International Journal of Information Technology Vol. 28 No. 2 2022

# AI-Driven Crowd Monitoring Platform for COVID-19 Pandemic Response

Hongchao Jiang[a], Wei Yang Bryan Lim[a] and Zhenchao Tao[b]

[a]Alibaba-NTU Singapore Joint Research Institute, Singapore
[b]University of Science and Technology of China, China

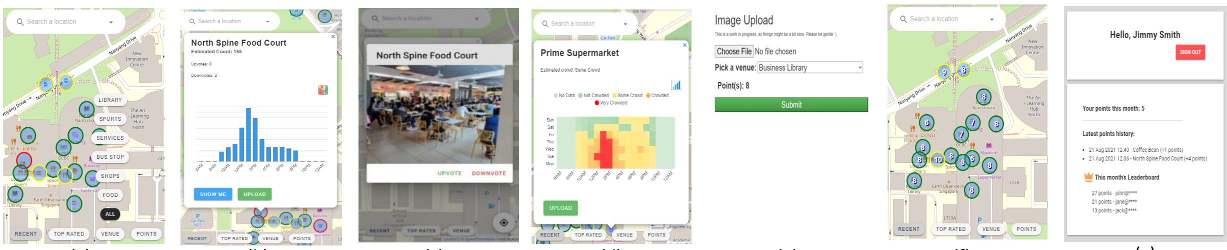
{hongchao001,limw0201}@e.ntu.edu.sg, taozhenchao@sina.com

## Abstract

The COVID-19 pandemic has disrupted the lives of millions across the globe. In Singapore, promoting safe distancing and managing crowds in public areas have been the cornerstone of containing the community spread of the virus. Social distancing requires monitoring of crowd levels for indoor and outdoor points of interest. In this paper, we report our experience developing and deploying an AI-empowered crowd counting platform. The platform relies on crowdsourced images and Unmanned Aerial Vehicles (UAVs) to collect data on the crowd situation. State-of-the-art deep learning algorithms in our AI engine perform analysis on the data to produce up-to-date crowd counts. To encourage user contributions, we deploy a dynamic incentive mechanism scheme. As more historical crowd data is collected, we are able to train models to forecast crowd trends. Initial deployment has been successfully carried out within the Nanyang Technological University (NTU) campus.

## I. Introduction

The Coronavirus Disease 2019 (COVID-19) has disrupted the lives of people all over the world. Governments had to make a tough decision between *lives* and *livelihood* when implementing lockdown measures to restrict movement and reduce social interactions within the community. With the development of vaccines, many countries are easing restrictions and transitioning to endemic




(a) (b) (c) (d) (e) (f) (g)

Fig 1: The *crowded.sg* user interaction design.

living. However, the number of infection cases remains high. A significant portion of the population is still vulnerable to the virus, as they are immuno-compromised and unable to get vaccinated. Furthermore, rising infection numbers could lead to virus mutations such as the more transmissible Delta variant.

Based on findings from epidemiological studies [12], the virus is known to spread quickly in confined and crowded areas. As such, there is an urgent need for the development of a crowd management platform to help the general public be aware of the crowdedness of the places they intend to visit, so that they can make informed decisions while safely transitioning to endemic living.

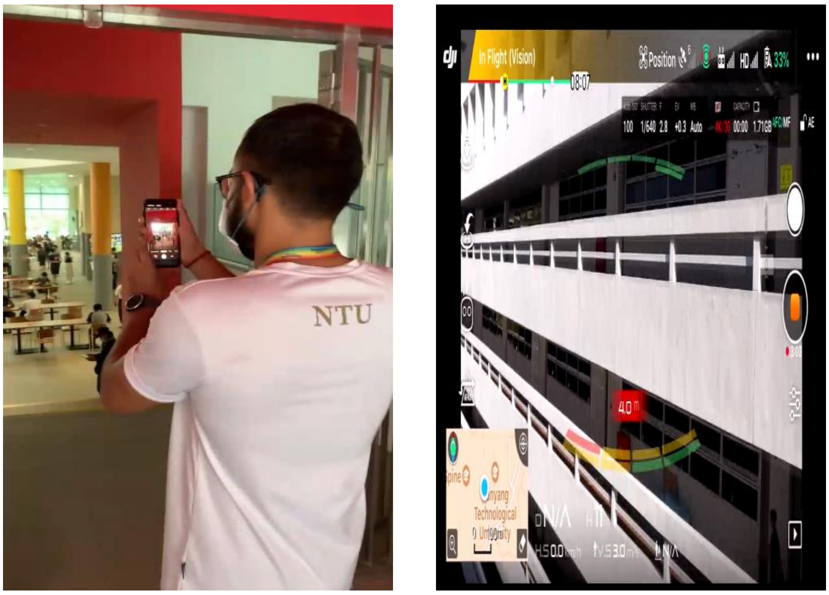
Methods to obtain crowd counts can largely be classified into i) device-based and ii) device-free approaches. For device-based approaches, users are required to carry along devices such as Radio-frequency Identification (RFID) tags or mobile phones. For example, Google relies on user location data to report the crowdedness of a place. However, this requires a high level of compliance from users to enable location sharing services. In contrast, device-free approaches utilize infrastructure such as network interface controllers or CCTV cameras. Network interface controllers broadcast beacon messages and receivers to measure Channel State Information (CSI) values for crowd counting [16]. Such an implementation is mainly suitable for indoor usage, as outdoor usage introduces obstructions to the signals in the form of scattering, fading, and multipath effects. Moreover, the maintenance cost is high.

Our solution[1] *crowded.sg* addresses the limitations of these approaches by adopting a crowd-sourcing approach (Figure 1). To count a room full of people, our approach requires just a single participant to contribute the image using their camera [6]. For larger outdoor venues such as running tracks or recreational parks, Unmanned Aerial Vehicles (UAVs) are utilized to capture aerial images for better coverage. State-of-the-art computer vision techniques then analyze the images collected to obtain accurate crowd counts. Users can review the uploads from others to ensure the reliability of information on the platform.

