UTRECHT UNIVERSITY
Artificial Intelligence

Rianne Lam 6888216

SHORT-TERM CROWDEDNESS PREDICTIONS FOR LOCATIONS IN AMSTERDAM

First Supervisor
Prof. Dr. Albert Salah
Second Examiner
Dr. Shihan Wang
External Supervisor
Shayla Jansen

Date
11-06-2021

Table of Contents

Acknowledgements

First, I would like to thank my supervisors, Albert Salah and Shayla Jansen, for their guidance throughout the thesis project. With Albert sharing his extensive knowledge, whether related to machine learning, the mathematics behind the models, experimental procedures, or writing tips, I was able to learn a lot and bring out the best when working on this thesis. I have learned a lot from all his feedback and ideas, which I am very thankful for. Shayla has helped me a lot with both my thesis and the internship. She was always there to help me navigate the different activities, keep a clear focus on my goals, and to brainstorm together based on any problems I had. I have learned a lot of new skills from her that I will use in my future career, which I am really grateful for.

Then, I would like to thank the whole IGOR team for welcoming me to the team and giving me the opportunity to share my ideas and work. Even though we had to work online, I felt as being part of the team and it was a great experience to learn all about working at the municipality and the work of a data scientist.

I would also like to thank the municipality's AI team, specifically Petra Ormel and Iva Gornishka, and the other AI and DS interns as well for being involved with my project and always available for any questions, advice, or brainstorming sessions. Our weekly walks were very helpful in keeping the team spirit and motivation for the thesis high.

Finally, I would like to thank Yannick Tamerius for being my feedback buddy when it came to writing the thesis, and sparring partner. It was really nice to be able to share experiences with another student from the UU. Lastly, I would like to thank Nardi Lam for helping me tackle the workings of the SARIMAX model, which was a challenge for me, but with his teaching skills it all worked out.

Abstract

The aim of this thesis was to investigate multiple models for short-term crowdedness predictions based on mobile phone data. For this, we used data of three different locations in the city of Amsterdam (a square, park and market). We examined the contribution of various external factors (such as the weather and COVID-19 regulations) and we compared different modelling techniques (such as regression and LSTM) when predicting crowdedness two hours ahead. We found that regression models obtained the highest prediction accuracy when used in combination with an oversampling technique to account for the sparsity of crowded samples. Furthermore, we found that historical values of the ground truth data (e.g. crowdedness of the previous time step) and information on temporal aspects (e.g. time of day and day of the week) were most influential in the prediction models. These prediction models could be used to support crowd management by providing the expected crowdedness for the near future.

1. Introduction

1.1 Crowd management

In the recent year, there has been increased attention to the field of crowd management because of the spread of COVID-19 [1]. The goal of crowd management is to plan and manage events or other circumstances under which many visitors tend to be present, so that safety can be guaranteed [2]. Especially during this pandemic, the need for crowd management has increased at many locations, as crowd forming currently poses a greater risk on the health of citizens and visitors than before the pandemic. At this moment, in many countries citizens should adhere to social distancing (also termed physical distancing) regulations, and as a consequence it is important to prevent crowd forming [3].

1.1.1 Problem statement

To support crowd management in tackling crowdedness, there is a need for predictions on future crowd forming at locations where many citizens might gather. In this way, crowd management can act based on these predictions and try to dissuade, prevent or cease crowd forming. For this, it is important that these predictions can be made within a short time frame, because in the current climate short term changes in for example COVID-19 related regulations can impact the locations that citizens will visit to a great extent. Thus, it is important to inform crowd management (and citizens) on crowdedness at certain locations within a short time frame so that they can act on this information timely.

1.1.2 Current limitations

Most studies on predicting crowdedness take into account some external factors that are known to influence crowdedness, such as the weather or holidays [4, 5, 6, 7]. The same could be true for other types of external factors (e.g. COVID-19 regulations, political demonstrations or sport events [8]). Hoang et al. (2016) proposed that in future work on predicting crowdedness, model performance could be further improved if other external factors were also included in the model [5]. However, none of the recent studies on this topic consider these type of external factors in their models.

