Published in AJSE, Vol:21, Issue: 2
Received on 12th January 2022
Revised on 14th October 2022
Accepted on 23rd November 2022

AIUB JOURNAL OF SCIENCE AND ENGINEERING
ISSN: 1608 – 3679 (print) 2520 – 4890 (Online)

# An Approach to Recognize Vehicles Context Flow for Smartphone-Based Outdoor Parking Using Supervised Machine Learning Classifiers

Md. Ismail Hossen, Michael Goh, Tee Connie, and Md. Nazmul Hossain

***Abstract*— Finding an available parking space in outdoor environments such as university campuses and roadsides need a good parking system. In a general parking system, detecting a vehicle entering leaving the parking premise is one of the major steps. Currently, there are parking systems that use cameras or external sensors to detect the leaving and entering of the automobiles. Both parking systems need very high cost of deployment and maintenance. To encounter the issues, this paper presents a parking system for outdoor parking systems using only smartphone-oriented sensors. The proposed approach does not require additional sensors installation nor manpower support. It takes the inputs from smartphones to detect the driver's context that is used to recognize the flow of the vehicle. Context flow recognition indicates whether a driver is parking or unparking his/her vehicle. Supervised classifiers like support vector machines (SMV) and decision trees (DT) are used to recognize the parking or unparking actions to enable vehicles tracking in the parking area. Outcome of the proposed approach is a significant contribution for outdoor parking as it solely utilizes the sensors smartphones embedded to detect parking behaviors.**

***Index Terms***—Sensors, Smart Parking, Machine Learning,

## I. INTRODUCTION

A smart parking system is a technology which helps to locate an available parking spot and broadcast the information to the drivers through a central server over the Internet. The conventional process of searching a available parking space is inefficient as it increases fuel consumption, driving time and traffic congestion which needs some deployment of parking system [1]. Automatic vacant parking spot search is crucial for the betterment of smart transportation. In most of the large cities around the world, parking problem is one of the major daily problems. A study [2] has revealed that vehicles looking for a parking spot produce 30% of the daily traffic. Another study [3] shows that jam-packed streets in US cost $78 billion annually on gas and time wastes. Another study [4] highlights that 45% of total jam is produced by vehicles rotating the area searching for a vacant park spot. In addition, a study [5] also shows that in the US around 42% of the people cannot reach their destination or meet appointments because they cannot find a parking space, and nearly 34% of the New York drivers cancel their trips due to parking frustration.

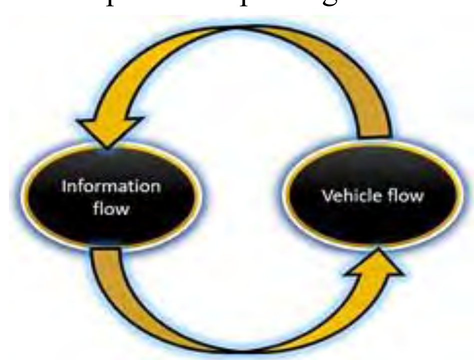


**Figure 1** A typical parking system.

Although numerous parking solutions have been proposed, most of them are not appropriate for outdoor parking areas due to their high installation and maintenance cost. Typically, a smart parking system consists of two main flows (as shown in Figure 1), namely information flow and vehicle

Md. Ismail Hossen, Lecturer, Dept. of Computer Science, AIUB,
email: ismail.hossen@aiub.edu
Michael Goh, Associate Professor, FIST, MMU, Malaysia
Tee Connie, Dean, Institute for Postgraduate Studies, MMU, Malaysia
Md. Nazmul Hossain, Lecturer, Dept. of Computer Science, AIUB,

flow. Firstly, the vehicle flow is responsible for detecting the parking action of the driver. The vehicle flow starts when the drivers look for a parking slot, get a parking slot and park his / her car. On the other hand, the information flow collects information about the driver's context, total parking slot, available parking slots and other relevant information. Information flow disseminates the information to the other drivers who are requesting the information. In a parking system, vehicle and information flows are equally important.

The focus of this study is on the first step which is detecting the entrance and leaving status of the vehicles. To detect a vehicle entering or leaving a parking spot, the existing systems use a camera-based solution [6]. The works by [7], [8] indicate that parking can be detected by external sensors under each parking spot, tightening sensors with the sides of the vehicle and setting sophisticated cameras. However, covering a large parking area with such sensors or cameras is impractical and almost impossible to implement.

For instance, a city of France implemented a parking system using ten thousand external parking sensors that cost 15 million euros for deployment [8]. Purchasing these huge numbers of sensors makes it expensive to cover a larger area. Another well know example is Los Angeles which deployed a parking system with external sensors. To deploy such system in one or more locations, various issues need to be considered such as soil testing, people's behaviors, suitability of sensors installation and so on [8]. Besides a huge amount of initial investment is also required.

Therefore, the proposed method that relies on smartphone's sensor represents a cost-effective parking solution as compared to external sensors or camera-based systems. To detect whether a vehicle is parking or leaving the area, only signals obtained from the smartphone embedded sensors are used. Neither external sensors nor additional cameras are required. To detect whether a vehicle is parking or leaving, the proposed system has two main components which are driver's context recognition and recognizing the flow of the context. To recognize the driver's context, supervised machine learning classifiers are utilized.

The rest of the paper is organized as follows: Section 2 illustrates the existing parking solutions, Section 3 presents the theoretical knowledge of proposed system, section 4 demonstrates the proposed methodology, Section 5 illustrates experiment results, and lastly Section 6 provides a conclusion.

