Phi = azimuth, \theta = pitch,$ and $\Psi = roll.$ Based on the rotation matrix, we can calculate the pure acceleration value by multiplying two of them from the following formulas where R_i is rotation matrix and ACC_i is 3D accelerometer vector. Equation (3-2) is used to obtain the X-axis corrected to the east–west direction; Equation (3-3) is used to determine the Y-axis corrected to the north–south direction, and Equation (3-4) is used to calculate the Z-axis corrected to the up–down direction. In this way, all three axes of the accelerometer show zero value while placing still in any direction and will lead us to gain the magnitude of gravity axis (Z-axis) when the phone is moved in up and down direction.

$$ProjectedAcc_x = \sum_{i=0}^{2} LinearAcc_i \times R_i^{-1} \tag{3-2}$$

$$ProjectedAcc_y = \sum_{i=0}^{2} LinearAcc_i \times R_i^{-1} \tag{3-3}$$

$$ProjectedAcc_z = \sum_{i=0}^{2} LinearAcc_i \times R_i^{-1} \tag{3-4}$$

Normally, sensor signal contains lot of noises. To smooth this, average moving filter with order 3 is applied. Figure 3-7 shows the before and after acceleration signal applying average moving filter. Green line shows the original signal while blue line shows the smoothed signal. As shown in the figure, big spikes which are noises are reduced. And also, the signal apex of the signal is smoothed from sharp point.

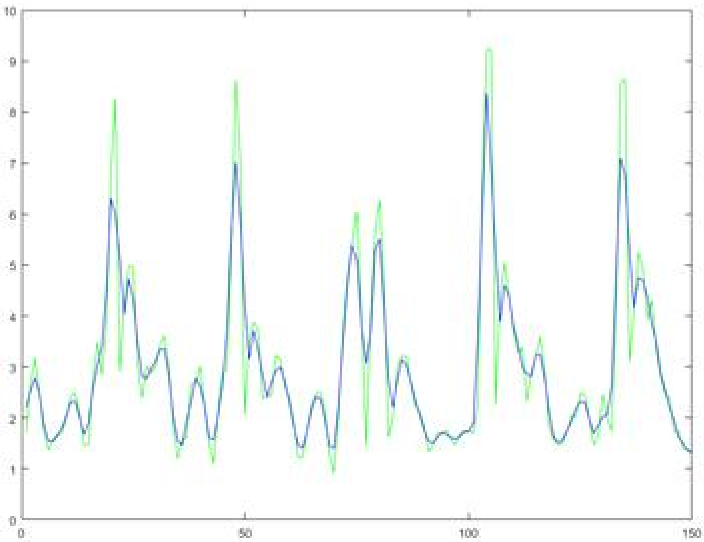


Figure 3-7. Example of Acc signal applying average moving filter
Green line is the original signal while blue line is the smoothed signal

As mentioned before, the acceleration signal must be converted to frequency signal to refer the natural vibration frequency. For this, we have applied Fast Fourier Transform (FFT) on acceleration signal [133] Frequency is how many times a periodic phenomenon has occurred in a unit of time, which means a number of repetitions occurring per second and represented as hertz (Hz). For reference, RPM (Round Per Minute) is the number of repetition that occurs per minute, and because the time of one repetition is called a period, the frequency is also the inverse of the period. Frequency analysis is a readout of patterns over time and is used as a fundamental data for finding the exact cause of the occurrence and establishing appropriate measures, which is why it is used. Discrete Fourier Transform (DFT), when given n different complex values, is the process of converting these into n different complex values in a certain way. Reverting to that reverse is called IDFT by attaching the word "inverse". DFT takes much time with complexity $O(N^2)$ where FFT reduces this to O(NlogN). There are many methods of FFT where Cooley-Tukey algorithm is widely used.

3.3 Feature Extraction

Several features are selected manually to discriminate activities. Mainly extracted activities are time and frequency domain features. Normally, time domain features are extracted with statistical features which can show the change of signal as time goes by. On the contrary, each frequency is a different dimension, where statistical feature extraction cannot be applied which makes no sense. Therefore, we only extracted natural vibration features.

3.3.1 Time domain features

For features we use to classify from time domain signal, we extracted 16 types of features including mean, standard deviation, max, min, zero crossing, mean crossing, range, interquartile range, median, median absolute, median absolute deviation, covariance, cross correlation, correlation coefficient, skewness, kurtosis, and trim mean. These methods are differently applied for each signal and signal sum. For example, TA from both acceleration signal and angular velocity signal, all of the feature extraction methods are applied except zero crossing, covariance, cross correlation and correlation coefficient. In TA signal, all of the values are over zero that zero crossing cannot be count. And the rest of the features require two different signals where TA has only one. Therefore, twelve features are extracted from TA signal. For each individual axis on both sensors such as accelerometer X, Y, Z and gyroscope X, Y, Z, all of the features are extracted except those which need two signals. These are extracted from each pair of each sensor, such as accelerometer X and Y, Y and Z, and X and Z. This is also applied to gyroscope. All of these features are extracted from 3 second based segmented signal. The number of extracted features are: 13 from TA accelerometer, 13 from TA gyroscope, 14 from accelerometer X, 14 from accelerometer Y, 14 from accelerometer Z, 9 from accelerometer pair, 14 from gyroscope X, 14 from gyroscope Y, 14 from gyroscope Z, 9 from accelerometer pair. So the total number is (13 x 2) + (14 x 6) + (9 x 2) = 128 features. Table 3-1 shows the list and description of each feature.

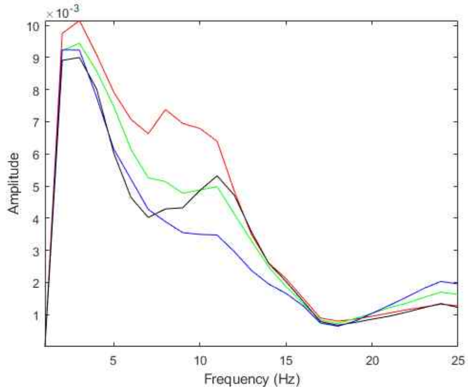
Table 3-1. Extracted time domain features

Feature	Description
Mean	The DC component (average value) of the signal over the window
Standard Deviation	Measure of the spreadness of the signal over the window
Max	The maximum signal value over the window
Min	The minimum signal value over the window
Zero crossing	The total number of times the signal changes from positive to negative or back or vice versa normalized by the window length
Mean crossing	The total number of times the signal changes from below average to above average or vice versa normalized by the window length
Range	The range of signal by the difference of maximum and minimum value over the window
Interquartile Range	Measure of the statistical dispersion, being equal to the difference between the 75^{th} and the 25^{th} percentiles of the signal over the window
Median	The median signal value of the signal over the window
Median Absolute Deviation	The median signal of the absolute deviations from the signal median over the window
Covariance	Measure of the joint variability of two signals
Cross correlation	A measure of similarity of two series as a function of the displacement of one relative to the other
Correlation Coefficient	Degree of statistical relationship between two signals

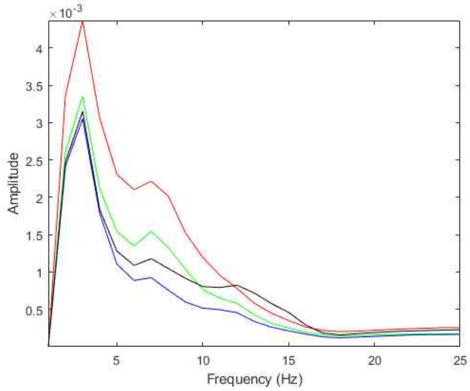
Skewness	The degree of asymmetry of the sensor signal distribution
Kurtosis	The degree of peakedness of the sensor signal distribution
Trim Mean	Calculation of the mean after discarding given parts of a probability distribution or sample at the high and low end, and discarding an equal amount of both.

3.3.2 Frequency domain features

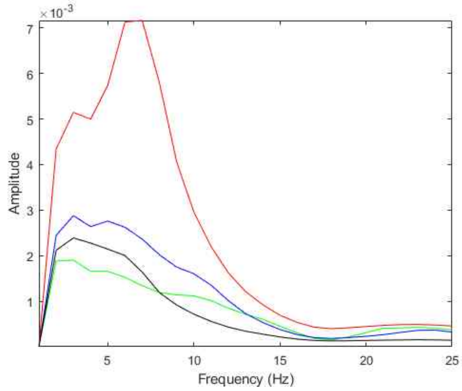
To extract features from frequency signal, we must analyze the data first. As the signal changes from time to time, we cannot pick random signal to check and decide it represents the whole. Therefore, an average distribution of each vehicle is drawn. This graph is drawn using only the Z axis of acceleration signal. This is because while inside vehicle, people do not move where back and forth or side acceleration signal will not change deeply. Instead, as explained in the introduction section, suspension mostly affects the vehicle, which is a movement of up and down. Therefore, gravity axis which is Z axis mainly affects on natural vibration, and is used. Figure 3-8 shows the average distribution of each vehicle from the collected data. For example in Figure 3-8 (a), average distribution of bus signal is drawn. The different color lines indicate the position of smartphone: Red line for hand, green line for backpack, blue line for top (jacket pocket) and black line for bottom (trousers pocket). From the graph, we can see some common phenomenon. First, all of the activities shows peak on lower frequency, under about 6Hz. But in the middle, they differ on each vehicle. Second, every signal line goes down to about 18Hz. This implies that vehicles do not produce high band frequencies. We can know three things from these signal, position independency, different natural vibration and different amplitude.



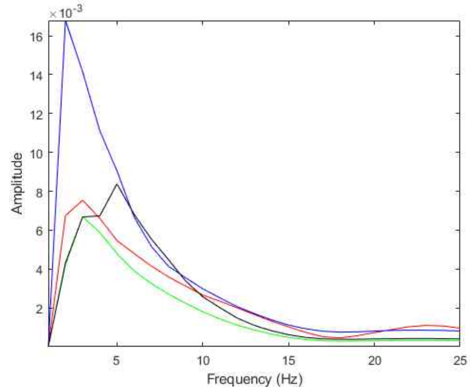
(a) Bus



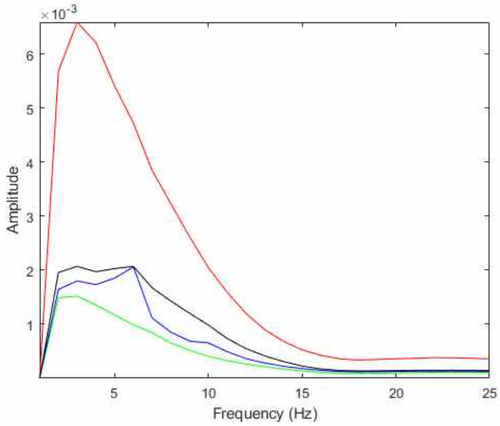
(b) Subway



(c) Car



(d) Bicycle



(e) Stay

Figure 3-8. Average distribution graph from different vehicles (Red: Hand, Green: Backpack, Blue: Top, Black: Bottom)

The first one is position, which doesn’t care if in the same vehicle. This is clear that in general activities, body parts move differently. For example while walking, the legs and arms will move back and forth but in cross while torso only goes up and down. But while in the vehicle, as it is static activity, user doesn’t move and the sensor on different body parts will be affected samely. Therefore, the phone on hand, trousers, jacket and backpack shows similar signals when in vehicle. But these are not identical that the characteristic is different and also the impact will be different. For example, when in stay, people will use their hand to manipulate the phone or wave hands with some gestures while talking. These causes more vibration on hand. Therefore, the signal has higher amplitude on hand than other positions. In summary, different positions will show different amplitude and have similar patterns of signal when inside the same vehicle.

The second one is signals on different vehicles show different natural vibration. Natural vibration is a unique signal of frequency of the object. There is not only one strict signal frequency but can be more than one, and also varies, not strictly an integer. For example in Figure 3-8 (a), We can see that the first natural frequency comes from about 3, and the second one comes from about 12, and the third one comes from about 24. From this, we can make a theory that the signal can be divided into three different parts. The first bandwidth section is from 1~5Hz. This is calculated by getting the average of natural frequency and the standard deviation of it. For example, the natural frequency of bus in the lower bandwidth is 3 having deviation of 2. This is also applied samely to other vehicles. The second bandwidth section is determined based on the signal showing lowest from the whole which is 18. All of the vehicles identically drops the amplitude on frequency 18. Therefore, the second bandwidth signal range is 5~18Hz. Automatically, the third bandwidth section comes from the last bandwidth section from the second to the end, 18~25Hz.

The average distribution is calculated by, adding the whole data with on the same frequency and divide it by number of data. For example, 1Hz signal magnitude is calculated by 1Hz value of first data, 1Hz value of second data, 1Hz value of third data, and so on. And then this is divided into number of samples. All of the other frequency value is calculated in same way, and then finally the average distribution graph is drawn. This procedure is shown in Algorithm 1.

Get Natural Vibration Frequency

Input: *A* – Acceleration data of gravity axis
 	
Output: *N* – Natural vibration frequency band
 	
F = Frequency
B = Frequency band
 	
for each frequency$\mathbf{F_i}$
        $\mathbf{FA_i}$= getAverage
end
 	
for each frequency band $\mathbf{B}_i$
        $\mathbf{BA}_i$= getAverageFrequency
        $\mathbf{BS}_i$= getStandardDeviation
        $\mathbf{NB}_i$= getBandofNaturalVibrationFrequency
end

Figure 3-9. Get natural vibration frequency

Table 3-2 shows the natural vibration frequency of different vehicles. The frequency band overlaps a lot in different vehicles. This is because the sampling rate of collected data was only 50Hz. With short section, the more overlap happens. But the magnitude of frequency, e.g. energy differs from vehicle where it is still distinguishable.

In summary, Table 3-3 shows the overall features extracted.

Table 3-2. Natural vibration frequency of different vehicles

	Section 1	Deviation	Section 2	Deviation	Section 3	Deviation
Bus	3	2	9	4	24	4
Subway	3	2	8	5	25	4
Car	3	2	8	5	25	4
Bicycle	4	2	7	6	24	3

Table 3-3. Total extracted features

	Time domain features	No.	Frequency domain features	No.
Features	Mean	8	Energy of each frequency in the range of natural vibration in section 1 (each 1~5Hz)	5
	Standard Deviation	8		
	Maximum	8		
	Minimum	8	Energy of each frequency in the range of natural vibration in section 2 (each 5~18Hz)	9
	Zero crossing	6		
	Mean crossing	8		
	Range	8	Energy of each frequency in the range of natural vibration in section 3 (each 18~25Hz)	7
	Interquartile range	8		
	Median	8	Energy sum of frequency range in section 1 (1~5Hz, 2~5Hz)	2
	Median absolute	8		
	Median absolute deviation	8	Energy sum of frequency range in section 2 (5~13Hz)	1
	Skewness	8		
	Kurtosis	8	Energy sum of frequency range in section 3 (20~25Hz, 21~25Hz)	2
	Trim Mean	8		
	Covariance	6	Energy sum of section 1 Energy sum of section 2	1 1
	Cross correlation	6	Energy sum of section 3	1
	Correlation coefficient	6	Spectral Entropy (X, Y, Z)	3
Target sensor	Accelerometer, Gyroscope		Accelerometer	
Number of features	128		32	
Total number	160			

3.4 Classification

Two stage classification is made which are GPS speed based and statistical classification based.

3.4.1 GPS speed based classification

From the GPS speed side, the activity is classified based on predefined jogging speed of 9.6km/h [134]. When in low speed, classification is made for every activity. When in high speed, static activities except walk and jog are in the range of classification.

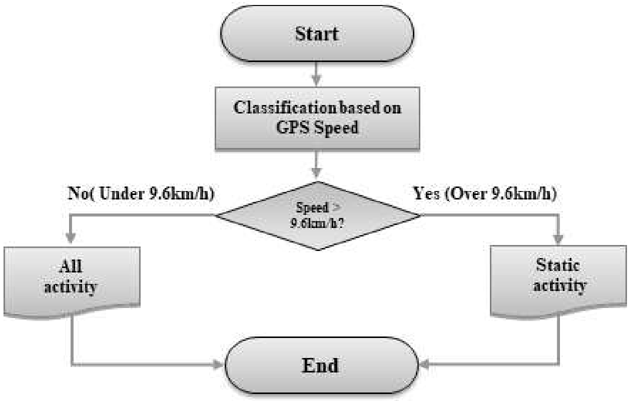


Figure 3-10. GPS speed based classification

3.4.2 Machine learning based classification

After GPS speed based classification is made, statistical classification is done. Several classifiers were compared to find showing the best performance. Following are brief description of classifiers used.