To encourage user uploads, we deploy a budget-aware dynamic incentive mechanism to award users for each upload. The mechanism aims to provide an appropriate point level that can attract sufficient uploads while keeping incentive costs low. Over time, as we collect more historical crowd data, we are able to train predictive models to provide forecasts on the crowd situation [14]. This enables users to plan in advance and reduces the platform's reliance on crowd-sourced contributions.

The deployment was carried out at the Nanyang Technological University (NTU) campus since August 2020. Within a short period of the launch of *crowded.sg*, the platform has attracted more

[1] Video demonstration: https://youtu.be/TZysnvYl4To




(a) (b)

Fig 2: Sources of images for crowd counting.

than 400 users. An assessment of the platform's usability and usefulness conducted across 100 users revealed that users found the system to be easy to use and have consciously avoided crowded places based on the information provided by the platform.

## II. User Interaction Design

*crowded.sg* utilizes user-contributed images and aerial images collected by UAV (Figure 2) to perform crowd counting. The platform provides the following user interaction features for decision support:

1. *Crowdedness at a glance*: At the homepage, users are presented with a map showing key locations of interest (Figure 1(a)), e.g., dining outlets. The circles are color coded based on how crowded they are. Red indicates that a location is very crowded and that users should avoid it. Yellow indicates that a location is moderately crowded, whereas green indicates that a location is not crowded. The rule-based color coding is optimized based on the historical data of crowd counts specific to each location. To access more details about how crowded a location is, users can click on the location marker to reveal an estimated count automatically derived based on the latest uploaded image (Figure 1(b)). The image is algorithmically blurred for privacy protection (Figure 1(c)). The platform automatically removes outdated images that have been uploaded for more than three hours.

2. *Crowdsourced curation of uploaded images*: To enhance its scalability, the platform enables users to upvote or downvote an uploaded image (Figure 1(c)). Images with high net down-votes are automatically removed.

3. *Easy search and filtering*: Users can toggle the venue filters such that only the locations that belong to a desired category are highlighted on the map. They are also given the option to search for an exact location using the auto-complete enabled search bar at the top left corner of the webpage. To get the most reliable crowd information, users can toggle the "top rated" (Figure 1(a-c)) filter to display images with the most upvotes in the past hour.

4. *Predicted crowd level based on historical data*: For locations without up-to-date uploads, we provide an estimate of the current crowd level (Not Crowded, Some Crowd, Crowded, Very Crowded). Predictions are also made for every hour in a week, displayed in the form of a heat map for weekly trends or bar graph for hourly trends to provide decision support on the opportune time to visit the given location (Figure 1(d)). For locations without historical data, we perform extrapolation by calculating the average crowd level of locations within 400 meter radius.

5. *Point reward system and leaderboard*: To incentivize users to upload photos to the platform, a point reward system and monthly leaderboard are incorporated. The number of points allocated for each locations are displayed inside the location markers and users have the option to toggle its display (Figure 1(f)). Logged in users can earn points when they make an upload and their accumulated points are used to compete with other users in the leaderboard (Figure 1(g)). Top users with the highest points can use them in exchange for monetary rewards.

## III. Use of AI Technology

The system architecture of *crowded.sg* is shown in Figure 3. The AI engine consists of three modules: 1) visual analytics module to perform crowd counting, 2) prediction module for forecasting the number of visitors at various locations, and 3) dynamic incentive module to determine the incentive points awarded to users for contributing to images at each location. When a user uploads an image, the visual analytics module produces a crowd count which is stored together with the location, time, and day of the week information in a database.

The prediction module uses the historical crowd data to train models for making forecasts about the current or future crowd counts. To reduce noises in the training data, we curate the data based on their upvote and downvote information. As more data is accumulated in the database, the prediction model is re-trained to reflect new trends. This automated pipeline enables the prediction module to get better with more contributions from the users both in the form of uploading images or curating uploads of others.

The incentive module serves the purpose of balancing user demand (i.e., interest in knowing the crowdedness of a location) and supply (i.e., information provided by the platform). The click rate for each location is stored in the database. The demand for image uploads is derived using the click rate, current crowd count from the visual analytics module, and the next hour crowd count from the prediction module. For supply, we measure the amount of up-to-date information on the platform based on the time since last image upload and user upload rate for the past hour. The incentive module subsequently allocates points to locations of interest to balance the demand and supply while keeping incentive costs (e.g., monetary rewards) low.