## II. LITERATURE REVIEW

Various advancements have been made for transportation system, especially in smart parking [9]. In the year 1905, the first automated parking system was invented which was also known as a semi-automated parking system [10]. From 1940s to 1960s, interests in parking system had grown in the United States [9]. In [11] a cloud-based and deep learning-based mobile smart parking system was proposed to reduce the problem of getting a parking place by developing a service based on deep learning with long short-term memory to predict parking slot. The recent studies about parking system will be discussed in the subsequent sections.

### *A. Vision-Based Parking Systems*

The continuous improvement in image processing and computer vision techniques have enabled -based parking systems [12]. Analyzing the video footages captured by CCTV to identify unallocated parking spaces is the main technique for parking systems that use image processing and computer vision techniques [12]. A deep convolutional neural network was presented in [13] to analyze the images inside the parking area. In a method introduced by [14], anyone with no technical knowledge can use and setup a parking system. A camera with wide-angle was deployed to monitor the whole parking space that cover a wide area.

### *B. Sensor-Based Parking Systems*

In [10], sensors were utilized to discover and track unoccupied parking spaces where the input from the AVM (Around View Monitor) and ultrasonic sensors were fused. Three stages namely making of the parking lot, occupancy classification of parking lot, and tracking parking slot had been proposed for the system. Another study by [13] identified free parking areas between the scanning laser radar and the vehicles. The proposed system contained corner detection, data processing and parking position recognition. Additionally, [15] suggested a solution for outdoor parking system to track the entrance and exit of cars. A similar type of procedure for communication was proposed by [16] where infrared sensors and radio frequency were utilized for locating parking space.

### C. Parking Systems Based on Smartphone

In the last couple of years, rapid development of hardware and sensors has made a great progress in mobile phone technology. A procedure proposed in [9] developed a Bayesian probabilistic framework called Probabilistic internal navigation (ProbIN) to regain genuine motion from noisy sensor readings. The purpose of the study was to obtain the user location for storing minimal-cost information. In [10], a phone-based driver tracking system was introduced to track the path of the driver. The idea was to provide a consciousness about when the driver was leaving the parking slot. Internal positioning system (IPS) was the main pillar for the system. It was used to track the users' movement within remote areas. IPS was integrated with different types of sensors in the smartphone like GPS, accelerometers, digital compasses, and gyroscopes to track whether the parking slot was allocated or unallocated. To obtain an accurate estimation result, a map-matching and waist-mounted unit was used. Here a different method was suggested where a map was used to display a probable allocation of available parking slots with the users' contextual details as input. Bluetooth and GPS were used to collect the users' information like position, path, and user's activities. At constant intervals, coordinates were collected from the user to estimate the user's path and position. A path contained several coordinates, latitude and longitude that are given in Equation (2.1):

$$\mathrm{path}_z = [(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_n, y_n)] \quad (2.1)$$

In $\mathrm{path}_z$ $x_n$ is the n-th latitude and $y_n$ is the n-th longitude. Reference [17] suggested to use two arrays for collecting longitude and latitude information in a constant time interval of the user movements:

$$lat_z = [(x_1, x_2, x_3, ..., x_n] \quad (2.2)$$

$$lon_z = [(y_1, y_2, y_3, ..., y_n] \quad (2.3)$$

In a system called Park Sense presented in [7], the parking action of the driver was tracked whether he was moving from the car or coming back to the car. Wi-Fi beacons was leveraged for tracking the parking actions. The set of Wi-Fi signatures are shown in Equations (2.4) and (2.5):

$$S_p = \{S_p(1), S_p(2), S_p(3), ...S_p(n), \} \quad (2.4)$$

$$W_p = \{W_p(1), W_p(2), W_p(3), ...W_p(n), \} \quad (2.5)$$

$S_p$ (i) represents the service set identifier (SSID) while $W_p$ (i) represents the ratio of the beacon at access point i. On the other hand, n indicates the total access points number using all the scans. Drivers' movement identification was performed by Pocket Parker in [12] for mobile phone's GPS and accelerometer. Centralized database was used in the system where the information was stored after collecting data and the data was broadcast using Wi-Fi and cellular network to the drivers. The existing methods for activity recognition was used to track the movement of the drivers like arrivals and departure information's.

### D. Summary of Existing Works

From the discussions about the different types of parking systems, the vision-based approach is more flexible, and the maintenance cost is low. Additionally, the configuration is not complex. The available parking spots can be identified by inspecting images of the parking areas. Nevertheless, the vision-based tracking system is not free from drawbacks. Sometimes, it provides inaccurate result due to lighting and obstruction problems. The camera should be installed at the right place so that the whole area can be observed from the location. Besides, additional costing is needed to setup camera with high resolution [18]. Vision-based parking becomes more complex for impulsive shadow, changes of illumination and sometimes for miscellany of ground materials.

On the other hand, sensor-based parking system requires external sensors. The configuration is more complex, and it is sometimes difficult to arrange the sensors in a practical scenario. Occasionally, it requires the whole infrastructure to be rebuilt in the worst-case scenario. Contrary, the smartphone-based system requires very low installation and maintenance cost. This technique does not require hardware installation and preservation. This motivates us to study and develop a smart sensor-based parking system in this paper.