3.4.2.1 K-Nearest Neighborhood (KNN)

KNN is a simple method which concludes the result with majority class within the boundary. K is the number to set how many observations will be count. Basically, Euclidean distance is measured to get the neareast observations. To avoid tie, K is normally set as odd number. KNN is lazy learning algorithm where it doesn’t learn in advance but start when the test data comes.

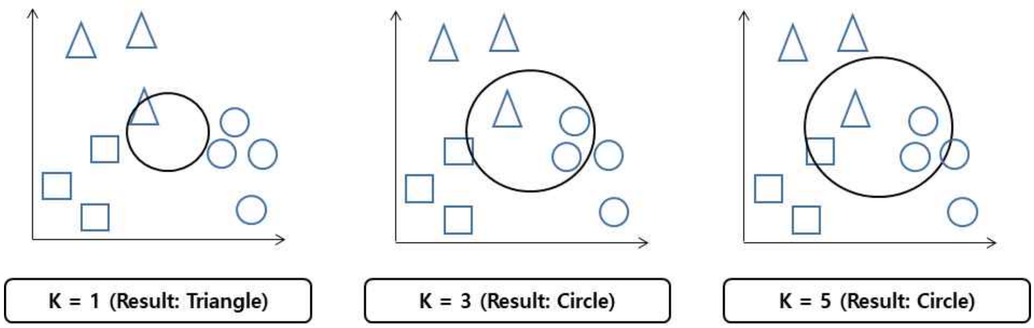


Figure 3-11. Concept of K-Nearest Neighbor

3.4.2.2 Decision Tree (DT)

DT is simple yet intuitive binary classification method. The order of nodes of the decision tree is set by descending order based on information gain of each element. Final conclusion is made on the class of the final leaf node.

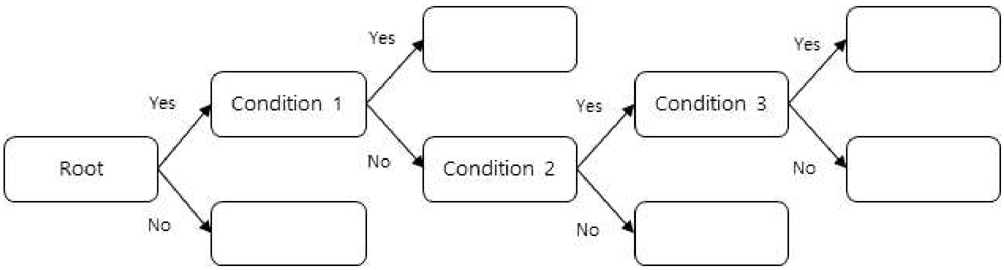


Figure 3-12. Concept of Decision Tree

3.4.2.3 Support Vector Machine (SVM)

SVM is a supervised learning based binary classifier which is well known and widely used, and is capable of performing classification and regression. It is known to show good performance before the emergence of deep learning. Basically, SVM draws an optimal boundary between two classes which can separate them in high dimensional feature space. This is called a hyperplane. Support vectors are two other boundaries which is drawn horizontality based on the hyperplane far with identical distance, where this distance is called margin. The goal of SVM is to find the hyperplnae having best margin which can lower the generalization error. Normally in real data, an absolute hyperplane to distinguish two classes is difficult. Therefore, some of the outliers should be permitted. The hyperparameter to change the margin to allow this is called cost. When the cost is low, the hyperplane is drawn roughly but may casue underfitting when too low. When the cost is high, the hyperplane is drawn strictly but may casue overfitting when too high.

SVM is basically a linear classification. But as the data from real world is non-linear, original method of SVM could not work well. To solve this problem, Kernel SVM (K-SVM) is introduced. Kernel projects the feature space into higher dimension from plain where it can draw the hyperplane in multi-dimension. Representative kernels are polynomial kernel, Gaussian kernel (or Radial basis function kernel), Pearson VII kernel, etc.

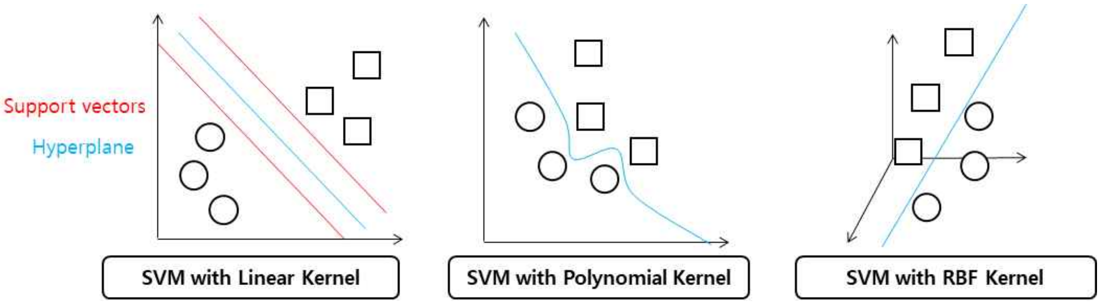


Figure 3-13. Concept of Support Vector Machine

3.4.2.4 Random Forest (RF)

RF is an ensemble learning method which is consittuded with multiple decision trees. RF's main idea is based on bagging, which is an abbreviation for Bootstrap aggregation. The samples constituting each decesion tree are selected randomly allowing repetition, where this is called Bootstrap. The advantage of Bootstrap is that it can generate different type of multiple samples when the size of training data is small, and can guarantee independence among trees which can increase the generalization. Each of the tree is a decision tree. The order of nodes of the decision tree is set by descending order based on information gain of each element. From the multiple tress with multiple conclusions, the final decision is made based on majority voting.

The size of the tree is the key of RF. If the size is small, the diversirty of the tree increases which means the chance of specific sample to be included is low, than the randomness increases and the overfitting decreases. If the size is large, it will become more similar to original decision tree which will have no effect on randomness and possess overfitting issues. The regularization parameters of RF are number of trees made, number of features selected on each tree and the depth of trees.

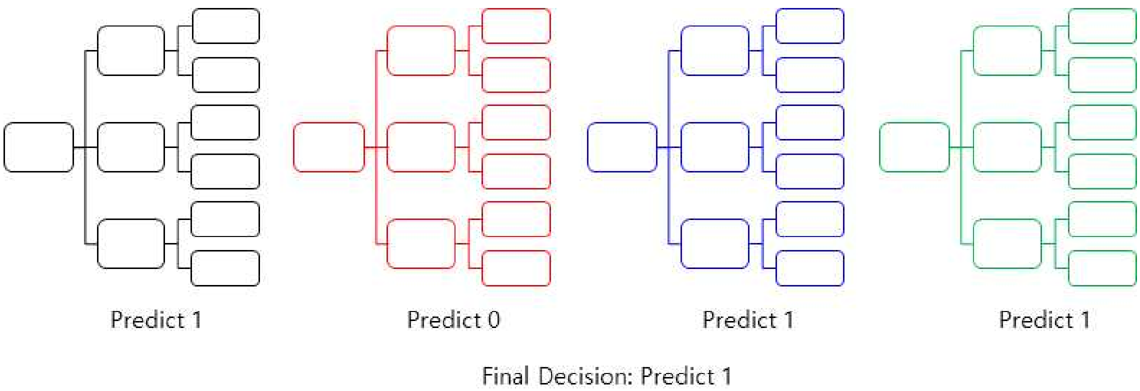


Figure 3-14. Concept of Random Forest

3.4.2.5 Recurrent Neural Network (RNN)

Normally, artificial neural network has a problem of vanishing gradient where the gradient disappears if the layer keeps going deeper. Meanwhile, RNN is a cyclic neural network which keeps the input data information and deliver them to the next repetitively. Therefore in RNN, the past event can affect the decision of the future by taking into account the correlation of past and current data and predict the future. Generally, sequential prediction is done in RNN such as text or audio. Still, RNN also possess the problem of vanishing gradient when the layer goes deeper. For example, if a word needs to be predicted to come next from a sentence, if the context to refer is close, there would be no problem. But if the context is far, it is hard to infer the relation between them which is called long term dependency problem. Therefore, Long Short Term Memory (LSTM) is proposed. In here, a hidden layer has a cell state which connects the cell and three gates which are input gate, output gate and forget gate. The input gate determines which new information do be stored in cell state. Forget gate determines which information to discard. Output gate outputs the result from the updated cell state.

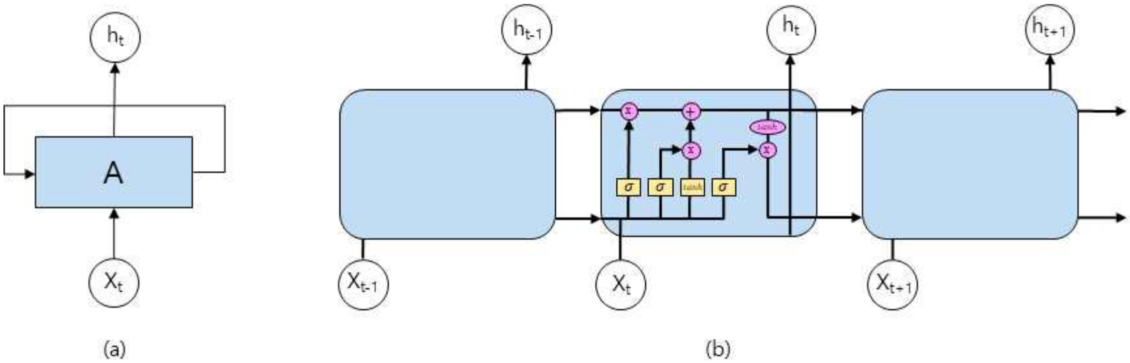


Figure 3-15. Concept of (a) RNN and (b) LSTM

3.4.2.6 Convolutional Neural Network (CNN)

CNN is a neural network which is a set of convolution procedures. CNN repeats this convolution task and finally input them to fully connected layer for output. It is widely used for image processing. Convolution is a mathematical operator that multiplies one function by another, and then integrates over the interval to find a new function. In the convolution process, a filter is set with a stride. A filter is a matrix which will extract features from the whole matrix of image. The stride is set by how many pixels the filter will move. For example, if the input image size is 10x10, filter size 3x3 and stride is 1, the filter will start from upper left considering 3x3 pixels. It will extract one feature from here and put that feature on the upper left of the newly created feature map. Then the filter moves only one pixel, and repeats this prodecure. So the size of the new feature map will be subtracting size of two from both vertical and horizontal. From the example, the new feature map size will be 8x8. Number of convolution layers differs based on the characteristic of dataset and what to achieve. An activation function is used to solve vanishing gradient problem caused by non-linearity data. A pooling layer is used which subtracts only meaningful feature from the feature map do reduce the feature dimension.

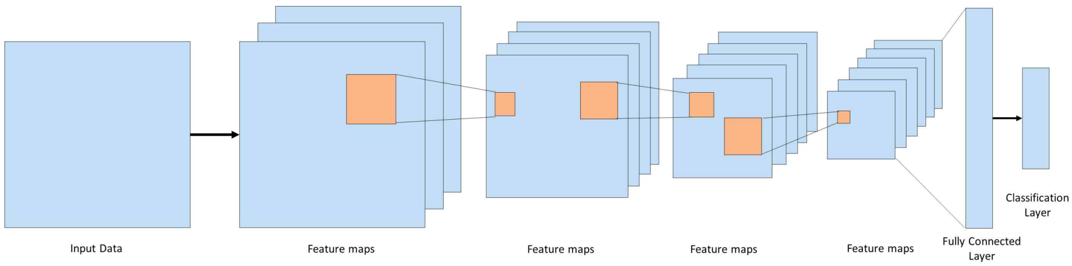


Figure 3-16. Concept of Convolutional Neural Network

3.5 Post-processing

3.5.1 Adding GPS Data

So far, this study has focused on general process for activity recognition. If we use GPS outdoors, we can distinguish other vehicles from subway more easily. Therefore, we have collected GPS data along with acceleration and angular velocity data to acquire coordinate (latitude and longitude) and speed information. The acquired acceleration and angular velocity data are used for feature extraction and classification for HAR, as described in Sections 3.3 and 3.4. GPS data are used for correction to recognize vehicles better. To distinguish vehicle from other activities, we set the threshold speed as 9.6 km/h as this is known to be average speed for jog. In other words, if the speed from GPS is greater than 9.6 km/h, we assume that the current activity is not stay, walk, or jog, but car, bus, subway or bicycle.

We also use GPS for ground-level sections of the subway. To distinguish between other vehicles and subway on similar courses, we collected five beacons between two subway stations, in case if they are on the ground. Our experiments were conducted in Seoul, the capital of South Korea. We input the GPS address of every beacon for the subway stations using Google Maps. To use this method in a different context, the GPS address should be changed. In Korea, the time from one station to the next takes 2–3 min on average. Therefore, the beacon is divided into five including departure station and arrival station. And then we set the time interval to 30 seconds among the beacons. Each beacon has its own latitude and longitude. If the subway passes from one beacon to another within 30 seconds, we determine that the user is on the subway. If the user does not pass from one beacon to another within 30 seconds or passes only one beacon, we determine that the user is using other vehicle. An illustration of this procedure is given in Figure 3-17.

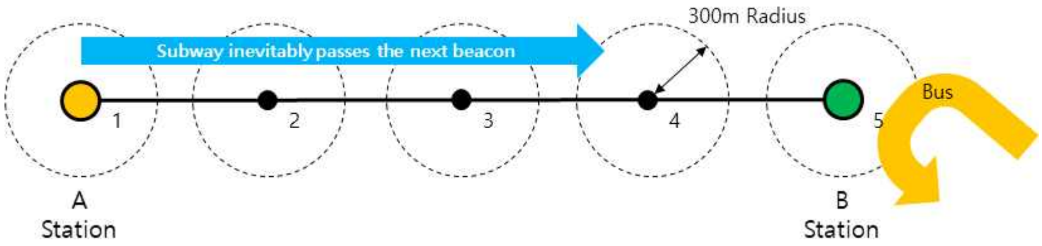


Figure 3-17. Distinguishing bus and subway on ground level using GPS coordinate.

3.5.2 Activity Correction

Guaranteeing accurate recognition requires a correction mechanism. During any activity, the user might handle the phone in a way that will change the pattern of the inertial data and greatly affect the accuracy. Vehicles recognition will also be affected if the GPS signal is not received properly or when they are stopped at a station. All of these occurrences constitute noise and could lead to incorrect results. To correct this problem, we make the final activity decision only if two sequential results (current and previous) are the same. If they are not the same, we maintain the activity as the last recognized activity. Figure 6 shows the flowchart for this process.

After the data is collected, we classify the activity based on inertial sensor signals. Then the classification based on speed is performed. A low speed could be any activity, but a speed greater than 9.6 km/h can only be the vehicles. In this first recognition cycle, there will be no correction result in the previous stage, so the correction result will be the same as the recognized result. The correction flow starts from the next cycle. If the recognized result differs from the previously recognized result, the final result will show the same one as previously corrected result, and the current corrected result will be set as the same activity and terminate the flow. If the recognized result is same as the previously recognized result and it is not staying, then the final result is shown as the current activity. Then, it saves the current activity to correction result and terminates the flow. On the other hand, if the recognized activity is the same as the previously recognized activity and both are stay, we take a look into the previously corrected result. If the previously corrected result was not a vehicle, the current activity will be stay. However, if the previously corrected result was a vehicle, we take a look whether the previously recognized results maintained as stay for 3 min. Additionally, if the stay showed for 3 min, the final activity is concluded as stay. If the stay did not show 3 min of stay continuously, the activity will be concluded as the previously corrected result, which will be a vehicle. The 3 min interval is set to prevent misrecognition as stay. When a vehicle run on smooth road and rail, the recognition result will frequently show stay. When either vehicle is stopped due to traffic or a station, it will also show stay due to the lack of movement. So, once the algorithm recognizes the activity as vehicle, and stay occurs in the middle, we disregard this result and force the algorithm to show the result as vehicle. However, we need a breakpoint or the activity will not be changed from vehicle. As mentioned above, the time from one station to another in Korea is 2–3 min for either bus and subway, and the longest traffic light standby time is about 3 min for the bus. So, we set the break time to 3 min. If the stay shows 3 min, now the algorithm determines that the user is actually in stay and escapes from the recognition as vehicle. Figure 3-18. shows the overall flow of activity decision and the according algorithm in Algorithm 2.