Moreover, most of these studies make use of the same data sets (data on bike trajectories in New York and taxi trajectories in Beijing [4, 5, 6]). Thus, another aspect that is missing in these studies is crowdedness data targeting pedestrians. It is valuable to also include information on the number of pedestrians in an area in a predictive model, especially with the current need for social distancing.

1.2 Objective

The goal of this thesis was to predict the crowdedness level at certain city locations within a short time frame. This was realised using information on the location of visitors based on mobile phone data, with aggregated visitor counts per location. Locations can be of varying size (e.g. a square, street or park). Some studies have shown that when predicting on a short time frame, the most recent data is most informative for the predictions [6, 9, 10]. Therefore, when selecting a prediction window, we had to find a balance in being able to make use of the most recent data, and still being able to use the predictions in practice [11]. As a result of this consideration, we decided to predict two hours in the future. In this way very recent data can be used for the predictions, and in most cases it is still possible to act based on the predictions.

With this research, we aimed to expand on the recent literature on this topic in two ways. First, we investigated the effects of a variety of external factors, among which COVID-19 regulations, on the expected crowdedness levels at certain city locations for the next two hours. Second, we included data on pedestrians, in addition to data on bikes and cars. This results in the first research question(s):

RQ 1: *What external factors have an influence on the level of crowdedness at certain city locations when predicting the next two hours?*

RQ 1a: *How can we evaluate the individual contribution of an external factor on the predicted level of crowdedness?*

Furthermore, there are many modelling techniques that one could use when trying to predict crowdedness based on multiple data sources, of which each has its advantages and disadvantages. Some examples of model types that can be used when predicting crowdedness are discussed in Section 2. This poses the second research question(s):

RQ 2: *What models are effective in predicting the level of crowdedness at certain city locations when predicting the next two hours?*

RQ 2a: *What are the advantages and disadvantages of using certain models for this prediction task?*

RQ 2b: *Is this prediction task more suited to frame as a regression problem or a classification problem?*

We formulate the prediction problem of this thesis as follows:

Problem definition. Given a set of historical observations of the number of visitors $\{Y_t|t = 0, 1, ..., k\}$ and a set of external factors $\{X_t|t = 0, 1, ..., k\}$, predict the number of visitors for the next two hours Y_{k+1}. Here, t represents the time step (with an interval of two hours) and k represents the current time step. An illustration of this problem is depicted in Figure 1.

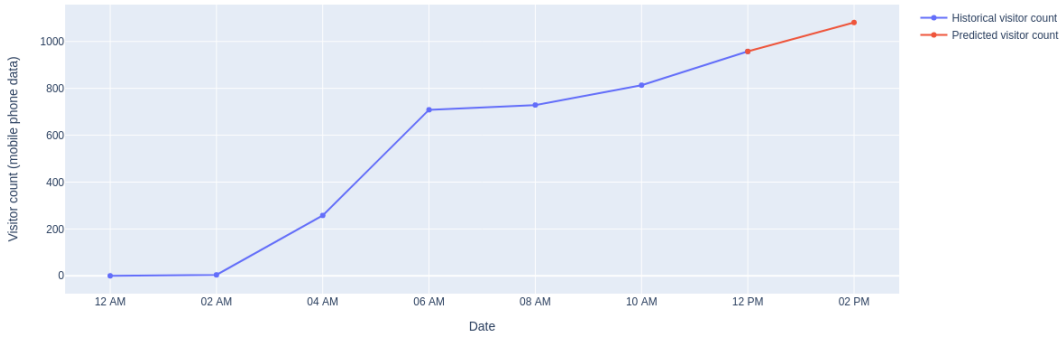


Figure 1. Illustration of the predicted number of visitors with a prediction window of two hours. In this example we predict the time slot 02:00 PM based on the historical data for 12:00 AM - 12:00 PM of the same day.