## III. LOGICAL IDEA BEHIND PROPOSED METHOD

Normally, there are four main modules in a smartphone-based parking system, namely parking zone, driver's device, connecting server and driver. The main responsibility for the server is to collect driver's information like speed of the vehicle, location, destination through automatically or manually through devices like smartphone or tablets in the time of driving. Based on the information collected, an accessible map is created in real time by the server. In a particular destination, the driver sends a request to the server for the real time map to know the available parking slots and the availability depends on the current position of the driver. Then the client device receives the search result returned by the server that there might be an available parking spot nearby. Figure 2 presents the overall architecture of a smartphone-based parking system along with a reference model.


**Figure 3.1** Typical smartphone-based parking system.

In this study, the main objective is to automatically detect if a vehicle is parking or unparking. It is conceivable to distinguish if a driver is parking or leaving a parking slot in a particular location which is very crucial for the system to count the number of available parking slots. With this information, the availability of the parking slots can be updated in real time and be presented in a real time map.

A driver needs to follow some eolithic actions to park or leave from the parking zone. As an example, the driver needs to firstly drive the vehicle to the parking zone. Then he/she needs to walk from the current location to leave the parking area. This implies an activity flow like driving walking. On the other hand, the reverse flow walking driving takes place when the driver leaves the parking area. In Figure 3.2a and 3.2b indicate the forward flow for car parking and the reverse flow for leaving the parking area. Therefore, it is easy to determine if a driver is parking or leaving the parking area by acknowledging the driver's activity context.

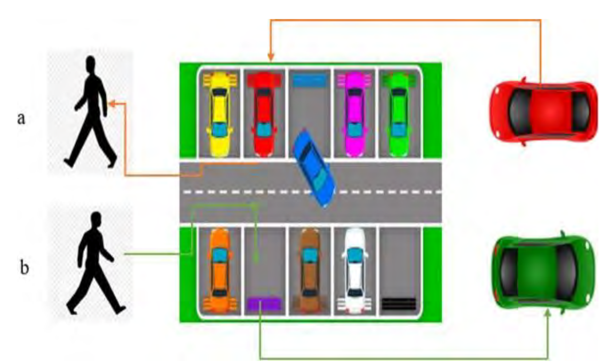


**Figure 3.2** (a) and (b) demonstrate the parking

and unparking actions of the proposed method.

Figure 3.3 depicts the flow that happens during parking and unparking through activity theory diagram. Figure 3.3a represents the flow that happens during the unparking event when a car leaves from the parking area. As opposed to Figure 3.3a, Figure 3.3b indicates the flows that happen when the driver parks the car in the parking area.

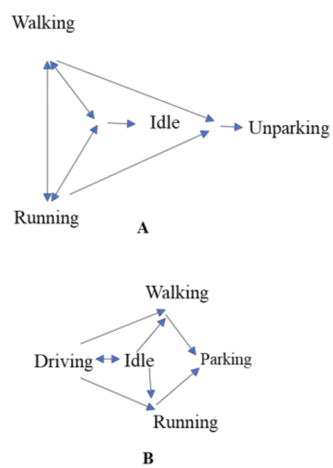


**Figure 3.3** Parking / Unparking possible actions flow

Figure 3.3 provides a clear overview about the unparking action activity that starts with any action except driving and ends with a driving action. On the contrary, the parking event starts with the driving action and ends with a walking or running action. To track the parking or unparking activity, it is crucial to detect the complete route of the driver's context. Here a complete route for a parking action indicates the flow that begins from the driving state and ends with the walking or running state. The same scenario happens for the unparking state where the operation begins a with walking, running or idle state and ends with the driving state. Throughout the paper, "parking" means allocating a slot in the parking area while "unparking" indicates unallocated parking space by exiting the parking area.

In Figure 3.3 the flow drawn indicates that a driver is parking or leaving the parking area. The "driving" action indicates an activity when a diver enters or exits the parking area. Sometime after parking, a driver may be slothful for some duration then the change to the activity is "idle" state. After moving to the parking zone, the driver parks and walks to the destination and the state is changed to "walking". Therefore, the flow for parking a vehicle can be shown as driving → idle → walking flow. Similar type of actions in the reverse direction is walking → idle → driving flow and this happens when the driver is leaving the parking zone after parking the vehicle.

Now the crucial part of the research is to distinguish the state if the driver is driving or walking or in idle state. According to the concept of velocity the diving and walking state can be distinguished effortlessly. Reference [19] indicates the walking velocity scales from 1 ms1 to 1.18 ms1 and the running velocity scales from 2.97 ms1 to 3.81 ms1. For the calculation of velocity Equation (3.1) can be used:

$$v = d / t \tag{3.1}$$

Here, v represents velocity, d represents distance in meters and t represents time in seconds.

## IV. PROPOSED METHOD

There are two main components in the proposed method which are context recognition (CR) and Context flow recognition (CFR) (Figure 4.1).


**Figure 4.1** Block diagram of proposed system.

Firstly, CR is the very first and the most important component of the proposed system. CR is responsible for detecting the context of the drivers. CR determines whether the driver is in driving, walking or idle mode. Secondly, CFR is responsible to track the flows of CR. It recognizes the context change and the flow of changes. It tells whether the vehicle is parked or unparked. Each of the components is consists of sub-components as shown in Table 4.1.

**Table 4.1** Major components and sub-components of the proposed system

| Components | Sub-components |
|---|---|
| CR | Pre-processing, context prediction and prediction regularization |
| CFR | Keep tracking of context change, detection of parking / unparking |

### A. Pre-processing

After the raw data have been collected, pre-processing is applied to treat the missing data, and perform feature translation, feature construction and gravity force elimination.

