Monetizing Parking IoT Data via Demand Prediction and Optimal Space Sharing

Thanchanok Sutjarittham, Hassan Habibi Gharakheili, Salil S. Kanhere, and Vijay Sivaraman

***Abstract*—Transportation is undergoing significant change due to advances in automotive technologies such as electric and autonomous cars and transportation paradigms such as car and ridesharing. Coupled with the rapid prevalence of IoT devices, this provides an opportunity for many organizations with large on-premise parking spaces, to better utilize this space, reduce energy footprint, and monetize data generated by IoT systems. This paper outlines our efforts to instrument our University's multi-storey parking lot with IoT sensors to monitor real-time usage, and develop a novel dynamic space allocation framework that allows campus manager to re-dimension the car park to accommodate both car sharing and existing private car users.**

Our first contribution describes experiences and challenges in measuring car park usage on the university campus and removing noise in the collected data. Our second contribution analyzes data collected during 15 months and draws insights into usage patterns. Our third contribution employs machine learning algorithms to forecast future car park demand in terms of arrival and departure rates, with a mean absolute error of 4.58 cars per hour for a 5-day prediction horizon. Lastly, our fourth contribution develops an optimal method for partitioning car park space that aids campus managers in generating revenue from shared cars with minimal impact on private car users.

***Index Terms*—IoT, machine learning, optimization, parking lot, smart campus, license plate recognition, forecasting**

I. Introduction

UNIVERSITIES worldwide are experiencing a surge in student enrollments [2], accompanied by an expansion in staff numbers, which together have contributed to an increase in demand for on-campus parking. Despite the increasing trend of private car usage to commute to campus [3], as many as 10-45% of available parking spaces are empty since they are distributed across a large campus area [4]. This problem has also been observed at our campus in UNSW Sydney, where one of the multi-storey parking lots fill up by 10 am while the other often has availability.

Parking spaces can be thought of as perishable goods with sunk cost and an empty space at any time resembles an unsold item that can not be resold later [5]. Hence, effective management is required to ensure their efficient use. The rapid advances in information technology in conjunction with data monetization have improved the efficiency of parking management systems with the push towards adopting dynamic and data-driven policies. A good example is the adoption of a dynamic pricing scheme for public parking spaces by Municipal Transport Agency in San Francisco [6], where parking rates are adjusted based on demand data, in order to maintain their target utilization. Capturing fine-grained dynamics of parking usage over a long-term period is likely to offer far-reaching benefits. In particular, high-resolution data can be used to inform strategic decisions such as expansion of parking capacity, developing new parking facility, or partitioning spaces to accommodate new paradigms of car use [7].

The medium-term future is likely to evolve around shared transport, leading to autonomous vehicles in the longer-term. This will not only enhance commuter experience, but also cut down the use of fuel, alleviate the number of cars on the road, and improve traffic congestion. Car sharing (offered by companies such as GoGet, ZipCar, Car2Go, etc.) is projected to grow at a rate of over 20% between 2018 and 2024 [8] and is becoming a more mainstream mode of transport. Accordingly, universities such as UNSW can leverage such trends to go green by encouraging car sharing schemes in order to reduce on-campus parking congestion and savings in infrastructure spending for new parking facilities to keep up with the growth of the campus population.

Many universities are moving towards this trend by partnering with car sharing companies to offer shared transport services to their campus community [9]. Current schemes are predominantly based on static allocation where a fixed number of parking bays are reserved for car sharing vehicles, and only support round-trip transport services where vehicles are required to be returned to their dedicated based station. In recent years, there has been a rapid adoption of one-way car sharing model that provides a more flexible service to users by allowing them to leave shared vehicles at locations different from their pick-up point [10]. This recent trend will likely capture new shared transport users, hence motivating universities to adopt a more efficient dimensioning method for their parking infrastructure. Accordingly, existing static allocation of parking spaces can potentially be replaced with a dynamic scheme where spaces allocation changes dynamically based on predicted usage demand.

This paper describes our experience in instrumenting an on-campus car park for real-time monitoring of space utilization, and develops a novel framework for monetizing the collected data by dynamically allocating parking spaces to car-sharing and private car users. We first comprehensively analyze the car park usage data that spans over a period of 15 months, covering both teaching and non-teaching periods, and highlight interesting insights into car arrival and departure patterns. We

T. Sutjarittham, H. Habibi Gharakheili, and V. Sivaraman are with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia (e-mails: t.sutjarittham@unsw.edu.au, h.habibi@unsw.edu.au, vijay@unsw.edu.au).

S. Kanhere is with the School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: salil.kanhere@unsw.edu.au).

This submission is an extended and improved version of our paper presented at the IEEE WoWMoM 2019 conference [1].

then apply machine learning models to forecast future car park usage demand and compare predictive performance of various learning algorithms across multiple forecast horizons, ranging from a day up to 10 weeks. Next, we develop a continuous-time non-homogeneous Markov model based on the predicted usage to simulate dynamic partitioning scheme for allocating parking spaces to private and shared cars. Finally, we formulate an optimization problem to assist campus managers in selecting an optimal allocation (*i.e.,* fraction of spaces allocated to car sharing vehicles) that aims at enhancing parking space utilization while avoiding situations where users are turned away due to lack of parking spots.

The rest of this paper is organized as follows: §II describes relevant prior work. §III outlines our experiences in implementing an IoT car park monitoring system for real-time data collection while §IV presents interesting insights obtained therein. §V compares machine learning algorithms in forecasting multi-step ahead arrival and departure rate of cars. We show how the predictions are used to approximate the number of rejected users (given their respective allocation) using Markov modeling and an optimization formulation that can aid car park dimensioning decisions in §VI. The paper is finally concluded in §VII.

II. Related Work

In this section, we present related work on car park monitoring technologies and methodologies for predicting parking demand.

A. Parking Management

According to Litman [7] parking problem is typically perceived as inadequate parking supply that has led planners to provide more supplies in order fulfill the growth of demand. Consequently, abundant parking facilities have resulted in urban sprawl, causing parking demand and supply to get further inflated. New parking management paradigms, on the other hand, aim to use parking facilities more efficiently, instead of expanding the physical infrastructure. This not only reduces development costs but also supports more strategic objectives such as reduction of motor vehicle use by encouraging the adoption of alternative modes of transport, thereby reducing traffic congestion. Parking management strategies can range from parking regulations (*i.e.,* controlling who, when and how long vehicles can park) to parking pricing (*e.g.,* performance-based pricing [11]) to mobility management which aims to change travel behavior in terms of travel frequency, mode, destination, or time. This paper focuses on promoting car-sharing programs which is an example of mobility management, providing commuters with richer transport options.

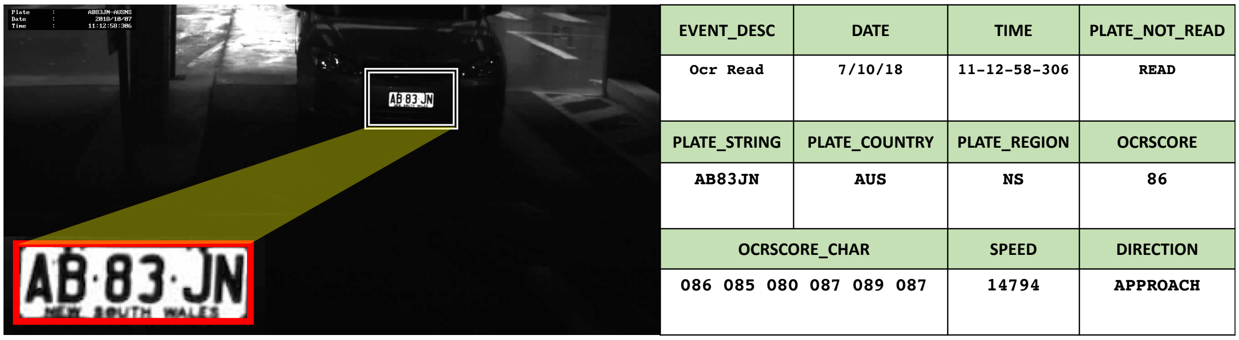
B. Sensing Technologies

Among various parking space monitoring technologies available in today's market, many smart parking deployments have adopted IoT sensors to detect the presence of vehicle at each parking bay. Some of the sensors that have been deployed include ultrasonic, light, temperature, acoustic, and magnetic sensor [12], which must be deployed in each parking slot. The approach to monitor each spot individually is expensive, especially for a large parking lot with hundreds of parking bays, hence such solutions are typically employed only in commercial parking areas such as shopping centers and airports. A more cost-effective solution is to use existing CCTV (closed-circuit television) camera to acquire images or videos of a car-park's view and apply image processing to obtain car occupancy data [13]. However, continuously recording images (or videos) of users' vehicles may raise privacy concerns, especially when images are collected without users consent. Radio frequency identification (RFID) technology can also be employed to identifying vehicles for parking management. The system involves installing an RFID reader at an entrance and exit of the car park in order to detect RFID tags of arriving and departing cars. The drawback of this method is that car park users need to have the tag attached to their vehicles. Casual visitors who have not installed the tags would be excluded. A simpler solution, which requires a smaller scale implementation, is to install a sensing device to capture arrivals and departures at every car park gate. For instance, license plate recognition (LPR) camera can be used to automatically record license plate number of each incoming and outgoing cars, providing a more cost-effective solution to measuring usage of the parking spaces.

C. Parking Prediction and Modeling

There is research in literature on predicting real-time parking availability, particularly for reducing congestion and eliminating inefficient cruising. One of the popular approaches is to use a stochastic model, where driver's arrival and departure behaviors are assumed to follow certain distributions. Many studies adopting this approach use information collected from vehicular ad-hoc network in order to predict car park occupancy, and explore different queuing models such as $M/M/m/m$ queue [15], [16] and $M/G/c/c$ queuing model [17]. The models allow blocking probability to be calculated and disseminated to users via vehicular communications, thus preventing more users from approaching the parking lot if it is fully occupied. Others model parking process as a birth-death stochastic process for revenue prediction [18], where the former corresponds to car arrivals and latter to departures. The aforementioned studies obtained numerical results through simulation, often through using fixed arrival and departure rate. This limits the efficacy of the method as real parking demand displays seasonal behavior, where parking demand is notably impacted by factors such as time-of-day and day-of-week.

With the emergence of IoT devices to collect real-time and historical parking information, many works have adopted data-driven approach for their occupancy estimation. An extensive range of methods that have been explored including classical time-series models like ARIMA [19]; non-parametric models such as regression trees and support vector regression [20]; and spatiotemporal models [21] when data of multiple parking locations are available. Furthermore, new techniques involving deep learning have been gaining popularity in the last decade, in additional to classical multi-layer perceptron (MLPs) for



EVENT_DESC	DATE	TIME	PLATE_NOT_READ
Ocr Read	7/10/18	11-12-58-306	READ
PLATE_STRING	PLATE_COUNTRY	PLATE_REGION	OCRSCORE
AB83JN	AUS	NS	86
OCRSCORE_CHAR		SPEED	DIRECTION
086 085 080 087 089 087		14794	APPROACH

Fig. 1. LPR camera outputs for author's car entering into the campus car park.

predicting future parking occupancy [22], several variants of recurrent neural networks (RNNs) have been proposed including LSTM [23] and RNN using evolutionary algorithms [24]. All aforementioned studies predict parking occupancies in the near future as to provide real-time or up to 1-hour in advance parking information to the users. Despite the fact that accurate short-term parking prediction can undoubtedly enhance informed parking decisions for users, longer term estimation is required for future space management such as space allocation problem that our work aims to address.