This research was performed in collaboration with the municipality of Amsterdam. Specifically, the goal was to predict the level of crowdedness at multiple locations throughout the city of Amsterdam, using mobile phone data as ground truth data. In doing so, we considered the following data sources as external factors: camera data, parking data, weather data, public transport data, holiday data, Twitter data and COVID-19 regulations data. Furthermore, we considered a set of different prediction models: linear regression models, non-linear regression models, Seasonal Auto-Regressive Integrated Moving Average with Exogenous variables (SARIMAX) models and Long Short-Term Memory (LSTM) models. More details on these models can be found in Section 3.3 (in relation to this thesis) as well as in Section 2 (in relation to the literature).

1.2.1 Thesis outline

This thesis is organized as follows: in Section 2, we discuss studies related to predicting crowdedness. Here, we outline what data sources, model types and evaluation methods have been used for this prediction task so far. In Section 3 we elaborate on the design of the experiments with detailed information on the data sources and models. In Sections 4, 5 and 6 we provide an overview of the results for each experiment. Then, in Section 7 we evaluate the results in relation to the research questions and related studies, discuss the most important limitations of this work and provide some suggestions for future work. Finally, in Section 8 we summarize the key findings and draw some conclusions.

2. Related work

In this section, we discuss related work on the topic of crowdedness predictions. First, we give an overview of the most often used data sources to infer crowdedness. Second, we discuss challenges that arise when working with crowdedness data. After this, we discuss different modelling techniques that have been used for predicting crowdedness. Then, we outline which evaluation methods can be used for this prediction problem. Finally, we briefly state the contribution of this research on the topic of crowdedness predictions with respect to the literature.

2.1 Data sources for predicting crowdedness

In a review paper, Sharma et al. (2018) discussed several techniques that are often used to gather data on crowdedness [12]. We will provide a brief overview of the most widely used data sources.

2.1.1 Vision based data sources

In vision based crowd counting, visitors are being counted based on detection techniques that rely on computer vision (e.g. detecting pedestrians using camera data) [12]. In this way, the density of a crowd can be estimated. Furthermore, in some cases crowd tracking can also be performed, when camera images are processed in a sequential manner, estimating the direction of a crowd (or individuals). Interestingly, since the start of the pandemic, a few studies have been published on detecting crowd density based on camera videos with the aim to monitor social distancing [3, 13]. These recent studies demonstrate that vision based data sources are useful for a variety of research questions related to crowdedness.

Santana et al. (2020) discussed some challenges when using vision based data sources to infer crowdedness levels. For instance, there is the risk of a loss of vision due to for example darkness, bad weather, or objects being in front of the camera. Another challenge is related to privacy concerns. For example, the possibility to detect individuals based on facial features could raise privacy issues. This type of data source was not the focus of this thesis. However, we did include visitor counts based on camera data as an external factor.

2.1.2 Wireless based data sources

The level of crowdedness can also be estimated based on mobile phone data [12]. The data that is gathered based on a mobile phone devices can take different forms: WiFi based location data (device connects to a WiFi access point, where the location of the access point is the inferred location), GPS based location data (GPS location of the device is recorded), or CDR based location data (location of the device is recorded when for example the user is making a phone call). With this kind of data, individual device holders can be either counted or tracked.

A challenge for wireless based data sources is that visitors that for example do not have a mobile phone or no WiFi connection are not being recorded, resulting in errors in the estimated level of crowdedness [1]. Privacy concerns related to using this type of data is typically dealt with by anonymizing and aggregating data of individual device holders.

Most studies discussed in this section use GPS based location data as the ground truth to infer the level of crowdedness [4, 6, 10, 14], as is the case in this thesis. This is followed by WiFi based location data [9, 15], and CDR based location data [7, 16].

2.2 Challenges for predicting crowdedness

The formation of crowds throughout a city is a dynamic process: the number of visitors at a certain location fluctuates over time, may be dependent on the number of visitors at other locations and is influenced by many different factors. This poses some challenges on the problem of predicting crowd forming, which are discussed next.