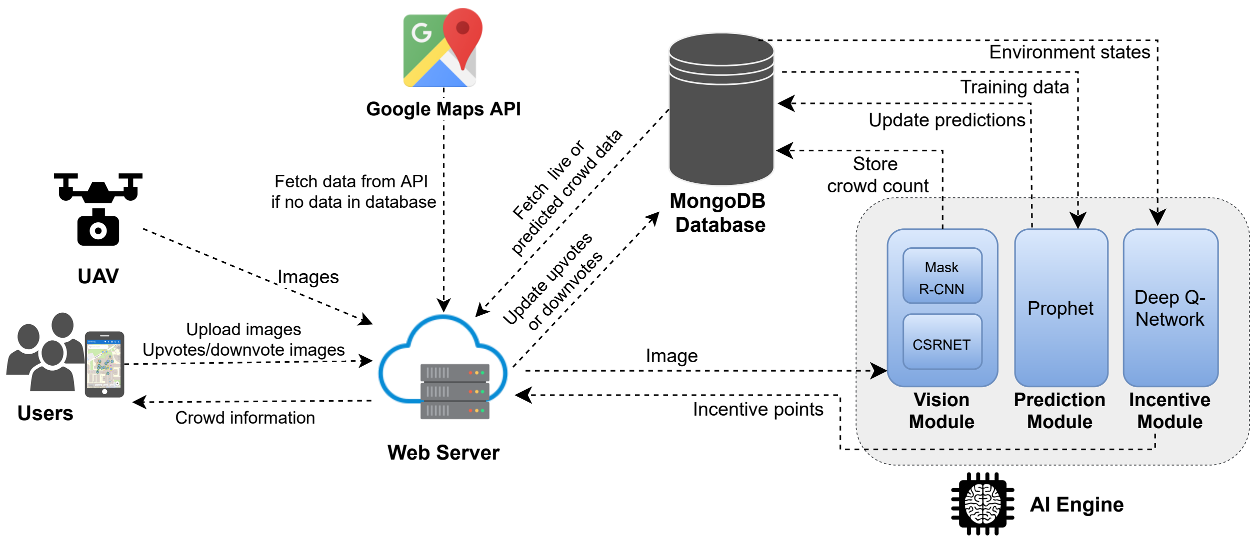



Fig 3: The system architecture of *crowded.sg*.

## A. *Hybrid Crowd Counting Model*

### A-1. *Person Detection with Mask R-CNN*

Mask R-CNN [4] is a two-stage detector used for instance segmentation of objects in an image. A Region Proposal Network (RPN) first finds parts of the image which are likely to contain an object. As objects are of varying sizes, the second stage involves aligning each proposal to a fixed shape for classification.

In our application, we make use of a Mask R-CNN with ResNet101 [5] backbone and Feature Pyramid Network [8]. The model is pre-trained on the MS-COCO dataset [9] which has a “person” category. During inference, we sum the number of unique segmentation of the person class to obtain the crowd count. However, object detection models have trouble dealing with occlusion and cluttered backgrounds [3]. As shown in Figure 4, an image with a large number of people (i.e., a dense crowd) will result in many missing counts.

### A-2. *Crowd Density Estimation with CSRNet*

CSRNet [7] is a fully convolutional neural network that uses dilated convolutional layers to increase the receptive field and aggregate the multi-scale contextual information for density estimation of dense crowds. Unlike Mask R-CNN, CSRNet does not try to detect instances of the object. It identifies regions in a given image that have a large concentration of people and produces a feature map output (Figure 5). The final crowd count is obtained by aggregating the feature map.

In *crowded.sg*, we adopt a CSRNet pre-trained on the ShanghaiTech Part A dataset [10], which contains images of congested places obtained from the Internet. CSRNet solves our problem of dealing with images that contain dense crowds. However, through usage in our platform, we observe that CSRNet has a tendency to overestimate the crowd count when the crowd is sparse.

### A-3. *Hybrid Model*

As discussed previously, each model has its own strengths and weaknesses. An effective way to heuristically combine the two models is to alternate between them based on the scenario in the


Fig 4: Detection results of Mask R-CNN: For sparse crowds, the model is able to accurately detect each person (top). For dense crowds, many of the people in the back are not counted correctly (bottom).

Fig 5: Image of a dense crowd (left). The corresponding feature map output for CSRNet shows larger weights for dense regions (right).

image. Thus, we first use Mask R-CNN to detect the number of people in the image. If the count is smaller than a stipulated threshold (e.g., 15 people), we use the output as our final count. However, if the threshold is exceeded, CSRNet is used to perform inference on the image with its estimated crowd count being used instead. By integrating both models into our AI engine, we are able to

| Symbol | Definition |
|---|---|
| $cc$ | crowd level (current hour) |
| $nc$ | crowd level (next hour ) |
| $cr$ | click rate (past one hour) |
| $lu$ | time since last upload |
| $ur$ | upload rate (past one hour) |
| $t$ | time slot |
| $p^t$ | posted point |

Table 1: Key notations.

handle both sparse and dense crowds.

### *B. Crowd Forecasting with Prophet*

The prediction module uses an additive regression model with components for modeling growth trends, seasonal patterns, and holidays. The model parameters are set based on the time span of data collected. The motivation behind this is to better deal with the irregular time series with missing data (i.e., no pictures uploaded by users) or outliers (i.e., inaccurate counts from the vision module or malicious users that upload irrelevant pictures). The regression model is implemented using Prophet [13], an open-source forecasting library designed by Facebook.

### *C. Dynamic Incentive Mechanism*

The goal of the dynamic incentive mechanism is to choose an appropriate amount of reward points to attract sufficient user uploads while minimizing incentive costs. Intuitively, awarding participants with higher points may attract more uploads but could lead to higher costs. On the other hand, awarding users too few points may reduce the user's interest to participate, resulting in a shortage of sensing data. Therefore, the dynamic incentive mechanism can be considered as a sequential decision problem where the platform's demand for image sensing data and the supply of user uploads determine the points allocated for each location. We first formulate the problem as a Markov Decision Process (MDP) [2]. The MDP is defined as follows using the notations in Table 1:

- *State space*: The state space is denoted as $\mathcal{S} = \{S^t; S^t = \langle cc^t, nc^t, cr^t, lu^t, ur^t \rangle, \forall t\}$ which represents the platform's sensing demand and image upload supply.
- *Sensing demand:* A higher value of $cc$ and $nc$ suggest that more people are at a location currently and more are expected to visit the location at the next hour respectively. The higher value of click rate $cr$ indicates that more users are interested in finding the crowd information of a location. These represent the demand for image uploads.
- *Image upload supply:* A higher value of $lu$ represents that a longer time has passed since the last upload, whereas a lower value of $ur$ represents that the upload rate of a location is lower. These constitute the supply of image uploads at a location.
- *State transition*: New states are polled every 10 minutes to derive $S^{t+1} = \langle cc^{t+1}, nc^{t+1}, cr^{t+1}, lu^{t+1}, ur^{t+1} \rangle$.

