Experiences with IoT and AI in a Smart Campus for Optimizing Classroom Usage

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

***Abstract*—Increasing demand for university education is putting pressure on campuses to make better use of their real-estate resources. Evidence indicates that enrollments are rising, yet attendance is falling due to diverse demands on student time and easy access to online content. This paper outlines our efforts to address classroom under-utilization in a real University campus arising from the gap between enrollment and attendance. We do so by instrumenting classrooms with IoT sensors to measure real-time usage, using AI to predict attendance, and performing optimal allocation of rooms to courses so as to minimize space wastage.**

Our first contribution undertakes an evaluation of several IoT sensing approaches for measuring class occupancy, and comparing them in terms of cost, accuracy, privacy, and ease of deployment/operation. Our second contribution instruments 9 lecture halls of varying capacity across campus, collects and cleans live occupancy data spanning about 250 courses over two sessions, and draws insights into attendance patterns, including identification of canceled lectures and class tests, while also releasing our data openly to the public. Our third contribution is to use AI techniques for predicting classroom attendance, applying them to real data, and accurately predicting future attendance with an RMSE error as low as 0.16. Our final contribution is to develop an optimal allocation of classes to rooms based on predicting attendance rather than enrollment, resulting in over 10% savings in room costs with very low risk of room overflows.

***Index Terms*—IoT, smart campus, classroom occupancy, AI, prediction.**

I. Introduction

HIGHER education institutes continue to experience steady growth in enrollment demand [2]. A major factor limiting universities in fulfilling this demand is real-estate, since enrollment in a course is capped by the capacity of the classroom to which the course is allocated. However, with recent trends towards student lifestyles that mix study with work and other commitments, as well as greater access to online content, there is ample anecdotal evidence that classroom attendance is often well below the enrollment number. This presents an opportunity for education institutes to better optimize the usage of classroom space based on attendance rather than enrollments. Since class attendance can vary significantly between courses and across weeks of semester, visibility into actual class attendance and ability to predict future attendance based on historical data are needed to dynamically re-allocate courses to rooms while minimizing risk of overcrowded lecture rooms where class attendance exceeds room capacity.

Several methods are available to count the number of people in an indoor space, such as WiFi-based approach [3], camera image processing, thermal imaging, ultrasound imaging, and beam counters affixed to entryways [1]. Each method has its own pros and cons across various dimensions such as cost, power, communications, ease of deployment and operations, privacy, and accuracy. For example, using WiFi data and cameras endanger privacy, thermal and ultrasound imaging have low accuracy, and camera-based image processing is computationally expensive. Furthermore, a method that works well in a small room may not be as effective in a larger lecture theater, and cost/accuracy may also be impacted by the layout of the room, the number/width of doorways, and the availability of power and wired/wireless network connections. Hence understanding both benefits and challenges of various approaches in order to adopt the most suitable methods for the nature of the room is important for the real deployment of classroom occupancy monitoring system.

This paper describes our experiences in adopting IoT to measure and AI to predict the attendance of lectures in courses at our University campus, and to use these to optimize the usage of lecture rooms. Our specific contributions are four-fold:

1) We begin by testing several sensing methods in a lab environment and characterizing their trade-offs in aspects such as cost, ease of installation, method of data extraction, privacy, and accuracy.
2) We then make appropriate sensor selections, build a full system, and deploy it across 9 lecture theaters of varying size across the university campus. We collect and clean the data to obtain visibility of occupancy across these rooms in real-time over a period of 18 weeks (*i.e.,* a full semester in 2017 and half a semester in 2018), integrate it with University timetabling data to infer attendance patterns of over 250 courses, and highlight interesting findings such as attendance trends, canceled lectures, and class tests. We also make our occupancy data openly available to the research community.
3) We develop machine-learning models to predict classroom attendance using three algorithms namely multiple regression, random forest, and support vector regression

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 ACM/IEEE IPSN 2018 conference [1].

(SVR). We employ quantile regression technique, allowing asymmetric penalties for under-prediction and over-prediction of attendance. Our models are able to predict attendance in advance with a root-mean-square error (RMSE) of less than 0.16. We also make our attendance dataset openly available to the research community.

4) Finally, we develop an optimization algorithm for allocating classes to rooms based on predicted attendance rather than static enrollments, and show potential saving of over 10% in room costs.

The rest of this paper is organized as follows: §II describes relevant prior work. We present our lab evaluation of various sensing methods and their trade-offs in §III, while §IV describes our field deployment across campus and interesting insights obtained therein. We present our techniques for predicting classroom attendance in §V. Our optimization formulation for dynamic classroom allocation is described in §VI, and the paper is concluded in §VII.

II. Related Work

Occupancy Counting: Various approaches have been proposed in the literature to measure occupancy. Many studies utilized special-purpose sensors to infer occupancy level of a given space – a work in [4] used a network of sensors to obtain various environmental parameters such as CO_2, carbon monoxide (CO), total volatile organic compounds (TVOC), acoustics, motion, temperature, and humidity, to derive occupancy count in an open office space using machine learning techniques. The method achieved an average accuracy of 73%, however it had only been tested in a space that accommodates only less than 10 people at a time. Works presented in [5], [6], [7] have also used indoor environmental sensors in combination with supervised learning methods to infer occupancy and achieved good accuracy results. Nevertheless, none of these studies have evaluated their methods in a larger room scenario where over 100 occupants can be accommodated. This is important for our case as typical lecture room size at a university can ranged from less than 100 to nearly 500 in capacity.

In addition, several studies have used video camera based approaches for people counting. The methods described in [8], [9] achieved good accuracy of result, but they rely on complex image processing algorithms which require significant computational resources. The authors in [10] successfully used image processing and Support Vector Machine (SVM) to measure classroom occupancy in large rooms with more than 100 occupants. However, their method only produces accurate results when there is minimal movement in the classroom. Privacy also remains an issue, especially if images and videos of people are taken without their explicit consent.

Another approach to deduce occupancy count is to leverage information from existing WiFi access points (AP) infrastructure where network connection parameters, such as connections count and received signal strength indicator (RSSI), are used to infer occupancy [11], [12], [13]. The advantage of this approach over the others is the fact that no additional hardware is required. However, there is a number of factors that may impact accuracy of the count. For instance, people who do not carry WiFi enabled devices would not get counted, individuals with multiple devices (e.g., laptop and phone) would get counted twice, and people outside the room may be connected to the AP inside the room and get included in the room occupancy count. Furthermore, obtaining WiFi connectivity data may also constitute a violation of privacy if the identities of connected users can be deduced.

Predicting Occupancy: Some studies have attempted to perform future forecasting of occupany from historical data. The authors in [14] compared two approaches of occupancy estimation based on indoor climate parameters and 3D stereo-vision camera. Both approaches were tested in 2 rooms (*i.e.,* a classroom and a study zone), and the camera-based approach was shown to outperform the indoor climate-based approach. Historical occupancy estimation with additional contextual features including day name, day type (weekday or week end), season, and holiday (binary) were used to perform future count prediction using decision tree and random forest. The prediction achieved the best accuracy of 3% error, yet it has to be noted that the models had only been trained using 20 days data with 1 day test data (using sliding window method). Moreover, for classroom occupancy, the dynamic variation of class attendance, which is likely to be influenced by factors such as courses and weeks of semester, were not captured by the models. In [15], the authors collected occupancy data at a commercial space using depth sensors (Kinect for XBOX One) for a duration of 9 months, future occupancy prediction was performed using historical data, however it had suffered from a very high error of up to 2100 %.

Classroom Scheduling and Allocations: The majority of existing works primarily use occupancy monitoring to improve energy efficiency of heating, ventilation, and air conditioning (HVAC) systems. To the best of our knowledge, there is no work on the application of occupancy measuring in dynamic allocation of classrooms.

The problem of course timetabling and classroom allocation have been studied extensively in the past where a variety of constraints, such as timing requirement of the events and lecturer availabilities were captured. Several optimization algorithms were employed to solve the problem, some of the popular ones include genetic algorithm [16], simulated annealing [17], and tabu search [18]. However, the number of students accommodated in the classroom were based on enrollment numbers rather than actual attendances, leading to a significant under utilization of room spaces in real-world scenarios.

We believe that our work is the first that combines occupancy monitoring system to solve class allocation problem where courses can be allocated to classrooms based on their predicted attendance rather than the traditional enrollment information.