III. Data Collection and Cleansing

In this section, we first outline our experience with license plate recognition (LPR) technology to measure car park usage. Since license plate information is private and confidential, we obtained appropriate ethics clearances for this study (UNSW Human Research Ethics Advisory Panel approval number HC171044) prior to conducting the experiment. We then briefly explain our system architecture for collecting, storing, and analyzing data collected by the LPR cameras. Lastly, we discuss measurement challenges, quantify errors and present our method for data cleansing.

A. *License-Plate-Recognition Camera*

We investigated several commercial sensors for our car park monitoring solution with two goals. First, we want to have a complete ownership of the data without risking it leaving our campus infrastructure. Second, we aim to access our own data without relying upon a vendor, hence freeing us from ongoing service costs. This would allow us to integrate the collected data into a central repository of our overarching Smart Campus project in order to facilitate better analytics across many data feeds we have on campus [25].

We selected Nedap's automatic number plate recognition (ANPR) camera [26] as our parking monitoring solution. To read license plates of vehicles, the camera uses LPR technology which involves two main stages of operation: (a) locating license plate in the captured image by isolating a rectangle area (of the license plate number), using physical characteristics such as the shape, symmetry, width to height ratio and alphanumeric characters; and (b) separating and recognizing characters inside the isolated image [27].

The camera unit consists of several components including a high definition camera, infrared (IR) illumination, and ANPR, allowing the camera to read complex number plate at various lighting conditions (due to weather or different hours of the day). The recognition engine relies on an on-board library that supports license plates readings from specific countries, each of which uses its own characters, colors, and designs. The camera also provides a management console (a web-page accessible via its IP address) that allows users to configure various parameters such as shutter time, strobo time (activation time of IR illuminator), and gain. We configured the camera to use the default "Autoiris" mode, which is recommended by the manufacturer guidelines, in order to allow the camera to automatically adjust the parameters based on the current lighting condition.

The ANPR system is capable of real time optical character recognition (OCR) processing, which runs when a license plate number is presented within the camera's frame. For each detection, the algorithm outputs two types of data; a JPEG image (with adjustable quality value between 0 and 100, where 100 is the highest resolution 1080p) of the vehicle, and a data record containing parameters such as timestamp, license plate string, OCR score, speed, country, state/region, type of vehicle, and up to 50 more fields. Fig. 1 shows an example of a real license plate (for the private vehicle of an author of this paper) recognized by the camera. On the left is the JPEG captured image showing the isolated license plate, and on the right is a list of selected key data fields generated by the OCR algorithm. The definition of the fields are as follows:

- EVENT_DESC: The value "OCR Read" means that the algorithm was able to recognize individual characters in the isolated license plate.
- DATE and TIME: Timestamp of the record.
- PLATE_NOT_READ: This field indicates whether the license plate was successfully isolated or not – the value could be either "READ" or "NOTREAD".
- PLATE_STRING: The output string of the recognized license plate. Australian cars and motorcycles have 6 and 5 alphanumeric characters on their license plate respectively.
- PLATE_COUNTRY and PLATE_REGION: Country and state of the license plate – AUS for Australia, and NS for New South Wales.
- OCRSCORE: Overall confidence value (between 0 and 100) given by the camera on how accurately the entire license plate number is recognized.

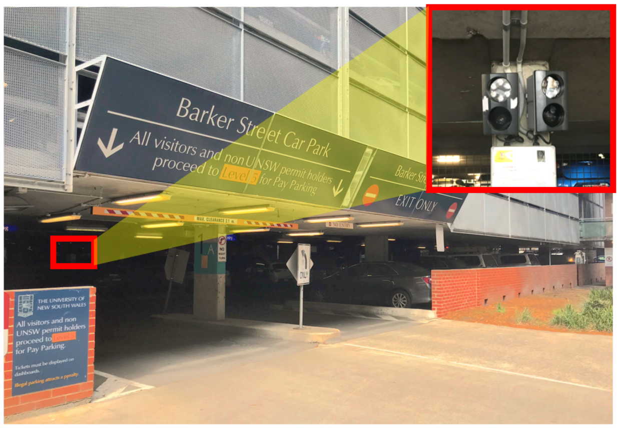

Barker Street Car Park
All visitors and non UNSW permit holders proceed to Level 5 for Pay Parking
EXIT ONLY

Fig. 2. A real picture showing entrance and exit of our campus car park and a pair of LPR cameras installed side-by-side close to the ceiling of the car park ground floor.

- `OCRSCORE_CHAR`: OCR score for individual characters of the string – in this example, character "J" has the highest score 89 and character "8" has the lowest score 80.
- `SPEED`: Speed of the vehicle in 100× actual speed (km/h) – we found that this data field is unreliable since reported values ranged from 1000 to 10,000,000.
- `DIRECTION`: Direction of the vehicle relative to the camera, *i.e.,* "APPROACH" for entry camera and "GO-AWAY" for exit camera.

During our field trial, we only captured and stored text output of the camera and not the JPEG image in order to maintain users' privacy and save space for data storage. In order to tune various parameters of cameras and collect ground-truth records (for measuring accuracy), we temporarily recorded a number of image captures. The camera supports both local storage on an Secure Digital (SD) card and an external File Transfer Protocol (FTP) for data collection. The later option was selected for continuous data collection, where the camera creates and automatically updates a text file in Comma Separated Values (CSV) format containing information about every detected vehicle by the ANPR.

For the deployment, we chose an on-campus 5-storey car park that serves 895 parking spaces for students, staff, visitors, and contract workers. The first four levels of the facility are reserved for permit holders and the top level for hourly-based paid parking. The car park has a one-lane entrance and exit on the ground floor, which we installed two LPR cameras to capture the arriving and departing cars as shown in Fig. 2. For installation, the devices require 24 volts direct current (VDC) power supply and communicates via an Ethernet port.

Hence, we (with help from our campus Estate Management) supplied new power points and provisioned Ethernet ports for the cameras. We note that the flow rate of cars at our deployed car park is fairly moderate, with an average of 3 cars per minute during peak hours on busy days, and hence the workload on the OCR system is not a concern, especially when the response time of cameras (for generating output records) is in the order of tens of milliseconds.

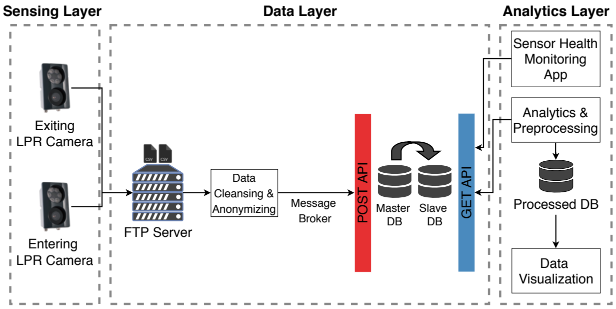


Fig. 3. System architecture of collecting, analyzing, and visualizing data from LPR cameras.

TABLE I
MEASUREMENT ACCURACY OF LPR CAMERA FROM GROUND-TRUTH OF CARS AND THEIR LICENSE-PLATES.

	Accuracy	
Camera	Capture rate	Read rate
Entry	94%	85%
Exit	88%	42%

B. Measurement System Architecture

Fig. 3 shows the architecture of our car park monitoring system, comprising of three main layers: (a) "sensing layer" is where the LPR cameras record license plates information of arriving and departing cars; (b) "data layer" is the core of our system, hosting an FTP server, a software engine for cleansing and anonymizing data, message broker, and multiple databases for backup and load balancing. Once the data is cleaned, it is passed onto the message broker for unifying records into a JSON format. Each record is first tagged with time-stamp and sensor UUID, and then is posted via a RESTful API to our master database; (c) "analytics layer" includes health check monitoring (whether cameras are active and functioning, or not), data analysis, and visualization modules – this layer retrieves raw data from the master database (DB) and writes computed occupancy (real-time) and stay duration (per vehicle) into another DB that is used as a backend for visualization.

The cameras are connected via a high-speed wired Ethernet cable to the campus network, sending real-time data records to backend servers on-premise. Given the reliable connection between sensing layer (cameras) and analytics layer (servers) we do not consider a stochastic process for the communication within our system.

C. Measurement Accuracy

There are various factors that can impact the accuracy of the LPR cameras including placement (*e.g.,* height and angle of installation), lighting conditions, speed of vehicles, angle of the license plate, and physical condition of the license plate. We quantify the accuracy using two metrics: (a) "capture rate" which is a fraction of cars correctly detected, and (b) "read rate" which is a fraction of license plate numbers that are recognized correctly by the OCR algorithm running inside the camera. According to the ANPR's manufacturer [28], it is expected to have capture rate and read rate of 98% and 95%, respectively.

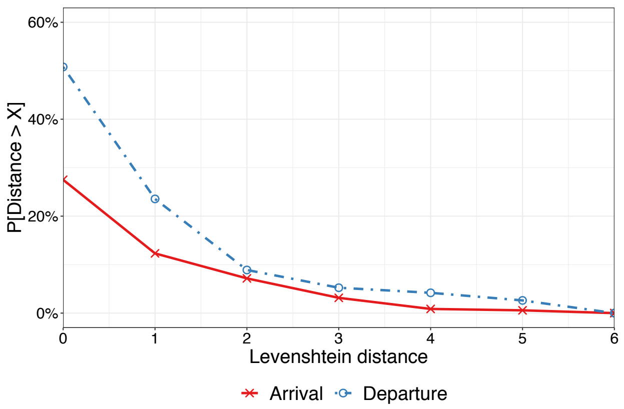


Fig. 4. CCDF of Levenshtein distance between captured and ground-truth plate numbers.

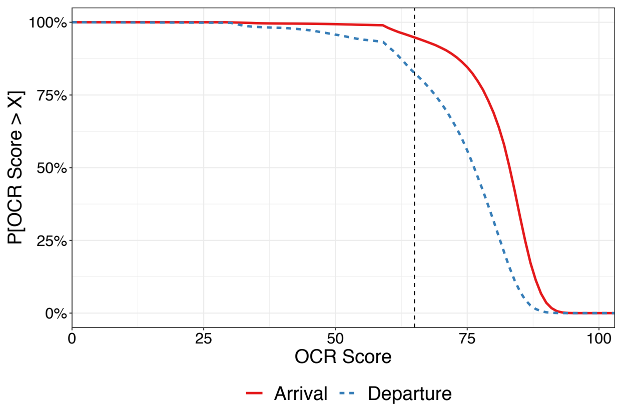


Fig. 5. CCDF of OCR score for entry and exit cameras.

TABLE II
DISTRIBUTION OF ERROR TYPES FOR EACH CAMERA (GROUND-TRUTH DATASET).