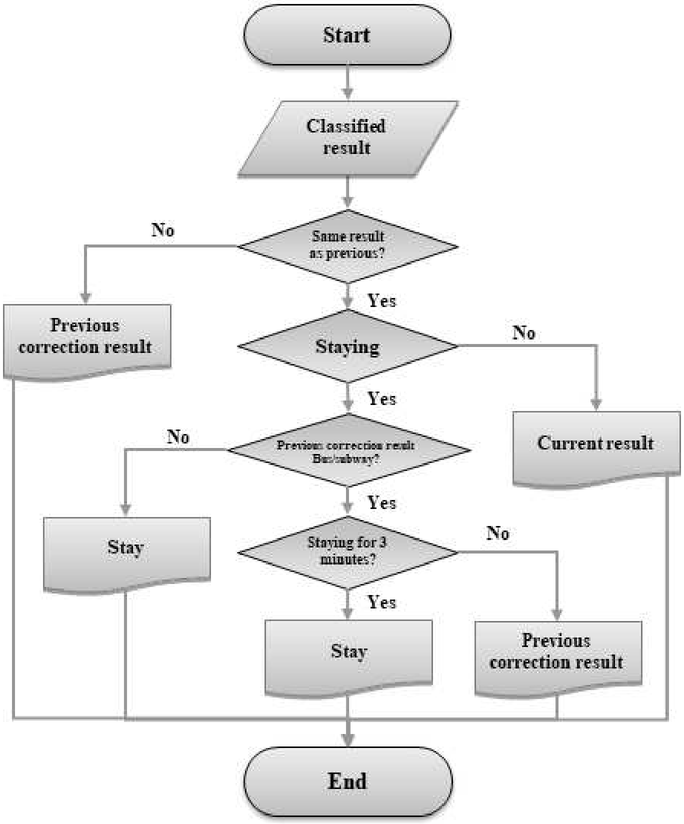


Figure 3-18. Flow of activity correction

```
Activity Correction
Input: C – Classified activity label

Output: F – Final activity label

while 1 do
      if currentActivity== previouslyCorrectedActivity then
            if currentActivity!= stay then
                  F =currentActivity
            else
                  if stay > 3minutes then
                        F =stay
                  else
                        F = previouslyCorrectedActivity;
                  endif
            endif
      else
            F = previouslyCorrectedActivity;
      endif
endwhile
```

Figure 3-19. Algorithm for activity correction

Figure 3-20 (a) shows an example operation of the correction algorithm. A total of 30 activities were recognized, and the correction was performed for each activity. Note that one activity label is concluded after 3 seconds and that we set the size of the window in this way. The first four correction results (12 seconds) are not shown because they are in initialization stage. The fifth result shows the same in both recognition and correction parts as there are no data to refer to from the previous. On the sixth result, the actual recognition changed from staying to walking. However, the previous correction result was walking, so the algorithm disregarded the recognition result and set the correction result as walking. This happened again on the 12th. On the other hand, the 13th recognition result was staying, which is the same as the previous (12th) recognition result. Although the correction result is different from walking, this is regarded as the ground truth activity and the correction result is changed to staying. Taking a look at the 23rd result, the recognition results of both 22nd and 23rd are staying. It seems that the correction result should be changed to staying even if the previous (22nd) correction result was bus. However, as mentioned above, we set a 3 min interval in the vehicle activity. So, the algorithm ignores the recognition result (staying) and concludes the correction result as bus.

Figure 3-20 (b) shows another example of the correction algorithm’s operation. Until the 140th result, it is similar to the previous example. After that, if the correction result changes to subway, it ignores the recognition result but keeps the correction result as subway. From the 149th, the recognition result turned to stay until the end. This means that the user actually got off the subway and was staying still. However, due to the 3-minutes interval problem, the correction result kept showing subway. After 3 minutes (which means 60 blocks after), the algorithm finally concluded the ground truth activity as stay. This case is the only disadvantage of our algorithm; however, we cannot eliminate this 3 minutes interval feature because it will lead to substantial error without it.

Number	Recognition	Correction
1	Walk	Walk
2	Walk	Walk
3	Stay	Walk
4	Walk	Walk
5	Walk	Walk
6	Walk	Walk
7	Stay	Walk
8	Stay	Stay
9	Stay	Stay
10	Stay	Stay
11	Walk	Stay
12	Walk	Walk
13	Walk	Walk
14	Bus	Walk
15	Bus	Bus
16	Bus	Bus
17	Bus	Bus
18	Bus	Bus
19	Stay	Bus
20	Stay	Bus
21	Bus	Bus
22	Bus	Bus
23	Stay	Bus
24	Stay	Bus
25	Bus	Bus

No. 3: Although current recognition is Stay, the activity is corrected to Walk

No. 7: Although current recognition is Stay, the activity is corrected to Walk

No. 8: As previously recognized activity and currently recognized activity is all Stay, the activity is concluded as Stay

No. 14: Although current recognition is Bus, the activity is corrected to Walk

No. 15: As previously recognized activity and currently recognized activity is all Bus, the activity is concluded as Bus

No. 20: Although previously recognized activity and currently recognized activity is Stay, Stay did not occurred continuously for 3 minutes. Therefore, the activity is corrected to Bus

(a)

Number	Recognition	Correction
1	Stay	Stay
2	Stay	Stay
3	Subway	Stay
4	Subway	Subway
5	Subway	Subway
6	Stay	Subway
7	Stay	Subway
8	Walk	Subway
9	Subway	Subway
10	Stay	Subway
11	Subway	Subway
12	Subway	Subway
13	Subway	Subway
14	Subway	Subway
15	Stay	Subway
16	Stay	Subway
17	Stay	Subway
18	Stay	Subway
	..	Subway
74	Stay	Subway
75	Stay	Subway
76	Stay	Stay

No. 6: Although current recognition result is Stay, the user is on Subway, so the activity is corrected to Subway

No. 7: Although previously recognized activity and currently recognized activity is Stay, Stay did not occurred continuously for 3 minutes. Therefore, the activity is corrected to Bus

No. 8: Current activity is recognized as Walk. But due to the state transition rule, the activity is corrected to Subway

No. 15~45: Activity is recognized to Stay continuously. But this can be the case when the subway has stopped at the station. Therefore, the activity is corrected to Subway

No. 46: Because Stay has occurred for 3 minutes, it concludes that it was misrecognizing as Subway. Therefore, the activity is corrected to Stay

Figure 3-20. Distinguishing bus and subway on ground level using GPS location

3.5.3 State Transition Rule

Figure 3-21 shows the state transition diagram. Stay, walk, and jog can alternate each other while vehicles cannot be changed to each other due to the nature of reality. Normally, vehicles cannot be changed each other before transferring them. A person should at lest walk to take another vehicle. One exceptional case is turning from vehicle to stay. In the case as we mentioned before, if the vehicle is misrecognized for 3 minutes, it should correct the result to stay. Figure shows the digram of state transition rule.

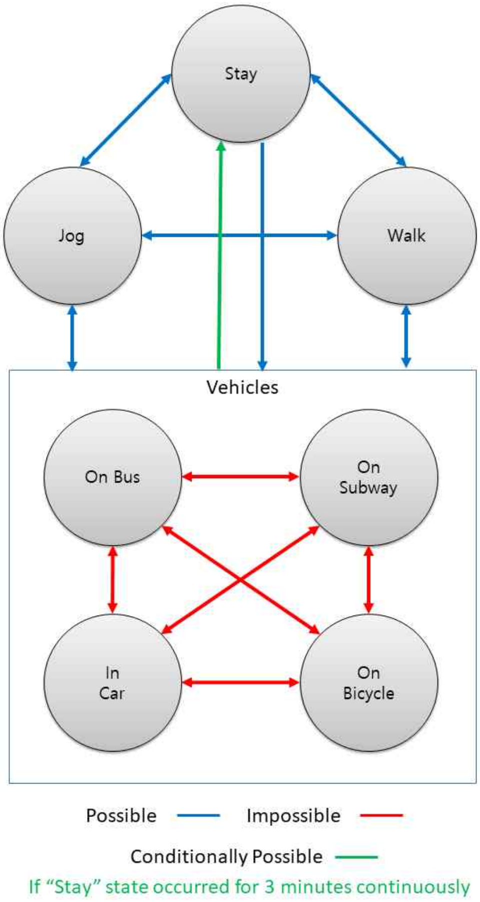


Figure 3-21. State Transition Rule

4. Experiment

In this section, we introduce the experiments from both traditional method and deep learning method, and with the result with applying post-processing. We also describe the devices used, data collection environment, performance evaluation method, and results.

4.1 Experimental Environment

The dataset is collected by ourselves using four Samsung Galaxy S7 smartphone. Both 3D axes of linear acceleration and gyroscope was collected with the frequency of 50Hz sampling rate and range 8G where the maximum range of accelerometer is $\pm 78 m/s^2$ and the gyroscope is $\pm 20 rad/s$. GPS was also collected with speed, latitude and longitude.

Seven commuting activities including ambulatory activities were collected: stay, walk, jog, riding bus, subway, car, and bicycle. For stay, walk, jog and bicycle, five minutes were collected for each subject. For bus, subway and car, more than ten minutes of data were collected. But to keep the balance of the data, the middle part of data was cut to only contain five minutes of data from these three activities. One observer followed the subject during data collection for checking whether the subject is performing totally different activity, record the time, and drive the car for them. Only one or two subjects have collected the data at a time due to only having four smartphones. When two subject went out for collection, they did it alternatively. The number of collected data for training and testing is shown in Table 4-1.

Table 4-1. Number of collected for train and test (instances)

Activity	Stay	Walk	Jog	Car	Bus	Subway	Bike	Total
Train data	6,973	7,343	6,823	7,073	7,121	7,016	7,139	49,488
Test data	618	360	80	3,865	2,492	4,047	388	11,848

Total 17 subjects has participated for data collection. The subjects ranged in age from 25 to 66 (average 35), in height from 167cm to 183cm (average 174cm), and in weight from 61kg to 89kg (average 72kg), where they were all male. None of the subjects had any kind of physical or mental disorder. Information of subjects are shown in Table 4-2.

Table 4-2. Subject information participated for data collection

No.	Gender	Age	Height (cm)	Weight (kg)
1	Male	25	180	80
2	Male	26	172	78
3	Male	27	167	67
4	Male	27	183	73
5	Male	28	179	73
6	Male	29	170	61
7	Male	29	180	78
8	Male	30	175	56
9	Male	32	167	74
10	Male	32	180	74
11	Male	34	178	87
12	Male	34	183	85
13	Male	35	167	63
14	Male	38	180	74
15	Male	42	175	89
16	Male	66	170	70
17	Female	62	156	56
Average	16 Male 1 Female	35.06	174.24	72.82

The smartphone was positioned in four different locations: trousers front pocket, jacket pocket, in the backpack, and holding in hand, where these positions represent common ways people carry their smartphones. The side location of pockets and hand was set free, in other words, one subject place the phone in right trousers pocket, right jacket pocket and hold on in left hand while other subject did vice versa. The orientation was told to set free. When in trousers pocket, the phone is totally fixed. When in jacket or backpack, the phone is loosely fixed. When in hand, the orientation changes more frequently than other positions.

The data is segmented in 3 seconds. When extracting feature, time and frequency domain features are differently extracted. In time domain, 3 seconds of data are applied. In frequency domain, at first, each 1 second data is transformed to frequency signal. Then each frequency from three different data are summed. For example, 1hz data on first, second and third data are summed, and this applies to all frequency until 25hz. Then natural vibration features are extracted from the summed 25hz data.

The specification of PC running the experiment was having i5 quad core cpu at 4.5Ghz, 16gb ram and a single GTX 1080 ti graphics card. Weka tool is used for traditional methods experiment and python with tensorflow is used for deep learning based experiments. All the experiments were run on Windows 10 environment.

The path for training data collection was set as follows:

- Subway: Bundang line (Yeongtong– Suwon station)
- Bus: City bus, Intercity bus
  - Highway: No. 5100 (Kyung Hee University – Gangnam)
  - City road: No. 310, 900, 7-2 (KHU – Suwon Station, All same route)

- Car: Gasoline car, Diesel car
  - Highway: Dongbu-daero (Kyung Hee University – Osan City hall)
  - City road: Kyung Hee University – Sungkyunkwan University
- Walk/Jog/Bicycle: Inside Kyung Hee University

The path for test data collection is set as follows:

- Subway: Line 1 (Suwon – Singil station)
- Bus: No. 641 (Singil – Gangnam station)
- Car: Gasoline car (Gangnam station – Kyung Hee University)
- Walk/Jog/Bicycle: Inside Kyung Hee University

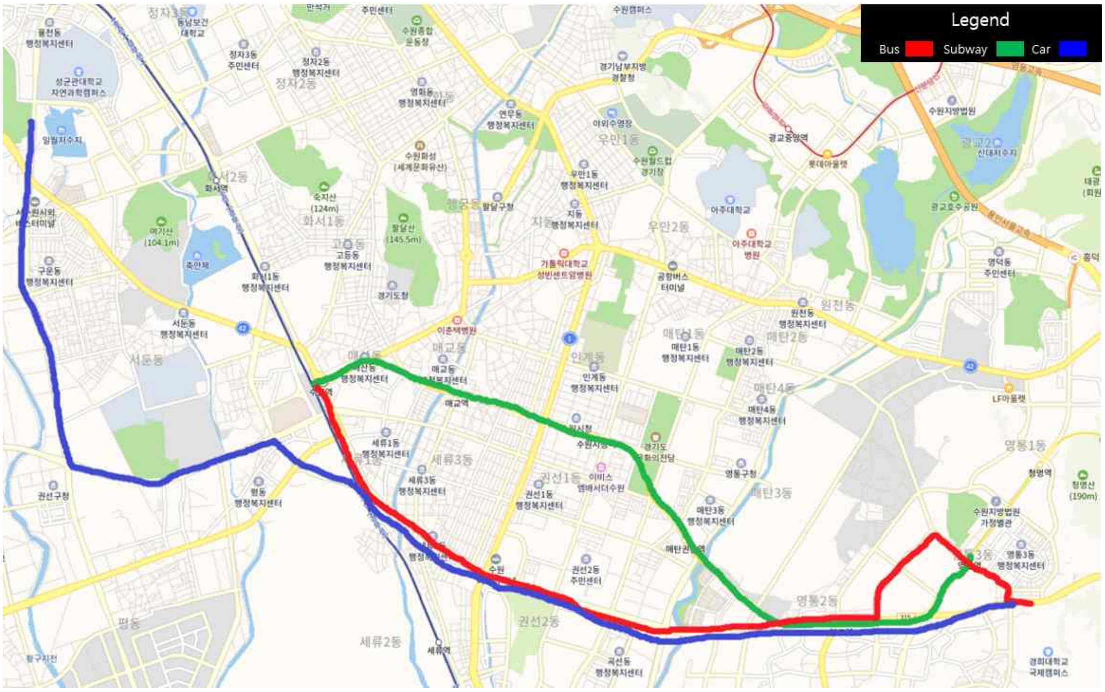


Figure 4-1. Route for subway, city road bus and city car for training data collection

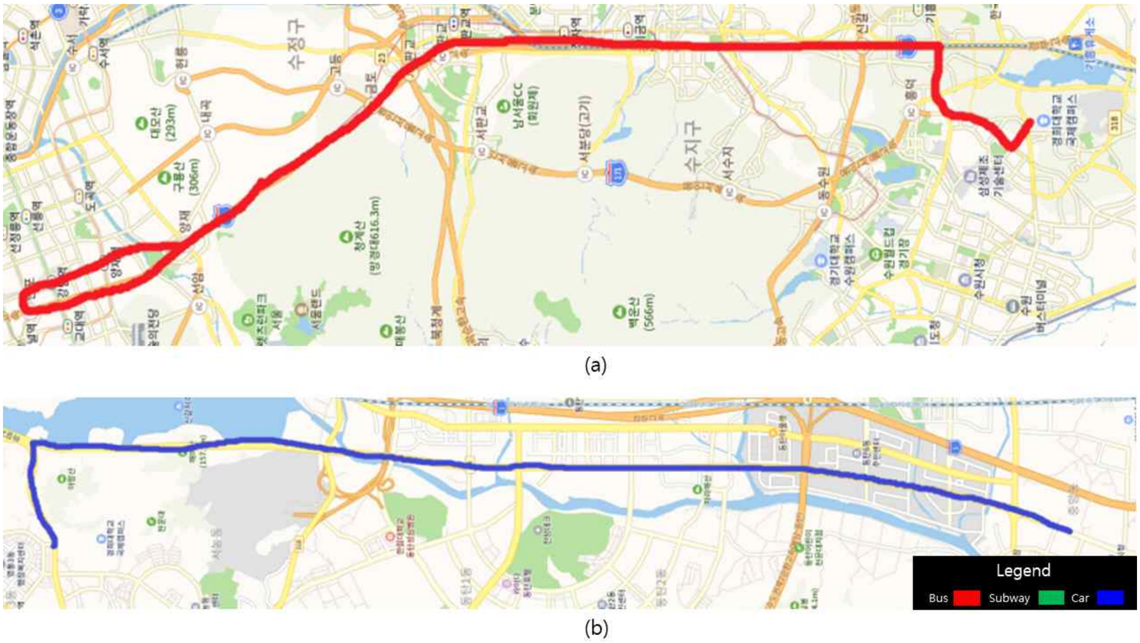


Figure 4-2. Route for (a) highway bus and (b) highway car for training data collection