III. Sensing Classroom Occupancy

In this section, we describe various sensing methods for counting people, outline their relative trade-offs with a view towards making appropriate selections suitable for a larger-scale deployment across the campus, and briefly explain our

system architecture for collecting, cleansing, and visualizing sensing data.

A. People Counting Methods

Sensors: We investigated several commercial sensors and straight-away eliminated those that send data to the vendor's cloud servers, since we wanted to: (a) keep the data entirely on-premises and not risk it leaving our campus infrastructure; and (b) not be beholden to a vendor to access our own data, hence freeing us from ongoing service costs. In other words, we wanted a "sale" model of the device so we could have unfettered access to our data without any ongoing "service" fees. We were quite happy to buy spares of the units to cover for device failures; further, this model allows us to integrate data into a centralized repository to facilitate better analytics across the many data feeds we have on campus.

We narrowed our lab trials to four types of commercial sensors: EvolvePlus Wireless Beam Counter [19], EvolvePlus Overhead Camera [20], Steinel HPD Camera (pre-market release), and Steinel Presence Detector [21]. In addition, the University IT department provided us with timestamped connections logs from two WiFi access points (one inside our lab and one just outside), so we could compare our approaches to those obtained from WiFi logs. We note that the WiFi logs gave us personal user information such as their device MAC address, user-ID, and connection durations; we therefore obtained ethics clearance (UNSW Human Research Ethics Advisory Panel approval number HC17140) for this experiment.

The **Beam Counter** comprises a pair of infrared (IR) break-beam sensors mounted on the door frame, and counts the number of people passing through in each direction. It communicates the counts (for "in" and "out" directions) to a gateway every 30 seconds using a propriety wireless protocol, and the gateway then posts these readings via Ethernet to an SQL database (DB) server hosted on a VM in our on-premises cloud infrastructure. The **Overhead Camera** is a thermal sensor mounted on the ceiling close to the entrance facing downwards, and counts the number of people passing below it. It also communicates the counts in each direction to the same gateway as the beam counter, which then forwards it on to the SQL DB. We wrote a script that pulls data from the SQL DB, stamps the data with the time and the unique UUID of the gateway, and posts as a JSON string to our master database (which holds data from many sources) via a REST API.

The **HPD Camera** (pre-market release) is a people counting sensor mounted in a corner with full view of the room. It uses built-in image processing to compute the number of people present within a configurable zone of interest. It is powered over Ethernet, and comes pre-configured with a server that be queried via a REST API. We wrote a "broker" script that polls the camera every 30 seconds to get the people count, and posts the time-stamped and sensor UUID-stamped data in JSON format to our master database. The **Presence Detector** is a passive infrared (PIR) sensor mounted on the ceiling in the middle of the room, and detects motion. Though it does not count the number of people in a room, it gives a binary indication on whether the room is occupied or not – this sensor can be used as a way to calibrate the other counting sensors which may accumulate errors with time. The PIR sensor sends its binary occupancy state every 60 seconds to its corresponding gateway via a propriety wireless interface, which then posts it to a broker script that again time- and sensor-UUID-stamps the data and posts to our master database.

Lastly, we receive a CSV file of daily WiFi connection logs for the two access points from our IT department every morning at 7am – real-time feed of data was not possible due to technical limitations of the AP vendor. We wrote a script to parse the log file and compute the number of unique users connected to each AP every 30 seconds – this was also posted to our master database.

With possibility of sourcing data from various sensing devices, one may want to perform sensor fusion for an accurate occupancy measurement. At a very minimum, a combination of PIR sensor and passing people counters (*i.e.,* beam counter and overhead camera) seems reasonable. PIR sensors are fairly accurate in detecting whether a room is empty which can be useful for resetting the errors accumulated over time via the people counting sensors. It is important to note that detecting presence is not a trivial task for a large lecture theater due to limited coverage of PIR sensors, and thus configuring non-overlapping zones for multiple units of PIR sensors can be quite challenging. For a second step of fusion, adding WiFi data or HPD camera would help infer an accurate occupancy since these methods measure occupants count instantaneously without keeping states (*i.e.,* not cumulative). But, as explained next, these sensors come with their own shortcomings. We note that deploying a collection of sensors at the scale of a university campus can significantly increase the cost. Therefore, our primary focus in this paper is to select and deploy one sensor type for each classroom, and demonstrate its value in optimal allocation of rooms to courses.

B. Sensor Evaluation and Selection

Our lab trial helped us compare the various counting methods in terms of their ease of installation, calibration, power and communications requirements, accuracy, cost, and privacy, as summarized in Table I.

Our comparison across these measures is qualitative rather than quantitative. Even aspects such as accuracy, that can be quantified, depend on factors like room size and layout, mounting position, number of doors, and width of doorways, which can vary widely across deployment environments. We therefore resort to qualitative measures (low, medium, and high) in this table, derived from our experience across the rooms we instrumented, and we back these up with several data points presented later in the paper.

Installation: The thermal camera, HPD camera, and PIR sensor needed professional installation by certified tradesmen, since each needed special mounting brackets and extra wiring for mounting on (or near) the ceiling. We could install the beam counter sensor easily by ourselves using two-sided adhesive strips on the door frame at around waist-height.

TABLE I
SENSORS COMPARISON

	Installation	Calibration	Power	Communications	Accuracy	Cost	Privacy
Beam counter	easy	easy	battery	wireless	high	medium	high
Thermal sensor	hard	medium	AC	wireless	low	high	high
HPD Camera	medium	hard	PoE	Ethernet	medium	medium	medium
PIR Sensor	hard	easy	AC	wireless	binary	medium	high
WiFi Data	existing	existing	existing	existing	existing	existing	low

Calibration and Positioning: Sensor positioning is another key factor in our comparison. The thermal camera needs to be positioned at a certain height range (i.e. 2.2m - 4.4m) recommended by the manufacturer and close to the entrance allowing the best coverage to count everyone that passes underneath. This requirement makes it hard or impossible to use the thermal camera in very large lecture halls with high ceilings. Beam counters require to be mounted at around waist-height (too low causes each leg to get counted separately, and too high causes the swinging arms to get counted!). Once an appropriate height is chosen for the beam counters, doors of all classrooms need to be outfitted in the same way. The HPD camera needs prior configuration for zone of interest that can vary across rooms depending on the room size and the place at which the camera is mounted. The PIR sensor is positioned at the center of the room (on the ceiling) to have a symmetrical coverage over an area that can also vary across rooms depending on their seating arrangement.

Power and Communications: Provisioning power was challenging for the thermal camera and PIR sensor, since the campus has pre-built and fixed wiring only in certain locations in each classroom. Therefore our Facilities Management was required to supply new exterior wiring for these three sensors. The beam counters are battery powered (with stated battery life in excess of a year), and the HPD camera required a special PoE switch that provides Ethernet for both power and communications. The corresponding gateways for the beam counter, thermal camera, and PIR sensor were hidden inside a closet with available power and Ethernet.

Accuracy: We performed several spot measurements in our lab to extract ground truth on occupancy. We found that the beam counter is the most accurate among the four techniques. We note that the beam counter has very good accuracy when the the door is narrow, like in our lab. However, for a wider doorway its accuracy is worse, since it does not always capture individuals walking in/out side-by-side (this became more evident in our field-trial, described in the next section). We found the accuracy of the thermal camera to be very sensitive to mounting position and distance from the entrance. Moreover, since the door of our lab opens inwards, it was not very conducive for the overhead thermal camera (mounting it on the outside of the room was not an option as it was a busy corridor). The HPD camera tended to have a non-zero absolute count error, which made its relative error high when the number of people in a room is small (e.g. less than 10) and low when the number of people is high (e.g. more than 40). We could not test its accuracy scaling to larger counts as our lab can only accommodate around 40 people. Lastly, the people count derived from the WiFi access points was wildly inaccurate, because our lab is adjacent to a busy corridor and study space that is busy with students during regular hours, and we could not distinguish who was inside versus outside the room.

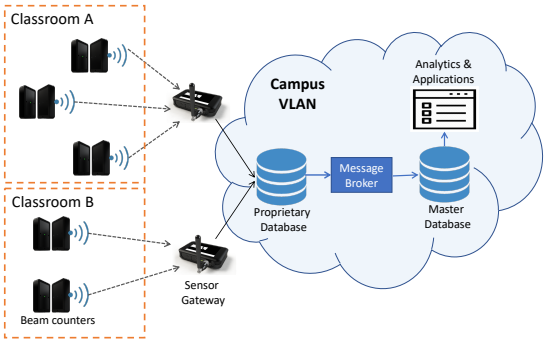

Fig. 1. System architecture of classroom occupancy monitoring.