### B. Missing data

Missing data is a common problem in real-life situations that must be dealt with carefully. One needs to identify which types of data are missing, how much data are missing, and reasons for the missing data. In the raw data, it is observed that some of the missing data come as 0 and null as shown in Figure 4.2 (the first table at the top) due to missing data during the start of the application. Every time the application is started, some data will be missed at the beginning. Therefore, rows with missing values were removed from the dataset. The result of missing data removal is shown in the second table at the bottom in Table 4.2.

**Table 4.2** Data Missing (Upper) and After Dealing with Missing Data (Lower)

| Initial data | | | | | | |
|---|---|---|---|---|---|---|
| aX | aY | aZ | gX | gY | gZ | Activity |
| 0 | 0 | 0 | 0 | 0 | 0 | still |
| 0.241 | 0.618 | 2.651 | 0.151 | 0.1545 | 0.154 | still |
| 0.502 | 0.744 | 0.282 | 0.274 | 0.22052 | 0.35491 | still |
| Data after dealing missing data | | | | | | |
| 0.241 | 0.618 | 2.651 | 0.151 | 0.1545 | 0.154 | still |
| 0.502 | 0.744 | 0.282 | 0.274 | 0.22052 | 0.35491 | still |

### C. Feature Translation

The machine learning classifiers used in this study require converting the non-numeric values to integer/floating-point numbers. The output column of the dataset consists of non-numeric values representing the Walking, Idle, Running, and Driving activities. For computation purposes, these values are converted to 1, 2, 3, and 4 to signify Walking, Idle, Running and Driving, respectively. The values 1, 2, 3 and 4 are used as the class labels for the different activities. The conversation of the non-numeric values is shown in Table 4.3.

**Table 4.3** Feature Translation

| Data Before Feature Translation | | | | | | |
|---|---|---|---|---|---|---|
| aX | aY | aZ | gX | gY | gZ | Category |
| 1.518 | 0.302 | 0.116 | -0.01 | 0.455 | 0.397 | Still |
| 0.688 | 0.350 | 1.390 | -0.27 | -0.311 | 0.166 | Walking |
| 2.230 | 0.812 | 2.296 | 0.065 | 0.340 | 0.003 | Running |
| 0.273 | 0.979 | 0793 | -0.11 | -0.265 | 0.021 | Driving |
| Data After Feature Translation | | | | | | |
| 0.241 | 0.618 | 2.651 | 0.151 | 0.1545 | 0.154 | still |
| 0.502 | 0.744 | 0.282 | 0.274 | 0.22052 | 0.35491 | still |

### D. Scaling

Scaling is a specific pre-processing step required by the SVM classifier. Data scaling is needed to avoid greater numeric ranged of attributes [13] to solve some numerical complications during calculation. In this study, the input data is scaled into [-1, +1] to standardize the inputs for SVM.

$$scaled_{score} = \frac{x}{Max\,(abs\,(x))} \quad (4.1)$$

where, x is the input value and $scaled_{score}$ represents a matrix value within [-1, +1].

*E. Context Prediction (CP)*

Supervised ML classifiers SVM, DT are used. In this paper SVM is a popular machine classifier. SVM constructs several hyperplanes to separate different classes. The better classification depends on choice of the optimal hyperplane. A hyperplane is considered optimal if it can separate classes without errors and if the distance among closest sample and hyperplane is utmost. The hyperplane of SVM is described in the following equation:

$$W^T x + b = 0, xR^d \tag{4.2}$$

where w is the vector, b is a scalar, the samples that are close to hyperplane boundaries are known as support vectors. Decision tree is another effective machine learning algorithm used for classification and prediction. The fundamental stage of decision trees algorithm is applying a selection measure to discover the correct test attribute at each node of decision while expanding the tree. Quinlan construed a measure named information gain also known as criterion to calculate the information gain for each attribute to examine how well each of the training attributes and selects the attributes with best performances. Indeed, this attribute provokes segregation where the classes of instances are as similar as possible inside each of the subsets conceived by the attribute. Let A be an attribute that has K outputs which partitions a training set T into k subset Tm where m = 1..... K. Assume there are n classes that are denoted by C1, C2, C3, . . . Cn. Then, proportion of objects pi is calculated as:

$$P_i = \frac{fre\ (Cit)}{|T|} \tag{4.3}$$

where T belongs to $C_i$ (I = 1.N) classes.

In context prediction, the dataset is divided into three sub- datasets which are train, test, and validation set. The train set is used for training models and the validation and test sets are kept unseen to the models. After fitting models with train set validation set is to do parameter tuning and lastly test set is used to measure the performance of the trained models.

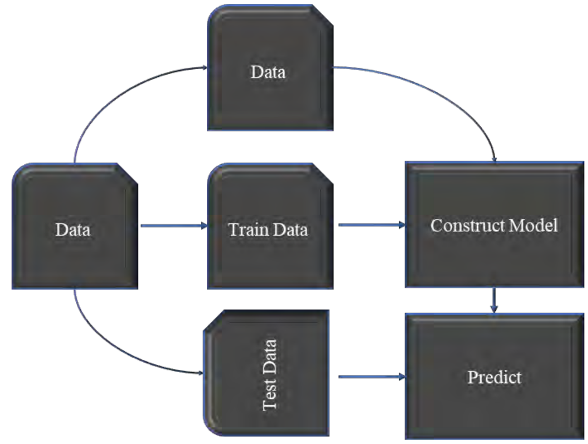


**Figure 4.4** Processes of context prediction

The performance of the driver's CP with respect to each of the classifier is evaluated on both validation and test dataset. Furthermore, k-fold cross-validation (KCV) is performed to validate the performance of CP [14]. Various splitting strategies lead to several CV estimations such as hold out, leave one out, k-fold CV and so on. In this research, k-fold validation is used as it is one of the commonly used validation methods.