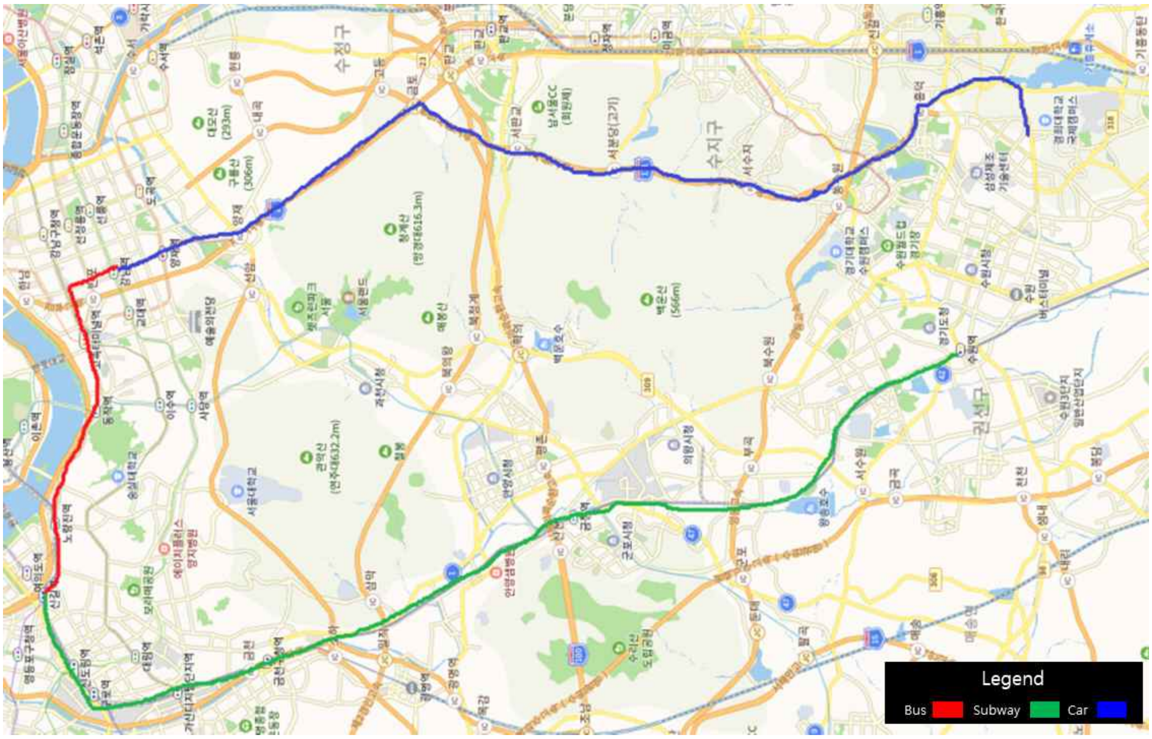


Figure 4-3. Route for subway, bus and car for test data collection

The experimental environment for statistical classifiers and DNN methods are as follows.

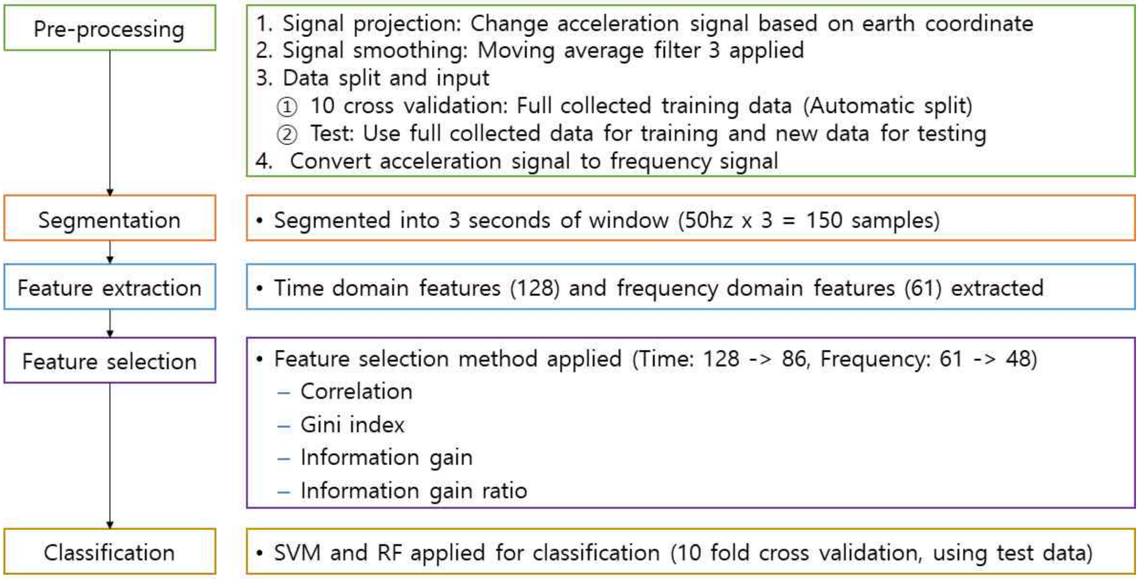


Figure 4-4. Experiment procedure for statistical classifiers

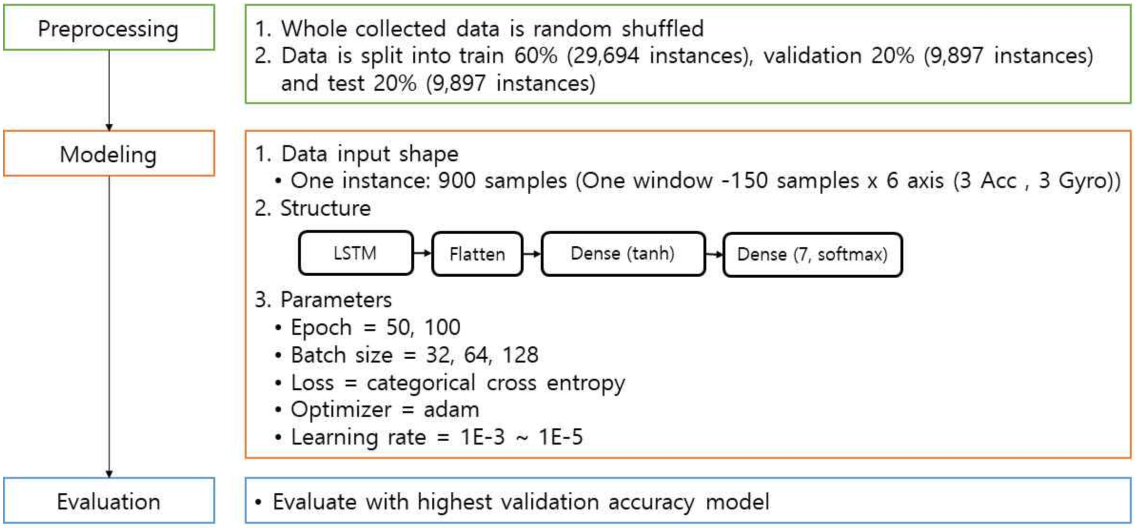


Figure 4-5. Experiment procedure for LSTM

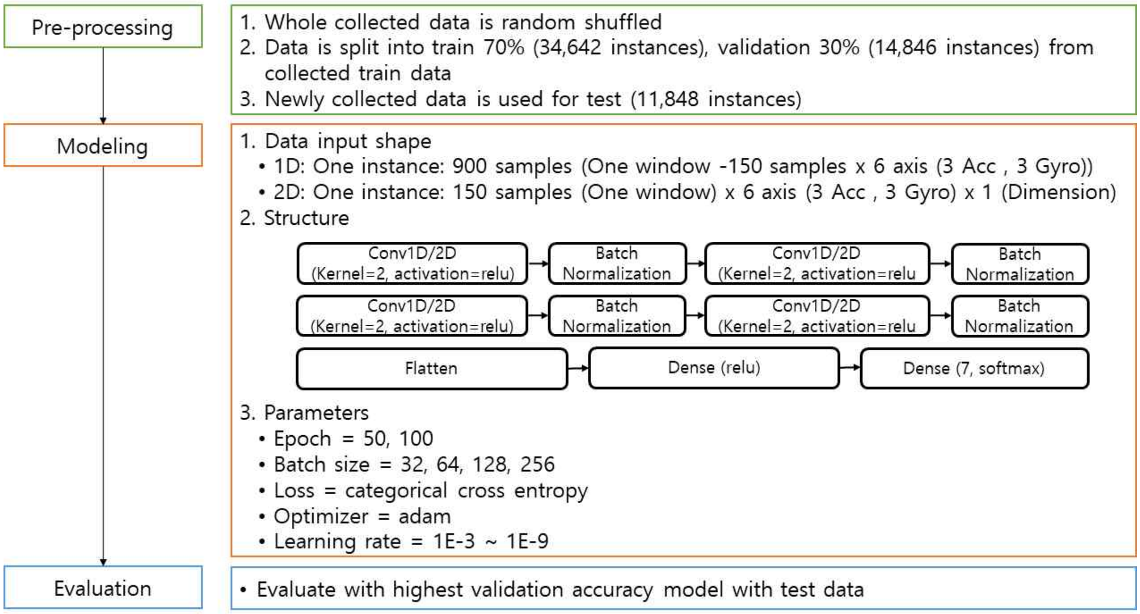


Figure 4-6. Experiment procedure for CNN 1D and 2D

4.2 Experimental Results

Total five kinds of experiments will be conducted which are ① Comparison of different classifiers, ② Comparison with different research, ③ Comparison with different post-processing research, and ④ regularization parameter tuning to overcome overfitting. All of the data is random shuffled first before input to classifiers. Table 4-3 shows the summarized list of experiments.

Table 4-3. List of experimentss

No.	Experiment Name	Details
1	Comparison of different classifiers	• Features input : Support Vector Machine, KNN, Decision Tree, Random Forest • Raw signal input: LSTM, CNN 1D, CNN 2D
2	Comparison with different research	• Widhalm et al. [135]: Using all of the frequency's energy • Hemminki et al. [136]: Energy of each frequency 1-10Hz • Wang et al. [137]: Energy of 0-2Hz and 2-4Hz • Fang et al. [138]: Peak frequency's energy
3	Comparison with different post-processing research	• Wang et al. [129]: Activity updating by majority voting looking 1 minute duration due to continuous activity • Grzeszick et al. [130]: Activity updating with majority voting in past five activities
4	Regularization parameter tuning to overcome overfitting	• Parameter tuning of SVM with parameter C • Parameter tuning of Random Forest with parameter tree depth, number of tree and number of features in a tree • Parameter tuning of DNNs

4.2.1 Comparison of different classifiers

In this section, different classification methods are compared to pick the classifier showing best performance. For traditional classifiers, Support Vector Machine, K-Nearest Neighborhood, Decision Tree and Random Forest are used. 10 fold cross validation is performed for these using the whole collected training data. In here, hand crafted features from proposed method including time domain statistical and frequency domain natural vibration features are used for input. Default parameters are used for each classifiers. For Deep Neural Networks method, Long Short Term Memory and Convolutional Neural Network with 1 dimension (1D) and 2 dimension (2D) is performed. In here, raw sensor data are input. The result is shown in Table 4-4.

Table 4-4. Comparison with different classification methods

Classification methods	Accuracy (%)
Support Vector Machine (SVM)	78.363
K-Nearest Neighborhood (KNN)	83.794
Decision Tree (DT)	85.400
Random Forest (RF)	93.203
Recurrent Neural Network (Long-Short Term Memory, LSTM)	89.963
Convolutional Neural Network (CNN) 1D	91.948
Convolutional Neural Network (CNN) 2D	92.351

From the experiment result in traditional methods, RF showed the best accuracy of 93.203%. The next comes DT, and KNN showed slightly lower. SVM showed lower than these, and then LSTM, CNN 2D, and CNN 1D comes next. Random forest randomly selects features and conclude the result with majority voting of all trees.

Therefore, in our case, as static activities show similar sensor values, data are gathered in feature space. The advantage of DT in this case is that unlike other methods considering whole features at the same time, DT only picks each feature at a time and distinguish the class. And with randomly selected features, it will perform better than DT and have strong point in overfitting. The KNN may pick the data well that the data for each class will be aggregated. But SVM may have problem drawing hyperplane within the aggregated data which showed poor result. Meanwhile, DNNs methods showed good result after RF. But these result cannot be trusted. The result of these has used the data split from collected train dataset where it may have high similarity, and used only basic setting without any parameter tuning. This will be handled in section 4.3. All of these result may contain overfitting even proved with cross validation. Therefore, new experiment with new test data where it was collected in different route and different user who hasn't participated in data collection.

4.2.2 Comparison with different research

In this section, different research trying to classify commuting activity is compared with proposed method. Fang et al. has extracted only the peak energy from frequency domain. On the contrary, Widhalm et al. have extracted energy from whole frequency and used as features. Wang et al. has extracted energy from two bandwidth, 0-2Hz and 2-4Hz. This is because the highest energy comes from low frequency band. Finally, Hemminiki et al. also extracts energy from low frequency band, 1-10Hz. Table 4-5 shows the result.

Table 4-5. Comparison with different research using RF

Research	Accuracy (%)
Widhalm et al.	92.846
Hemminki et al.	91.591
Wang et al.	91.308
Fang et al.	90.927
Proposed method	93.203

Proposed method showed the best accuracy of 93.203%. But other methods also showed similar results where all of them exceeds 90% accuracy. Widhalm et al. used all the frequency energy where each may have their own meaning. Wang and Fang method used low frequency band energy where vehicles show most energy in these area. Fang only used one which is the peak energy but still will have the value in low frequency band. From this result, we cannot say the proposed natural vibration feature outperforms the other feature extraction methods. But as it will be later shown, it has more generalization and overfitting prevention power than others.

4.2.3 Comparison with different post-processing research

In this section, different post-processing methods are compared with proposed method. Wang et al. updates the activity by majority voting looking for one minute duration of data. The reason for setting such a long time is that they have claimed that because commuting activity such as vehicles lasts long once the user is inside. Grzeszicket et al. also uses majority voting scheme looking for past 5 activities. The experiment was conducted using the propose method, using RF for classifier and time domain and natural vibration features are extracted. For Wang, since the data in this paper is segmented in 3 seconds, looking for one minute means looking for past 20 activities. But in here, the author didn’t exactly mentioned on which activity they refer, the inferred label or the updated label. In the latter case, there is a problem when the activity is changing. Even new order of activity starts, it will keep changing to previously recognized activities due to seeing long duration of previous activity. Eventually, the activity never changes but fixed to one, and the accuracy will decrease dramatically until same activity is performed again. Therefore, the experiment is conducted with former case, updating activity with classified result. The same was is also applied for Grzeszicket’s method. Only the post-processing method is compared in this experiment, e.g. accuracy before applying post-processing is same with our proposed method and comparison on only after applying post-processing is done. Table 4-6 shows the result.

Table 4-6. Real world scenario test result

Research	Accuracy before applying post-processing (%)	Accuracy after applying post-processing (%)
Wang et al.	93.203	93.821
Grzeszicket et al.		94.109
Proposed method		95.598

From the experiment result, proposed method showed more increase in accuracy than other two methods. As expected, Wang’s method increased most little and then Grzeszicket’s method but still lower than proposed methods. This is because that sometimes tie happens by looking multiple activities. Meanwhile, proposed methods only refers previously recognized activity which may not have the problem aforementioned.