Cost: The beam sensors and PIR sensors are priced in the range of a few hundreds of dollars, while the cameras are in excess of a thousand. The beam counter and thermal camera both need a gateway to send their readings to the back-end server, and each gateway is priced in at nearly a thousand dollars. Bear in mind that each gateway can connect up to 20 sensors (though our deployment described in the next section maps at most 4-5 sensors to a gateway in large lecture theaters). The beam counters therefore end up as a more cost-effective solution than the cameras for large-scale deployment across campus.

Privacy: Among the four sensing techniques, WiFi clearly endangers students privacy as their IDs are visible (due to PEAP authentication their devices perform to connect with the campus WiFi network). The HPD camera does on-board processing and does not store or transmit any images of people (though it is possible to log in to it to view the current image), and can hence be deemed to preserve privacy. The beam counter and the thermal camera are truly privacy-preserving, since they can only sense the number of people passing through the doorways without sensing any private attributes of the individuals.

Summary: The trade-offs discussed above are summarized in Table I. WiFi is not an option as it compromises privacy and is inaccurate. The cameras are eliminated as being expensive, difficult to install/position, and poor in accuracy (though we are considering them for open spaces that do not have doorways). The PIR sensor has only binary output (*i.e.,* 0 for unoccupied and 1 for occupied), and is used for re-calibration rather than counting. We therefore decided on a larger-scale deployment of the beam counter, based on its relatively lower cost, easy deployment, high accuracy, and good protection of privacy. Our deployment in classrooms is described next.

TABLE II
MEASURED ERROR FROM GROUND-TRUTH OF OCCUPANCY.

Room	#seats	#doors	Error	
			Room-based	Course-based
BUS105	35	1	27.7 %	13.0 %
BUS115	53	1	34.2 %	17.3 %
CLB7	497	4	89.5%	4.6 %
CLB8	231	3	26.3 %	16.1 %
MAT A	472	6	25.5 %	8.0 %
MAT B	246	3	6.3 %	9.1 %
MAT C	110	2	14.6 %	24.4%
MAT D	110	2	16.9 %	9.2 %
PhTh	369	4	NA	NA

C. System Architecture and Data Collection

Figure 1 shows a high-level system architecture of the classroom occupancy monitoring system using beam counters. First the beam counters communicate their count data to sensor gateways that are installed in each room via a proprietary wireless protocol. The gateway is directly connected to an Ethernet port which has been provisioned to allow connection to our university private VLAN, where the data is being stored in a proprietary database. We then wrote a message broker script to unify the data format into a commonly agreed structure, where the sensor data is converted into a JSON string as well as being tagged with timestamp and sensor UUID, before getting posted to our master database via a REST API. This provides the feasibility for the platform to collect data from various sensors which generate heterogeneous data format.

Similarly, the retrieval of data for analysis or as an input to applications can be easily performed through a GET RESTful API. This raw beam counter data only contains timestamped count in and count out from each sensor installed at the doorway, hence data preprocessing is required to infer room occupancy and finally class attendance number.

IV. Data Processing and Visualization

We worked with campus staff to identify appropriate classrooms for a field trial, and picked 9 rooms of varying size, as shown in the first column in Table II. Some of the doorways to the lecture-halls posed a challenge as they were very wide, increasing the likelihood that multiple students walking out side-by-side get counted as one. The data collected over the first few days was manually verified (volunteers were used to do head-counts) so as to obtain ground-truth and calibrate the errors. In what follows we describe our methods for data cleansing, linking with class-timetabling information, processing, and visualization using a web-UI.

A. Occupancy and Attendance Calculation

We compare two methods of data processing to deduce the occupancy from the number of entries and exits at each door:

Method 1: Room-based: Our first (naive) method for deriving occupancy is to set it to the cumulative number of entries minus the cumulative number of exits across all doorways of a classroom. However, errors arise when students walk in/out in groups; though we reset counts to zero at midnight each day, errors accumulating during the day can become significant.

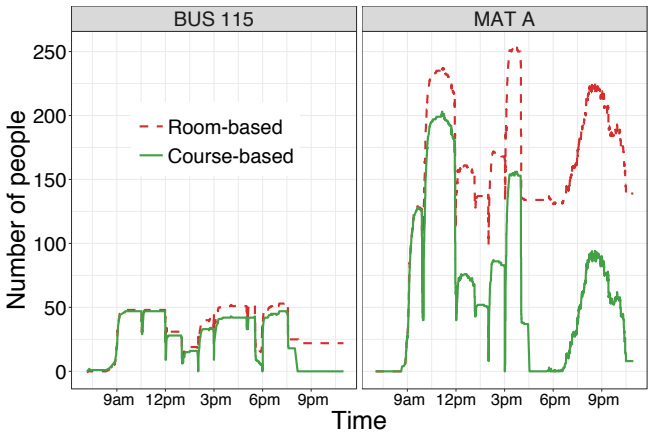

Fig. 2. Error of room-based method is higher than course-based and increases with the size of classroom; 'BUS 115': small room and 'MAT A': large theater.

Method 2: Course-based: To reduce the errors accumulating during the day, we enhance our method by computing course attendance independent of each other by linking our sensor data with course timetable databases obtained from our University. We assume that students may enter the room up to 10 minutes prior to start of the scheduled lecture time, and may leave up to 10 minutes after the scheduled lecture time. Attributing each entry and exit to a specific lecture therefore allows us to compute attendance per-course, and errors are not carried over from one lecture to the next even if they are adjacent in time to each other.

Accuracy of Counting: To evaluate the accuracy of our counting methods, we obtained ground-truth information by having volunteers physically count attendance during the lectures. We collected a total of 50 samples covering 31 lectures over 4 days. The ground-truth samples were collected from 8 out of 9 classrooms in which the sensors have been deployed. Table II shows the average error of the computed occupancy using the two methods described above applied to the various rooms. As expected, course-based occupancy computation yields lower errors (average: 12.71%) compared to room-based occupancy computation (average error: 30.60%). This is because the room-based method gradually built-up errors over the course of a day, whereas the course-based method had a stable error irrespective of time-of-day (since the errors do not accumulate). However, it should be noted that the course-based method requires access to timetabling information, which may not be generalizable to other environments.

Typological Analysis of Error in Calculating Occupancy: We now analyze the impact of room characteristic (*i.e.,* size and number of doorways) on the accuracy of estimating occupancy. Fig. 2 shows the occupancy computed by room-based and course-based methods, for two representative rooms (one small classroom 'BUS 115' with 53 seats and one large lecture theater 'MAT A' with 472 seats) on 9-August-2018. Our first observation is that the room-based method, shown by dashed red lines, results in higher residuals (*i.e.,* accumulated error in calculated occupancy) than the course-based method, shown by solid green lines, at the end of the day. Additionally,

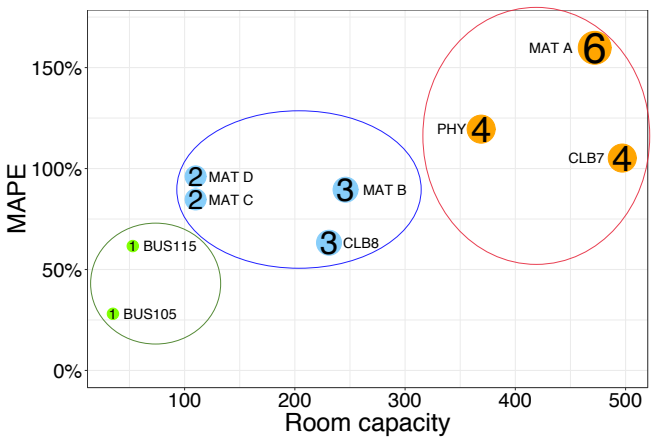

Fig. 3. The mean absolute percentage error (MAPE) versus room size, highlighting three clusters.

it is seen that the gap between room-based and course-based is dilated in larger classrooms – for example a significant gap of more than 100 people is observed in 'MAT A' (on the right plot), compared to a relatively smaller gap in 'BUS 115' (on the left plot).