	Error types		
Camera	Multiple recognitions	Incorrect locating	Incorrect recognition
Entry	59.3%	4.2%	36.4%
Exit	13.5%	34.9%	51.6%

To measure the capture rate, we performed several spot measurements over 5 days (Monday to Friday), each for a period of 5 hours (11am to 4pm) in order to collect ground-truth data of car park usage. We used a GoPro camera to record video logs and captured a total of 1400 vehicles entering and exiting the car park. For measuring read rate, we first enabled the JPEG recording on both cameras for a week and manually inspected 700 images for each of the LPR cameras (randomly selected) in order to obtain the ground-truth license plate labels.

Table I summarizes the accuracy of both entry and exit cameras. It is seen that the exit camera underperforms by both accuracy metrics, especially with a very low read rate of 42%. This means that more than half of the departing vehicles are not correctly recognized. The poor read rate of the exit camera was mainly due to its non-ideal placement which causes the detection to happen at a slight angle, and thus negatively affecting the performance of the OCR algorithm. Re-positioning the camera was a nontrivial task due to difficulty (and cost) of provisioning Ethernet port and power outlet for the new position. With the majority of read rate errors resulted from misinterpretation of only one of the six characters, this can be used to our advantage in the cleansing process.

D. Measurement Errors

The errors observed from the outputs of cameras can be categorized into three types: (a) multiple recognitions of the same license plate, (b) incorrect recognition of license plate location, and (c) incorrect recognition of license plate characters.

Multiple recognitions: This error type occurs when the camera takes multiple images of a single vehicle (possibly because of its speed or moving pattern), and thus triggers the OCR algorithm multiple times. This generates multiple data records for the vehicle in the CSV file, resulting in over-counting of vehicles. These multiple records do not necessarily have the exact same license plate string – it may output slightly different strings due to the angle of the moving car and its distance to the camera in a sequence frames captured.

Incorrect locating: This type of error occurs when the OCR algorithm incorrectly locates a license plate in the captured image and attempts to recognize characters. The output license plate strings from this error are almost always "`OCR NOT READ`" as the read plate does not match any of known formats of license plates available in its embedded library. There are rare cases where a non-license plate object gets partially recognized that their output appears as incomplete strings with low OCR scores of below 60.

Incorrect recognition: This type of error occurs when the camera successfully locates a license plate, but fails to recognize the characters correctly. To quantify the severity of the read errors, we employ Levenshtein distance [29] to measure the difference between two string words, *i.e.,* the minimum amount of single character addition, substitution or deletion required to make two strings identical. For example, the Levenshtein distance between the string "`ABC123`" and "`AC123`" is 1, two strings will be identical by inserting a character "`B`" into the string "`AC123`". We compute the Levenshtein distance between ground-truth and recognized strings, the complementary cumulative distribution function (CCDF) plot of the distance is shown in Fig. 4. It is seen that up to 28% of records from the entry camera and 52% from the exit camera have at least 1 character misrecognized (*i.e.,* distance of more than 0 character). We also observe that the majority of these errors are caused by 1 misread character, accounting for 15% for entry and 30% for exit of the total observed records. Note that incorrect recognition also results in lower OCR scores for the output record. Fig. 5 shows the distribution the OCR score of our deployed cameras. It can be seen that for the exit camera, 83% of the data records come with an OCR score more than 65. For the entry camera, on the other hand, records seem more reliable where 85% of them have the score greater than 75.

We summarize the distribution of each error type for both cameras in Table II. As can be observed, the errors from entry records are largely stemmed from multiple-recognitions which accounted for 59.3% of the total errors. This is attributed to

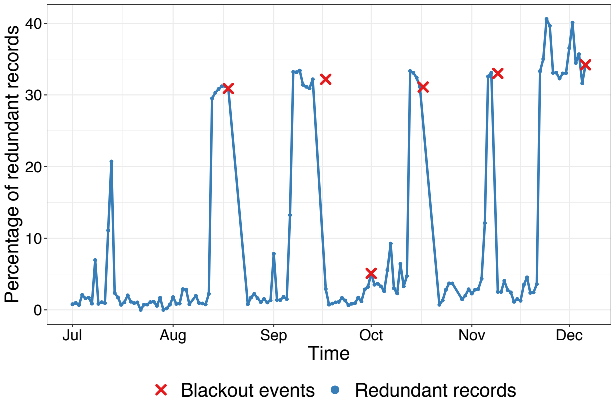


Fig. 6. Rate of error due to redundant (multiple) records rises prior to camera black-outs.

the recurrence of a peculiar problem where the camera's view unexpectedly blacked out, resulting in no data being recorded (*i.e.,* black-out events). To understand this relationship, we plot in Fig. 6 the time-trace of daily error rate due to redundant records, overlaid by black-out events which required camera reboot. It is interesting to observe that the rate of redundant records steeply rises a few days prior to black-out events (evidenced by red cross markers in the middle of August to middle of December). The errors in data records of the exit camera, on the other hand, are mostly (51.6%) from incorrect recognition. We believe this is primarily due to the non-ideal placement of the camera that causes it to capture license plates at an angle which is non-ideal, hence resulting in poor performance of the OCR algorithm. Furthermore, we can see that 34.9% of the errors are due to incorrectly locating of license plates, while the same measure is only accounted for 4.2% of the total errors from the entry camera. By manually inspecting the collected images, we found that moving grass (close to the exit point of the car park, as visible in Fig. 2) gets occasionally detected as a moving object by the exit camera. This issue does not affect our entry camera, and thus it displays a much lower rate of incorrectly locating errors.

E. Data Cleansing and Preprocessing

We cleanse raw data collected from the two cameras with the following objectives: (a) removing multiple records to obtain the correct count of arrivals/departures, (b) removing records of non-vehicle objects incorrectly captured by cameras, and (c) matching license plates captured by both cameras to deduce the distribution of stay-duration in our campus car park.

Fig. 7 shows our cleansing process with various stages involved. We first remove duplicate records caused by multiple recognitions. A license plate is considered as "redundant" if it re-appears within the next five records after its first appearance with the Levenshtein distance between the plates of two or less. The distance threshold of two is selected based on our ground-truth analysis (Fig. 4) – applying this filter eliminates only 8% and 10% of entry and exit records, respectively. We found that increasing the threshold to three will only improve the coverage by less than 10%, which is not substantial especially when the chance of plates getting mismatched increases (two different plates get incorrectly matched and treated as the same plate). For license plates identified as redundant, the one with the highest OCR score is kept and others are discarded.

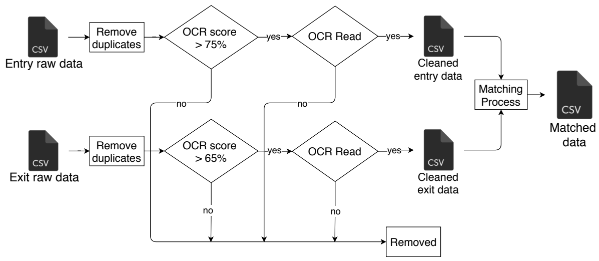


Fig. 7. Data cleaning process.

TABLE III
PERCENTAGE OF RECORDS REMOVED FROM THE ENTIRE DATASET AT EACH STAGE OF CLEANSING.

Camera	Error types		
	Multiples (redundant)	low OCR score	OCR-Not-Read
Entry	9.7%	12.9%	2.5%
Exit	7.9%	8.5%	8.2%

We then remove records with low OCR scores. Most of these records relate to the erroneous captures (redundant records) of one vehicle. As discussed earlier in Fig. 5, we use filtering threshold of 75 and 65 for OCR scores for the entry and exit cameras respectively. Lastly, we remove all records with `OCR NOT READ` value – those with incorrectly located license plate in the captured image. The pair of cleaned data will be used for arrival/departure counts. A summary of records removal due to each error type is shown in Table III.

As mentioned earlier, the goal of the last stage of our cleansing process is to match vehicles from the entry and the exit datasets. The output of this stage will be used to compute stay duration of users. Similar to the multiple recognitions removal process, we use a Levenshtein distance of two or less for the matching. For one-to-many matched events, which rarely occur, we select the pair that yields the lowest Levenshtein distance and the highest OCR score. By running the matching process on daily data from July to the end of December, we were able to match 86% of the records on average. The remaining unmatched records correspond to: (a) overnight parking, (b) vehicles that are captured by one camera (mostly entry camera), but not the other.

We found from our spot measurements that only a small number of vehicles stayed overnight (*i.e.,* 23 cars per day on average for a sample size of 11 nights), and thus they would not have significant impact on the car park usage patterns. We therefore analyze our dataset on a per-day basis (*i.e.,* midnight-midnight).

IV. Analysis and Insights into Usage Pattern

In this section, we analyze our cleansed data (obtained from §III) which spanned 15 months of teaching and non-teaching periods in 2018 and 2019, to highlight the usage pattern of the campus car park across various temporal dimensions including time-of-day, day-of-week, week-of-semester, and semester break/exam periods.

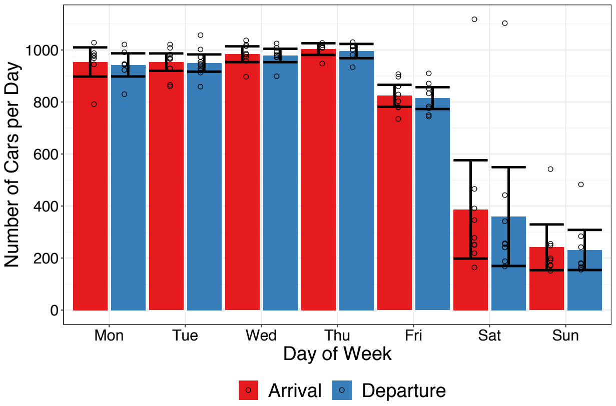


Fig. 8. Number of arriving and departing cars per day during teaching period.

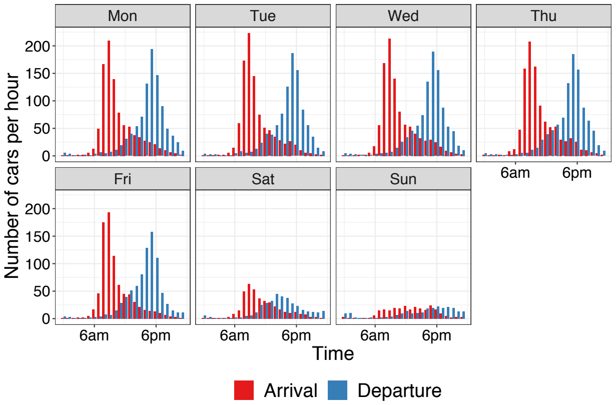


Fig. 9. Average hourly arrival and departure rate of cars for each day-of-week.

A. Arrival and Departure Pattern

We begin with Fig. 8 that depicts the average daily number of cars entering to (red bar) and exiting from (blue bar) the car park for each day-of-week during teaching periods. The black points represent actual measurement values and the error bars represent 95% confidence interval of data-points. On average, it is seen that there are about 950 to 1000 cars using the car park on weekdays, except for Friday, where the number drops to about 830 due to fewer number of classes running. During weekends, the number of cars occupying the car park are significantly lower, with an average of below 400. Furthermore, we observe narrower error bars on weekdays compared to weekends, suggesting predictable usage patterns for weekdays. In contrast, the car park usage varies significantly on weekends ranging from 200 to 600. Interestingly, it goes beyond 1000 cars for one particular Saturday (*i.e.,* 1 September 2018). After checking the University event calendar, we found that this corresponds to the University Open Day which typically attracts a large number of high-school students and their families. The University provides free parking for all Open Day Visitors.