4.2.4 Regularization parameter tuning to overcome overfitting

Experiments result above have used the data collected for training. In here, all of the route for collection is same. This means that, for example, the subject has taken the subway of same line, starting from one station to another and vice versa. Similarly, the bus goes the same route. In this case, the data might be well fitted with the same route, having similar road conditions or traffics even though the bus and subway itself varies. And for the car, two types of car was used in the same route, which would be more well fitted. Therefore, to prove that the proposed model is well generalized, test data is collected in different route with different user who hasn't participated the training data collection. Figure shows the evaluation result using test data and compared with different research, same as in section 4.2.2.

Table 4-7. Comparison with different research using different test data using RF

Research	Accuracy (%)
Widhalm et al.	73.850
Hemminki et al.	72.857
Wang et al.	72.448
Fang et al.	71.890
Proposed method	77.578

As shown in the result, proposed method showed the best accuracy than other research. This means that natural vibration features are more robust to overfitting. But still model adaptation process is required that testing accuracy is lower than training accuracy, which is an overfitting problem. There several methods to overcome overfitting. First way is to increase the training data. With more data, it has more chance to learn different environments, having higher probability that unseen data will have same or similar properties. Second is to reduce features. Some of the features may not work well classifying but still affect in some sense. In this case, it cannot classify general cases but only biased to that case. Therefore, eliminating meaningless feature may also improve the performance well. The third method is regularization. This object is to constrain the learning too much by tuning hyper parameters. Regularization is applied differently for each classification methods. Forth method is normalization, which converts the data into range zero to one. This will equalize the representation of the data, having more reliability. For DNNs methods, there are many methods such as early stopping, which stops learning when there is no change, or dropout where the ratio of neuron is wiped out, an so on. In the following section, SVM and RF based regularization is performed, and then the DNNs.

4.2.4.1 SVM based regularization

For SVM based regularization, changing complexity parameter c is applied. Firstly, kernel is selected which shows the best result when the c is fixed to 1. Among 4 different kernels such as polynomial, Gaussian (radial basis function) and Pearson Universal Kernal (PUK), polynomial showed the best accuracy. Generally, RBF and PUK kernel are reported to show the highest result. But due the characteristic of the dataset in this work, polynomial kernel showed the best. Therefore, polynomial kernel is selected. Multiple experiemtns are performed changing the parameter c without chaning the parameters of kernel itself. All of the experiment is performed using the training set collected from 17 users and test dataset collected from different user. The results are shown in Table 4-8 and Table 4-9.

As shown in the result, the performance using the default c parameter which is one shows 61.166% accuracy in polynomial kernel. 17.197% of accuracy has been reduced from 78.363% where it was the result using only the training dataset collection with 10 fold cross validation. By increasing parameter c which tolerates outliers has increased the performance. First the c was increased with minimum gap and then larger gap. As a result, when parameter c was 1000, it showed the best accuracy of 71.046%. After that, it started to decrease where we can assume that c with 1000 is the optimized value in this experiment which has increased the accuracy of 9.880% from 61.166% to 71.046%.

Table 4-8. Comparison of different kernels in SVM fixing parameter C with test data

Kernel	C-value	Accuracy (%)
Polynomial	1	61.166
RBF	1	49.452
PUK	1	59.568

Table 4-9. Comparison by changing C-value in SVM with test data using polynomial kernel

C-value	Accuracy (%)
0.5	58.379
1	61.166
1.5	62.690
2	63.335
2.5	64.198
3	64.706
3.5	65.033
4	65.278
10	66.485
15	66.957
20	67.057
30	68.628
50	69.282
100	69.529
200	70.182
300	70.339
500	70.857
1000	71.046
2000	70.914

4.2.4.2 RF based regularization

For RF based regularization, changing parameter of tree depth, number of feature and number of tree is important. Therefore, with the same way, best accuracy is searched changing three different parameters with separated train data and test data. Table 4-10 shows the experimental results. As shown in the result, the overall accuracy did not changed much while changing the parameters. This means that random forest is already generalized which is strong to overfitting. For each parameters, we can observe different phenomena. The depth of tree could not affect the accuracy after 100 where it doesn’t change. The number of features could not affect the accuracy after 14 where it doesn’t change. Finally, the number of trees over 200 will start to decrease the accuracy while it increases until 200. Besides the depth and number of trees, the accuracy variation of the number of features proves that normal DT may possess overfitting but not the RF as DT will use the whole feature at once. From the experimental result, the best performance showed when the tree depth is 20, number of tree is 200 and the used feature is 7 as 78.106%.

Table 4-10. Comparison by changing different parameters with test data in RF

Tree depth	Number of features	Number of trees	Accuracy
10	7	10	72.168
10	10	10	72.168
10	14	10	70.280
10	7	20	73.851
10	7	30	73.763
10	7	50	74.369
10	7	100	74.702

10	7	200	74.770
10	10	20	72.168
10	10	30	72.637
10	10	50	73.195
10	10	100	72.637
10	10	200	73.684
10	14	20	71.199
10	14	30	71.914
10	14	50	71.287
10	14	100	71.473
10	14	200	71.943
20	7	10	72.706
20	10	10	70.926
20	14	10	71.180
20	7	20	75.269
20	7	30	76.482
20	7	50	77.187
20	7	100	78.047
20	7	200	78.106
20	10	20	74.800
20	10	30	75.416
20	10	50	76.375
20	10	100	77.421
20	10	200	77.431
20	14	20	74.047
20	14	30	74.760
20	14	50	75.025

20	14	100	75.709
20	14	200	76.022
30	7	10	71.835
30	10	10	71.728
30	14	10	70.749
30	7	20	74.731
30	7	30	75.836
30	7	50	76.668
30	7	100	77.323
30	7	200	77.323
30	10	20	75.445
30	10	30	75.504
30	10	50	76.766
30	10	100	77.509
30	10	200	77.656
30	14	20	73.831
30	14	30	74.663
30	14	50	74.897
30	14	100	75.602
30	14	200	75.934
50	7	10	71.835
50	10	10	71.728
50	14	10	70.749
50	7	20	74.604
50	7	30	75.934
50	7	50	76.550
50	7	100	77.196

50	7	200	77.412
50	10	20	75.445
50	10	30	75.504
50	10	50	76.766
50	10	100	77.500
50	10	200	77.685
50	14	20	73.831
50	14	30	74.663
50	14	50	74.897
50	14	100	75.602
50	14	200	75.934
100	7	10	71.835
100	10	10	71.728
100	14	10	70.749
100	7	20	74.604
100	7	30	75.934
100	7	50	76.551
100	7	100	77.196
100	7	200	77.412
100	10	20	75.445
100	10	30	75.504
100	10	50	76.766
100	10	100	77.500
100	10	200	77.685
100	14	20	73.831
100	14	30	74.663
100	14	50	74.897

100	14	100	75.602
100	14	200	75.934
200	7	10	71.835
200	10	10	71.728
200	14	10	70.749
200	7	20	74.604
200	7	30	75.934
200	7	50	74.604
200	7	100	77.196
200	7	200	77.412
200	10	20	75.445
200	10	30	75.504
200	10	50	76.766
200	10	100	77.500
200	10	200	77.685
200	14	20	73.831
200	14	30	74.663
200	14	50	74.897
200	14	100	75.602
200	14	200	75.934

4.2.4.3 DNN based regularization

For DNN based regularization, using adam optimizer, only the parameter learning rate is changed while maintaining other parameters (beta_1=0.9, beta_2=0.999, epsilon=1e-08). And also the batch size, epoch size and layers are changed. Showing the highest validation accuracy with minimum difference of train loss and validation loss model is saved. And then the test data is evaluated with the stored model. Table 4-11 shows the result of LSTM, Table 4-12 shows the result of CNN 1D and Table 4-13 shows the result of CNN 2D respectively.

Table 4-11. Comparison by changing different parameters with test data in LSTM

Learning rate	Batch	Epoch	Layers	Train loss	Validation loss	Validation accuracy	Test accuracy
1E-3	32	50	LSTM(50), Dense(100)	0.152	0.258	0.923	92.715
1E-3	32	50	LSTM(100), Dense(100)	0.065	0.221	0.947	94.817
1E-3	32	100	LSTM(100), Dense(100)	0.018	0.239	0.959	95.898
1E-4	32	50	LSTM(50), Dense(100)	0.583	0.606	0.757	76.700
1E-4	64	50	LSTM(50), Dense(100)	0.576	0.598	0.756	76.377
1E-4	128	50	LSTM(50), Dense(100)	0.721	0.738	0.686	70.638
1E-4	128	50	LSTM(100), Dense(100)	0.414	0.429	0.846	84.915
1E-4	128	100	LSTM(100), Dense(100)	0.230	0.298	0.900	89.724
1E-5	32	50	LSTM(50), Dense(100)	0.805	0.810	0.667	68.273
1E-5	64	100	LSTM(100), Dense(100)	0.753	0.760	0.686	69.718
1E-5	128	100	LSTM(100), Dense(100)	0.797	0.810	0.680	69.304

The result from the LSTM shows that learning rate 1E-3, batch size 64 is the turning point of the loss. When learning rate is bigger, the train loss is much higher than validation loss which shows underfitting. When the learning rate is smaller, the loss increases. When the batch size is smaller or larger than 64, the loss increases. Having the minimum gap between training loss and validation loss and showing the highest validation accuracy is 75.590% when the learning rate is 1E-3, batch size 64, epoch 50, and LSTM hidden layer 50 with dense layer 100. In this stage, the test data accuracy showed 76.377%. From this experiment, it is clear that RNN can well handle time series data.

Table 4-12. Comparison by changing different parameters with test data in CNN 1D

Learning rate	Batch	Epoch	Layers	Train loss	Validation loss	Validation accuracy	Test accuracy
1E-3	32	50	con(32), con(64), con(128), con(256), Dense(100)	0.052	0.369	0.920	53.863
1E-3	32	100	con(32), con(64), con(128), con(256), Dense(100)	0.029	0.466	0.905	53.818
1E-4	32	50	con(32), con(64), con(128), con(256), Dense(100)	0.021	0.523	0.911	50.724
1E-4	32	100	con(32), con(64), con(128), con(256), Dense(100)	0.007	0.592	0.910	54.279
1E-5	32	50	con(32), con(64), con(128), con(256), Dense(100)	0.030	0.349	0.899	53.962
1E-5	32	100	con(32), con(64), con(128), con(256), Dense(100)	0.023	0.379	0.893	53.546
1E-6	32	50	con(32), con(64), con(128), con(256), Dense(100)	0.433	0.695	0.725	46.481
1E-6	32	100	con(32), con(64), con(128), con(256), Dense(100)	0.223	0.379	0.873	56.939
1E-6	256	100	con(32), con(64), con(128), Dense(100)	0.560	0.601	0.774	52.913

1E-6	256	100	con(32), con(64), Dense(100)	0.750	0.766	0.692	40.474
1E-7	32	50	con(32), con(64), con(128), con(256), Dense(100)	0.908	0.847	0.661	45.468
1E-7	32	100	con(32), con(64), con(128), con(256), Dense(100)	0.854	0.810	0.664	53.881
1E-7	64	50	con(32), con(64), con(128), con(256), Dense(100)	0.841	0.797	0.680	50.823
1E-7	64	100	con(32), con(64), con(128), con(256), Dense(100)	0.747	0.715	0.722	52.560
1E-7	128	50	con(32), con(64), con(128), con(256), Dense(100)	0.877	0.868	0.633	36.485
1E-7	128	100	con(32), con(64), con(128), con(256), Dense(100)	0.738	0.707	0.688	49.159
1E-7	256	50	con(32), con(64), con(128), con(256), Dense(100)	0.908	0.900	0.626	40.528
1E-7	256	100	con(32), con(64), con(128), con(256), Dense(100)	0.784	0.786	0.690	48.354
1E-7	256	100	con(32), con(64), con(128), con(256), con(512),Dense(100)	0.640	0.684	0.732	47.449
1E-7	256	100	con(32), con(64), con(128), con(256), con(512),con(1024),Dense(100)	0.476	0.555	0.788	54.026
1E-7	256	100	con(32), con(64), con(128), Dense(100)	0.895	0.887	0.631	37.290
1E-8	32	50	con(32), con(64), con(128), con(256), Dense(100)	1.344	1.203	0.453	50.208
1E-8	32	100	con(32), con(64), con(128), con(256), Dense(100)	1.170	1.071	0.537	31.120

1E-8	256	100	con(32), con(64), con(128), con(256), con(512),con(1024),Dense(100)	0.953	0.957	0.606	38.990
1E-9	32	50	con(32), con(64), con(128), con(256), Dense(100)	2.300	2.194	0.192	16.790
1E-9	32	100	con(32), con(64), con(128), con(256), Dense(100)	1.799	1.686	0.337	36.611

The result from the CNN 1D shows that learning rate 1E-7, batch size 256 is the turning point of the loss. When learning rate is bigger, the train loss is much higher than validation loss which shows underfitting. When the learning rate is smaller, the loss increases. Increasing the batch size and epoch both affected to lower the loss. Increasing the layer also had increased the loss and decreasing the layer had increased underfitting. Having the minimum gap between training loss and validation loss and showing the highest validation accuracy is 69.030% when the learning rate is 1E-7, batch size 256, epoch 100, and four convolution layers where each layer has filter size 32, 64, 128 and 256, and dense layer 100 and 7. In this stage, the test data accuracy showed 48.354%. This results show that RNN shows better perfomance on handling time series data. And it also shows that time signal of commuting activity cannot be distinguished well even using deep learning approach as there are minimum difference among vehicles.

Table 4-13. Comparison by changing different parameters with test data in CNN 2D

Learning rate	Batch	Epoch	Layers	Train loss	Validation loss	Validation accuracy	Test accuracy
1E-3	32	50	con(32), con(64), con(128), con(256), Dense(100)	0.1309	0.3089	0.8924	55.075
1E-3	32	100	con(32), con(64), con(128), con(256), Dense(100)	0.1524	0.3238	0.8904	56.495
1E-4	32	50	con(32), con(64), con(128), con(256), Dense(100)	0.1475	0.9569	0.8573	48.724
1E-4	32	100	con(32), con(64), con(128), con(256), Dense(100)	0.1507	0.3254	0.8841	55.428
1E-5	32	50	con(32), con(64), con(128), con(256), Dense(100)	0.5346	0.8006	0.7916	50.914
1E-5	32	100	con(32), con(64), con(128), con(256), Dense(100)	0.5788	0.5891	0.7675	52.379
1E-6	32	50	con(32), con(64), con(128), con(256), Dense(100)	0.9885	0.6054	0.651	51.348
1E-6	32	100	con(32), con(64), con(128), con(256), Dense(100)	0.9561	0.8402	0.6689	46.282
1E-7	32	50	con(32), con(64), con(128), con(256), Dense(100)	1.6208	1.4789	0.4005	34.947
1E-7	32	100	con(32), con(64), con(128), con(256), Dense(100)	1.5412	0.3867	0.4266	19.522
1E-7	64	50	con(32), con(64), con(128), con(256), Dense(100)	1.3966	1.3335	0.4579	19.540
1E-7	64	100	con(32), con(64), con(128), con(256), Dense(100)	1.3935	1.315	0.4587	29.917
1E-7	128	50	con(32), con(64), con(128), con(256), Dense(100)	1.3828	1.3606	0.4616	38.466

1E-7	128	100	con(32), con(64), con(128), con(256), Dense(100)	1.4421	1.3953	0.4165	18.337
1E-7	256	50	con(32), con(64), con(128), con(256), Dense(100)	1.2619	1.2385	0.4984	36.711
1E-7	256	100	con(32), con(64), con(128), con(256), Dense(100)	1.2801	1.2593	0.5012	33.038
1E-7	256	100	con(32), con(64), con(128), con(256), con(512), Dense(100)	1.2785	1.2695	0.4758	38.05
1E-7	256	100	con(32), con(64), con(128), con(256), con(512), Dense(100) paddingsame	0.8494	0.8489	0.6551	42.817
1E-7	256	100	con(32), con(64), con(128), con(256), con(512), con(1024),Dense(100),paddingsame	0.714	0.7283	0.7201	48.607
1E-7	256	100	con(32), con(64), con(128), con(256), con(512), con(1024),con(2048),Dense(100),paddingsame	0.6141	0.6427	0.7625	53.329
1E-7	256	100	con(32), con(64), con(128), con(256), con(512), con(1024),Dense(500),Dense(100),paddingsame	0.5973	0.6277	0.7655	52.108
1E-8	32	50	con(32), con(64), con(128), con(256), Dense(100)	3.1244	3.1107	0.1361	9.960
1E-8	32	100	con(32), con(64), con(128), con(256), Dense(100)	2.4581	0.1767	0.1801	28.297
1E-9	32	50	con(32), con(64), con(128), con(256), Dense(100)	2.1854	2.1036	0.1896	6.649
1E-9	32	100	con(32), con(64), con(128), con(256), Dense(100)	2.4211	0.1459	0.151	25.882

The result from the CNN 2D also shows that learning rate 1E-7, batch size 256 is the turning point of the loss. When learning rate is bigger, the train loss is much higher than validation loss which shows underfitting. When the learning rate is smaller, the loss increases. Increasing the batch size and epoch both affected to lower the loss. Increasing the layer has decreased the loss even further but increased underfitting. Having the minimum gap between training loss and validation loss and showing the highest validation accuracy is 65.51% when the learning rate is 1E-7, batch size 256, epoch 100, and five convolution layers where each layer has filter size 32, 64, 128, 256 and 512, and dense layer 100 and 7. In this stage, the test data accuracy showed 42.817%. This results also show that RNN shows better perfomance on handling time series data, and that time signal of commuting activity cannot be distinguished well even using deep learning approach. In addition, CNN 2D showed lower accuracy than CNN 1D about 5.537%. The reason for this might be that each axis of sensor are not correlated in commuting activities.