- *Action space:* The action space is represented by $A = \{1, 2, \ldots, 10\}$, which corresponds to the incentive point level $p^t$ for a specific location.
- *Reward function*: The reward function is given by $R^t(S^t, p^t) = -|\frac{cc^t+nc^t+cr^t}{3} + lu^t - ur^{t+1} - p^t|$ if $ur^{t+1} < \frac{cc^t+nc^t+cr^t}{3}$ or $lu^t >= 60min$, penalizing the agent for reaching an undesirable state where the observed upload rate is less than the sensing demand rate or when the location does not receive any upload for more than an hour. $R(S^t, p^t) = \frac{cc^t+nc^t+cr^t}{3} + lu^t - \frac{ur^{t+1} \cdot p^t}{10}$ is used otherwise, to reward the agent for reaching a state where supply meets demand. The agent receives a higher reward if fewer points are used to reach the desirable state.

The MDP with unknown transition probability is solved using the Deep Q-Network (DQN) algorithm [11], which uses the neural network to find the optimal policy by learning the state-action values, , $Q^*(S^t, p^t)$. To address large state space in *crowded.sg*, we adopt the Deep Q-Network (DQN) algorithm [11] which uses the neural network in Figure 6 to estimate the Q-values [15]. The DQN outputs Q-values of each incentive point level for a specified location and the point level with the maximum Q-value will be displayed in the corresponding location marker to be rewarded when user make their upload.

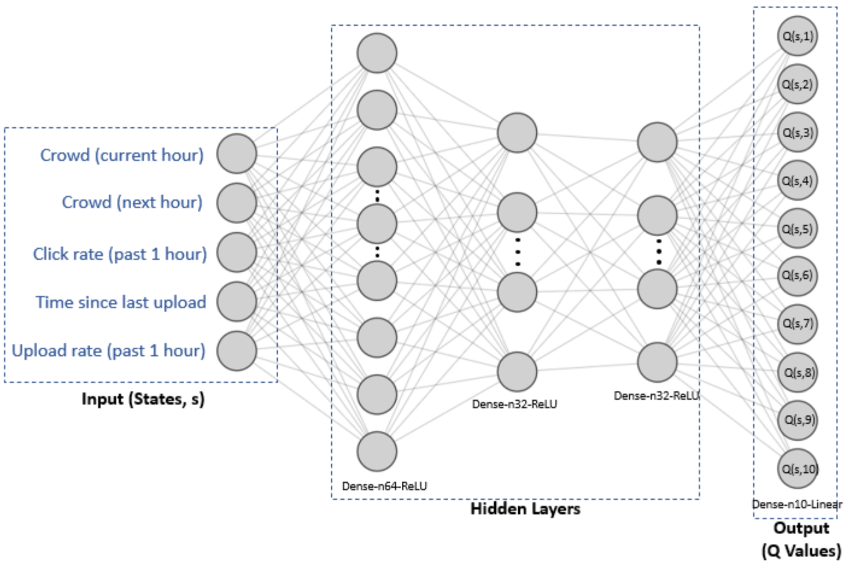



Fig 6: The DQN architecture.

The DQN is trained on platform usage data and external data from the Google Popular Times API. When the platform receives more image uploads providing historical and up-to-date data, the DQN can be retrained based on those data to account for changing image upload patterns (i.e vacation break, examination period).

## IV. Application Development and Deployment

The *crowded.sg* platform was developed using Javascript. Vue.js is used for the frontend, with a mobile-first design approach so as to ensure a seamless user experience across multiple types of devices (Figure 7). We launched the *crowded.sg* platform in early August 2020 on the NTU campus, just in time for the reopening of the school so as to cater to the increase in crowd density brought about by the returning student population. The website was publicized to more than $1,300$ students from the School of Computer Science and Engineering (SCSE) in NTU.

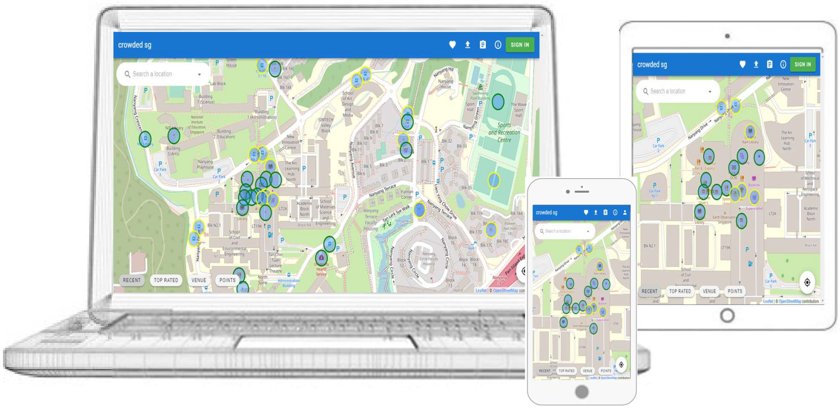


Fig 7: *crowded.sg* across different devices.

### A. *Forecasting model performance*

To evaluate the prediction module's real-world forecasting ability, we collected data at two popular eateries on campus: McDonald's and Pioneer canteen over the period of 16 November 2020 to 20 November 2020. Data was collected at hourly intervals for a total of 151 samples. To obtain the ground truth crowd numbers, we first used the vision module to obtain a preliminary crowd count before carrying out manual inspection to correct for inaccuracies.

We report the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Median Absolute Percentage Error (MdAPE), and Coverage in Table 2. Our model achieves accurate predictions with the ground truth count being within the confidence intervals 90% of the time.

| Venue | RMSE | MAE | MdAPE | Coverage |
|---|---|---|---|---|
| Pioneer Canteen | 3.05 | 2.48 | 0.253 | 0.934 |
| McDonald | 3.96 | 3.07 | 0.221 | 0.890 |

Table 2: Crowd forecasting performance.