Fig. 9 illustrates an average hourly count of arriving cars (red bars) and departing cars (blue bars) by time-of-day, for each day of the week (separate graphs per day). It is seen that during weekdays, the arrival rate starts rising steeply from 6am, peaks at 8am-10am, and falls slowly afterwards. The departure process displays a similar bell-shape pattern but shifted in time by about 8 hours, *i.e.,* rising in the afternoon, peaking at 5pm-6pm, and falling afterwards. During peak demand times, about 200 cars enter and exit the parking lot per hour. This number is slightly lower for Friday (*i.e.,* 194 and 165 cars per hour). We also observe an irregular pattern for weekends where car park usage heavily depends on events hosted on campus. Our findings of weekday arrival/departure pattern corroborate with other studies [30].

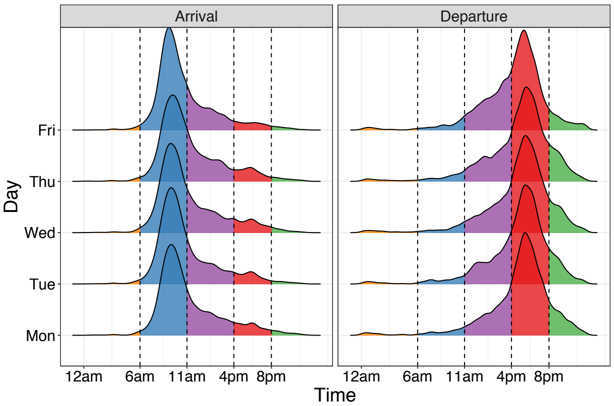


Fig. 10. Distribution of arrival and departure time during teaching period, across each weekday

Fig. 10 depicts the distribution of arrival and departure rate for the five weekdays (in a stacked representation with Friday on top and Monday at the bottom). To better illustrate the pattern, we color-coded five time intervals: orange for prior-sunrise (12am-6pm), blue for morning-peak (6am-11am), purple for afternoon-offpeak (11am-4pm), red for evening-peak (4pm-8pm), and green for night-time (8pm-12am). Note that the difference in absolute numbers is not seen for Friday due to normalization. Unsurprisingly the distribution plot shows that both arrival and departure patterns are consistent across the majority of time-slots on weekdays, with the strongest similarity observed during peak hours. As before, we observe some different trends for Friday with the red part of the arrival curve (4-8pm) showing a less pronounced peak compared to other weekdays. This suggests fewer arrivals to campus on Friday evening, which can be attributed to the fact that there are very few evening lectures running on Friday. A closer examination of the exit curve for Friday reveals that the area in purple is larger than the corresponding regions of other weekdays. The green part of the curve is also considerably flatter than the other days. This suggests higher percentage of cars leaving during early afternoon time and lower percentage of cars exiting the car park after 8 pm on Fridays compared to Mondays-Thursdays.

Additionally, we looked at the arrival and departure patterns during different periods of the academic calendar including orientation week (O-Week) which is largely geared for new entrants to get acquainted with the university, regular teaching weeks, a week long mid-semester break, a study-break week right before final exams, exam periods and a lull period before the end of year holiday shutdown during which undergraduate students are away and campus attendance for everyone else progressively reduces. The general patterns are similar to Fig. 9 for all periods. One main difference observed was the morning arrival peak occurred one hour earlier (*i.e.,* 8am-9am)

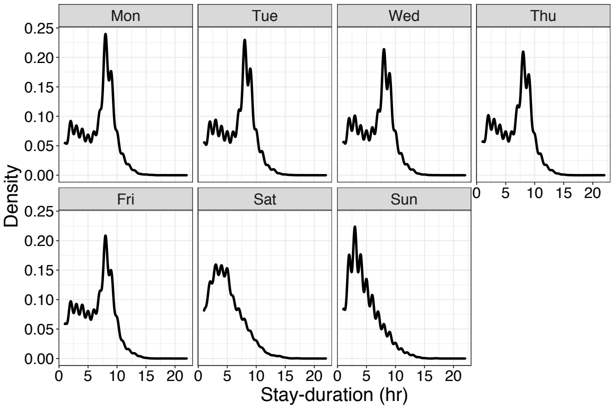


Fig. 11. Density distribution of stay-duration (hour) for each day-of-week.

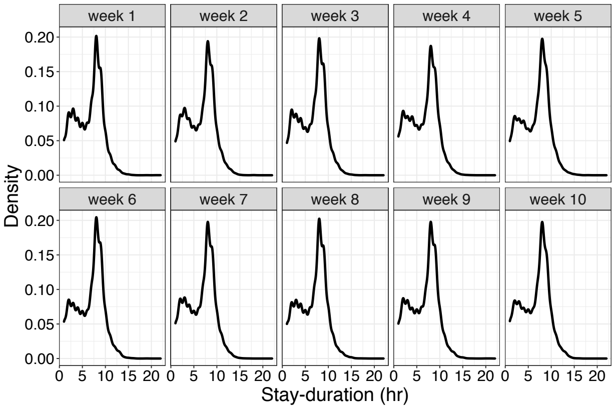


Fig. 12. Density distribution of stay-duration (hour) for each week of term.

during the O-Week and other non-teaching periods. The former can be attributed to the fact that a portion of students and administration staff arrive early during the O-week for setting up various activities. The latter can be ascribed to the fact that students are less likely to be on campus during the non-teaching period and thus the morning peak time is largely determined by the arrival pattern of staff who typically arrive earlier (between 8-9 am) than students. Furthermore, it can be observed that car park usage drops (*i.e.,* around 50 cars) during mid-semester break and other non-teaching periods, suggesting that the majority of car park users are staff members. This is not surprising due to high price of parking permits in our university.

B. Stay Duration Pattern

Our dataset allows us to obtain insight into stay-duration of car park users given that the license plate numbers are captured at entry and exit points. We choose to analyze the distribution of these patterns across various user groups, instead of on an individual basis. For this analysis we use the cleansed dataset after the matching process mentioned in §III-E.

Fig. 11 shows the distribution of stay duration (in hours) for each day-of-week. It is clearly seen that the distribution is peak at around 8 hours for Mondays to Fridays – this is consistent with the standard working hours in Australia, which is 7.6 hours a day [31]. We also observe that users tend to use the car park for about 2-4 hours during weekends - these users are likely to attend events hosted on or near-by the campus or Postgraduate students attending Saturday lectures.

We further look at the stay duration pattern for each week of term in Fig. 12. A consistent pattern of bi-modal (double-peaked) distribution is observed for each week. The peak stay duration, as expected, centers at 8 hours, a typical full-time work day. The second peak centered between 2 to 3 hours, highlighting usage patterns for weekends as well as usage from students and visitors.

C. Parking Behavior Users

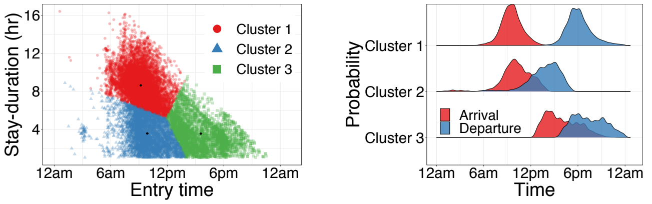
We now cluster car park users using k-means algorithm [32] to identify parking behavior of certain user groups on weekdays during teaching period. For each car we extract three features including arrival time, departure time, and stay duration. Three number of clusters (optimal value is $k = 3$) were selected based on the elbow method. Fig. 13 shows the result of our clustering. We show in Fig. 13(a) the scatter plot of stay-duration versus entry time for individual vehicles in our dataset – clusters are color-coded. It is seen that "Cluster-1" (shown by red region) corresponds to users who enter the car park early and stay more than 8 hours (center of cluster-1 is located at 9am entry time and 8.8 hours of stay duration) – this cluster represents full-time staff. "Cluster-2" users enter the car park at about the same time as Cluster-1 and stay shorter (*i.e.,* less than 5 hours) – morning visitors and typical students. Lastly, "Cluster-3" users are those who enter late and stay for a short period (with center at 3pm entry time and 3.5 hours of stay) – this cluster is likely to denote afternoon visitors and postgraduate coursework students who attend evening classes.

In Fig. 13(b), we show the distribution of arrival and departure time for each cluster. We observe that the distributions for Clusters-2 and -3 are relatively wider than of Cluster-1. Again, it is seen that full-time staff in Cluster-1 typically enter at about 9am and exit at about 5pm. Similar to the scatter plot, Cluster-2 users arrive early and leave early too. Lastly, as expected Cluster-3 users enter in the afternoon and exit in the evening. We also observe a significant overlap in the exit and entry distributions for these users, which suggests that they tend to stay on campus for a short period (about 3 hours).

Lastly, to quantify the impact of choosing a larger number of clusters (*i.e.,* more than the optimal value 3), we experiment with $k = 4$. In doing so, we observe that two of the original clusters, *i.e.,* Cluster-2 and Cluster-3 in Fig. 13, remain intact with the same arrival and departure characteristics. However, the original Cluster-1 gets divided into two sub-clusters, both representing the behavior of full-time staff. The first sub-cluster-1 displays a pattern fairly similar to Cluster-1; arrival time centered at 9am and departure time at 6pm. The other one, sub-cluster-2, which accounts for half of the size of the first one, presents a slightly different characteristic, with arrival and departure times shifted by an hour (centered at 10am and 7pm, respectively).

V. Forecasting Parking Demand

We apply three machine learning models to forecast long-term parking usage, specifically, arrival and departure rate of



(a) Stay-duration versus arrival time for each cluster.

(b) Distribution of arrival and departure time for each cluster.

Fig. 13. Clustering of car park users using arrival time, departure time, and stay duration.

cars per hour, and evaluate their performance across various forecast horizons.

A. Demand Time-Series Forecasting

Having an accurate forecast of user demand is essential in planning for future space allocation of the car park. In particular, long-term forecasting is useful for campus managers to implement car park partitioning schemes, where spaces are dynamically allocated to private and shared-car users, in order to improve the utilization of parking facility. In this work, we adopt multi-step ahead time-series forecasting for predicting arrival and departure rates of cars. We perform the prediction for a range of time horizons, from one day to ten weeks, in order to determine an appropriate time horizon which provides sufficient demand forecast and yields an acceptable prediction accuracy.