For our analysis, we use deployed sensors data (without ground-truth) collected in entire Semester 2, 2018 to investigate the impact of the two key factors (*i.e.,* rooms size and number of doorways). We employ the mean absolute percentage error (MAPE) metric computed by the average of absolute difference between room-based and course-based divided by course-based occupancy (*i.e.,* ratio of the gap between green and red lines in Fig. 2 to green lines). We illustrate in Fig. 3 the value of MAPE as a function of room characteristics. Note that x-axis is the room capacity and each circle on the plot shows the number of doorways for the corresponding classroom – size of each circle is proportional to doors count. We observe that the MAPE is positively correlated with both the room capacity and the number of doorways, highlighted by three clusters namely small rooms with less than 100 seats and one door (green circles), medium-size rooms with 100 to 300 seats and 2 to 3 doors (blue circles), and large theaters with more than 300 seats and more than 4 doors (orange circles). We note that for larger classrooms, the chance of students walking in/out side-by-side is relatively higher (such instances are usually counted as a single individual by the beam counters) which results in a larger MAPE value.

Occupancy and Attendance data: Our weekly occupancy dataset, computed using the Method-2 above, is openly available for download [22]. Each row in a CSV file represents the real-time measurement from beam sensors comprising *time-stamp*, *week* of semester, room information including *room name*, *number of doorways*, and *number of seats*, course information including *course-id* (we have intentionally obfuscated the actual names of courses), *course start-time*, and *course end-time*, sensor measurements including *count-in*, *count-out*, and computed number of attendance (i.e. *occupancy*). Note that count-in and count-out are available for the entire day (even during times with no lectures scheduled), whereas occupancy is available only when a course is scheduled.

Additionally, we release class attendance dataset [23] which will be used for prediction in §V. We derive class attendance by taking the maximum value of occupancy count in the room during the period when the class is operational.

B. Data Visualization

Tool: To provide an intuitive user interface (UI) for real-time occupancy monitoring, we developed a web application using R Shiny – our tool is available at [24]. The tool allows the user to view the attendance pattern of a course (by choosing from the course dashboard tab), as well as the utilization rate (number of attendees divided by the total number seats available for each classroom) for different time-slots.

Insights: Our UI provides some interesting insights into attendance patterns. Fig. 4 shows our UI output for an occupancy pattern of a selected room (CLB8) on a selected day, comparing the number of attendees (red line) and the associated enrollments (blue line) for 7 courses scheduled between 9am-9pm. From the plot, attendance is seen to vary widely across courses, in the range of 10% to over 90% of the enrollments. Interestingly, we observe that the lecture scheduled between 1pm-2pm has an enrollment of 211 but close to zero attendance; this indicates the cancellation of lecture which has led to a wastage of room spaces on the day. The visibility of room occupancy monitoring allows us to quantify space utilization that is otherwise largely unknown to facility managers.

Our visualization tool also provides visibility into attendance pattern of all courses scheduled in the classrooms where sensors were installed. Figure 5 shows an example of attendance patterns for 3 selected courses (we have obfuscated course names) across the whole semester from week 1 to week 12. We can observe some interesting trends such as a general decline in attendance over week, represented in blue line; class cancellation on week 7, represented in green line; and mid session class examination during week 6 (red line), which has been verified by looking up the course web page.

Furthermore, the tool allows us to generate a room utilization heat-map on a chosen day, as shown in Figure 6. Bright yellow cells represent time slots where rooms are being mostly occupied with utilization rate approaching 1, whereas color scale towards dark blue represents time slots when rooms are under utilized. On the web interface, hovering over a cell shows further details on the usage of classroom and the scheduled course. For instance, we can see from the plot that course `df0c74e1e5` has an enrollment number of 193 but only 10 students attending the lecture, leading to a poor utilization of 4.1%. The interface allows campus managers to track classroom utilization, with a view towards more optimal allocation, as described in §VI.

V. Prediction of Classroom Attendance

In previous section, we observed that the falling attendance pattern results in underutilization of classrooms. In order to improve the overall utilization, universities may want

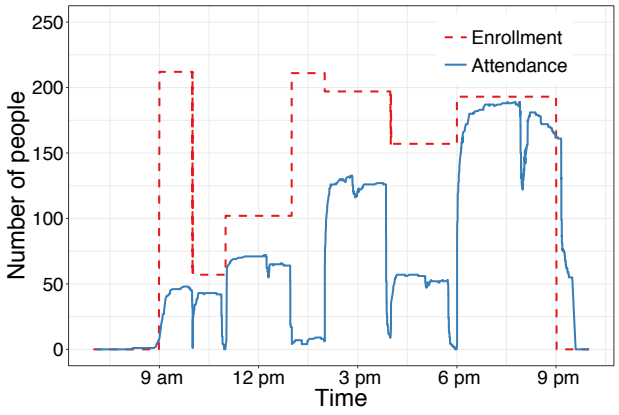

Fig. 4. Occupancy pattern of a classroom on 16 August 2017.

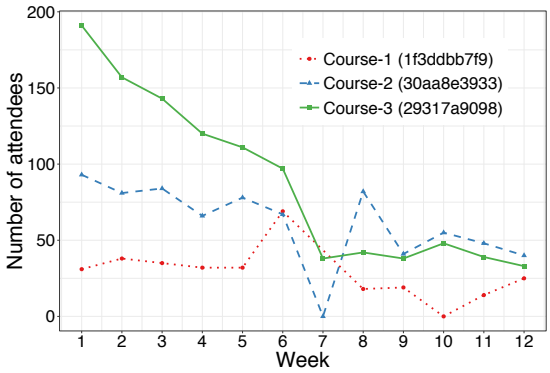

Fig. 5. Attendance pattern of three courses across weeks.

to dynamically re-allocate classrooms in advance based on attendance – if attendance is much lower than the number of enrollments of a class then a smaller room may potentially be allocated to it, thus saving cost. We note that while our system is typically useful to obtain real-time data, room scheduling is based on the predicted attendance from historical data and should be decided several weeks prior to actual classes. This entails a fairly accurate prediction of attendance for all classes operated on campus. It is important to note that underestimated prediction may lead to class overflow (*i.e.,* significant discomfort for students), and overestimation would lead to wasted capacity and thus not achieving optimal cost reductions.

In this section, we compare three learning-based algorithms in predicting class attendance, each using two different functions of regression. We train models using historical labeled dataset from semester 2, 2017 and test the models with attendance dataset from semester 2, 2018. Note that it is infeasible to perform spot measurement at scale to collect ground-truth data needed for training of our prediction models. We, therefore, use course-based attendance count (which is deduced from data measured by sensors) to generate supervised learning models. The error in sensor measurement and thereafter in prediction, will be considered in §VI when the predicted attendance is used for dynamic allocation of classrooms.

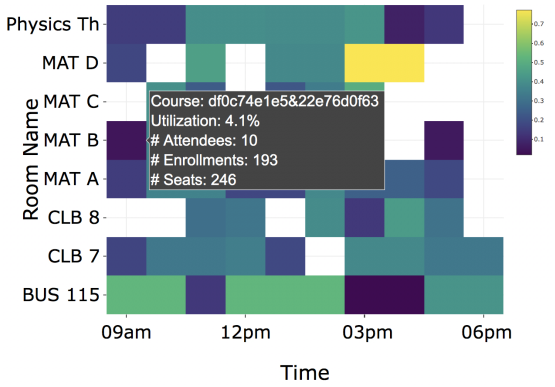

Fig. 6. Heat-map of classrooms occupancy on a chosen day.

A. Attributes Impacting Attendance

There are several factors that can influence class attendance in universities, especially student motivations, quality of teaching, and characteristics of class lectures [25]. However it is infeasible to measure students motivation without conducting extensive surveys of a large population of students. Further, quality of teaching largely depends on the course lecturer, which can vary from semester to semester. We instead consider attributes related to individual course (*e.g.,* engineering faculty, undergraduate, tutorial) and temporal aspects (*e.g.,* week 3, Monday) which can be readily obtained.

In order to generalize our prediction model, we do not use specific course attributes such as course code or course instructor as inputs to our model. This allows us to perform prediction for courses for which no training data is available. Note that since our data collection started from semester 2 in 2017, there are insufficient data samples that span across multiple "semester" and "year" combinations. Thus, we do not include these attributes for our prediction model in this paper – even though they may have impacts on class attendance.