We further analyze seasonality components in our additive model for the two venues (Figure 9a, Figure 9b) and observe that some interesting trends have been picked up. McDonald's has a larger lunch crowd compared to dinner crowd. This is likely due to McDonald's convenient location at the main campus building which draws more foot traffic consisting of students and staff during office hours. For off office hours like dinner time, the crowd is considerably smaller with less staff. In contrast, the dinner and lunch crowd for Pioneer canteen is generally similar. The canteen is situated near student hostels and has a steady stream of student visitors who typically have lunch and dinner on campus during weekdays.

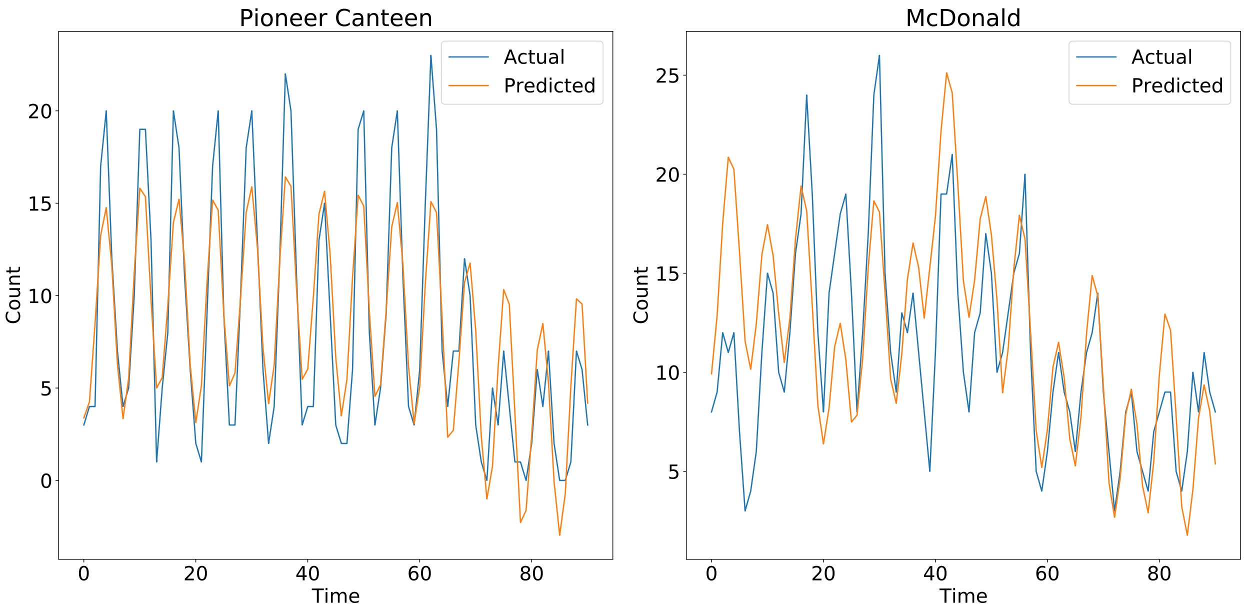
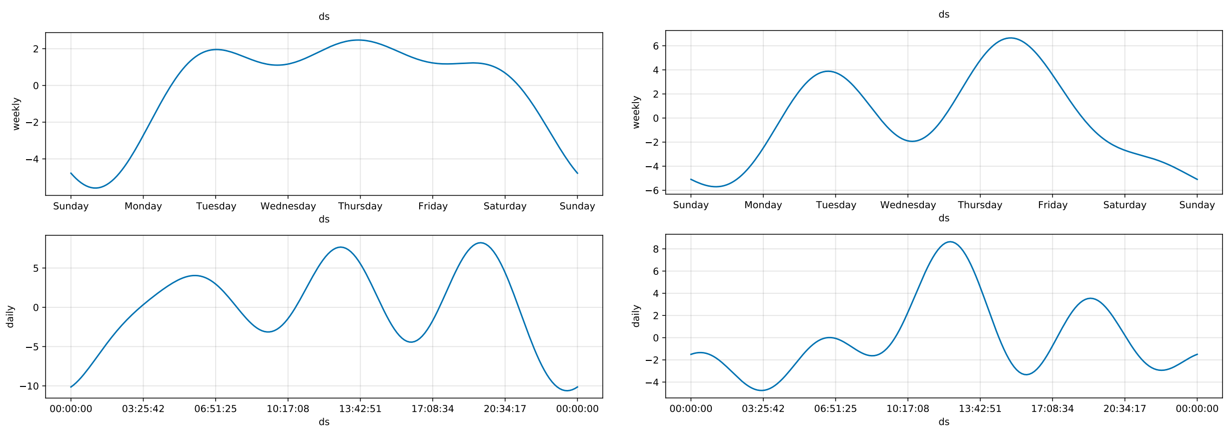



Fig 8: Comparison of actual and predicted crowds at two eateries on campus.

### B. Dynamic Incentive Mechanism Evaluation

We perform simulations based on a school canteen on a weekday from 00:00 AM to 23:59 PM. The crowd level can be observed in Figure 10, with peak crowds observed during lunch hours. The number of participants using the platform are modelled to be a function of the crowd level. There are three tranches of participants, each with a willingness to upload only if the points awarded exceed their preference threshold. The preference threshold is arranged in an ascending order. At every 10 minutes interval, the agent will sense the environment and adjust the points offered.

The incentive scheme is observed to respond to the current system states. The points chosen by the agent can be visualised in Figure 11. States with higher sensing demand and lower image upload supply are allocated higher points and vice versa. In contrast, static incentive schemes offered to users do not vary according to the current system states. Moreover, the dynamic incentive scheme also takes into account the *next hour* crowd level derived from historical data (Figure 10).




(a) Pioneer Canteen (b) McDonald's

Fig 9: Plots describing the seasonality at different venues.

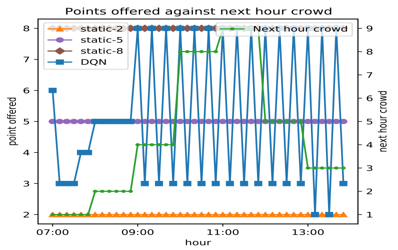



Fig 10: Points allocation for different incentive schemes.