2.2.1 Temporal and spatial components

Data on the amount of visitors at different locations display some periodic patterns that are inherent to time series data. For example, typically the count of visitors is expected to be higher during the day than during the night. This also holds for broader time scales: an example could be that in the summer months it is busier in the city than in the winter months due to an increase in tourism. In time series modelling, these periodic patterns are referred to as seasonality [17]. Thus, when we are for example trying to predict the number of visitors at a park on a Sunday afternoon, it might be valuable to not only consider the number of visitors an hour ago, but also yesterday afternoon and last week's Sunday afternoon.

Furthermore, in some cases there is also a spatial component to account for. For instance, spatial information is valuable for the problem of predicting traffic flows or crowd flows. In this case, the trajectories of visitors (or vehicles) are being modelled. For example, Zhang et al. (2017) predicted the inflow and outflow of crowds in multiple regions of a city [4]. For this they used several months of data on taxi trajectories in Beijing and bike trajectories in New York, both based on GPS location data. They argued that the in- and outflow of both nearby and distant regions will have an influence on each other. Specifically, if for example region A and region B lie next to each other, and the crowd outflow of region A increases, the inflow of region B is expected to increase as well.

To incorporate this spatial information in their model, they used a neural network called ST-ResNet (Spatio-Temporal Residual Network), in which complex spatial relationships can be modelled using many layers. In the first set of layers of such a network the spatial relationship of regions that are near each other are modelled, while at further layers the spatial relationship of regions that are farther from each other are modelled. They defined grid maps for the inflow and outflow, consisting of a $I \times J$ matrix with n regions (where $n = I \times J$). By combining these, one observation could be represented as $X_t \in \mathbb{R}^{2 \times I \times J}$, where t represents the current time step.

They first combined these grid maps with temporal information. Different seasonal effects were incorporated: closeness (dependence on recent crowd flow), period (daily cycles of crowd flow) and trend (increase or decrease in crowd flow over time). The grid maps were divided based on the three temporal aspects and used as input in the network. Alongside this, some external factors (weather and holiday data) were fed into a two-layer feed forward neural network. Finally, both outputs were combined and further processed, resulting in predicted crowd in- and outflows for individual regions based on a predicted grid map. They showed that their model outperforms a set of baseline models (e.g. linear time series models such as ARIMA (see Section 2.3.1)).

2.2.2 External factors

External factors that could have an influence on the problem of predicting crowdedness were already briefly discussed in Section 1. Here, we further distinguish between three categories of external factors: environmental factors, social events, and COVID-19 regulations.

First, environmental factors that could have an effect on crowdedness are for example the weather or air quality [6, 7]. These are factors that may influence citizen's behaviour. An intuitive example is that citizens might prefer to walk outside in the city on a sunny day

compared to a rainy day. In the study of Yuan et al. (2020) weather data and air quality data were included in their predictive model for crowd flows [7]. They used location data based on anynomized Call Detail Records (CDR) (provided by a large telecom operator in China) as an indicator of crowd flow and public data from meteorological websites on the weather and air quality. They compared a set of baseline models to a ST-ResNet model in a similar fashion as in Zhang et al. (2017) [4]. They found that the inclusion of each of the external factors led to an increase in prediction accuracy, compared to models without these factors, with the largest improvement when all external factors were included.

Second, the occurrence of social events can lead to abnormal behavioural patterns that disrupt the usual periodic patterns of behaviour. For example, on a regular Monday evening we expect the city center to be relatively quiet, but if there is a planned demonstration on the Monday evening a week after, we would expect the city center to suddenly be very busy compared to last week. An illustration of the effect of social events on crowdedness is a recent anti-racism protest on the Dam square on June 1st 2020 in the Netherlands [18]. Due to this event many more citizens gathered at this location than was expected, which resulted in drastic overcrowding. This shows the importance of incorporating social events in a prediction model for crowdedness.

There are some studies that focus on the role of social events on crowd forming, however, these studies are limited to predicting crowdedness within the scope of individual social events taking place. For example, Furletti et al. (2017) examined which type of mobile phone users were present at certain social events (such as festivities) based on mobile phone call data [16]. They found that for example political events mostly attracted local residents, while sport events or musical concerts mostly attracted visitors from outside the city.