4.2.5 Comparison of different research using optimized parameter

Based on the optimized parameter, comparision experiment was conducted. As shown in the result in Table 4-14, proposed method shows the highest accuracy.

Table 4-14. Comparison with different research using different test data using RF

Research	Accuracy (%)
Widhalm et al.	74.234
Hemminki et al.	73.287
Wang et al.	72.957
Fang et al.	73.190
Proposed method	78.106

5. Conclusion

In this research, a human commuting activity recognition method using inertial sensors and GPS in a smartphone environment is proposed. Total seven activities were collected including stay, walk, jog, car, bus, subway and bicycle which all are common activities performed for commuting.

Summing sensor signals of each axis is conducted to only refer magnitude for orientation independency. Also offsetting the gravity axis is performed to only refer the gravity axis for detecting natural vibration. Features are extracted from time domain and frequency domain. In time domain, general statistical features were selected and extracted. In frequency domain, natural vibration features were extracted. Before classfication based on statistical classifier, GPS speed based classification is undergone. If the speed is low, it could be one of all activities where whole activity based classification is made. If the speed is high, it would be one of car, bus, subway and bike. Stay is also included in here due to the possibility of misclassification. When the speed is high, coordinate data is used to distinguish subway and other vehicles when running ground section. After classifying activity, to revise errors from pre-defined classification, post-processing is undergone, namely activity correction. Then with proposed correction rule, current activity, previous activity and previously corrected activity result is referred for final revision.

Natural vibration features are proved to show better result when new test data is input, while other method showed lower accuracy. It is also proved that correction method increases the accuracy by correcting the error from classification.

The future work will be collecting more activity in different route for improved generalization, and setting the window in variable size that fixed size may contain erroneous data where the activity might be changed in the same window. Another future work is to increase the type of commuting activities such as motorcycle or kickboard. This may require to use additional sensors that smartphone cannot reflect the movement of limbs. Therefore, future work will try to use smartwatch, which complies the usage of minimum sensor devices with unobtrusiveness.

6. Bibliography

1. Kim, E., Helal, S., & Cook, D. (2009). Human activity recognition and pattern discovery. IEEE pervasive computing, 9(1), 48-53.
2. Banos Legren, O. (2014). Robust expert systems for more flexible real-world activity recognition. Universidad de Granada.
3. Lara, O. D., & Labrador, M. A. (2012). A survey on human activity recognition using wearable sensors. IEEE communications surveys & tutorials, 15(3), 1192-1209.
4. Shoaib, M., Bosch, S., Incel, O. D., Scholten, H., & Havinga, P. J. (2015). A survey of online activity recognition using mobile phones. Sensors, 15(1), 2059-2085.
5. Turaga, P., Chellappa, R., Subrahmanian, V. S., & Udrea, O. (2008). Machine recognition of human activities: A survey. IEEE Transactions on Circuits and Systems for Video technology, 18(11), 1473.
6. Chen, L., Hoey, J., Nugent, C. D., Cook, D. J., & Yu, Z. (2012). Sensor-based activity recognition. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(6), 790-808.
7. Banos, O., Amin, M. B., Khan, W. A., Afzal, M., Hussain, M., Kang, B. H., & Lee, S. (2016). The Mining Minds digital health and wellness framework. Biomedical engineering online, 15(1), 76.
8. Popoola, O. P., & Wang, K. (2012). Video-based abnormal human behavior recognition—A review. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(6), 865-878.
9. Aggarwal, J. K., & Xia, L. (2014). Human activity recognition from 3d data: A review. Pattern Recognition Letters, 48, 70-80.
10. Ke, S. R., Thuc, H. L. U., Lee, Y. J., Hwang, J. N., Yoo, J. H., & Choi, K. H.

(2013). A review on video-based human activity recognition. computers, 2(2), 88-131.

11. Robertson, N., & Reid, I. (2006). A general method for human activity recognition in video. Computer Vision and Image Understanding, 104(2-3), 232-248.
12. Ribeiro, P. C., Santos-Victor, J., & Lisboa, P. (2005, September). Human activity recognition from video: modeling, feature selection and classification architecture. In Proceedings of International Workshop on Human Activity Recognition and Modelling (pp. 61-78).
13. Mohamed, R., Perumal, T., Sulaiman, M. N., & Mustapha, N. (2017). Multi resident complex activity recognition in smart home: A literature review. Int. J. Smart Home, 11(6), 21-32.
14. Mehr, H. D., Polat, H., & Cetin, A. (2016, April). Resident activity recognition in smart homes by using artificial neural networks. In 2016 4th international istanbul smart grid congress and fair (ICSG) (pp. 1-5). IEEE.
15. Alsinglawi, B., Nguyen, Q. V., Gunawardana, U., Maeder, A., & Simoff, S. J. (2017). RFID systems in healthcare settings and activity of daily living in smart homes: A review. E-Health Telecommunication Systems and Networks, 6, 1-17.
16. Qi, J., Yang, P., Waraich, A., Deng, Z., Zhao, Y., & Yang, Y. (2018). Examining sensor-based physical activity recognition and monitoring for healthcare using Internet of Things: A systematic review. Journal of biomedical informatics, 87, 138-153.
17. Ravi, N.; Dandekar, N.; Mysore, P.; Littman, M.L. Activity recognition from accelerometer data. In Proceedings of the Twentieth National Conference on Artificial Intelligence, Pittsburgh, PA, USA, 9–13 July 2005; Volume 5, pp. 1541–1546.
18. Maurer, U.; Smailagic, A.; Siewiorek, D.P.; Deisher, M. Activity recognition and monitoring using multiple sensors on different body positions. In Proceedings of

the International Workshop on Wearable and Implantable Body Sensor Networks, Cambridge, MA, USA, 3–5 April 2006.

19. Banos, O.; Damas, M.; Pomares, H.; Prieto, A.; Rojas, I. Daily living activity recognition based on statistical feature quality group selection. Expert Syst. Appl. 2012, 39, 8013–8021.
20. Banos, O.; Damas, M.; Pomares, H.; Rojas, F.; Delgado-Marquez, B.; Valenzuela, O. Human activity recognition based on a sensor weighting hierarchical classifier. Soft Comput. 2013, 17, 333–343.
21. https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/ (accessed on 4 April 2019).
22. Wang, L., Gjoreski, H., Ciliberto, M., Mekki, S., Valentin, S., & Roggen, D. (2019). Enabling reproducible research in sensor-based transportation mode recognition with the Sussex-Huawei dataset. IEEE Access, 7, 10870-10891.
23. Gudivada, V., Apon, A., & Ding, J. (2017). Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. International Journal on Advances in Software, 10(1), 1-20.
24. https://en.wikipedia.org/wiki/Natural_frequency (accessed on 5 April 2019)
25. Izumi, J. (1999). Combined environmental testing for equipment used on automobiles (No. 9). ESPEC Technology Report.
26. Banos, O.; Villalonga, C.; Garcia, R.; Saez, A.; Damas, M.; Holgado-Terriza, J.A.; Pomares H.; Lee, S.; Rojas, I. Design, implementation and validation of a novel open framework for agile development of mobile health applications. Biomed. Eng. Online 2015, 14, S6.
27. Kaghyan, S.; Sarukhanyan, H. Activity recognition using K-nearest neighbor algorithm on smartphone with tri-axial accelerometer. Int. J. Inf. Models Anal. 2012, 1, 146–156.
28. Berchtold, M.; Budde, M.; Gordon, D.; Schmidtke, H.R.; Beigl, M. Actiserv:

Activity recognition service for mobile phones. In Proceedings of the 2010 International Symposium on Wearable Computers, Seoul, Korea, 10–13 October 2010; pp. 1–8.

29. Ayu, M.A.; Mantoro, T.; Matin, A.F.A.; Basamh, S.S. Recognizing user activity based on accelerometer data from a mobile phone. In Proceedings of the 2011 IEEE Symposium on Computers & Informatics, Kuala Lumpur, Malaysia, 20–23 March 2011; pp. 617–621.
30. Du, H.; Yu, Z.; Yi, F.; Wang, Z.; Han, Q.; Guo, B. Group mobility classification and structure recognition using mobile devices. In Proceedings of the 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom), Sydney, Australia, 14–18 March 2016; pp. 1–9.
31. Wang, S.; Chen, C.; Ma, J. Accelerometer Based Transportation Mode Recognition on Mobile Phones. In Proceedings of the 2010 Asia-Pacific Conference on Wearable Computing Systems, Shenzhen, China, 17–18 April 2010; pp. 44–46.
32. Siirtola, P.; Röning, J. Ready-to-use activity recognition for smartphones. In Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Data Mining, Singapore, 16–19 April 2013; pp. 59–64.
33. Han, M.; Lee, Y.K.; Lee, S. Comprehensive context recognizer based on multimodal sensors in a smartphone. Sensors 2012, 12, 12588–12605.
34. Karantonis, D. M., Narayanan, M. R., Mathie, M., Lovell, N. H., & Celler, B. G. (2006). Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE transactions on information technology in biomedicine, 10(1), 156-167.
35. Mathie, M. J., Celler, B. G., Lovell, N. H., & Coster, A. C. F. (2004). Classification of basic daily movements using a triaxial accelerometer. Medical and Biological Engineering and Computing, 42(5), 679-687.
36. Yang, J. Y., Wang, J. S., & Chen, Y. P. (2008). Using acceleration measurements

for activity recognition: An effective learning algorithm for constructing neural classifiers. Pattern recognition letters, 29(16), 2213-2220.

37. Bonomi, A. G., Goris, A. H., Yin, B., & Westerterp, K. R. (2009). Detection of type, duration, and intensity of physical activity using an accelerometer. Medicine & Science in Sports & Exercise, 41(9), 1770-1777.

38. Gupta, P., & Dallas, T. (2014). Feature selection and activity recognition system using a single triaxial accelerometer. IEEE Transactions on Biomedical Engineering, 61(6), 1780-1786.

39. Masse, F., Gonzenbach, R. R., Arami, A., Paraschiv-Ionescu, A., Luft, A. R., & Aminian, K. (2015). Improving activity recognition using a wearable barometric pressure sensor in mobility-impaired stroke patients. Journal of neuroengineering and rehabilitation, 12(1), 72.

40. Frank, K., Nadales, M. V., Robertson, P., & Angermann, M. (2010, September). Reliable real-time recognition of motion related human activities using MEMS inertial sensors. In Proceedings of the 23rd International Technical Meeting of the Satellite Division of the Institute of Navigation (pp. 2906-2912).

41. Garcia-Ceja, E., Brena, R., Carrasco-Jimenez, J., & Garrido, L. (2014). Long-term activity recognition from wristwatch accelerometer data. Sensors, 14(12), 22500-22524.

42. Khan, A. M., Lee, Y. K., Lee, S. Y., & Kim, T. S. (2010). A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. IEEE transactions on information technology in biomedicine, 14(5), 1166-1172.

43. Casale, P., Pujol, O., & Radeva, P. (2011, June). Human activity recognition from accelerometer data using a wearable device. In Iberian Conference on Pattern Recognition and Image Analysis (pp. 289-296). Springer, Berlin, Heidelberg.

44. Long, X., Yin, B., & Aarts, R. M. (2009, September). Single-accelerometer-based

daily physical activity classification. In 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 6107-6110). IEEE.

45. Parkka, J., Ermes, M., Korpipaa, P., Mantyjarvi, J., Peltola, J., & Korhonen, I. (2006). Activity classification using realistic data from wearable sensors. IEEE Transactions on information technology in biomedicine, 10(1), 119-128.

46. Lyons, G. M., Culhane, K. M., Hilton, D., Grace, P. A., & Lyons, D. (2005). A description of an accelerometer-based mobility monitoring technique. Medical engineering & physics, 27(6), 497-504.

47. Yeoh, W. S., Pek, I., Yong, Y. H., Chen, X., & Waluyo, A. B. (2008, August). Ambulatory monitoring of human posture and walking speed using wearable accelerometer sensors. In 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 5184-5187). IEEE.

48. Salarian, A., Russmann, H., Vingerhoets, F. J., Burkhard, P. R., & Aminian, K. (2007). Ambulatory monitoring of physical activities in patients with Parkinson's disease. IEEE Transactions on Biomedical Engineering, 54(12), 2296-2299.

49. Mannini, A., Intille, S. S., Rosenberger, M., Sabatini, A. M., & Haskell, W. (2013). Activity recognition using a single accelerometer placed at the wrist or ankle. Medicine and science in sports and exercise, 45(11), 2193.

50. Olguın, D. O., & Pentland, A. S. (2006, October). Human activity recognition: Accuracy across common locations for wearable sensors. In Proceedings of 2006 10th IEEE international symposium on wearable computers, Montreux, Switzerland (pp. 11-14). Citeseer.

51. Chamroukhi, F., Mohammed, S., Trabelsi, D., Oukhellou, L., & Amirat, Y. (2013). Joint segmentation of multivariate time series with hidden process regression for human activity recognition. Neurocomputing, 120, 633-644.

52. Moncada-Torres, A., Leuenberger, K., Gonzenbach, R., Luft, A., & Gassert, R. (2014). Activity classification based on inertial and barometric pressure sensors at

different anatomical locations. Physiological measurement, 35(7), 1245.

53. Attal, F., Mohammed, S., Dedabrishvili, M., Chamroukhi, F., Oukhellou, L., & Amirat, Y. (2015). Physical human activity recognition using wearable sensors. Sensors, 15(12), 31314-31338.

54. Pirttikangas, S., Fujinami, K., & Nakajima, T. (2006, October). Feature selection and activity recognition from wearable sensors. In International symposium on ubiquitious computing systems (pp. 516-527). Springer, Berlin, Heidelberg.

55. Banos, O.; Damas, M.; Pomares, H.; Rojas, F.; Delgado-Marquez, B.; Valenzuela, O. Human activity recognition based on a sensor weighting hierarchical classifier. Soft Comput. 2013, 17, 333–343.

56. Gao, L., Bourke, A. K., & Nelson, J. (2014). Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems. Medical engineering & physics, 36(6), 779-785.

57. Gjoreski, H., Lustrek, M., & Gams, M. (2011, July). Accelerometer placement for posture recognition and fall detection. In 2011 Seventh International Conference on Intelligent Environments (pp. 47-54). IEEE.

58. Bao, L.; Intille, S.S. Activity recognition from user-annotated acceleration data. In Proceedings of the International Conference on Pervasive Computing, Vienna, Austria, 21–23 April 2004; pp. 1–17.

59. Atallah, L., Lo, B., King, R., & Yang, G. Z. (2011). Sensor positioning for activity recognition using wearable accelerometers. IEEE transactions on biomedical circuits and systems, 5(4), 320-329.

60. Cleland, I., Kikhia, B., Nugent, C., Boytsov, A., Hallberg, J., Synnes, K., ... & Finlay, D. (2013). Optimal placement of accelerometers for the detection of everyday activities. Sensors, 13(7), 9183-9200.