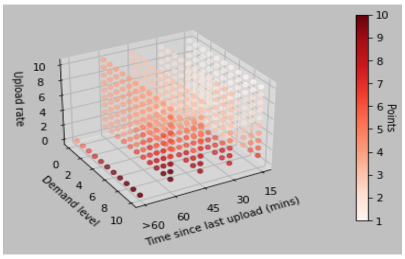



Fig 11: Incentive points allocated for different environment states.

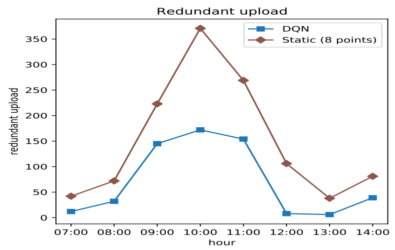



Fig 12: Redundant uploads by hour for DQN vs. Static scheme.

For example, the points awarded are adjusted upwards when the location is expected to become more crowded. This serves to encourage more uploads at the location so that users of the platform can make up-to-date informed decisions on whether to avoid the place.

The dynamic incentive scheme is also budget-aware. Less points are allocated for off-peak hours with less crowds. For places with up-to-date information, there will be fewer points offered for the next period to reduce the number of redundant uploads (Figure 12).

## V. Maintenance

Our system is built based on a modular approach to achieve separation of concerns. The frontend, backend, and AI engine are separate modules that can be freely modified without affecting each other. For example, we can improve the accuracy of our crowd counting models by changing to more effective models or retrain the current models with new data collected. Such changes to the AI engine are independent of other modules and will not disrupt the overall workflow. As the platform is deployed as a software system (i.e., a website) rather than a software product (e.g., a mobile app), any updates made to it will be immediately reflected in subsequent user accesses.

Our system has been able to handle the user traffic to date and no maintenance has been required since deployment in August 2020. The storage of user images and historical data could increase with time. Potential solutions like cloud-based storage or archiving of old data could be used to ensure the smooth running of our servers.

## VI. Application Use and Payoff

The main goal of *crowded.sg* is to help users make informed decisions when planning their trips to locations of interest in order to avoid overcrowding. Within a month of its deployment, the platform has attracted over $400$ unique users and $2,000$ page views (Figure 13). From Figure 14, it can be observed that *crowded.sg* receives the most traffic during the middle of the week, which is when most of the student population are on campus. Peak usage is mainly concentrated in peak hour periods, namely lunch time (12pm - 1pm) and the end of lessons or office hours for the day (5pm - 6pm).

We also conducted a comprehensive user study to assess the efficacy of our platform. A total of 100 users participated in the online survey and gave feedback on various aspects of the platform. The questions asked were designed to assess the platform's usability and usefulness, following the System Usability Scale (SUS) [1]. The results of the studies are presented in Figure 15.