*F. Process of Prediction Regularization*

Prediction regularization (PR) improves the results of CP. PR is very important for the next process which is CFR. To determine whether the driver is parking or unparking, it needs to know a complete flow of the driver's activities. By knowing only, the partial activities, one cannot conclude whether the driver is parking or leaving. CP might detect an activity wrongly. Any single classification mistake made by CP could lead to a wrong transition flow. Therefore, the misclassifications need to be corrected by PR. PR consists of two steps known as partitioning rolling window (w), and output regularization for each partition.

*a) Partitioning rolling window*

In this step, the total predicted outputs (*o*) are divided into number of windows (*w*) with *m* consecutive samples. Since four possible outputs might occur, a mode (*o*) needs to be selected with at least a double of classes. Thus, the selection of size $m$ is taken into consideration with the following equation.

$$m = n * 2 + 1 \tag{4.4}$$

where $m$ stands for output sample size and n stands for possible CP output.

*(b) Predicted Output Regularization*

The mode of each rolling window is calculated with the following equation,

$$Mode(o) = max_{repetation}(walking, idle, running, driving) \tag{4.5}$$

### G. *Process of Context Flow Recognition*

Context flow recognition (CFR) is an important step to recognize the parking and unparking status. This step compares the current and previous activities achieved by mode (o). If both the previous and current contexts are the same, it does not update anything. Otherwise, it keeps track of the context changes and update the current activity. CFR is responsible for concluding the parking or unparking actions. For instance, if the CFR pattern is walking > idle > driving then based on the nature of parking / unparking actions it concludes it is a parking unparking action and increases the total parking availability by 1 (refer Figure 4.5).

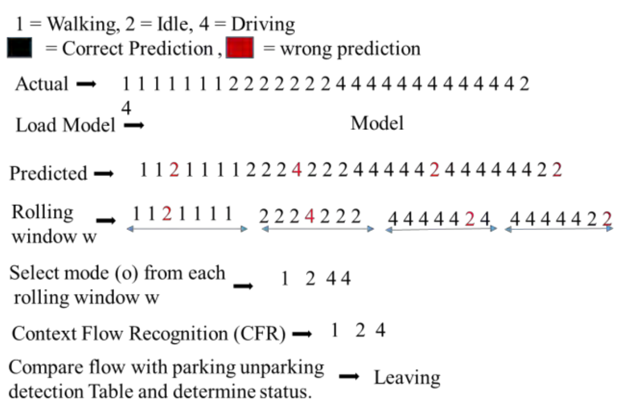


**Figure 4.5** Sample of Parking / Unparking (Leaving) Detection with Demo Data

From the example provided in Figure 4.5, the actual values that correspond to each feature vector are given to the model for prediction. It is observed that all the predicted values are not the same as the actual values due to classification error (colored as red). To avoid classification errors, the CP outputs are divided into rolling window ($w$) and takes mode($o$) as the output and the flow of activities is detected by CFR. It is important to know that a complete transitions sequence is mandatory to decide a parking / unparking activity. Partial detection is not sufficient for concluding parking or unparking. The pseudocode of the proposed algorithm is shown as follow.

PSEUDOCODE: PARKING EVENT (PARKING/UNPARKING) DETECTION ALGORITHM

- Step 1: Receive input from driver's phone
- Step 2: Pre-process input signal
- Step 3: Construct meaningful matures
- Step 4: Detect driver's context using supervised machine learning classifiers
- Step 5: Improve context recognition using context regularization
- Step 6: Context flow recognition
- Step 7: Detect Parking/unparking actions

### H. *Research Assumptions*

Generally, the aim of parking system is to notify the drivers about parking occupancy in the nearby parking area as soon as possible. In case of smartphone sensor-based parking, the system collects data from the user's smartphone and uses the data to navigate the drivers to suitable parking lots. There are three main modules in a smartphone-based outdoor parking system which are drivers, driver's devices, and a central server. For the convenience of this research, some assumptions are made to explain how the components interact in an outdoor parking system. The assumptions are explained below.

*a) Central Servers*

Central servers collect information about the driver's destination, current location, and vehicle speed while looking for a parking place. The server maintains and updates the information about different parking lots. The server searches for a right parking lot when the driver is close to his destination and inform driver through drivers' device about the available parking spot. Besides, it is assumed that the server has the right to access the parking database. The parking database is used to store parking relevant information such as total capacity of each parking zone, current occupancy, available parking slots, pricing, legal period of parking, duration of parking, and so on.

### *(b) Drivers and Devices*

The drivers communicate with the server through their devices that are responsible for sending coordinates (latitude and longitude) of location, query about availability of parking lots. It is reasonable to assume that the driver's devices have GPS and internet connection since most of the devices such as smartphones, tablet PCs versatile GPS nowadays can play the navigation role. The recent devices have the capability to automatically send the information to the server about their movements without human intervention. The drivers receive a recommended parking lot upon sending their destination to the server.

### *(c) Parking Zones*

It is also assumed that the area of the city is segmented into a grid with a unique identifier i. It assumes that the total capacity of parking in each grid is Ni. The overall information about parking in a grid i is denoted as:

$$N_{it} = N_{io} + N_{ia} \tag{4.6}$$

where, $N_{it}$ is the total parking spaces for parking zone it, $N_{io}$ is the occupied space and $N_{ia}$ is the available space for that parking zone at certain time t. The total number of parking, available parking, and occupied parking are updated and stored to database whenever a driver comes in or goes out. The server broadcasts this information to drivers who are within a range R from that cell of parking zone i. This is motivated by the fact that most drivers start searching for parking when they are adjacent to their destination. Therefore, the drivers might only be interested in parking spaces that are close to their destination.