61. Shoaib, M., Bosch, S., Incel, O., Scholten, H., & Havinga, P. (2016). Complex human activity recognition using smartphone and wrist-worn motion sensors.

Sensors, 16(4), 426.

62. Khan, A. M., Lee, Y. K., Lee, S. Y., & Kim, T. S. (2010, May). Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In 2010 5th international conference on future information technology (pp. 1-6). IEEE.
63. Khan, A. M., Tufail, A., Khattak, A. M., & Laine, T. H. (2014). Activity recognition on smartphones via sensor-fusion and kda-based svms. International Journal of Distributed Sensor Networks, 10(5), 503291.
64. Shoaib, M., Bosch, S., Incel, O., Scholten, H., & Havinga, P. (2014). Fusion of smartphone motion sensors for physical activity recognition. Sensors, 14(6), 10146-10176.
65. Martín, H., Bernardos, A. M., Iglesias, J., & Casar, J. R. (2013). Activity logging using lightweight classification techniques in mobile devices. Personal and ubiquitous computing, 17(4), 675-695.
66. Ryder, J., Longstaff, B., Reddy, S., & Estrin, D. (2009, August). Ambulation: A tool for monitoring mobility patterns over time using mobile phones. In 2009 International Conference on Computational Science and Engineering (Vol. 4, pp. 927-931). IEEE.
67. Kim, T. S., Cho, J. H., & Kim, J. T. (2013). Mobile motion sensor-based human activity recognition and energy expenditure estimation in building environments. In Sustainability in Energy and Buildings (pp. 987-993). Springer, Berlin, Heidelberg.
68. Ouchi, K., & Doi, M. (2012, September). Indoor-outdoor activity recognition by a smartphone. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing (pp. 600-601). ACM.
69. Zhao, K., Du, J., Li, C., Zhang, C., Liu, H., & Xu, C. (2013, October). Healthy: A diary system based on activity recognition using smartphone. In 2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems (pp.

290-294). IEEE.

70. Anjum, A.; Ilyas, M.U. Activity recognition using smartphone sensors. In Proceedings of the 2013 IEEE 10th Consumer Communications and Networking Conference, Las Vegas, NV, USA, 11–14 January 2013; pp. 914–919.

71. Henpraserttae, A.; Thiemjarus, S.; Marukatat, S. Accurate activity recognition using a mobile phone regardless of device orientation and location. In Proceedings of the 2011 International Conference on Body Sensor Networks, Dallas, TX, USA, 23–25 May 2011; pp. 41–46.

72. Lu, H.; Yang, J.; Liu, Z.; Lane, N.D.; Choudhury, T.; Campbell, A.T. The Jigsaw continuous sensing engine for mobile phone applications. In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, Zurich, Switzerland, 3–5 November 2010; pp. 71–84.

73. Khan, A. M., Lee, Y. K., Lee, S. Y., & Kim, T. S. (2010, May). Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In 2010 5th international conference on future information technology (pp. 1-6). IEEE.

74. Chen, Y., & Shen, C. (2017). Performance analysis of smartphone-sensor behavior for human activity recognition. Ieee Access, 5, 3095-3110.

75. Thiemjarus, S., Henpraserttae, A., & Marukatat, S. (2013, May). A study on instance-based learning with reduced training prototypes for device-context-independent activity recognition on a mobile phone. In 2013 IEEE International Conference on Body Sensor Networks (pp. 1-6). IEEE.

76. Khan, A., Siddiqi, M., & Lee, S. W. (2013). Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones. Sensors, 13(10), 13099-13122.

77. Yang, J.; Nguyen, M. N.; San, P. P.; Li, X.; Krishnaswamy, S. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity

Recognition. 24th International joint conference on artificial intelligence, Buenos Aires, Argentina, 25–31 July 2015, Vol. 15, 3995-4001.

78. Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2012, December). Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In International workshop on ambient assisted living (pp. 216-223). Springer, Berlin, Heidelberg.

79. Lee, Y. S., & Cho, S. B. (2011, May). Activity recognition using hierarchical hidden markov models on a smartphone with 3D accelerometer. In International Conference on Hybrid Artificial Intelligence Systems (pp. 460-467). Springer, Berlin, Heidelberg.

80. Reyes-Ortiz, J. L., Oneto, L., Samà, A., Parra, X., & Anguita, D. (2016). Transition-aware human activity recognition using smartphones. Neurocomputing, 171, 754-767.

81. Lu, Y., Wei, Y., Liu, L., Zhong, J., Sun, L., & Liu, Y. (2017). Towards unsupervised physical activity recognition using smartphone accelerometers. Multimedia Tools and Applications, 76(8), 10701-10719.

82. Keally, M., Zhou, G., Xing, G., Wu, J., & Pyles, A. (2011, November). Pbn: towards practical activity recognition using smartphone-based body sensor networks. In Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (pp. 246-259). ACM.

83. He, Y., & Li, Y. (2013). Physical activity recognition utilizing the built-in kinematic sensors of a smartphone. International Journal of Distributed Sensor Networks, 9(4), 481580.

84. Reiss, A., Hendeby, G., & Stricker, D. (2013). A competitive approach for human activity recognition on smartphones. In European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013), 24-26 April, Bruges, Belgium (pp. 455-460). ESANN.

85. Capela, N. A., Lemaire, E. D., & Baddour, N. (2015). Feature selection for wearable smartphone-based human activity recognition with able bodied, elderly, and stroke patients. PloS one, 10(4), e0124414.

86. Ronao, C. A., & Cho, S. B. (2014, August). Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models. In 2014 10th International Conference on Natural Computation (ICNC) (pp. 681-686). IEEE.

87. Yu, M. C., Yu, T., Wang, S. C., Lin, C. J., & Chang, E. Y. (2014). Big data small footprint: the design of a low-power classifier for detecting transportation modes. Proceedings of the VLDB Endowment, 7(13), 1429-1440.

88. Xia, H., Qiao, Y., Jian, J., & Chang, Y. (2014). Using smart phone sensors to detect transportation modes. Sensors, 14(11), 20843-20865.

89. Sankaran, K., Zhu, M., Guo, X. F., Ananda, A. L., Chan, M. C., & Peh, L. S. (2014, November). Using mobile phone barometer for low-power transportation context detection. In Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (pp. 191-205). ACM.

90. Yang, J. (2009, October). Toward physical activity diary: motion recognition using simple acceleration features with mobile phones. In Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics (pp. 1-10). ACM.

91. Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., & Srivastava, M. (2010). Using mobile phones to determine transportation modes. ACM Transactions on Sensor Networks (TOSN), 6(2), 13.

92. Fang, S. H., Fei, Y. X., Xu, Z., & Tsao, Y. (2017). Learning transportation modes from smartphone sensors based on deep neural network. IEEE Sensors Journal, 17(18), 6111-6118.

93. Su, X., Caceres, H., Tong, H., & He, Q. (2016). Online travel mode identification

using smartphones with battery saving considerations. IEEE Transactions on Intelligent Transportation Systems, 17(10), 2921-2934.

94. Siirtola, P., & Röning, J. (2012). Recognizing human activities user-independently on smartphones based on accelerometer data. IJIMAI, 1(5), 38-45.

95. Sonderen, T. (2016). Detection of transportation mode solely using smartphones. In Proc. 20th Student Conf. IT (pp. 1-7).

96. Feng, T., & Timmermans, H. J. (2013). Transportation mode recognition using GPS and accelerometer data. Transportation Research Part C: Emerging Technologies, 37, 118-130.

97. Lu, H., Yang, J., Liu, Z., Lane, N. D., Choudhury, T., & Campbell, A. T. (2010, November). The Jigsaw continuous sensing engine for mobile phone applications. In Proceedings of the 8th ACM conference on embedded networked sensor systems (pp. 71-84). ACM.

98. Shafique, M. A., & Hato, E. (2015). Use of acceleration data for transportation mode prediction. Transportation, 42(1), 163-188.

99. Shin, D., Aliaga, D., Tunçer, B., Arisona, S. M., Kim, S., Zünd, D., & Schmitt, G. (2015). Urban sensing: Using smartphones for transportation mode classification. Computers, Environment and Urban Systems, 53, 76-86.

100. Nick, T., Coersmeier, E., Geldmacher, J., & Goetze, J. (2010, July). Classifying means of transportation using mobile sensor data. In The 2010 International Joint Conference on Neural Networks (IJCNN) (pp. 1-6). IEEE.

101. Jahangiri, A., & Rakha, H. A. (2015). Applying machine learning techniques to transportation mode recognition using mobile phone sensor data. IEEE transactions on intelligent transportation systems, 16(5), 2406-2417.

102. Han, M.; Lee, Y.K.; Lee, S. Comprehensive context recognizer based on multimodal sensors in a smartphone. Sensors 2012, 12, 12588–12605.

103. Ashqar, H. I., Almannaa, M. H., Elhenawy, M., Rakha, H. A., & House, L.

(2018). Smartphone transportation mode recognition using a hierarchical machine learning classifier and pooled features from time and frequency domains. IEEE Transactions on Intelligent Transportation Systems, 20(1), 244-252.

104. Lari, Z. A., & Golroo, A. (2015, January). Automated transportation mode detection using smart phone applications via machine learning: Case study mega city of Tehran. In Proceedings of the Transportation Research Board 94th Annual Meeting, Washington, DC, USA (pp. 11-15).

105. Balli, S., & Sağbaş, E. A. (2017). Diagnosis of transportation modes on mobile phone using logistic regression classification. IET Software, 12(2), 142-151.

106. Lee, S., Lee, J., & Lee, K. (2017, June). VehicleSense: A reliable sound-based transportation mode recognition system for smartphones. In 2017 IEEE 18th international symposium on a world of wireless, mobile and multimedia networks (WoWMoM) (pp. 1-9). IEEE.

107. Chen, S., Luo, H., Zhao, F., Yuan, W., & Jiang, M. (2017, September). Transportation mode recognition algorithm based on multiple support vector machine classifiers. In 2017 5th International Conference on Enterprise Systems (ES) (pp. 253-259). IEEE.

108. Yanyun, G., Fang, Z., Shaomeng, C., & Haiyong, L. (2017, September). A convolutional neural networks based transportation mode identification algorithm. In 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN) (pp. 1-7). IEEE.

109. Byon, Y. J., Abdulhai, B., & Shalaby, A. (2009). Real-time transportation mode detection via tracking global positioning system mobile devices. Journal of Intelligent Transportation Systems, 13(4), 161-170.

110. Byon, Y. J., & Liang, S. (2014). Real-time transportation mode detection using smartphones and artificial neural networks: Performance comparisons between smartphones and conventional global positioning system sensors. Journal of

Intelligent Transportation Systems, 18(3), 264-272.

111. Dabiri, S., & Heaslip, K. (2018). Inferring transportation modes from GPS trajectories using a convolutional neural network. Transportation research part C: emerging technologies, 86, 360-371.

112. Hedemalm, E. (2017). Online transportation mode recognition and an application to promote greener transportation.

113. Manzoni, V., Maniloff, D., Kloeckl, K., & Ratti, C. (2010). Transportation mode identification and real-time CO2 emission estimation using smartphones. SENSEable City Lab, Massachusetts Institute of Technology, nd.

114. Zheng, Y., Chen, Y., Li, Q., Xie, X., & Ma, W. Y. (2010). Understanding transportation modes based on GPS data for web applications. ACM Transactions on the Web (TWEB), 4(1), 1.

115. Xiao, Z., Wang, Y., Fu, K., & Wu, F. (2017). Identifying different transportation modes from trajectory data using tree-based ensemble classifiers. ISPRS International Journal of Geo-Information, 6(2), 57.

116. Endo, Y., Toda, H., Nishida, K., & Ikedo, J. (2016). Classifying spatial trajectories using representation learning. International Journal of Data Science and Analytics, 2(3-4), 107-117.

117. Stenneth, L., Wolfson, O., Yu, P. S., & Xu, B. (2011, November). Transportation mode detection using mobile phones and GIS information. In Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems (pp. 54-63). ACM.

118. Semanjski, I., Gautama, S., Ahas, R., & Witlox, F. (2017). Spatial context mining approach for transport mode recognition from mobile sensed big data. Computers, Environment and Urban Systems, 66, 38-52.

119. Bolbol, A., Cheng, T., Tsapakis, I., & Haworth, J. (2012). Inferring hybrid transportation modes from sparse GPS data using a moving window SVM

classification. Computers, Environment and Urban Systems, 36(6), 526-537.

120. Gong, H., Chen, C., Bialostozky, E., & Lawson, C. T. (2012). A GPS/GIS method for travel mode detection in New York City. Computers, Environment and Urban Systems, 36(2), 131-139.

121. Gonzalez, P. A., Weinstein, J. S., Barbeau, S. J., Labrador, M. A., Winters, P. L., Georggi, N. L., & Perez, R. (2010). Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks. IET Intelligent Transport Systems, 4(1), 37-49.

122. Nitsche, P., Widhalm, P., Breuss, S., & Maurer, P. (2012). A strategy on how to utilize smartphones for automatically reconstructing trips in travel surveys. Procedia-Social and Behavioral Sciences, 48, 1033-1046.

123. J.Wenand Z.Wang, ``Learning general model for activity recognition with limited labelled data," Expert Syst. Appl., vol. 74, pp. 1928, May 2017.

124. Wang, Changhai, et al. WOODY: A Post-Process Method for Smartphone-Based Activity Recognition. IEEE Access, 2018, 6: 49611-49625

125. San-Segundo, R., Blunck, H., Moreno-Pimentel, J., Stisen, A., & Gil-Martín, M. (2018). Robust Human Activity Recognition using smartwatches and smartphones. Engineering Applications of Artificial Intelligence, 72, 190-202.

126. L. Zheng et al., ``A novel energy-efficient approach for human activity recognition," Sensors, vol. 17, no. 9, p. 2064, 2017

127. M. Zhong, J. Wen, P. Hu, and J. Indulska, ``Advancing android activity recognition service with Markov smoother," in Proc. Int. Conf. Pervasive Comput. Commun., Mar. 2015, pp. 3843.

128. M. Zhong, J. Wen, P. Hu, and J. Indulska, ``Advancing android activity recognition service with Markov smoother: Practical solutions," Pervasive Mobile Comput., vol. 38, pp. 60-76, Jul. 2017.

129. Wang, Lin, and Daniel Roggen. "Sound-based transportation mode recognition

with smartphones." ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.

130. Grzeszick, Rene, et al. "Deep neural network based human activity recognition for the order picking process." Proceedings of the 4th international Workshop on Sensor-based Activity Recognition and Interaction. ACM, 2017.

131. Han, C. W., Kang, S. J., & Kim, N. S. (2010). Implementation of hmm-based human activity recognition using single triaxial accelerometer. IEICE transactions on fundamentals of electronics, communications and computer sciences, 93(7), 1379-1383.

132. Karantonis, D. M., Narayanan, M. R., Mathie, M., Lovell, N. H., & Celler, B. G. (2006). Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE transactions on information technology in biomedicine, 10(1), 156-167.

133. Heckbert, P. (1995). Fourier transforms and the fast Fourier transform (FFT) Algorithm. Computer Graphics, 2, 15-463.

134. http://news.bbc.co.uk/sport2/hi/health_and_fitness/4286146.stm

135. P. Widhalm, P. Nitsche, and N. Brändie, ``Transport mode detection with realistic smartphone sensor data," in Proc. 21st Int. Conf. Pattern Recognit., Tsukuba, Japan, Nov. 2012, pp. 573576.

136. S. Hemminki, P. Nurmi, and S. Tarkoma, ``Accelerometer-based transportation mode detection on smartphones," in Proc. ACM Conf. Embedded Netw. Sensor Syst., Roma, Italy, 2013, pp. 114.

137. S. Wang, C. Chen, and J. Ma, ``Accelerometer based transportation mode recognition on mobile phones," in Proc. AsiaPacic Conf. Wearable Comput. Syst., Shenzhen, China, Apr. 2010, pp. 4446.

138. S.-H. Fang et al., ``Transportation modes classication using sensors on smartphones," Sensors, vol. 16, no. 8, pp. 13241339, Aug. 2016.