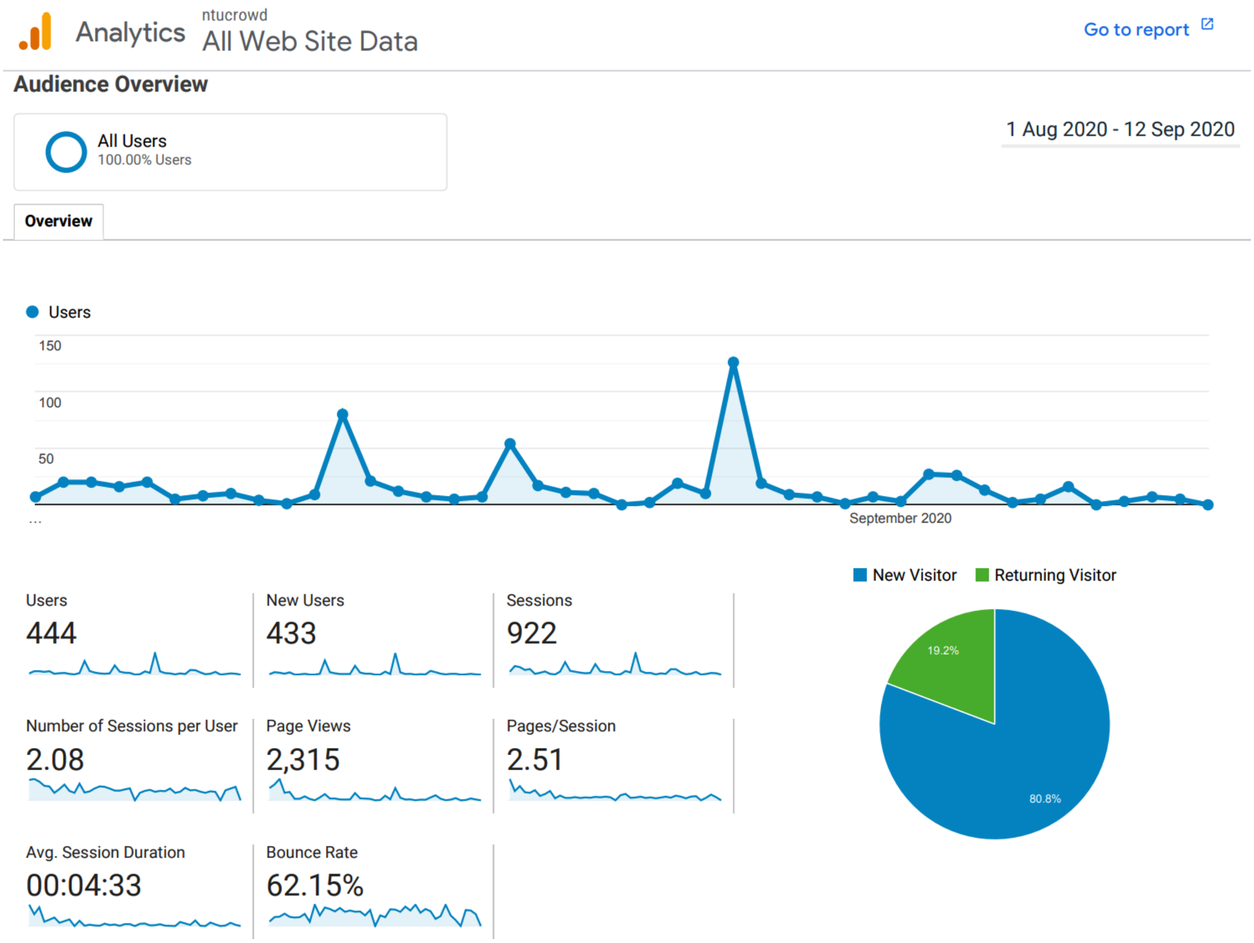



Fig 13: Google Analytics results for *crowded.sg* from 1st August, 2020 to 12th September, 2020.

In terms of usability, 80% of respondents found our interaction design and workflow easy to use. For those who did not, an open-ended feedback was elicited. The most common reasons cited involve the map interface. It was suggested to swap the map interface with a list view instead which shows the least or most crowded venues for each category. We will implement this as a complementary page to the current map view. The reason is that a list view is only useful for users who are familiar with the location. For new users or visitors to the school, the name of the venue alone does not provide many useful insights that they can act on.

Another key concern is whether the venues listed on the platform are comprehensive enough. A missing venue would mean that users are unable to receive any update about the crowdedness in the corresponding venue. Our decision to have a fixed list of curated venues was to ensure that the map interface is easily accessible and not cluttered with location markers. In fact, 90% of users agreed that no further addition is required. In summary, these results validate the usability of our interface and workflow, giving us a strong foundation to build on for future improvements.

In terms of usefulness, 81% of users made a conscious decision to avoid crowded locations based on results provided by crowded.sg. A common use case of the platform is for deciding which campus canteen to have lunch at, before having access to the website, 83% of users needed to visit multiple locations before deciding on one, especially during peak hours, in order to avoid

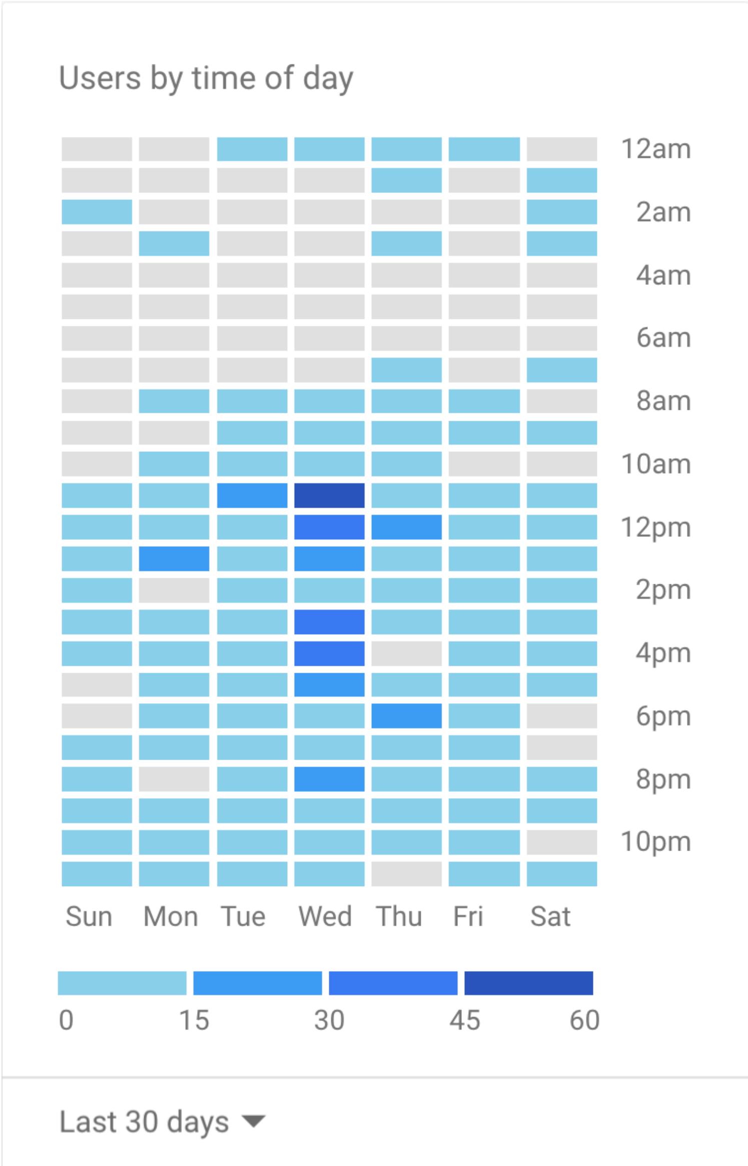



Fig 14: Usage patterns of *crowded.sg*

the crowd. With advice from crowded.sg, the users do not have to physically check different locations. In terms of absolute time saved, 86.7% of users reported that the platform helped them cut down on the time needed to make a decision, with 71.4% of users stating that crowded.sg helped them save more than 10 minutes per trip on average.

## VII. Lessons Learned During Deployment

The value of a COVID-19 crowd management decision support platform lies in the timeliness and quality of the information provided to users. This poses three challenges to the future large-scale deployment.