In summary, the following attributes are used in our prediction models:

- ***class type****:* type of the class, *e.g.,* lecture, lab, and tutorial.
- ***faculty****:* faculty the course belongs to, *e.g.,* engineering, medicine, and science.
- ***school****:* school the course belongs to (*e.g.,* Material Science).
- ***enrollment****:* number of students enrolled in the course.
- ***course duration****:* duration of the course (in hours).
- ***degree****:* degree of study (*e.g.,* undergraduate, postgraduate).
- ***course status****:* enrollment status of the course (*e.g.,* open, full).
- ***joint****:* a binary indicating if the course is combined with other courses.
- ***week****:* week of semester (*e.g.,* week 3).
- ***day****:* day of week (*e.g.,* Monday).
- ***time-of-day****:* categorical value of time the class begins (*e.g.,* morning: 9am-12pm, afternoon: 12pm-3pm, evening: 3pm-6pm, and night: 6pm-9pm).

For the output of our prediction model, we use normalized attendance which is the ratio of maximum classroom occupancy (from course-based method in §IV-A) to the enrollment

TABLE III
SUMMARY OF DATASETS FOR TRAINING AND TESTING.

Dataset	Description	Sample Size
DS1	Sem2, 2017 - train set	1497
DS2	Sem2, 2017 - test set	639
DS3	Sem2, 2018 - test set	940

count. Hence the output varies between 0 (no student attended, or class cancellation) and 1 (all enrolled students attended).

Prior to generating and testing a model, we cleansed our data instances. This process involved removing classes that have no attendance (or canceled class), removing classes with excessive overflow (*i.e.,* normalized attendance more than 1.5) due to probably over-counting, and capping the normalized attendance to 1. Our attendance dataset with corresponding attributes for both training (*i.e.,* semester 2, 2017) and testing (*i.e.,* semester 2, 2018) is openly available for download [23]. Each row in a CSV file represents a class comprising all attributes described above along with the actual attendance number obtained from the occupancy sensors data (*i.e.,* the attendance field is not normalized in our released dataset).

B. Prediction Modeling

We choose supervised machine learning algorithms to perform attendance prediction given our labeled dataset from 2017. We considered three common regression learning algorithms including multiple linear regression, random forest, and support vector regression. For each of these algorithms we apply two regression functions namely ordinary least square and quantile regression. The models are trained using caret package [26] in R [27].

1) Algorithms: We first explain the algorithms used to build our prediction models:

Multiple Regression: Multiple linear regression (MLR) is one of the simplest prediction algorithms. It is the most common form of linear regression where the value of a variable is predicted based on the value of two or more attributes. The algorithm finds the best fit for the training data by minimizing the sum of the squares of residues to obtain the resulting model.

Random Forest: Random forests (RF) is an ensemble learning method that can be used for both classification and regression problems. During training, multiple decisions trees are created from the training set, where a random selection of features is used to split each node of a tree. The final prediction result is obtained from majority voting, where mean is used for regression trees [28].

Support Vector Regression: Support vector regression (SVR) applies the same principles as support vector machine (SVM) to the data but for a regression problem. The algorithm involves transforming the training data into a higher dimensional feature space, where linear regression is performed with a tolerance margin provided by the boundary lines [29]. In our model, radial basis kernel is used to map data to higher dimensions.

2) Regression Loss Function: Machine learning algorithms attempt to minimize a loss function on the training data as part of their modeling process. Commonly used loss functions for regression models, *i.e.,* mean square error (MSE) and mean absolute error (MAE), are symmetric. They treat error of under-prediction and over-prediction equally. This may not be desirable for class attendance estimation as under-prediction (*i.e.,* class allocated to a room with capacity lower than actual attendance) is more harmful than over-prediction since it leads to students' discomfort.

To address unequal treatment of errors (*i.e.,* differentiating under-prediction and over-prediction), we employ quantile loss function [30] for generating prediction models. Quantile loss function is defined by:

$$L(y, \hat{y}) = \sum_{i=1}^{n} (y_i - \hat{y_i})(\tau - \mathbb{1}_{y_i - \hat{y_i} < 0}) \quad (1)$$

where y is the actual attendance and $\hat{y}$ is the predicted attendance, $\mathbb{1}$ is an indicator function that equals to 1 in case of over-prediction and 0 otherwise, quantile τ is a tuning parameter that takes a value from 0 to 1 allowing us to ascribe varying emphasis on over/under-prediction (*i.e.,* $\tau < 0.5$ causes over-prediction to impose higher cost than under-prediction while $\tau > 0.5$ increases the cost of under-prediction). Our choice of quantile τ will be explained next.

TABLE IV
SUMMARY OF COURSES PER FACULTY.

Faculty	Count
Faculty of Science	698
UNSW Business School	662
Faculty of Engineering	436
Faculty of Arts & Social Science	203
Faculty of Medicine	97
Faculty of Built Environment	22
Faculty of Engineering & Science	18

C. Performance Evaluation of Prediction Models

We build our prediction models using the three learning algorithms mentioned earlier and consider both ordinary regression and quantile regression as loss functions for each, thus a total of 6 models are evaluated. We use 10-fold cross validation to select the best tuning parameters that yield the lowest error for each model. This subsection explains the dataset we used for model training and testing. We also compare the prediction results of the six models using a range of metrics.

1) Dataset: We partitioned the attendance data from entire semester 2 of 2017 into a train set and test set at a ratio of 70% (*i.e.,* DS1) and 30% (*i.e.,* DS2) respectively. Additionally, we obtained attendance data for the first 6 weeks from semester 2 of 2018 (*i.e.,* DS3) and used it as another test set to evaluate the performance of the models in predicting future attendance. A summary of our datasets along with their sample sizes is shown in Table III. We note that only data from semester 2 (and not semester 1) was used for the prediction, eliminating any biases that could arise from the effect of semester on our attendance prediction. Additionally, a summary of courses per various faculties is shown in Table IV. It is also important

TABLE V
PERFORMANCE OF PREDICTION MODELS.

Models	2017 train set DS1 (cross-validation)			2017 test set DS2 (testing)			2018 test set DS3 (testing)		
	RMSE	MAE	WMAE	RMSE	MAE	WMAE	RMSE	MAE	WMAE
Multiple Linear Regression (MLR)	0.163	0.123	0.060	0.149	0.118	0.060	0.193	0.145	0.063
Random Forest (RF)	0.120	0.086	0.043	0.121	0.089	0.044	0.157	0.122	0.051
Support Vector Regression (SVR)	0.135	0.094	0.041	0.125	0.086	0.042	0.193	0.148	0.062
Quantile Linear Regression	0.188	0.142	0.048	0.179	0.135	0.045	0.240	0.183	0.059
Quantile Regression Forests	0.147	0.102	0.033	0.154	0.108	0.034	0.217	0.167	**0.047**
Quantile Regression using SVM	0.141	0.095	0.035	0.147	0.100	0.036	0.216	0.170	0.061

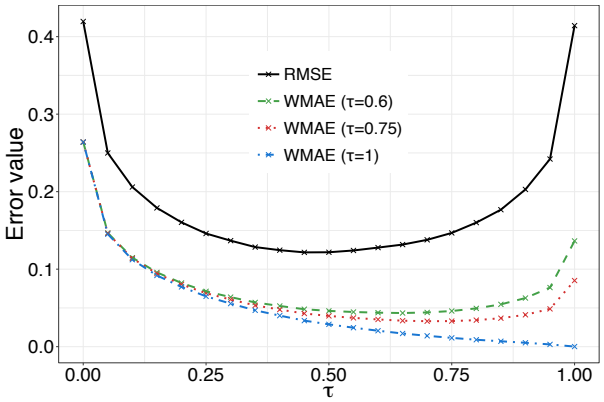

Fig. 7. Impact of quantile parameter on prediction error.

to note that not all courses are run every semester and the number of enrollment for courses that run every semester may vary vastly (for example the course "Computer Networks and Applications" in our university may have around 300 students in semester 1, but over 600 students in semester 2).