The car park demand displays seasonal characteristics affected by several temporal factors like hour-of-day and day-of-week (as discussed in §IV). Additionally, including past observations (lagged dependent variables) as part of features in a prediction model is a standard practice in time-series analysis which is known to produce robust estimation [33]. Below we summarize features that we used in our forecasting models:

- Time-of-day (Fourier representation)
- Day-of-week (Fourier representation)
- Teaching/Non-teaching period
- Lagged variables (daily seasonal lags of last 10 days, *e.g.,* to forecast arrival rate between 9am-10am, historical arrival rates of the same time slot from the last 10 days were used as features)

We use Fourier representation as features to capture seasonality (*e.g.,* time-of-day and day-of-week) in our time-series data. Given a seasonal period of p hour, the Fourier terms with K sine and cosine pairs is defined as:

$$\left[\sin\left(\frac{2\pi kt}{p}\right), \cos\left(\frac{2\pi kt}{p}\right)\right]_{k=1}^{K} \tag{1}$$

Fourier terms for time-of-day and day-of-week seasonality are obtained by setting $p = 24$ and $p = 120$ (5 working days) respectively. We show an example of the first sine terms for both daily and weekly seasonality, comparing with the typical representations of seasonal variables in Fig. 14. It can be seen that Fourier terms (given the smooth nature of sinusoidal waves) allow us to better model the relationship between the starting and ending of a day or week (*e.g.,* continuity of the rates profile from 12am on a day to 1am on next day). We tested up to eight Fourier terms in our models and found that inclusion of two terms is sufficient for the prediction.

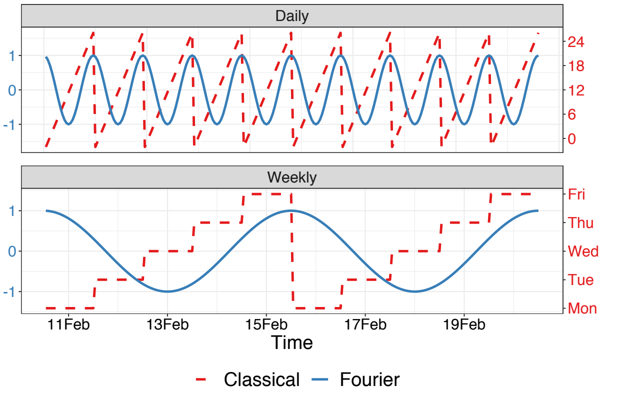


Fig. 14. Comparison of classical representation and Fourier terms representation of seasonality.

B. Strategies for Long-Term Forecasting

We predict future arrival and departure rates using machine learning models [34]. In what follows we describe our strategy to predict long-term usage demands and the learning algorithms we employ.

1) Multi-Step Ahead Prediction Strategies: In general, long-term forecasting is performed to predict the future behavior of an observed time series over a long horizon. Common strategies include: (a) recursive approach [35], where the one-step ahead prediction is fed back as an input to predict the next steps recursively (the predictor progressively takes estimated values as inputs, instead of actual observations); (b) direct approach [36], where multiple forecasting models are built, each corresponding to a forecasting horizon (steps); (c) hybrid strategy, which combines the previous two strategies [37] (models specific to each forecast horizon with recursive inclusion of the prediction from all previous horizons); and (d) Multiple-Output strategy, where a single model predicts multiple steps of forecasting in one go (the predicted value is not a scalar quantity but a vector of future values of the time series). This strategy, however is more complex and requires a longer time to train.

For our purposes, we adopted a direct approach as it does not suffer from the propagated errors like in recursive strategy and does not involve training complexity like in hybrid/multi-output strategy. However, our choice of approach requires multiple models for multiple forecast horizons.

2) Forecasting Algorithms: We employ three widely-used algorithms to build our forecasting models:

Random Forest (RF) Regression is an ensemble learning method where a collection of decision trees are combined to make the final prediction in order to give better accuracy while reducing the likelihood of overfitting. Each decision tree is generated by a random sampling of training observations. Also, random subsets of features are used for splitting nodes.

Support Vector Regression (SVR) applies the same principles as Support Vector Machine (SVM) by using the concept

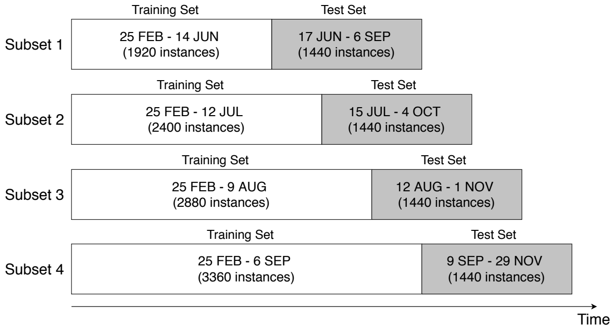


Fig. 15. Summary of four train/test subsets.

of maximal margin hyperplane. In SVR, a decision boundary with distance ϵ from the hyperplane can be tuned and the objective of the algorithm is to maximize the number of points (support vectors) within the boundary line. Additionally, hyper parameter C is used to tune the tolerance of data points that fall outside the decision boundary. For a non-linear SVR, a kernel trick is employed to transform the training data into a higher dimensional feature space prior to performing linear regression. In our model, radial basis kernel is used for this mapping.

Multiple Linear Regression (MLR) is one of the simplest prediction algorithms that involves multiple input variables (features) for the prediction. There are several techniques that can be used to train linear regression from the data. The most common one is ordinary least squares (OSL), where the objective of the algorithm is to minimize sum of squared errors when fitting the data.

C. Performance Evaluation of Forecasting Models

1) Evaluation Methods and Dataset: We use historical data from 2019 (from 25 Feb to 29 Nov) for training and testing our models. We employ nested cross validation procedure where the dataset is divided into multiple training and testing subsets, each training set is further divided into sub-cross-validation sets for tuning hyper parameters. Use of multiple train-test subsets allows for an unbiased and robust assessment of the performance of the models in predicting unseen data. Note that we preserve the temporal order of observations during the train-test splitting and cross validation in order to prevent data leakage so that future data is not used to train the model that predicts past data. Hence, we emulate and demonstrate a real-world forecasting scenario.

Fig. 15 shows a summary of how we divide our dataset into 4 subsets, each consisting of varying size data (from 16 weeks to 28 weeks worth of instances) used to train the model along with fixed-size data corresponding to test periods of 12 weeks starting right after their respective training period.

We also use a simplistic baseline model in our comparisons to confirm that the more sophisticated approach that we have adopted is worth the effort. Our baseline model uses a mean average rate of each specific hour-slot (*e.g.,* 9am-10am) computed over historical data as a prediction of future values for the corresponding hour-slot. In other words, past observations are used to compute a daily profile of hourly arrival and departure rates which will be used for any day in the future.

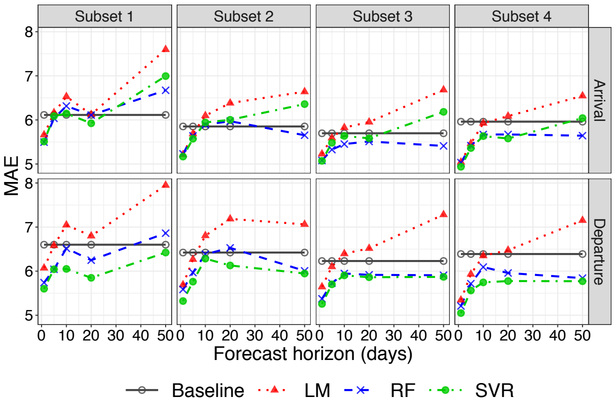


Fig. 16. Performance of forecasting models per individual subsets during cross-validation phase.

2) Evaluation Results: For each train-test subset, we first perform cross-validation on the training set using time slicing in order to tune the hyper parameters of our machine learning models. We then train the final model of each algorithm using the parameters that minimize the error and apply the model to predict the test set. Note that we will use Mean Absolute Error (MAE) as our performance metric as we intend to penalize every error equally (*i.e.,* mispredicting 10 cars per hour is twice as costly as mispredicting 5 cars per hour).

Table IV shows the numerical results in both cross-validation and testing phases. We observe that the MAE of prediction generally reduces from top (Subset1) to bottom (Subset4). For instance, RF model gives MAE of 5.29 on Subset1 and 4.56 on Subset4 for predicting the departure rate with 5-day horizon (highlighted in green). This improvement can be ascribed to the increase in size of training data, *i.e.,* 1920 and 3360 instances for Subset1 and Subset4, respectively. It also suggests that with more data, overfitting can be minimized and our models generalize better.

To visualize cross-validation results, we plot Fig. 16 to illustrate the MAE for the three ML algorithms under consideration. For short term forecasting, we can see that all three algorithms exhibit similar performance, with SVR yielding slightly better results for 1-day ahead prediction (average MAE of 5.17 and 5.31 for arrival and departure rate respectively) and RF model performing marginally better over 5 days arrival rate (average MAE of 5.61). For longer term prediction (10+ days), where predictive performance is observed to deteriorate, RF (blue lines with cross markers) is seen to outperform the other two algorithms for predicting arrivals while SVR (green lines with circle markers) yields the best results in predicting departures. On the other hand, LM yields the worst predictive power, suggesting that the parking rate data can be better explained using tree-based (*i.e.,* RF) or non-linear (*i.e.,* SVR) models as opposed to a linear one.

Next, we evaluate the performance of our models on the test set (unseen data). Fig. 17 shows the average MAE across all four subsets. In general, we can see that the forecast errors rise as the time horizon increases. For instance, the average MAE of departure prediction is as low as 4.67 for LM model when predicting at 1 day forecast horizon, but increases to 6.32 when

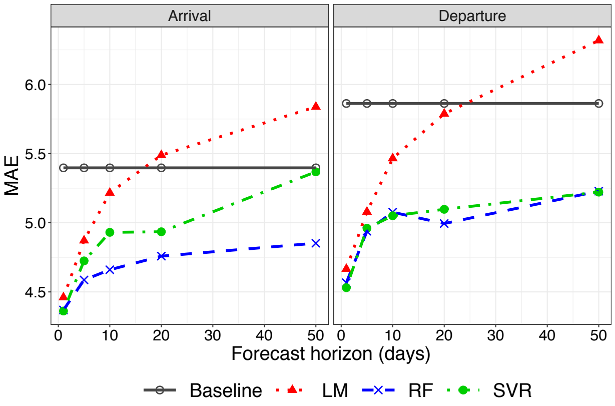


Fig. 17. Performance of forecasting models (averaged over the four subsets) during testing phase.

predicting at 50 day horizon. This highlights the fact that our models learn the pattern better when more recent observations are available. In predicting the arrival rates (left plot), RF is the clear winner with an MAE lower than 5.0 across all the horizons. To predict the departures, however, both RF and SVR models perform fairly similarly. Corroborating the validation results, LM yields the highest MAE for both arrival and departure among the three algorithms and even worse than the baseline for time horizons greater than 20 days.

From the results in Fig. 17, we can see that RF and SVR models perform better in forecasting parking demand on our university campus, with an average prediction error of less than 5 cars per-hour for a medium-term horizon (5-day), and less than 5.5 cars per-hour for a longer-term horizon (10-week).

Note that the rate of car arrival and departure varies significantly (between 0 and more than 200 cars per hour) throughout the day. Therefore, we analyze the dynamics of MAE (for the RF model) during non-peak hours versus peak hours for a medium-term prediction horizon. During non-peak hours, with an average rate of 18 cars (entering/exiting) per hour, the model yields MAE of 3.60 for both arrival and departure rates. We observe that more than half of the predictions come with an absolute error of less than 2. Also, it is found that about 20% of predictions give an error of more than 5 – these high errors predominantly correlate with a sudden change in parking usage due to irregular events/functions organized on campus during evening hours (between 4pm and 11pm), and hence can not be captured by our model. During peak hours (8am-10am for arrival and 4pm-7pm for departure), when an average of 180 and 130 cars per hour enter and exit the car park, the model yields MAE of 13, translating to about 10% of the total traffic. Note that large errors during peak hours are predominantly observed during non-teaching period when academics have flexible working arrangements, and hence a higher uncertainty in the parking usage is introduced.