Firstly, the UAVs are banned from flying in sensitive areas near military bases and airports in Singapore. For large-scale deployment of the platform, the UAVs have to be restricted to operating in parks and sports facilities where people might congregate. Fortunately, several UAVs have

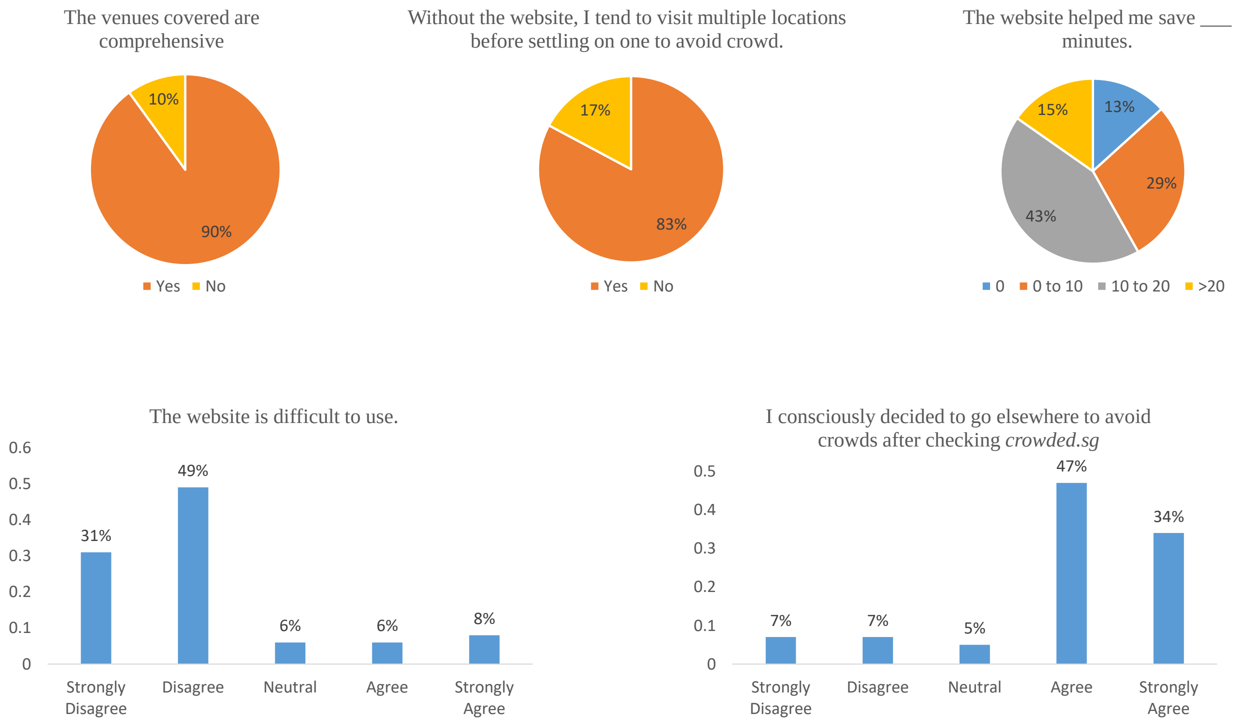



Fig 15: Results of the usability and usefulness study of *crowded.sg*.

already been deployed by government bodies and private companies for purposes such as the monitoring of nature reserves. For future deployment, partnerships need to be forged to seek support from multiple stakeholders.

Secondly, while *crowded.sg* provides crowd information on key locations of interest, the routes leading to them, such as corridors, are found to be congested. For future deployment, we aim to recommend users to take the less crowded routes to locations.

Thirdly, a concern raised is the potential privacy infringements with regard to the uploaded pictures. We have tried to increase the amount of blur in the image according to its resolution. However, in some cases, the image is blurred to the point where the location is indistinguishable. This defeats the purpose of having the functionality in the first place which is to give users a chance to decide for themselves, on top of the count generated by the AI engine. Instead of blurring the entire image, we have experimented with using facial detection and only blurring the region which contains the face. This function will be incorporated into future versions of the platform.

## VIII. Conclusions and Future Work

In conclusion, the current iteration of the platform has achieved our main objective of helping users make more informed decisions of places to visit during the pandemic to maintain proper social distancing. There are many more exciting future works that can be explored. One key area is on automated decision making. Functionalities such as recommending the best path from one venue to the next (i.e., a route planner optimized for users to avoid crowds when traveling to and from venues) will be explored to enhance our platform.

## References


[1] Aaron Bangor, Philip T Kortum, and James T Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594, 2008.

[2] Richard Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684, 1957.

[3] Guangshuai Gao, Junyu Gao, Qingjie Liu, Qi Wang, and Yunhong Wang. Cnn-based density estimation and crowd counting: A survey. *arXiv preprint arXiv:2003.12783*, 2020.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] Hongchao Jiang, Wei Yang Bryan Lim, Jer Shyuan Ng, Harold Ze Chie Teng, Han Yu, Zehui Xiong, Dusit Niyato, and Chunyan Miao. Ai-empowered decision support for covid-19 social distancing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 16044–16047, 2021.

[7] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1091–1100, 2018.

[8] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[10] W. Liu, W. Luo, D. Lian, and S. Gao. Future frame prediction for anomaly detection – a new baseline. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[12] Hussin A Rothan and Siddappa N Byrareddy. The epidemiology and pathogenesis of coronavirus disease (covid-19) outbreak. *Journal of Autoimmunity*, page 102433, 2020.

[13] Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

[14] Harold Ze Chie Teng, Hongchao Jiang, Xuan Rong Zane Ho, Wei Yang Bryan Lim, Jer Shyuan Ng, Han Yu, Zehui Xiong, Dusit Niyato, and Chunyan Miao. Predictive analytics for covid-19 social distancing. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 5016–5019, 8 2021.

[15] Liang Xiao, Yanda Li, Guoan Han, Huaiyu Dai, and H. Vincent Poor. A secure mobile crowdsensing game with deep reinforcement learning. *IEEE Transactions on Information Forensics and Security*, 13(1):35–47, 2018.

[16] Han Zou, Yuxun Zhou, Jianfei Yang, Weixi Gu, Lihua Xie, and Costas Spanos. Freecount: Device-free crowd counting with commodity wifi. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.