2) Evaluation metrics: Both mean absolute error (MAE) and root mean squared error (RMSE) measure the average magnitude of the errors in a set of predictions. RMSE is most useful when large errors are particularly undesirable since errors are squared, giving a relatively higher weight to larger errors, before being averaged. Additionally, we employ the average weighted absolute error (WMAE) [31] to evaluate the performance of prediction when under-prediction is considered more costly than over-prediction. Our custom WMAE is defined as the mean of the quantile loss function and is written as:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})(\tau - \mathbb{1}_{y_i - \hat{y_i} < 0}) \tag{2}$$

3) Impact of Quantile Parameter: As noted in §V-B, we consider a quantile loss function to differentiate under-prediction and over-prediction. We now build quantile regression models by varying the value of quantile parameter τ in order to quantify its impact on performance metrics. Fig. 7 shows error values of RMSE and WMAE versus the quantile parameter for a random-forest-based model as an example. Note that for WMAE, the τ value is fixed to 0.60, 0.75, and 1.00 as shown by dashed green, dotted red, and dashed dotted blue lines respectively. As expected, it is seen that RMSE values, shown by solid black lines, produce a symmetrical curve with a central dividing line at the second quantile (*i.e.,* $\tau = 0.50$), as this metric treats under-prediction and over-prediction equally. On the other hand, we observe asymmetric curves for WMAE that have the minimum error values at its corresponding τ (*e.g.,* WMAE with $\tau = 0.75$, shown by dotted red lines, gets minimum at 0.75 quantile).

As mentioned earlier, the quantile parameter τ with a value greater than 0.50 will penalize under-prediction more than over-prediction. We note that for $\tau = 1$, only under-estimation is penalized in the prediction process with no penalty for over-estimation. This may provide a very safe margin for campus managers during dynamic classroom allocation. But, this choice yields a high RMSE of over 0.4 which makes it unattractive. On the other hand, $\tau = 0.5$ gives the minimum RMSE but it weights over-prediction and under-prediction equally which does not match our requirement. Hence, we choose a quantile value of 0.75, which still results in a reasonable value of RMSE (*i.e.,* below 0.2) while giving more weight to under-prediction. This value is used as the quantile parameter for both loss function and WMAE performance metric.

Attendance Prediction within Semester: We first apply our prediction models to a testing set containing attendance data of classes from the same semester/year (*i.e.,* semester 2, 2017). For this, we use DS1 to train the model and DS2 for testing. The performance results of our six models are shown in Table V. In terms of RMSE, we can see that RF and SVR algorithms achieve the best predictive performance in both cross-validation and testing, with values within a range of 0.120 and 0.135. Linear models (both MLR and quantile MLR), on the other hand, yield the highest errors of over 0.16 RMSE on the validation set. This suggests that attendance and course attributes are not linearly correlated and such non-linear relationship is better modeled by RF and SVR. By imposing higher cost to under-prediction, we can see that quantile regression methods give better performance in terms of WMAE when compared to their default regression counterparts. RF achieves the best performance with WMAE of 0.033 during cross validation and 0.034 during testing.

To visualize the performance of the six models, we show in Fig. 8 scatter plots of the predicted normalized attendance (x-axis) versus the actual attendance (y-axis) with a blue fitted line of $y = x$ (*i.e.,* predictions are expected to match the actual output) – points on the left side of this line represent under-prediction, and right side points represent over-prediction. We also note that the predicted attendance can sometimes go beyond 1. From the plot we can obviously see that points are more dispersed for linear regression models (red dots) compared to SVR (green dots) and RF (blue dots), which matches the results of RMSE metrics discussed earlier. Furthermore, we observe that quantile regressions (shown by

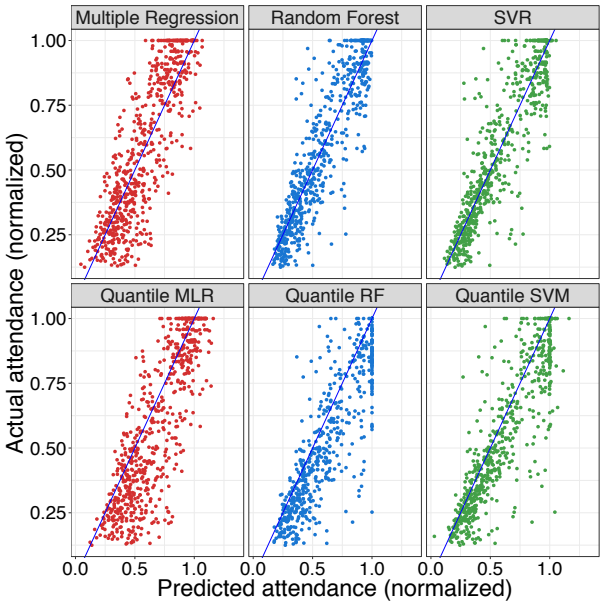

Fig. 8. Predicted vs. actual attendance for test set in 2017 (*i.e.,* DS2).

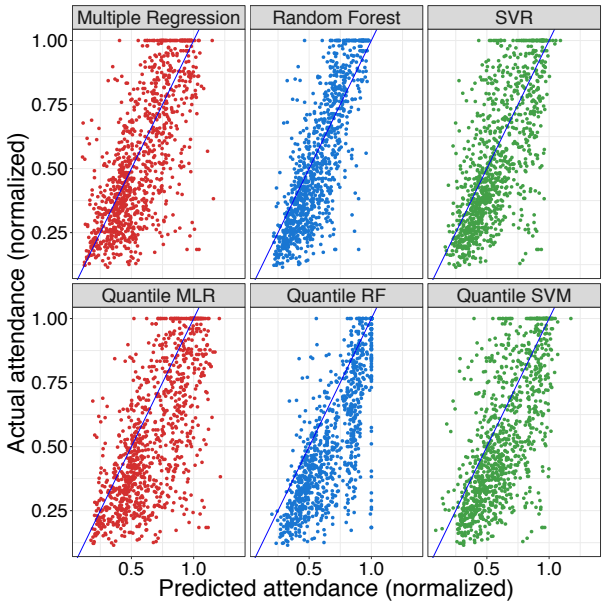

Fig. 9. Predicted vs. actual attendance for test set in 2018 (*i.e.,* DS3).

bottom plots in Fig. 8) result in more over-predictions than under-predictions, which is an expected output.

We note that the prediction result is fairly consistent in cross-validation and testing, indicating that a good prediction can be obtained if part of the data from one semester/year are used to predict the attendance for the remainder of the same semester. A more practical implementation that involves using historical data to predict future year/semester attendance is discussed next.

Attendance Prediction for Future Semester: We now predict the attendance for semester 2, 2018 (*i.e.,* DS3) while the model was trained by the data from the same semester but in 2017 (*i.e.,* DS1). Results of all the models are shown by the last column in Table V. In terms of RMSE, it is seen that RF achieves the best performance with a value of 0.157 – slightly higher than the result from testing DS2. Looking at WMAE, quantile RF give the most accurate result with the error of 0.047 (highlighted in green).

From the predictions vs. actual attendance plot in Fig. 9, as expected, we observe wider dispersions from the fitted line compared to the scatter plots in Fig. 8 for the within-semester attendance prediction. RF and quantile RF (blue dots) show the lowest deviation from the fitted lines compared to predictions from linear regression models (red dots) and SVR models (green dots).

The lower performance in prediction of future-semester attendance is caused by several factors when changing year/semester – for example, variations in enrollments from year to year due to popularity of courses, changes made to the programs (*e.g.,* a certain course may no longer be considered a core course while another course may become a core), changes in lecturing staff, changes in students lifestyle (motivation for attending classes), and increases in availability (or better quality) of online lecture recordings. Despite all these factors, we believe that the future-semester prediction results obtained from quantile RF (with WMAE of 0.047 and RMSE of 0.217) is acceptable given the non-trivial variations mentioned above.

VI. Optimal Allocation of Classrooms

Our empirical results from Section IV highlight the significant variations in occupancy on a weekly basis which leads to under-utilization of classrooms. This presents an opportunity for campus managers to employ a dynamic allocation scheme to save cost. In this section, we develop an optimization formulation to determine the potential cost savings. Our formulation uses prospective attendance counts computed from our prediction model (discussed in §V). A practical implementation could develop a dynamic schedule for a course using our optimization algorithm, while leaving some margin for error arising from the prediction algorithm.

We first formulate a problem with an objective function and related constraints. We then apply the optimization to our dataset and quantify the cost savings versus students discomfort (due to room overflows) when adopting dynamic classroom allocation in comparison to the traditional fixed enrollment-based allocation.