Another example is the study of Fan et al. (2015), in which the authors performed two case studies on predicting crowdedness in Tokyo: Comiket (a widely known comic fair) and New Year's Eve [10]. They predicted the movements of visitors based on mobile phone location data (GPS). The authors argued that a crucial aspect in accurately predicting the number of visitors during such events was the importance of the most recent observations (e.g. the number of visitors that arrived during the previous hour). Thus, when a social event is occurring, the most recent historical data is an important indicator of crowdedness in the near future.

Lastly, a recent and very specific category of external factors are COVID-19 regulations. The impact of these regulations on predicted crowdedness levels is expected to be large, as some regulations pose strong limitations on crowd forming. For example, in the

Netherlands all restaurants, bars, and other public establishments such as theaters, museums and libraries had to close for a long time period [19]. Consequently, for locations in the cities that are usually very busy (e.g. streets where restaurants and museums are located) there will suddenly be a large drop in the expected number of visitors as visitors are less inclined to visit these locations. There is one recent study by Mu et al. (2020) on predicting crowdedness that took place during the pandemic [9] and is discussed in Section 2.3.1. However, to the best of our knowledge, no studies so far have been published on predicting crowdedness that incorporate COVID-19 regulations directly as an external factor in their predictive model.

2.3 Methodologies for predicting crowdedness

In the literature, many different model types have been used to tackle the problem of predicting crowd forming. Next, we will go over a range of studies using different model types, briefly explaining the model's workings on a conceptual level.

2.3.1 Time series models

Since predicting crowdedness typically concerns time series data, time series models are often used for this task. Often Auto-Regressive Integrated Moving Average (ARIMA) models have been used [20], or variations of this model type [21]. However, it is important to note that in recent studies, this model is typically used as a baseline model to compare to a more complex model such as a neural network [4, 5, 6, 7, 22]. This model type makes use of autoregression (future observations are predicted using past observations), a moving average (future observations are predicted using the error of past observations) and a differencing term (the observations are made stationary by removing trends) [23]. A trend indicates a long-term increase or decrease of the variable to predict.

Variations of ARIMA models such as the SARIMA (seasonal ARIMA) model handles seasonality in addition to trend. As mentioned before, time series data often display periodic patterns that follow seasonal effects at different levels (daily, weekly, yearly, and so on; see Figure 2) that could be used to enhance prediction performance [17]. The different aspects of a time series can be decomposed and visualized as in Figure 3.

Another variation that is especially relevant for this thesis is the (S)ARIMAX model. In this model exogenous variables can also be modelled, meaning that in this case the target variable is not only being predicted by using its past observations, but also by using one or more exogenous variables. These variables could be any of the external factors that were