We note that building RF and SVR models incurs an overhead of tuning their hyper-parameters, while MLR gives a closed-form solution with no hyper-parameters.

VI. Optimization of Carpark Partitioning

As car sharing continues to grow in popularity, especially with new emergence of service models like one-way car sharing, university estate managers may want to set aside some of car park spaces to accommodate shared vehicles. Such a forward looking strategy is likely to increase the utilization of campus parking facility, and hence generate new revenue. However, it is important to maintain the satisfaction of both private and shared vehicle users by minimizing situations where they are turned away due to lack of parking spots (rejected users).

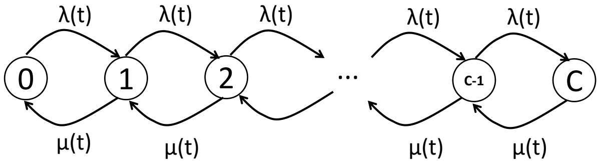


Fig. 18. Usage dynamics of carpark with capacity C – cars arriving at rate $\lambda(t)$, and departing at rate $\mu(t)$.

Since parking usage is highly seasonal depending on hour-of-day and day-of-week (§IV), we envision a novel partitioning scheme that dynamically allocates a certain fraction of total spaces to car sharing vehicles, as opposed to the existing static allocations. This dynamic allocation can be adjusted using predicted demands (§V), so that rejected users are minimized while a better utilization of the entire facility is achieved. In order to highlight the value of dynamic partitioning (driven by ML-based forecast), we compare its efficacy with a baseline static approach.

This section describes our methodology to decide an optimal space partitioning. We first develop a Markov model that estimates the number of rejected cars for a given partitioning scheme (fraction of cars allocated to car sharing companies). We then formulate an optimization problem to formally choose an optimal scheme based on minimizing the total cost of rejections for both types of users, *i.e.,* existing private-vehicle (PV) users and prospective shared-vehicle (SV) users. Finally, we evaluate the efficacy of our Markov modeling and optimization framework via simulation of real usage behaviors recorded through our IoT parking system discussed in §III.

A. Carpark usage as a Markov Process

A car park can be abstracted as a queuing system, with an incoming and outgoing cars considered as arrival and service events respectively, and the number of parking spaces as the queue capacity. We therefore model the dynamics of car park usage as a M/M/1/C queue whose operation is visualized in Fig. 18 by a continuous-time Markov chain. Each state of this chain represents the number of occupied spaces in the car park which has a total capacity C. An arriving car to the park at state C (full) is rejected. An hourly arrival and departure profile on an example weekday (11 Feb 2019) is illustrated in Fig. 19, where arrival rate peaks at 240 cars during 8am-9am and departure rate peaks at 200 cars during 5pm-6pm.

In continuous-time Markov process, transition rate matrix Q can be used to calculate the transition probabilities at any time $t \geq 0$. The Q-matrix, with dimension $(C+1) \times (C+1)$, consists of elements $q_{i,j}$ denoting the transition rate from state

TABLE IV
MEAN ABSOLUTE ERROR (MAE) OF PREDICTION MODELS.

	Arrival										Departure									
	Cross-Validation Phase					Testing Phase					Cross-Validation Phase					Testing Phase				
model	1-day	5-day	10-day	20-day	50-day	1-day	5-day	10-day	20-day	50-day	1-day	5-day	10-day	20-day	50-day	1-day	5-day	10-day	20-day	50-day
Subset 1																				
Baseline	6.11	6.11	6.11	6.11	6.11	5.65	5.65	5.65	5.65	5.65	6.60	6.60	6.60	6.60	6.60	6.14	6.14	6.14	6.14	6.14
LM	5.66	6.16	6.53	6.12	7.60	4.51	4.84	5.30	5.79	6.45	6.07	6.58	7.05	6.80	7.95	4.80	5.32	5.83	6.07	6.80
RF	5.50	6.04	6.32	6.11	6.67	4.62	4.85	4.98	5.08	5.22	5.74	6.04	6.52	6.25	6.86	4.83	5.29	5.45	5.53	5.53
SVR	5.51	6.09	6.15	5.92	6.99	4.43	4.98	5.05	5.42	5.74	5.60	6.04	6.05	5.85	6.42	4.73	5.15	5.34	5.50	5.81
Subset 2																				
Baseline	5.85	5.85	5.85	5.85	5.85	5.70	5.70	5.70	5.70	5.70	6.42	6.42	6.42	6.42	6.42	6.08	6.08	6.08	6.08	6.08
LM	5.21	5.69	6.10	6.38	6.64	4.59	5.11	5.46	5.50	6.22	5.69	6.27	6.81	7.19	7.06	4.73	5.21	5.60	5.78	6.47
RF	5.23	5.64	5.89	5.97	5.65	4.47	4.71	4.75	4.89	5.15	5.59	5.97	6.38	6.53	6.01	4.58	5.04	5.19	5.00	5.34
SVR	5.17	5.57	5.94	6.00	6.36	4.74	4.98	5.21	4.98	5.83	5.32	5.76	6.28	6.13	5.95	4.66	4.93	5.03	5.08	5.50
Subset 3																				
Baseline	5.70	5.70	5.70	5.70	5.70	5.59	5.59	5.59	5.59	5.59	6.23	6.23	6.23	6.23	6.23	5.99	5.99	5.99	5.99	5.99
LM	5.23	5.59	5.83	5.95	6.68	4.54	5.01	5.31	5.57	5.42	5.64	6.10	6.40	6.52	7.28	4.56	5.04	5.40	5.82	5.85
RF	5.08	5.33	5.45	5.51	5.41	4.45	4.69	4.76	4.87	4.83	5.37	5.73	5.95	5.92	5.91	4.53	4.94	5.07	4.91	5.03
SVR	5.07	5.48	5.64	5.58	6.18	4.41	4.76	5.00	4.75	4.89	5.26	5.70	5.90	5.86	5.87	4.59	4.92	4.92	5.00	5.02
Subset 4																				
Baseline	5.96	5.96	5.96	5.96	5.96	4.65	4.65	4.65	4.65	4.65	6.39	6.39	6.39	6.39	6.39	5.24	5.24	5.24	5.24	5.24
LM	5.04	5.46	5.93	6.09	6.55	4.21	4.53	4.81	5.10	5.26	5.35	5.93	6.35	6.47	7.15	4.57	4.75	5.04	5.48	6.14
RF	5.03	5.42	5.67	5.67	5.64	4.09	4.06	4.15	4.18	4.23	5.21	5.72	6.09	5.96	5.84	4.32	4.56	4.62	4.60	4.99
SVR	4.94	5.36	5.63	5.58	6.04	4.39	4.63	4.89	4.51	4.97	5.05	5.56	5.75	5.78	5.77	4.53	4.78	4.80	4.89	5.06

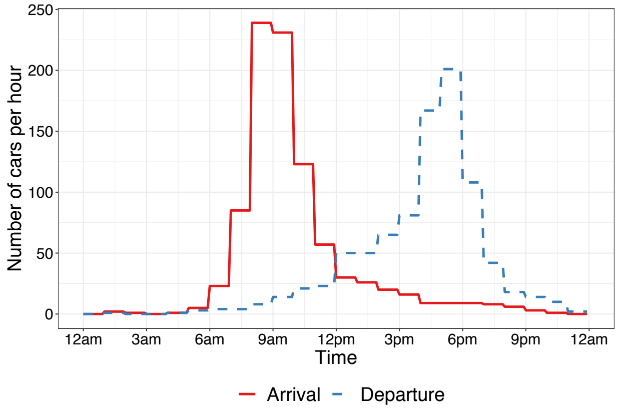


Fig. 19. Hourly rate of arrival and departure on 11 Feb 2019.

i to state j. For Poisson arrival and departure rate of λ and μ, the Q-matrix is represented by:

$$Q = \begin{pmatrix} -\lambda & \lambda & & & \\ \mu & -(\lambda+\mu) & \lambda & & \\ & \ddots & \ddots & \ddots & \\ & & \mu & -(\lambda+\mu) & \lambda \\ & & & \mu & -\mu \end{pmatrix} \tag{2}$$

Kolmogorov backward equations are used to characterize stochastic processes. In particular, they describe how the transition probability $P(t)$ that a stochastic process is in a certain state changes over time by differential equations [38]:

$$P'(t) = P(t)Q \tag{3}$$

The solution of the differential equations is given by:

$$P(t) = e^{Qt} = \sum_{n=0}^{\infty} \frac{(Qt)^n}{n!} \tag{4}$$

In order to compute an estimate of the solution above (infinite sum of matrix powers), we use Krylov subspace projection method [39] which is capable of coping with sparse matrices of very large dimension. The probability of the Markov process being at state n can be obtained by $\pi(t) = \pi(0)P(t)$ where $\pi(0)$ is the initial state vector.

Since a daily profile of transition rates is a step function (piece-wise constants) of hourly slots, as shown in Fig. 19, our matrices Q and P become time-varying and hence are represented by Q_k and P_k, with $k \in \mathbb{Z} : 0 \leq k \leq 23$, each corresponding to an hour-slot. Note that, we aim to track the dynamics of state probabilities over time during the forecast horizon. We first discretize time into fixed-size epochs, δt, each of 5 minute duration. We update fine-grained real-time state probabilities after every epoch δt (probability matrix P_k remains consistent across all epochs of each hour-slot) by:

$$\pi(t_k + n\delta t) = \pi(t_k + (n-1)\delta t)P_k \tag{5}$$

where $n \in \mathbb{Z} : 1 \leq n \leq 12$ for 12 epochs (each 5-min) in an hour-slot, and t_k is the timestamp at the beginning of hour-slot k.

To quantify car park user experience, we consider the number of rejected users, *i.e.,* those who cannot find a parking spot since their allocated partition is full at their time of arrival. In our Markov model, we compute the number of rejections after each epoch time by:

$$r(t_k + n\delta t) = \pi_C(t_k + (n-1)\delta t) \times \lambda(k) \times \delta t \tag{6}$$

where $\pi_C(t_k + (n-1)\delta t)$ is the probability of the car park being full at time $t_k + (n-1)\delta t$ within the hour-slot k and $\lambda(k)$ is the predicted arrival rate over that hour-slot. We accumulate rejected cars per each epoch across a day, and feed the daily count of rejected cars into our optimization formulation, explained next.

B. Optimization Formulation

The goal of our optimization is to select the best partitioning scheme (*i.e.,* what fraction of capacity to allocate to car sharing services) that can minimize the cost of rejected users, while maintaining a certain level of revenue from space leased to

TABLE V
CONSTANT PARAMETERS OF OUR OPTIMIZATION SCHEME.