## V. RESULTS AND DISCUSSIONS

To justify the performance of the proposed approach, sensors input is received from six participants of various ages who had completed 4 parking relevant actions such as walking, running, driving and idle by holding their smartphones in different positions and orientations. Some of the positions are phone in right pocket, left pocket, left hand right hand, handbag, chest bag, back bag, phone on their car. The participants were free to keep the phone as their wished. No restriction was placed on the orientation or position of the phone. The trained models were given different numbers of train / test data to experiment CR. The dataset is divided into three ratios which are 70:30, 80:20 and 90:10. The data in each ratio is given in Table 5.1. The accuracy of each classifier with respect to different train: test ratio is shown in Figure 5.1.

**Table 5.1** Number of training and test samples for different train / test split ratio

| Train test split ratio | number of training samples | number of testing samples |
|---|---|---|
| 70:30 | 23,417 | 10,036 |
| 80:20 | 26,762 | 6691 |
| 90:30 | 30,107 | 3346 |

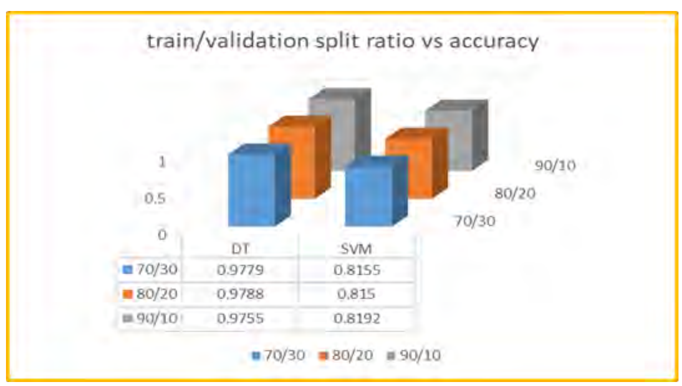


**Figure 5.1:** Accuracy of models with respect to different train: test split ratio

Furthermore, to justify a model on unseen data, KCV is used. The value of K is taken 10 a many of the of machine researchers suggests taking k = 10. The outcome of KCV shows a identical achievement. The accuracy of KCV of this research is shown in Figure 5.

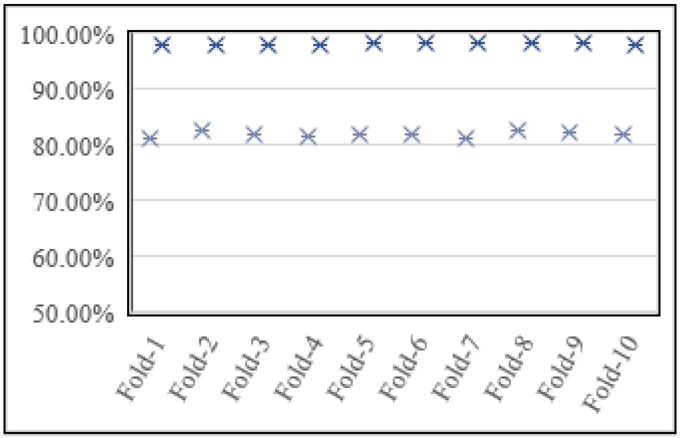


**Figure 5.2** Results of KCV where k = 10

From Figure 5.2, it can be observed that all folds get closely alike accuracy score. Hence, all the folds of KCV are mutually well aligned with unseen data. Furthermore, to justify the effectiveness of the models, precision, f1-score, and recall SVM and DT have been calculated. The calculations are based on TP, FP, TN and FN where TP, FP, TN, and FN stand for true positive, false positive, true negative and false negative respectively. The equations for these measurements are given as, stands for true positive, false positive, true negative and false negative respectively. The equations for these measurements are given as,

$$TP = |s_k | y(s_k) = +1, y_e(s_k = +1)| \quad (5.1)$$

$$TN = |s_k | y(s_k) = -1, y_e(s_k = -1)| \quad (5.2)$$

$$FP = |s_k | y(s_k) = -1, y_e(s_k = +1)| \quad (5.3)$$

$$FN = |s_k | y(s_k) = +1, y_e(s_k = -1)| \quad (5.4)$$

$$Precision = TP/((TP + FP)) \quad (5.5)$$

$$F1{-}Scor = (2 * recall * precision)/((recall + precision)) \quad (5.6)$$

$$Recall = TP/((TP + FN)) \quad (5.7)$$

Here $s_k$ represents testing sample y($s_k$) real label of testing sample of $s_k$, $y_e$ ($s_k$) the denoted prediction level of $s_k$. -1 and +1 represents the negative and positive labels for the sample respectively. Most of the performance measurement of machine learning depends on these preliminary (TP, TN, FP, FN) measures. The results of precision, f1-score and recall are shown in figure 5.3.