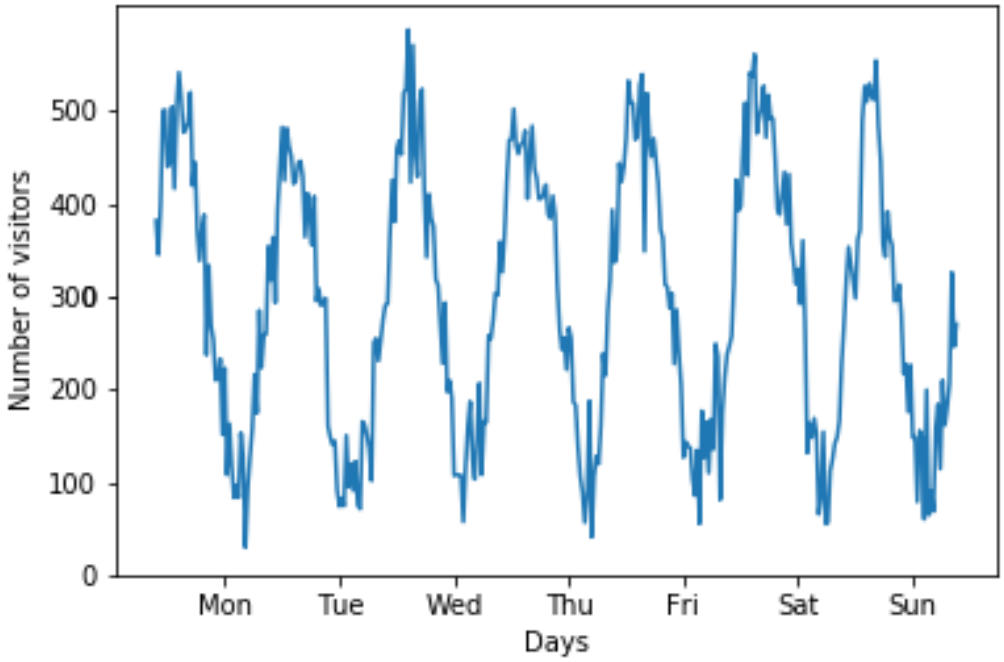


Figure 2. Example of time series data with seasonal effects (daily). This data has been generated using the TimeSynth library in Python [24].

discussed in subsection 2.2.2. More details on the SARIMAX model and its mathematical derivation can be found in Section 3.3.5.

Bouuaert (2013) discusses some advantages and disadvantages of time series models [26]. Advantages of time series models are that they usually show a sufficiently high performance, it is possible to draw statistical conclusions based on the model output, and they are useful for data that is on a macro scale (data that reflect the behaviour of many individuals). Conversely, disadvantages of these models are that the data should adhere to some statistical assumptions, and that often quite some data manipulation is required (e.g. removing trend or seasonality effects).

An example study on predicting crowdedness using a time series model that was performed recently is the study by Mu et al. (2020) [9]. In this study, the authors predicted the number of students present at a university campus for a certain day. The number of students was inferred based on a WiFi based crowd monitoring system at the university campus. Over twenty WiFi access points were located throughout the campus. Every minute a sample was recorded, consisting of the total amount of devices connected to the system, indicating the total number of students on the campus. The goal was to predict the number of students one day in the future, based on past observations of either previous days (intra-week), or observations of the same weekday a week before (inter-week).

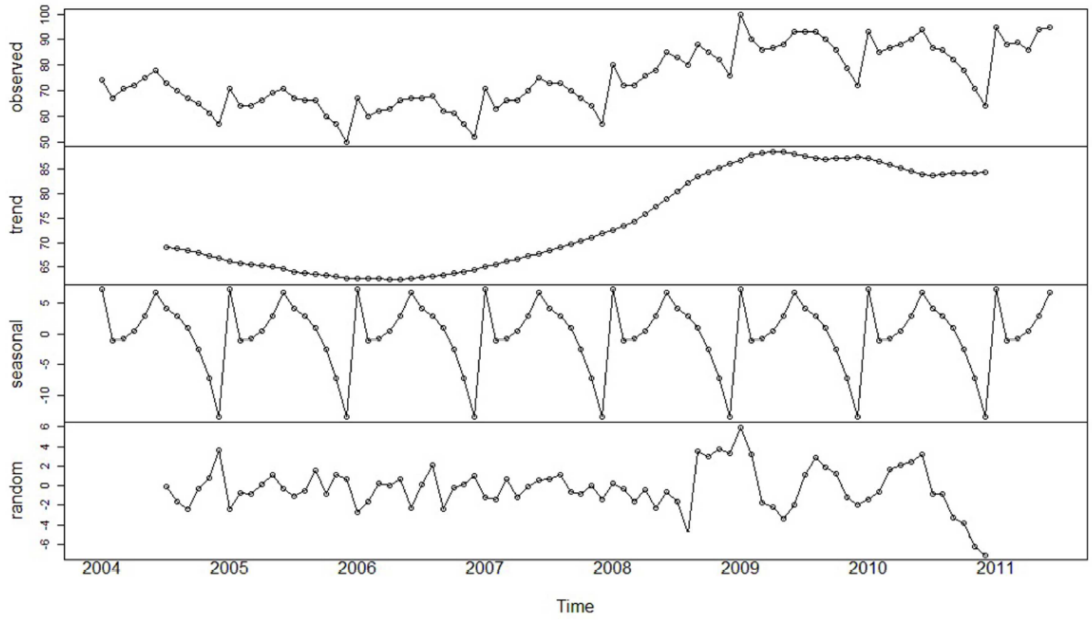


Figure 3. Example of the different temporal aspects of time series data: the observed signal, trend, seasonal cycles and a random signal (residuals). This image is taken from the paper by Jebb et al. (2015) [25].