Parameter	Value	Description
M	\$15.8 /car /day	Renting price per car space, computed from annual market rate for city of Sydney [40].
W^{SV}	\$15.8 /car	Opportunity cost of rejecting a shared-vehicle user is equal to the lease price per space.
W^{PV}	\$26 /car	Opportunity cost of rejecting a private-vehicle user is equal to day permit rate for parking on campus [41].
R	\$36468.75 /week	Minimum revenue to cover the fixed cost of operating a Space Sharing Scheme, including planning & administration (0.5 FTE staff at \$100,000 annual rate) + system installation and maintenance (\$1,900/space/annum) + 5% safety margin.

car sharing companies. We formulate an optimization problem over a period of D days.

Let there be P capacity partitioning schemes and D number of days within the scope of our optimization. The number of rejected shared-vehicle and private-vehicle users for selecting scheme j on day i are respectively denoted by r_{ij}^{SV} and r_{ij}^{PV}, where $1 \leq j \leq P$ and $1 \leq i \leq D$. Further, each scheme is identified by a number of spaces allocated to car sharing users, this number is denoted by s_j.

Our decision variable is denoted by x_{ij}, indicating if scheme j is selected on day i. This can be represented by the following equation:

$$x_{ij} = \begin{cases} 1 & \text{if } \textit{scheme } j \textit{ is selected on day } i \\ 0 & \text{if } \textit{otherwise} \end{cases} \tag{7}$$

We assign a constant cost for each rejected customer as W^{SV} for shared vehicles and W^{PV} for private vehicles. Since our aim is to minimize the total cost of space allocation across D days, the objective function can be written as:

$$\min \quad \sum_{i=1}^{D}\sum_{j=1}^{P}\left\{x_{ij} \times (W^{SV} r_{ij}^{SV} + W^{PV} r_{ij}^{PV})\right\} \tag{8}$$

On a given day, one partitioning scheme will be selected, and hence the following constraint:

$$\sum_{j=1}^{P} x_{ij} = 1 \;\forall\; i \tag{9}$$

Furthermore, campus managers would expect a minimum amount of revenue to be generated by leasing out parking spaces to car sharing companies in order to cover the investment made for implementing such an allocation scheme. Given a daily dollar price M for each space leased out and the minimum revenue R over D days expected by the University, the revenue constraint is given by:

$$\sum_{i=1}^{D}\sum_{j=1}^{P} M x_{ij} s_j > R \tag{10}$$

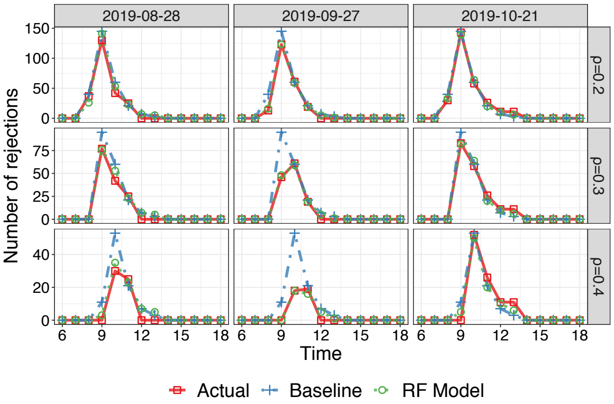


Fig. 20. Time-trace of hourly rejections across three representative days for tighter allocation policies ($\rho = 0.2$, $\rho = 0.3$, and $\rho = 0.4$) by which PV rejections may occur.

We set our optimization period D to 5 working days (1 week) as we can obtain a fairly accurate prediction of demand at this forecast horizon in §V. We use a set of constant parameters in our optimization that are summarized in Table V. The renting price (for car sharing companies) of each parking space is estimated by converting the annual market price for car space lease in city of Sydney [40] to a daily rate. The cost of rejecting a shared vehicle user is considered to be equal to opportunity cost of losing the rent per car space, and the cost of rejecting a private user is equal to daily parking fee on our university campus [41]. Also, the revenue from parking space lease (R) is expected to at least cover the investment cost of operating a space sharing scheme. This includes (a) cost of planning & administration: 0.5 full-time equivalent (FTE) staff at rate $\$100,000$ per annum, (b) cost of system installation and maintenance: $\$1,900$ per annum per car space, and (c) 5% safety margin.

C. Evaluation Results

To simulate the dynamics of shared vehicles, we assume that 20% of the existing users subscribe and use shared-vehicles (SV) service on a regular basis and 80% of the current users continue using their private vehicles (PV) to commute to the University. Furthermore, car sharing have been found to serve other uses in addition to daily work commute, including shopping, business, and leisure [42]. Hence, we further assume additional demand of SV users by considering a constant usage profile sampled from a binomial distribution with hourly λ rate of $B(200, 0.5)$ and hourly μ rate of $B(200, 0.4)$. This corresponds to an assumption of 200 new subscribers with 0.5 and 0.4 probability of entering and exiting the car park in each 1-hour slot.

Since we defined the scope of our allocation scheme as 1-week (equivalent to 5 working days), the 5-day ahead demand forecast obtained from the best performing model (RF) from Section V, are used in our optimization formulation, where partitioning of spaces is done dynamically. In addition, the results are compared with static partitioning using the forecast demand from the baseline prediction model, where arrival and departure profile is constant for each day.

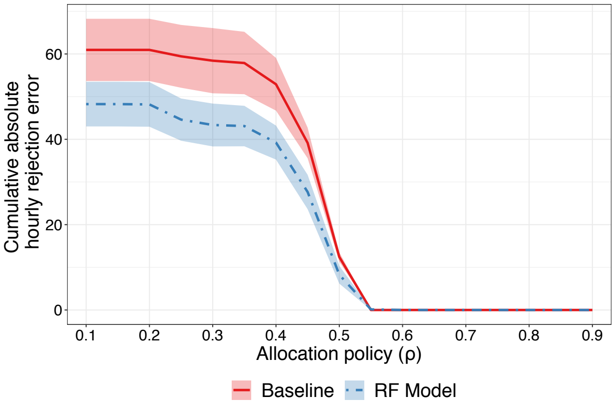


Fig. 21. Sum of absolute hourly rejection errors per day (average) as a function of allocation policy ρ.

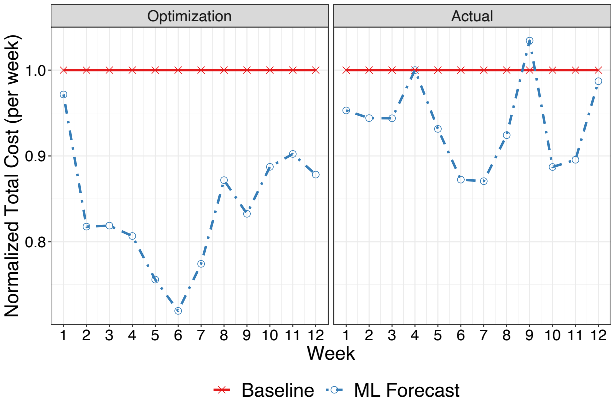


Fig. 22. Weekly (normalized) cost: dynamic (ML forecasting) versus static (baseline forecasting) partitioning.

Using the aforementioned assumptions and data, we first evaluate the accuracy of our Markov model in estimating the number of rejection per each partitioning scheme. We then use the computed rejection number to find the optimal space partitioning scheme by applying the optimization formulation from §VI-B and compare rejection cost incurred from both static and dynamic partitioning.

1) Evaluation of Markov Model: We evaluate the performance of our Markov model in estimating the total number of rejected PV users per day by comparing the output from Markov model to the actual rejected number calculated from the actual arrival and departure count (obtained from sensor data).

Fig. 20 compares hourly rejection number computed from baseline count data (blue line) and forecast count data from RF model (green line) with the actual rejections (red line). The results are shown across 3 different capacity partitioning schemes (row wise) where ρ indicates the fraction of parking spaces allocated to PV users, for 3 example days. From observations, we can see that for $\rho = 0.2$, predictions from baseline and RF model can give a decent estimate of the rejections number with all the three lines aligning closely. On the other hand, for $\rho = 0.4$, we can clearly see that rejection profile calculated using the baseline model stays constant across the 3 example days due to its static arrival and departure rate profile, while the actual rejection trend has changed across the days. This illustrates an example where the superior predictive power from RF model for usage demand can lead to a better approximation of rejection numbers.

In order to quantity the outcome of the Markov modeling, we accumulate the absolute hourly error of rejections across each day, and obtain the daily average error. The average daily sum of errors (PV user case) for both baseline and RF model with 95% confidence interval of mean (area bounding the line) are shown in Fig. 21 for different capacity partitioning schemes (ρ). Undoubtedly, the errors decrease as the fraction of spaces allocated to PV users increases due to the lower number of rejected users. From the graph, we can see that the error becomes zero when ρ reaches 0.55 as there are sufficient spaces allocated to accommodate the PV users demand. By comparing the two lines, it is evident that the rejection error is lower for RF model (blue line) when compared with baseline model (red line) across all partitioning schemes. The highest average daily errors for RF model is 48.25 cars while baseline model 60.95 when $\rho = 0.1$.

2) Evaluation of Partitioning Scheme Optimization: We employed Mixed Integer Linear Programming (MILP) algorithm, which is conducive for a problem that has a linear objective function subjected to a set of linear constraints, to solve our optimization problem described in Section VI-B. We calculated the total cost incurred from the optimum allocation scheme across each week for the two demand prediction: one using the predicted demand from the baseline mean model (static partitioning) and another using the 5-day ahead forecast from RF model (dynamic partitioning). In addition to the minimum cost obtained from the optimum allocation, we calculated the actual cost incurred if the optimum partitioning scheme was selected.

Fig. 22 compares weekly total costs for the baseline (red) and ML forecast (blue) approaches - the costs were normalized with respect to the baseline case. The left plot represents the expected cost from the optimum partitioning scheme, with the actual cost incurred shown on the right plot. By looking at the minimum cost obtained from our optimization (left plot), it is apparent that the ML forecast approach yields lower cost compared to the baseline approach. This suggests that the variation in usage demand, which can be captured through ML forecasting, can provide flexibility in allocating the partitioning scheme and hence yield a lower optimum cost.

When examining the actual cost incurred from the selected partitioning scheme (right plot), we can see that there is a smaller gap between the ML and baseline approaches, indicating lower actual cost savings than expected. In particular during Week 9, the cost saving from dynamic partitioning using ML forecast is expected to be 16.74% but the actual incurred cost is higher by 3.43%. The reasons for this discrepancy is due to errors from predicting parking demand (arrival and departure rate) which can be attributed to the inaccuracy in approximating rejections. For instance, we have one particular day during week 9 where the ML predicted the rejections to be significantly greater than reality, leading the optimization to select an alternate partitioning scheme that is not the optimum choice in reality and hence magnifying the actual objective function cost.

The results of our analysis show that dynamic dimensioning of car park spaces can provide a better user experience when compared to the static partitioning approach, with an average of 6.3% lower weekly customer rejections cost. Our proposed framework shows a potential benefit of dynamic allocation for future car park dimensioning which can be adopted by universities with an aim to optimize the usage of their parking facilities while keeping up with the evolution of shared mobility market.