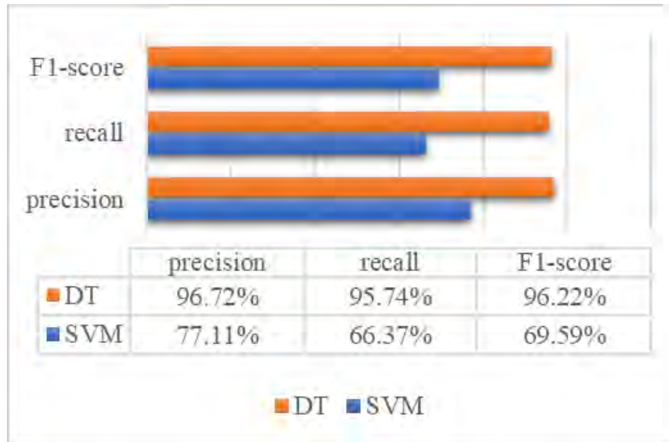


**Figure 5.3** Precision, recall and f1-score

*A. RESULT OF CONTEXT FLOW RECOGNITION*

Unlike context recognition, one hundred and twenty unparking / parking actions data have been experimented for CFR. Among 120 experiments, 60 were performed in a controlled environment where proper instruction to place the phone at four fixed positions. On the other hand, 60 actions data were collected with random position. The users were free to keep the phone in any places such as right hand, left hand, inside the bag and so on. There are three main types of parking / unparking actions which are parking, unparking and unknown. If CFR matches with any of the parking flows, the status of that car is considered parking. If CFR matches with the unparking list, then it is labeled unparking or leaving parking space. If none of these matches, an unknown status is given to that vehicle. Table 5.2 shows the result of CFR.

**Table 5.2** Experiment result of DT and SVM respect to different positions of the phone

| Smartphone placement location | Num of Correctly identified exp | |
|---|---|---|
| | DT | SVM |

| | | |
|---|---|---|
| Number of experiments with Radom location | 48, 57 | 54 |
| Number of experiments with Fixed location | 48 | 55 |
| Number of overall experiment | 105 | 109 |
| Overall accuracy | 87.5% | 90.83% |

Table 5.2 shows that the proposed algorithms can detect whether a vehicle is parking or unparking regardless of the phones' position. For example, SVM can correctly detect 55/60 for fixed positions and 54/60 for random position. Such a result shows the satisfactory level of accuracy regardless how the phone is hold by the drivers. The overall accuracies of CFR for SVM and DT are 90.83% and 87.5%, respectively.

Furthermore, it is important to check how the proposed method behaves with fault or wrong inputs. To check the result of CFR, 10 wrong actions that are not relevant to parking or unparking are taken and fed to the model. Instead of taking data from car, the data is taken from a bicycle to check what does the model think about this. As cycling data is not relevant to a parking action, this type of data was not used for training the model. Naturally the models were expected to classify them as unknown. The result of the models against wrong inputs are shown in the Table 5.3.

**Table 5.3** Result of parking / unparking detection with wrong input data

| No. Of experiments | DT | SVM |
|---|---|---|
| | 10 | 10 |
| Unknown | 9 | 9 |
| Wrongly detected | 1 | 1 |

Table 5.3 shows that both SVM and DT accurately detect the action as unknow 90% of the time. It clearly confirms that the proposed is clever enough to deal with wrong inputs. As expected, all the10 experiments unknown to the algorithm confirm that the system is smart enough to differentiate wrong inputs.

Lastly, the overall performance of all the classifiers for CFR is shown in Figure 5.4. NPCD stands for the number of percentages for correct detection and NPWD stands for the number of percentages of wrong detection.

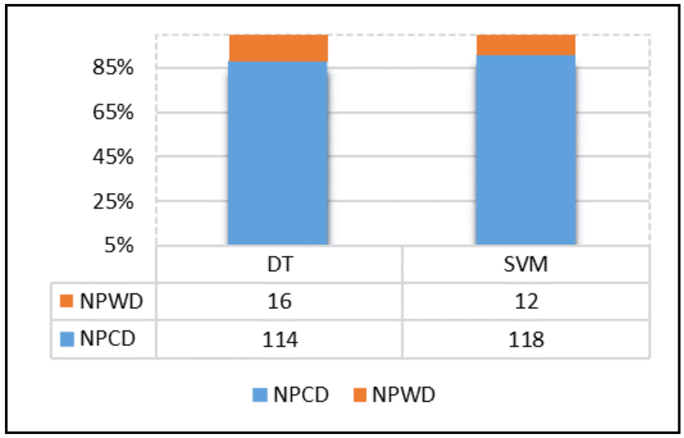


**Figure 5.4** Experiment of parking / unpacking actions

NPCD or NPWD = 130 (120 experiment with correct data + 10 wrong data) / 100. The experiment result indicates that both SVM and DT provides satisfactory outcomes. Figure 5.4 indicates that the overall performance of DT and SVM is 88% and 91% respectively

## VI. CONCLUSION

Detection of vehicles entering and leaving is a very essential stage towards tracking available parking space in a parking system. The existing solutions use either expensive sensors like ultrasonic, RFID or cameras which need a huge implementation and maintenance cost. Due to the high initial and maintenance cost, it is very difficult to be implemented in large-scale. This research proposes a system to substitute external sensors and cameras to detect relevant actions by utilizing the blessings of smartphone's-oriented sensors. The aim of the proposed method is to detect whether the vehicles are being parked or have left the parking zone. Supervised machine learning classifiers like SVM and DT have been used in the proposed system. The proposed method can correctly predict parking or unparking actions of a vehicle using only smartphone embedded accelerometers and gyroscope sensors. Future works will be devoted to broadcasting the parking availability information such as how many parking slots are occupied; how many are available and how many slots could be occupied or free by a certain time. Public drivers will greatly benefit from such information dissemination system.

## REFERENCES


[1] H. Canli and S. Toklu, "Deep Learning-Based Mobile Application Design for Smart Parking," vol. 9, pp. 61171-61183, 2021.