Appendix: A. List of publications

A-1. Journal Papers

SCI Journal Papers

1. Jamil Hussain, Fahad Ahmed Satti, Wajahat Ali Khan, Muhammad Afzal, Hafiz Syed Muhammad Bilal, Muhammad Zaki Ansaar, Hafiz Farooq Ahmad, **Taeho Hur**, Jaehun Bang, Jee-In Kim, Gwang Hoon Park, Hyonwoo Seung, and Sungyoung Lee, "Exploring the dominant features of social media for depression detection", Journal of Information Science (SCIE, IF: 2.327), Online First Publish, 2019
2. **Taeho Hur**, Jaehun Bang, Thien Huynh-The, Jongwon Lee, Jee-In Kim and Sungyoung Lee, "Iss2Image: A Novel Signal-Encoding Technique for CNN-Based Human Activity Recognition", Sensors (SCIE, IF: 2.475), Vol.18, Issue 11, pp.1-19, 2018
3. Jaehun Bang, **Taeho Hur**, Dohyeong Kim, Thien Huynh-The, Jongwon Lee, Yongkoo Han, Oresti Banos, Jee-In Kim and Sungyoung Lee, "Adaptive Data Boosting Technique for Robust Personalized Speech Emotion in Emotionally-Imbalanced Small-Sample Environments", Sensors (SCIE, IF: 2.475), Vol.18, Issue 11, pp.1-21, 2018
4. Maqbool Ali, Syed Imran Ali, Dohyeong Kim, **Taeho Hur**, Jaehun Bang, Sungyoung Lee, Byeong Ho Kang, and Maqbool Hussain, "uEFS: An efficient and comprehensive ensemble-based feature selection methodology to select informative features", PLoS ONE (SCIE, 2.766), Vol.13 No.8, 2018
5. Jamil Hussain, Wajahat Ali Khan, **Taeho Hur**, Hafiz Syed Muhammad Bilal, Jaehun

Bang, Anees Ul Hassan, Muhammad Afzal and Sungyoung Lee, "A Multimodal Deep Log-Based User Experience (UX) Platform for UX Evaluation", Sensors (SCIE, IF:2.677), Vol.18, Issue 5, pp.1-31, 2018

6. Taqdir Ali, Maqbool Hussain, Muhammad Afzal, Wajahat Ali Khan, **Taeho Hur**, Muhammad Bilal Amin, Dohyeong Kim, Byeong Ho Kang and Sungyoung Lee, "Clinically Harmonized Wellness Concepts Model for Health and Wellness Services", IEEE Access (SCIE, IF: 3.244), Vol.6, pp.26660-26674, 2018
7. Maqbool Ali, Rahman Ali, Wajahat Ali Khan, Soyeon Caren Han, Jaehun Bang, **Taeho Hur**, Dohyeong Kim, Sungyoung Lee, and Byeong Ho Kang, "A data-driven knowledge acquisition system: An end-to-end knowledge engineering process for generating production rules", IEEE Access (SCIE, IF: 3.244), Vol.6, pp.15587-15607, 2018
8. Thien Huynh-The, Cam-Hao Hua, Anh Tu Nguyen, **Taeho Hur**, Jaehun Bang, Dohyeong Kim, Muhammad B. Amin, Byeong Ho Kang, Hyonwoo Seung, Soo-Yong Shin, Eun-Soo Kim, Sungyoung Lee, "Hierarchical Topic Modeling With Pose-Transition Feature For Action Recognition Using 3D Skeleton Data", Information Sciences (SCI, IF:4.832), Vol.444, pp.20-35, 2018
9. Thien Huynh-The, Cam-Hao Hua, Anh Tu Nguyen, **Taeho Hur**, Jaehun Bang, Dohyeong Kim, Muhammad Bilal Amin, Byeong Ho Kang, Hyonwoo Seung and Sungyoung Lee, "Selective Bit Embedding Scheme For Robust Blind Color Image Watermarking", Information Science, (SCI, IF:4.832), Vol. 426, pp1-18, 2018
10. Muhammad Asif Razzaq, Claudia Villalonga, Sungyoung Lee, Usman Akhtar, Maqbool Ali, Eun-Soo Kim, Asad Masood Khattak, Hyonwoo Seung, **Taeho Hur**, Jaehun Bang, Dohyeong Kim and Wajahat Ali Khan, "mlCAF: Multi-Level Cross-Domain Semantic Context Fusioning for Behavior Identification", Sensors, (SCIE, IF:2.677), Vol.17, Issue 10, 2017
11. **Taeho Hur**, Jaehun Bang, Dohyeong Kim, Oresti Banos, Sungyoung Lee,

“Smartphone Location-Independent Physical Activity Recognition with Transportation Natural Vibration Analysis”, Sensors, (SCIE, IF:2.033), Vol.17, Issue 4, 2017

12. Oresti Banos, Claudia Villalonga, Jaehun Bang, **Taeho Hur**, Donguk Kang, Sangbeom Park, Thien Huynh-The, Vui Le-Ba, Muhammad Bilal Amin, Muhammad Asif Razzaq, Wahajat Ali Khan, Choong Seon Hong and Sungyoung Lee, "Human Behavior Analysis by Means of Multimodal Context Mining", Sensors, (SCIE, IF: 2.033), Vol.16, Issue 8, 2016

Non-SCI Journal Papers

1. **Taeho Hur**, Sungyoung Lee, “Multi-Level Fusion Activity Recognition Framework using Smart Devices”, Journal of KIISE (JOK), Vol. 45, No. 9, pp.950-956, 2018
2. Wajahat Ali Khan, Syed Imran Ali, Muhammad Bilal Ali, Shujaat Hussain, **Taeho Hur**, Sungyoung Lee, "Healthcare standard interoperability”, Information & communications magazine, Vol. 35, No. 2, pp. 73-84, 2018
3. **Taeho Hur**, Haneul Yeom, Sungyoung Lee, “Accurate Step Count Detection based on Recognizing Smartphone Hold Position”, Journal of KIISE (JOK), Vol.44, Issue 4, pp374-382, 2017
4. Muhammad Bilal Amin, Wajahat Ali Khan, Bilal Ali, Jaehun Bang, Taqdir Ali, **Taeho Hur**, Shujaat Hussain, Imran Ali, Dohyeong Kim “Health and Wellness platforms: A Survey on Services and Enabling Technologies”, Communications of the Korean Institute of Information Scientists and Engineers, Vol.35 No.7 (Wn.338), pp. 9-25, 2017
5. Oresti Baonos, Wajahat Ali Khan, Muhammad Bilal Amin, **Taeho Hur**, Jaehun Bang, Donguk Kang, Maqbool Hussain, Muhammad Afzal, Taqdir Ali and Sungyoung Lee, “Big Data based Mining Minds Healthcare Framework”, Journal of Korea Institute Of Communication Sciences, Vol.32, No.11 , pp.12 -pp.20, 2015

A-2. Conference Papers

International Conference Papers

1. Jaehun Bang, Dong Uk Kang, **Tae Ho Hur**, Thien Hyun The, Muhammad Asif Razzaq, Wajahat Ali Khan, Oresti Banos and Sungyoung Lee, "IoT based Human Centric Context Awareness Framework for Healthcare and Wellness Platform", 2016 International Symposium on Perception, Action and Cognitive Systems (PACS), Seoul, Korea, Oct 27-28, 2016
2. Wajahat Ali Khan, Muhammad Bilal Amin, Oresti Banos, Taqdir Ali, Maqbool Hussain, Muhammad Afzal, Shujaat Hussain, Jamil Hussain, Rahman Ali, Maqbool Ali, Dongwook Kang, Jaehun Bang, **Tae Ho Hur**, Bilal Ali, Muhammad Idris, Asif Razzaq, Sungyoung Lee and Byeong Ho Kang, "Mining Minds: Journey of Evolutionary Platform for Ubiquitous Wellness", 12th International Conference on Ubiquitous Healthcare (u-Healthcare 2015), Osaka, Japan, Nov 30- Dec 2, 2015
3. Oresti Banos, Jaehun Bang, **Taeho Hur**, Muhammad Hameed Siddiqi, Huynh-The Thien, Le-Ba Vui, Wajahat Ali Khan, Taqdir Ali, Claudia Villalonga, Sungyoung Lee, “Mining Human Behavior for Health Promotion”, International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2015), Milano, Italy, Aug 25-29, 2015
4. Oresti Banos, Muhammad Bilal Amin, Wajahat Ali Khan, Muhammad Afzel, Mahmood Ahmad, Maqbool Ali, Taqdir Ali, Rahman Ali, Muhammad Bilal, Manhyung Han, Jamil Hussain, Maqbool Hussain, Shujaat Hussain, **Tae Ho Hur**, Jae Hun Bang, Thien Huynh-The, Muhammad Idris, Dong Wook Kang, Sang Beom Park, Hameed Siddiqui, Le-Ba Vui, Muhammad Fahim, Asad Masood Khattak, Byeong Ho Kang and Sungyoung Lee, "An Innovative Platform for Person-Centric

Health and Wellness Support", 3rd International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO 2015), Granada, Spain, April 15-17, 2015

5. Kisung Park, Yongkoo Han, **Tae ho hur**, Young-Koo Lee, “Correlated subgraph search for multiple query graphs in graph streams”, The 9th International Conference on Ubiquitous Information Management and Communication (IMCOM ’15), BALI, Indonesia, Jan 08-10, 2015

Domestic Conference Papers

1. **Taeho Hur**, Sungyoung Lee, “Multi-model activity recognition using smart devices”, Korea Computer Congress (KCC 2017), Jeju, Korea, June 18-20, 2017
2. **Taeho Hur**, Sungyoung Lee, “Multi-Label Activity Recognition based on Inertial Sensors”, Korea Information Processing Scoiety (KIPS 2017), Jeju, Korea, April 28-29, 2017
3. **Taeho Hur**, Sungyoung Lee, “Two sensors based Hierarchical Activity Recognition for General and Sports Activity”, Korea Computer Congress (KCC 2016), Kangwon, Korea, Dec 21-23, 2016
4. **Taeho Hur**, Sungyoung Lee, “Upper/Lower Body Classficiation based Low Level Activity Recognition with smart devices”, Korea Computer Congress (KCC 2015), Jeju, Korea, June 24-26, 2015
5. Wonju Lee, **Taeho Hur**, Sungyoung Lee, “Sports Activity Recognition System using Accelerometer in Wrist Type Wearable Sensor”, Korea Computer Congress (KCC 2014), Busan, Korea, June 25-27, 2014
6. **Taeho Hur**, Ho Sung Lee, Sungyoung Lee, “Personalized Service Recommendation by Real-time Activity Recognition Revision with Prompt Method”, The 40th Korea Information Processing Scoiety (KIPS 2013), Jeju, Korea, Nov 8-9, 2013
7. **Taeho Hur**, Sungyoung Lee, Tae Choong Chung, “A Study on efficient user

emotion extraction based on MCRDR under smart phone environment”, Korea Information Processing Scoiety (KIPS 2013), Busan, Korea, May 10-11, 2013

8. **Taeho Hur**, Hyun Woo Kim, Chang Hyun Lee, Sungyoung Lee, “Context Aware based Smart home monitoring and controling using Smart phone”, Korea Computer Congress (KCC 2011), KyungJu, Korea, June 29 - July 1, 2011
9. **Taeho Hur**, Sungyoung Lee, “A Study on providing service by network sensitivity on Cloud Mobile OS based on Thin Client Systems”, The Korean Institute of Communications and Information Sciences (KICS 2011), Kangwon, Korea, Feb 23-25, 2011.
10. Lee Seongho, **Hur Taeho**, Lee Sungyoung, “Utility of Smartphone Application through using Falls Alert Application for elderly persons in u-Healthcare Service”, The Korean Institute of Communications and Information Sciences (KICS 2010), Jeju, Korea, June 23-25, 2010.
11. Lim DongJin, Kim Hyungil, An JaeYoon, Jung ChanMin, **Taeho Hur**, Lee SungHo, Han ManHyung, Ku KyoHo, Kim WooYong, Jin HyunMok, Lim YeJoon and Sungyoung Lee, “Mobile Activity Sensor Logger”, Korea Computer Congress (KCC 2009), Seoul, Korea, Nov 27-28, 2009.

A-3. Patents Registration

Domestic Patent Registration

1. **Taeho Hur**, Sungyoung Lee, Kyung Hee University, “METHOD, APPARATUS AND COMPUTER PROGRAM FOR RECOGNITION OF A USER ACTIVITY”, Registration No. 1020128040000, 08/14/2019

Domestic Patent Application

2. **Taeho Hur**, Sungyoung Lee, Kyung Hee University, “THE METHOD AND DEVICE FOR CONVERSION FROM SIGNAL OF INERTIAL SENSOR TO IMAGE”, Application No. 10-2019-0048291, 04/25/2019

Appendix: B. Korean Abstract

초 록

기존의 사용자 행위인지는 웨어러블 센서를 방향과 위치를 고정한 채 몸에 부착하는 형태로 진행되었다. 그러나 기술의 발전으로 스마트폰이 출시되었으며, 이는 다양한 종류의 센서를 지니고 있어 기존 행위인지에서 사용되던 센서를 대체하게 되었다. 스마트폰은 특성상 사람들이 위치와 방향을 고려하지 않고 가지고 다니기 때문에 기존 센서와는 달리 위치와 방향 독립성을 보장해야 한다.

한편 현대의 사람들은 평일에는 핸드폰을 소지하고 차량을 이용하여 통근하는 등의 비슷한 생활양식을 보이고 있다. 이에 따라 통근 활동 인지를 위한 최적의 환경이 구성되었다. 통근 활동은 통근 중에 일어나는 행위로써 서기, 앉기, 걷기, 조깅 및 자동차, 버스, 지하철, 기차, 오토바이, 자전거 등을 탑승하는 행위를 일컫는다. 대부분의 사람들이 스마트폰을 소지하고 다니기 때문에, 스마트폰의 센서로부터 가속도, 각속도, 속도, 좌표, 소리, 영상 등의 데이터를 습득할 수 있다. 이러한 데이터로부터 특징을 추출하고 분류하여 사용자의 행위가 무엇인지 파악 가능하다.

통근 활동을 자동으로 인지할 경우의 장점으로 1) 이동 수단 이용에 대한 통계 데이터를 자동으로 획득 가능하여 설문 등의 수동 방법에 비해 시간과 비용 절약이 가능하고, 2) 사용 중인 이동 수단에 따라 사용자 맞춤형 서비스나 광고가 즉시에 전달 가능하며, 3) 건강 상태, 안전 문제, 소모 칼로리량 등이 모니터링 가능하여 건강 관련 서비스가 가능하고, 4) 온실 가스 탄소 발자국 예측 등이 있다.

통근 활동 중에서는 차량 인지가 가장 큰 문제를 야기한다. 이는 차량 안에서는 사람이 서 있거나 앉아 있는 등 움직임을 보이지 않으며, 이는 정적 행위이기 때문이다. 이에 따라 시계열 센서 신호에서는 수치에 큰 변화가 나타나지 않으며 특정한 패턴을 가지지도 않는다. 이러한 문제를 해결하기 위해 몇몇 연구에서는 주파수 신호에서 특징을 추출하였다. 주파수는 단위 시간 당 반복적인 이벤트 발생 횟수를

나타내는 단위이다. 주로 추출되는 특징으로는 통계적 특징인 평균이나 표준 편차, 그리고 최대 진폭을 보이는 주파수 등이 있다. 그러나 통계적 특징의 경우 주파수 자체가 서로 다른 차원을 나타내기 때문에 이에 대한 통계를 내어서는 안되며, 최대 진폭 주파수의 경우 고정되지 않고 변화하는 문제점을 가지고 있다. 또다른 고려 사항으로는 실제 환경에서 발생하는 예측되지 못한 변수에 대하여 기계학습 기법에서는 오류가 발생한다는 사실이다. 이러한 오류를 보정하기 위해서 기계학습 기반 분류 이후에 휴리스틱 기반 후처리 방법이 요구된다.

따라서 본 논문에서는 이를 해결하기 위한 두 가지 방안을 제안한다. 첫 번째는 소지위치 문제와 차량 분류 문제를 위한 차량의 고유진동 특징을 이용하는 것이다. 모든 물체는 고유진동을 가지고 있으며, 이는 차량도 마찬가지로써 소지위치에 영향을 끼치지 않는다. 두 번째는 행위 보정으로 현실 세계를 반영하여 분류 결과는 조정하는 것이다. 여기서는 GPS를 이용하여 지하철과 지상 차량을 분류하는 것, 상호 변환 규칙의 사용, 그리고 현재 행위, 이전 행위, 이전 보통 행위를 이용한 행위 보정 방법이 사용된다. 실험 결과에 따라 제안하는 방법이 기존의 통근 활동 인지 방법보다 높은 결과를 나타냄을 입증하였다.