VII. Conclusion

Digital transformation in transport industry demands large organizations like universities to revisit the operation of their expensive on-campus parking facilities. With the rapid prevalence of IoT technologies, universities can benefit from the immense amount of data generated from smart devices, assisting them with planning and restructuring the usage of their parking facility. In this paper we have outlined our experiences in designing and deploying a monitoring system for a real car park on our university campus. We collected data over 15 months (covering both teaching and non-teaching periods) and cleaned it for analysis. We then analyzed the usage data and highlighted insights into car arrival and departure patterns as well as users parking behavior. Finally, we developed a novel framework for the university to optimally partition their parking infrastructure through dynamically allocating a fraction of parking spaces to car sharing operators. We showed that the University can reduce the cost of customer rejections by 6.3% per week when adopting dynamic allocation using predicted arrival and departure rates.

References

[1] T. Sutjarittham, G. Chen, H. Habibi Gharakheili, V. Sivaraman, and S. S. Kanhere, “Measuring and Modeling Car Park Usage: Lessons Learned from a Campus Field-Trial,” in *Proc. IEEE WoWMoM*, Washington DC, USA, June 2019, pp. 1–10.

[2] B. Council, “The shape of things to come: higher education global trends and emerging opportunities to 2020,” British Council, Tech. Rep., Jan. 2012.

[3] L. dell’Olio, R. Cordera, A. Ibeas, R. Barreda, B. Alonso, and J. L. Moura, “A methodology based on parking policy to promote sustainable mobility in college campuses,” *Transport Policy*, vol. 80, pp. 148–156, Aug. 2019.

[4] A. Filipovitch and E. F. Boamah, “A systems model for achieving optimum parking efficiency on campus: The case of minnesota state university,” *Transport Policy*, vol. 45, pp. 86–98, Jan. 2016.

[5] G. Pierce, H. Willson, and D. Shoup, “Optimizing the use of public garages: Pricing parking by demand,” *Transport Policy*, vol. 44, pp. 89–95, 2015.

[6] (2017) SFpark. Accessed on: 2020-09-17. [Online]. Available: http://sfpark.org/

[7] T. Litman, *Parking management: strategies, evaluation and planning*. Victoria Transport Policy Institute Victoria, BC, 2016.

[8] P. W. Wadhwani and P. Saha, “Car Sharing Market Size By Model (P2P, Station-Based, Free-Floating), By Business Model (Round Trip, One Way), By Application (Business, Private), Industry Analysis Report, Regional Outlook, Application Potential, Price Trend, Competitive Market Share & Forecast, 2020 – 2026,” Global Market Insights, Market Report, Apr. 2020. [Online]. Available: https://www.gminsights.com/industry-analysis/carsharing-market

[9] T. H. Stasko, A. B. Buck, and H. O. Gao, “Carsharing in a university setting: Impacts on vehicle ownership, parking demand, and mobility in ithaca, ny,” *Transport Policy*, vol. 30, pp. 262–268, 2013.

[10] S. A. Shaheen, N. D. Chan, and H. Micheaux, “One-way carsharing’s evolution and operator perspectives from the americas,” *Transportation*, vol. 42, no. 3, pp. 519–536, 2015.

[11] D. C. Shoup, “The high cost of free parking,” *Journal of planning education and research*, vol. 17, no. 1, pp. 3–20, 1997.

[12] E. Polycarpou, L. Lambrinos, and E. Protopapadakis, “Smart parking solutions for urban areas,” in *Proc. IEEE WoWMoM*, Madrid, Spain, June 2013, pp. 1–6.

[13] D. Bong, K. Ting, and K. Lai, “Integrated approach in the design of car park occupancy information system (COINS),” *IAENG International Journal of Computer Science*, vol. 35, no. 1, Feb. 2008.

[14] Z. Pala and N. Inanc, “Smart parking applications using rfid technology,” in *Proc. IEEE Annual RFID Eurasia*, Istanbul, Turkey, Sep 2007, pp. 1–3.

[15] M. Caliskan *et al.*, “Predicting parking lot occupancy in vehicular ad hoc networks,” in *Proc. IEEE Vehicular Technology Conference*, Dublin, Ireland, Apr. 2007, pp. 277–281.

[16] A. Klappenecker, H. Lee, and J. L. Welch, “Finding available parking spaces made easy,” *Ad Hoc Networks*, vol. 12, pp. 243–249, Jan. 2014.

[17] R. Lu, X. Lin, H. Zhu, and X. Shen, “An intelligent secure and privacy-preserving parking scheme through vehicular communications,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 2772–2785, July 2010.

[18] G. Yan, W. Yang, D. B. Rawat, and S. Olariu, “Smartparking: A secure and intelligent parking system,” *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 1, pp. 18–30, Apr 2011.

[19] M. R. Burns and D. J. Faurot, “An econometric forecasting model of revenues from urban parking facilities,” *Journal of Economics and Business*, vol. 44, no. 2, pp. 143–150, May 1992.

[20] Y. Zheng, S. Rajasegarar, and C. Leckie, “Parking availability prediction for sensor-enabled car parks in smart cities,” in *Proc. IEEE ISSNIP*, Singapore, Singapore, May 2015.

[21] T. Rajabioun and P. A. Ioannou, “On-street and off-street parking availability prediction using multivariate spatiotemporal models,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2913–2924, May 2015.

[22] E. I. Vlahogianni, K. Kepaptsoglou, V. Tsetsos, and M. G. Karlaftis, “A real-time parking prediction system for smart cities,” *Journal of Intelligent Transportation Systems*, vol. 20, no. 2, pp. 192–204, June 2016.

[23] W. Shao, Y. Zhang, B. Guo, K. Qin, J. Chan, and F. D. Salim, “Parking availability prediction with long short term memory model,” in *Int. Conf. Green, Pervasive, and Cloud Computing*. Springer, May 2018, pp. 124–137.

[24] A. Camero, J. Toutouh, D. H. Stolfi, and E. Alba, “Evolutionary deep learning for car park occupancy prediction in smart cities,” in *Int. Conf. Learning and Intelligent Optimization*. Springer, 2018, pp. 386–401.

[25] T. Sutjarittham, H. Habibi Gharakheili, S. S. Kanhere, and V. Sivaraman, “Realizing a Smart University Campus: Vision, Architecture, and Implementation,” in *Proc. IEEE ANTS*, Indore, India, Dec 2018.

[26] (2019) ANPR Access HD - HD license plate camera for vehicle access control. [Online]. Available: https://bit.ly/2HixxU5

[27] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, “Automatic license plate recognition,” *IEEE Trans. Intell. Transp. Syst*, vol. 5, no. 1, pp. 42–53, Mar. 2004.

[28] V. Lyons, *Guidance on ANPR Performance Assessment and Optimisation*, 1st ed., Home Office of UK government, Centre for Applied Science and Technology, Sandridge, St Albans, AL49HQ United Kingdom, Mar. 2014.

[29] S. Takahashi and T. Izumi, “Travel Time Measurement by Vehicle Sequence Matching Method - Evaluation Method of Vehicle Sequence using Levenshtein Distance,” in *Proc. SICE-ICASE*, Busan, Korea, Oct. 2006.

[30] T. Rajabioun, B. Foster, and P. Ioannou, “Intelligent parking assist,” in *Proc. IEEE Mediterranean Conference on Control and Automation*, Chania, Greece, June 2013.

[31] (2019) Maximum weekly hours. Australian Government Fair Work Ombudsman. Accessed on: 2020-07-08. [Online]. Available: https://bit.ly/1I14M9L

[32] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[33] A. S. Wilkins, “To lag or not to lag?: Re-evaluating the use of lagged dependent variables in regression analysis,” *Political Science Research and Methods*, vol. 6, no. 2, pp. 393–411, 2018.

[34] G. Bontempi, S. B. Taieb, and Y.-A. Le Borgne, “Machine learning strategies for time series forecasting,” in *European business intelligence summer school*. Springer, 2012, pp. 62–77.

[35] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse, “Methodology for long-term prediction of time series,” *Neurocomputing*, vol. 70, no. 16-18, pp. 2861–2869, 2007.

[36] G. Chevillon, “Direct multi-step estimation and forecasting,” *Journal of Economic Surveys*, vol. 21, no. 4, pp. 746–785, 2007.

[37] L. Zhang, W.-D. Zhou, P.-C. Chang, J.-W. Yang, and F.-Z. Li, “Iterated time series prediction with multiple support vector regression models,” *Neurocomputing*, vol. 99, pp. 411–422, 2013.

[38] W. J. Anderson, *Continuous-time Markov chains: An applications-oriented approach*. Springer Science & Business Media, 2012.

[39] R. B. Sidje, “Expokit: A software package for computing matrix exponentials,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 24, no. 1, pp. 130–156, 1998.

[40] “The impact of car share services in australia,” Phillip Boyle & Associates, Tech. Rep., 2016.

[41] Unsw estate management parking rates 2020. Accessed on: 2020-06-16. [Online]. Available: https://www.estate.unsw.edu.au/getting-here/parking/parking-rates-2020

[42] H. Becker, F. Ciari, and K. W. Axhausen, “Comparing car-sharing schemes in switzerland: User groups and usage patterns,” *Transportation Research Part A: Policy and Practice*, vol. 97, pp. 17–29, 2017.

Thanchanok Sutjarittham received her B.Eng. degree in Electrical Engineering and Telecommunications from UNSW Sydney in 2016. She is currently pursuing her Ph.D. in Electrical Engineering at UNSW. Her primary research interests include Internet of Things, sensor data analytics, and applied machine learning.

Hassan Habibi Gharakheili received his B.Sc. and M.Sc. degrees of Electrical Engineering from the Sharif University of Technology in Tehran, Iran in 2001 and 2004 respectively, and his Ph.D. in Electrical Engineering and Telecommunications from UNSW in Sydney, Australia in 2015. He is currently a Senior Lecturer at UNSW Sydney. His current research interests include programmable networks, learning-based networked systems, and data analytics in computer systems.

Salil S. Kanhere (Senior Member, IEEE) received the M.A. and Ph.D. degrees from Drexel University, Philadelphia, PA, USA.

He is a Professor of Computer Science and Engineering with UNSW Sydney, Sydney, NSW, Australia. His research interests include Internet of Things, blockchain, pervasive computing, cybersecurity, and applied machine learning.

Prof. Kanhere serves as the Editor in Chief for *Ad Hoc Networks* and as an Associate Editor for IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, *Computer Communications*, and *Pervasive and Mobile Computing*. He has served on the organizing committee of many IEEE/ACM international conferences, including PerCom, CPS-IOT Week, MobiSys, WoWMoM, MSWiM, and ICBC. He was awarded the Friedrich Wilhelm Bessel Research Award (2020) and Humboldt Research Fellowship (2014) from the Alexander von Humboldt Foundation in Germany. Salil is a Senior Member of the ACM and an ACM Distinguished Speaker.

Vijay Sivaraman received his B. Tech. from the Indian Institute of Technology in Delhi, India, in 1994, his M.S. from North Carolina State University in 1996, and his Ph.D. from the University of California at Los Angeles in 2000. He has worked at Bell-Labs as a student Fellow, in a silicon valley start-up manufacturing optical switch-routers, and as a Senior Research Engineer at the CSIRO in Australia. He is currently a Professor at the University of New South Wales in Sydney, Australia. His research interests include Software Defined Networking, network architectures, and cyber-security particularly for IoT networks.