A. Problem Formulation

First, let there be L courses, each with its own start time, duration, and the attendance number. Each course is allocated to one of R classrooms available on campus, and each room has a cost associated with it (proportional to its capacity). The cost of room j is denoted by C_j where $1 \leq j \leq R$. We consider our optimization problem over a one-day window with a total of S slots – each slot accounts for an hour period.

We define our variables to be $x_{i,s}$ that indicates the room to which the course i is allocated during timeslot s, where $1 \leq i \leq L$ and $1 \leq s \leq S$:

$$x_{i,s} \in \{0, 1, 2, ..., R\} \tag{3}$$

Note that $x_{i,s} = 0$ indicates that course i is not allocated to any room during s. From variables $x_{i,s}$, we derive another variable denoted by $y_{i,j}$ which is a binary value indicating whether or not course i is allocated to room j during any given slot:

$$y_{i,j} = \begin{cases} 1 & \text{if } \textit{course } i \textit{ allocated to room } j \\ 0 & \textit{otherwise} \end{cases} \quad (4)$$

Furthermore, room-slot allocation variables, $z_{j,s}$ indicate the number of courses that are allocated to room j at timeslot s:

$$z_{j,s} \in \{0, 1, 2, ..., L\} \quad (5)$$

Therefore, the total cost of allocation for a given day is specified as:

$$J = \sum_{j=1}^{R} \left\{ C_j \sum_{s=1}^{S} z_{j,s} \right\} \quad (6)$$

where C_j is the capacity of room j.

Our aim is to minimize the total cost J in Eq. (6). Note that allocation of a course to a room incurs a full cost of that room, and an unallocated room incurs no cost. Thus, the total cost of the allocation is the sum of capacities of rooms that are used across all timeslots of the day.

We have four sets of constraints in our optimization problem listed as follows.

- *Course Constraint*: Each course can only be allocated to one room during a timeslot. These L constraints are captured by:

$$\sum_{j=1}^{R} y_{i,j} = 1 \ \forall \ i \quad (7)$$

- *Room Constraint*: A room cannot be occupied by more than one course at a time. Hence, Eq. (5) can be expressed as:

$$z_{j,s} \in \{0, 1\} \quad (8)$$

- *Capacity Constraint*: Course i with attendance number of o_i needs to be assigned to room j that has sufficient capacity C_j accommodating students attended. These constraints are captured by:

$$o_i \leq \sum_{j=1}^{R} C_j y_{i,j} \ \forall \ i \quad (9)$$

- *Schedule Constraint*: Each course has a fixed schedule (*e.g.,* course 1 is scheduled from slot 3 to slot 5, equivalent to 11 am to 1pm), thus room allocation can not change over these consecutive slots (*i.e.,* slots 3 to 5 for this example).

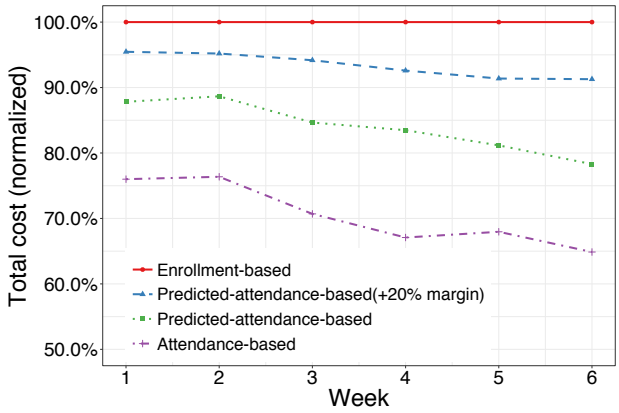

Fig. 10. Allocation cost across weeks in 2018.

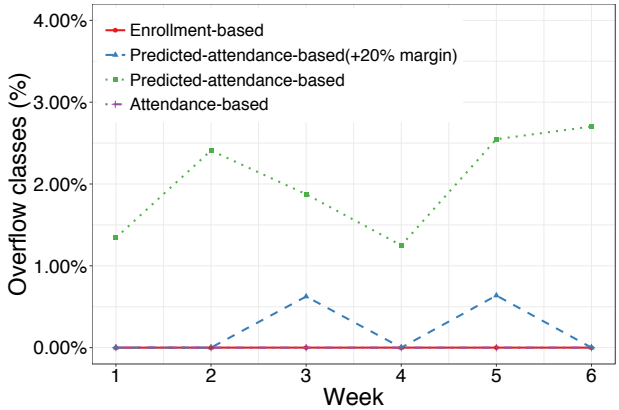

Fig. 11. Students discomfort across weeks in 2018.

B. Optimization Algorithm

To solve the optimization problem, we employed constraint programming (CP) algorithm using Google Optimization Tools for Python [32] as a solver. CP is widely used for solving combinatorial problem, where search space with possible values of variables (domains) are defined and constraints are declaratively stated in order to limit all possible assignments to a set of feasible solutions. There are two phases involved in CP: (a) a propagation phase where infeasible regions are methodically removed from the search space, (b) a search phase where the browsing of the search space is performed using a complete search algorithm such as backtracking, or an incomplete search such as local search algorithm [33]. CP can be used to solve for all feasible solutions, or an near optimal solution. Due to time complexity, we choose the latter option.

For our optimization problem, we used the real data obtained from semester 2, 2018 (*i.e.,* DS3) consisted of 748 courses (as mentioned in §V-C1) operating in the 9 classrooms. We assume that there are some spare rooms available for optimal allocation – we use only one spare room of 100 seats capacity. There are 12 timeslots for each operation day, starting from 9 am (corresponding to timeslot 1) to 9 pm (corresponding to timeslot 12). The optimization was performed for a complete set of courses accounted for a period of 6 weeks. We considered several scenarios, both enrollment-based and attendance-based. For the latter scenario, we use different attendance count including actual attendance (measured by our

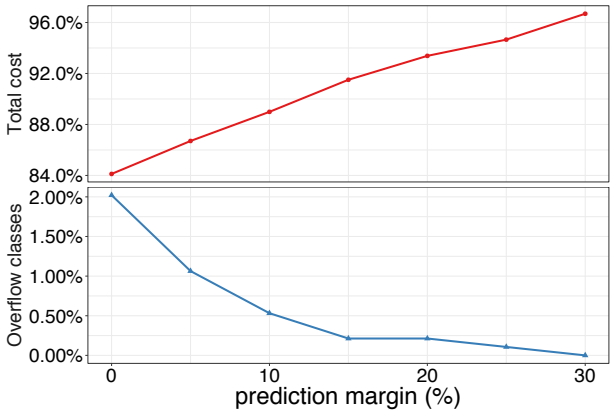

Fig. 12. Allocation cost versus students discomfort.

sensors), predicted attendance, and also predicted attendance with additional margin (*i.e.,* 5%, 10%, 15%, 20%, 25%, and 30%). For each round of our optimization, we obtain the total cost of allocation. Note that for some cases when predicted attendance is used, the classroom may overflow due to under-prediction and cause students discomfort. We, therefore, measure the number of "overflow classes" in each round of allocation.

C. Optimization Results:

We run our optimization algorithm and obtain the cost of allocation for each day. We then compute the weekly cost of allocation by adding daily costs across the week. We plot the weekly total cost in Fig. 10 – total costs are normalized with respect to the enrollment-based scenario as a baseline (solid red line). Unsurprisingly, the enrollment-based approach results a constant cost (*i.e.,* upper bound) as it tries to meet the fixed constraints every week. On the other hand, total cost obtained from various scenarios of attendance-based allocation (dashed lines) falls gradually due to falling pattern of attendance for majority of the courses. We note that even though allocation based on actual attendance yields the best result with lowest cost, it is not feasible in practice. It is seen that the prediction-based allocation yields a cost slightly higher than when actual attendance is considered, but still it is beneficial to the campus by saving on average of 12% per week of operation (as shown by dotted green lines with no margin). Obviously, adding margin to the predicted attendance count would reduce the cost saving, for example 20% margin gives a saving of about 5% per week.

In addition, we capture the number of overflow classes (*i.e.,* the room capacity is lower than the actual attendance of the class) as a proxy for the students experience (or discomfort) from dynamic allocation of classrooms. Fig. 11 shows the fraction of overflow classes (out of all allocated classes) across weeks for various scenarios. As we expect, allocations based on enrollment and actual attendance count yield no overflow since rooms capacities are well provisioned in advance. Allocation based on predicted attendance, on the other hand, causes overflow between 1.25% to 2.70% of classes per week when no margin is considered (as shown by dotted green lines). The measure of overflow is reduced to less than 1% by applying a margin of 20% (as shown by dashed blue lines).