[2] L. Stenneth, O. Wolfson, B. Xu, and P. S. Yu, "PhonePark: Street Parking Using Mobile Phones," in *2012 IEEE 13th International Conference on Mobile Data Management*, Jul. 2012, pp. 278–279.

[3] S. Mathur, S. Kaul, M. Gruteser, and W. Trappe, *ParkNet: A Mobile Sensor Network for Harvesting Real Time Vehicular Parking Information*. 2009.

[4] Paul Steely White, "No Vacancy Park Slope's Parking Problem And How to Fix It," 2007.

[5] Kevin McCoy, "Drivers spend an average of 17 hours a year searching for parking spots," 2017. .

[6] M. I. Hossen, M. Goh, T. Connie, A. Aris, and W. L. Pei, *A Review on Outdoor Parking Systems Using Feasibility of Mobile Sensors*, vol. 488. 2018.

[7] S. Nawaz, C. Efstratiou, and C. Mascolo, "Parksense: A smartphone based sensing system for on-street parking," *Proc. Annu. Int. Conf. Mob. Comput. Networking, MOBICOM*, pp. 75–86, 2013.

[8] T. Lin, H. Rivano, and F. Le Mouel, "A Survey of Smart Parking Solutions," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–25, 2017.

[9] M. Fraifer and M. Fernström, "Investigation of Smart Parking Systems and their technologies," 2016.

[10] J. K. Suhr and H. G. Jung, "Sensor fusion-based vacant parking slot detection and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 21–36, Feb. 2014.

[11] Mikel Celaya-Echarri; Iván Froiz-Míguez; Leyre Azpilicueta; Paula Fraga-Lamas; Peio Lopez-It, "Building Decentralized Fog Computing-Based Smart Parking Systems: From Deterministic Propagation Modeling to Practical Deployment," 2020.

[12] M. Kłosowski, M. Wójcikowski, and A. Czyzewski, "Vision-based parking lot occupancy evaluation system using 2D separable discrete wavelet transform," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 63, no. 3, pp. 569–573, Sep. 2015.

[13] L. Zhang, J. Huang, X. Li, and L. Xiong, "Vision-Based Parking-Slot Detection: A DCNN-Based Approach and a Large-Scale Benchmark Dataset," *IEEE Trans. IMAGE Process.*, vol. 27, no. 11, 2018.

[14] S. E. Shih and W. H. Tsai, "A convenient vision-based system for automatic detection of parking spaces in indoor parking lots using wide-angle cameras," *IEEE Trans. Veh. Technol.*, vol. 63, no. 6, pp. 2521–2532, 2014.

[15] J. A. Vera-Gómez, A. Quesada-Arencibia, C. R. García, R. S. Moreno, an d F. G. Hernández, "An Intelligent Parking Management System for Urban Areas," *Sensors (Basel).*, vol. 16, no. 6, Jun. 2016.

[16] W. : Www, S. V Reve, and S. Choudhri, "International Journal of Emerging Technology and Advanced Engineering Management of Car Parking System Using Wireless Sensor Network," vol. 2, no. 7, 2012, Accessed: Sep. 05, 2021.

[17] A. Nandugudi, T. Ki, C. Nuessle, and G. Challen, "PocketParker: Pocketsourcing parking lot availability," *UbiComp 2014 - Proc. 2014 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput.*, pp. 963–973, 2014.

[18] N. Moses and Y. D. Chincholkar, "IJARCCE Smart Parking System for Monitoring Vacant Parking," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 5, 2016.

[19] L. L. Long and M. Srinivasan, "Walking, running, and resting under time, distance, and

average speed constraints: optimality of walk–run–rest mixtures," *J. R. Soc. Interface*, vol. 10, no. 81, Apr. 2013.

**Md Ismail Hossen** is a lecturer at Dept. of Computer Science, Faculty of Science and Technology, American International University-Bangladesh (AIUB). He obtained his B.Sc. and M.Sc. degrees from Multimedia University, Malaysia in 2017 and 2019 respectively. He teaches C, C++, Java, C#, Web Tech, Data Structures, Computer Graphics, Algorithms and computer organization and architecture. His research interests in machine learning, computer vision and image processing. He is now actively involved in supervising B.Sc. thesis students in the area of machine learning, data science, and image processing. He could be reached at ismailmmu@gmail.com.

**Michael Goh** is an Associate Professor in the Faculty of Information Science & Technology (FIST) at Multimedia University (MMU), Malaysia. His current research interests include pattern recognition, image processing, data classification, video analytics, fusion and computer vision, specifically automated hand-based multimodal biometrics.

**Tee Connie** received her Bachelor of Information Technology (Hons) degree in 2003, Master of Science (IT) degree in 2005, and PhD in IT in 2015 from Multimedia University Malaysia. Currently, she is the Dean of Institute for Postgraduate Studies. She also serves as an Associate Professor at Faculty of Information Science and Technology at Multimedia University. Her research interests cover the areas of computer vision, image processing and machine learning. She is the principal investigator of several research funding in the areas of gait recognition and human activity recognition.

Md Nazmul Hossain is a Lecturer of the Department of Computer Science, Faculty of Science and Technology, American International University-Bangladesh (AIUB). He obtained his B.Sc. and M.Sc. degrees from American International University-Bangladesh, Bangladesh in 2018 and 2019 respectively. He teaches C++, Java, C#, Data Structures and Computer Organization and Architecture. His research interests in Machine Learning and Data Science. He could be reached at nazmul5670@gmail.com