Lastly, we focus on the impact of margin on our evaluation metrics. We plot in Fig. 12 the total normalized cost (top) and the total overflow fraction of classes (bottom) for a duration of 6 weeks by varying the prediction margin from 0% to 30%. We can see that the normalized cost monotonically increases from 84% (with no margin) to 96% (with 30% margin). Conversely, the overflow falls from 2% (accounted for 19 impacted classes) to 0% where no class is impacted. This suggests that campus can benefit from at least 4% cost saving over 6 weeks of operation by employing a dynamic allocation of classrooms with no impact on student experience.

VII. Conclusion

In this paper we have outlined our efforts to address classroom under-utilization in a real University campus arising from the gap between enrollment and attendance. We instrumented classrooms with IoT sensors to measure real-time usage, used AI to predict attendance, and performed optimal allocation of rooms to courses minimizing space wastage. We undertook a lab evaluation of various commercial IoT sensors and compared them in terms of cost, ease of operation, and accuracy. We then deployed our sensors in 9 real classrooms of varying sizes across campus and collected data over two semesters covering 250+ courses, which we release to the public. Our data and visualization reveal interesting insights into course attendance patterns and class utilization measures. Based on this real data, we developed AI based methods to predict classroom attendance which fed into our optimization algorithm for dynamic allocation of classes to classrooms based on predicted attendance rather than enrollments, and showed gains of 10% in room costs with a very low risk of room overflows.

References

[1] T. Sutjarittham *et al.*, "Data-Driven Monitoring and Optimization of Classroom Usage in a Smart Campus," in *Proc. ACM/IEEE IPSN*, Porto, Portugal, 2018.

[2] B. Council, "The shape of things to come: higher education global trends and emerging opportunities to 2020," British Council, Tech. Rep., 01 2012.

[3] I. P. Mohottige *et al.*, "Estimating Room Occupancy in a Smart Campus Using WiFi Soft Sensors," in *Proc. IEEE LCN*, Chicago, USA, October 2018.

[4] B. Dong, B. Andrews, K. P. Lam, M. Hö Ynck, R. Zhang, Y. Chiou, and D. Benitez, "An information technology enabled sustainability test-bed (itest) for occupancy detection through an environmental sensing network," *Energy and Buildings*, vol. 42, no. 7, pp. 1038–1046, July 2010.

[5] A. Dey, X. Ling, A. Syed, Y. Zheng, B. Landowski, D. Anderson, K. Stuart, and M. E. Tolentino, "Namatad: Inferring occupancy from building sensors using machine learning," in *Proc. IEEE WF-IoT*, Reston, VA, USA, 2016.

[6] L. Zimmermann, R. Weigel, and G. Fischer, "Fusion of Non-Intrusive Environmental Sensors for Occupancy Detection in Smart Homes," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2343–2352, 2017.

[7] Z. Yang, N. Li, B. Becerik-Gerber, and M. Orosz, "A multi-sensor based occupancy estimation model for supporting demand driven hvac operations," in *Proc. Symposium on Simulation for Architecture and Urban Design*, 2012.

[8] Y. Hou and G. K. H. Pang, "People Counting and Human Detection in a Challenging Situation," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 41, no. 1, pp. 24–33, January 2011.

[9] J. Li, L. Huang, and C. Liu, "Robust people counting in video surveillance: Dataset and system," in *Proc. IEEE AVSS*, Klagenfurt, Austria, 2011.

[10] F. Paci, D. Brunelli, and L. Benini, "0, 1, 2, many – a classroom occupancy monitoring system for smart public buildings," in *Proc. Conference on Design and Architectures for Signal and Image Processing*, Madrid, Spain, 2014.

[11] G. Fierro, O. Rehmane, A. Krioukov, and D. Culler, "Demo abstract: Zone-level occupancy counting with existing infrastructure," in *Proc. ACM BuildSys*, Toronto, Ontario, Canada, 2012.

[12] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, and Y. Agarwal, "Sentinel: Occupancy Based HVAC Actuation using Existing WiFi Infrastructure within Commercial Buildings," in *Proc. ACM SenSys*, Roma, Italy, 2013.

[13] S. K. Ghai, L. V. Thanayankizil, D. P. Seetharam, and D. Chakraborty, "Occupancy detection in commercial buildings using opportunistic context sources," in *Proc. IEEE PerCom*, Lugano, Switzerland, March 2012.

[14] F. C. Sangogboye, K. Arendt, A. Singh, C. T. Veje, M. B. Kjærgaard, and B. N. Jørgensen, "Performance comparison of occupancy count estimation and prediction with common versus dedicated sensors for building model predictive control," vol. 10, no. 6, pp. 829–843, 2017.

[15] K. S. Liu, J. Francis, C. Shelton, and S. Lin, "Long Term Occupancy Estimation in a Commercial Space : An Empirical Study," in *ACM/IEEE IPSN*, Pittsburgh, USA, April 2017.

[16] E. A. Abdelhalim and G. A. E. Khayat, "A utilization-based genetic algorithm for solving the university timetabling problem (uga)," *Alexandria Engineering Journal*, vol. 55, no. 2, pp. 1395–1409, 2016.

[17] P. Kostuch, "The university course timetabling problem with a three-phase approach," in *International Conference on the Practice and Theory of Automated Timetabling*. Springer, 2004, pp. 109–125.

[18] E. Burke, Y. Bykov, J. Newall, and S. Petrović, "A time-predefined approach to course timetabling," *Yugoslav Journal of Operations Research*, vol. 13, no. 2, pp. 139–151, 2003.

[19] (2017) EvolvePlus: Wireless People Counters with Remote Data Viewing. https://goo.gl/dAUe8G.

[20] (2017) EvolvePlus: Overhead Thermal Counter. https://goo.gl/ugtW1L.

[21] (2017) Steinel Australia: Presence Control PRO IR Quattro HD. https://goo.gl/T8zvNF.

[22] (2018) UNSW Smart Campus - Classroom Usage Monitoring and Optimization. https://smartcampus.unsw.edu.au/roomoccupancy/data.

[23] (2018) UNSW Smart Campus - Classroom Usage Monitoring and Optimization. https://smartcampus.unsw.edu.au/roomoccupancy/data/IoT-J/.

[24] (2018) Visualization of classroom occupancy. https://smartcampus.unsw.edu.au/roomoccupancy/.

[25] S. Devadoss *et al.*, "Evaluation of factors influencing student class attendance and performance," *American Journal of Agricultural Economics*, vol. 78, no. 3, pp. 499–507, 1996.

[26] M. Kuhn *et al.*, "Caret package," *Journal of statistical software*, vol. 28, no. 5, pp. 1–26, 2008.

[27] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013. [Online]. Available: http://www.R-project.org/

[28] L. Breiman *et al.*, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[29] H. Drucker *et al.*, "Support vector regression machines," in *Advances in neural information processing systems*, 1997, pp. 155–161.

[30] R. Koenker and K. F. Hallock, "Quantile regression," *Journal of economic perspectives*, vol. 15, no. 4, pp. 143–156, 2001.

[31] H. Haupt, K. Kagerer, and J. Schnurbus, "Cross-validating fit and predictive accuracy of nonlinear quantile regressions," *Journal of Applied Statistics*, vol. 38, no. 12, pp. 2939–2954, 2011.

[32] (2018) Google optimization tools. [Online]. Available: https://developers.google.com/optimization/

[33] F. Rossi *et al.*, *Handbook of constraint programming*. Elsevier, 2006.

Thanchanok Sutjarittham is currently pursuing her Ph.D. in Electrical Engineering and Telecommunications at the University of New South Wales (UNSW Sydney), where she has also received her B.Eng. in Electrical Engineering and Telecommunications in 2016. 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 lecturer at UNSW Sydney. His current research interests include programmable networks, learning-based networked systems, and data analytics in computer systems.

Salil S. Kanhere received his Ph.D. from Drexel University. He is a Professor in the School of Computer Science and Engineering at UNSW. His research interests include the Internet of Things, pervasive computing, blockchain, crowdsourcing, sensor networks, and security. He has published 200 peer-reviewed articles and delivered over 30 tutorials and keynote talks. He is a Senior Member of ACM. He is a recipient of the Humboldt Research Fellowship.

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 now 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.