They implemented a SARIMA model to account for seasonal effects, in this case modelling daily seasonality. Predictions of the numbers of students during a "lockdown" period, when strict regulations applied, were compared to a period during which the regulations were less strict. They found that predictions based on intra-week data were a better fit to the actual observations in the lockdown period than predictions based on inter-week data, while for the period with no lockdown a combination of both predictions resulted in the best fit. Therefore, it seems that when rare scenarios occur, such as a lockdown, the most recent observations become increasingly important when predicting the near future. This effect was also found in the previously mentioned study by Fan et al. (2015) on mobility patterns during big events [10].

2.3.2 Markov models

Markov models are probabilistic models that model the stochastic (random) process of state transitions [27]. A specific type of Markov model that has often been used for the task of predicting crowdedness is the Hidden Markov model (HMM) [28]. In this model, there is a finite number of states that are not observable (within the topic of predicting crowds the states could for example be "quiet" or "busy"). The states are inferred based on the values of other variables that are observable (e.g. the number of visitors). The underlying idea is that the states generate the observations. Over time, a sequence of state transitions is produced.

A HMM consists of the following elements: states, observations, state transition probabilities, emission probabilities (probability that a certain observation will be generated by a certain state) and initial state probabilities [29]. The components are shown in an example HMM (see Figure 4). In this figure, "Quiet", "Busy" and "Crowded" are the states, "100", "200", and "300" (and more) visitors are the observations, the arrows between the states represent the state transition probabilities, B represents the emission probabilities for each state, and Π represents the initial state probabilities.

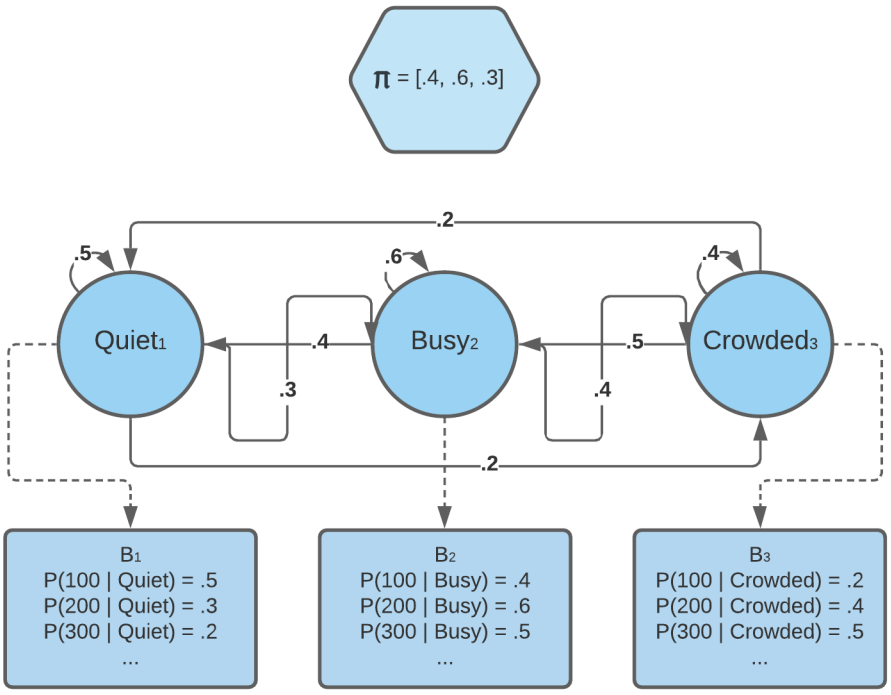


Figure 4. An example HMM for the problem of predicting crowdedness. This figure is based on an example HMM from the book of Jurafsky and Martin (2019) [30].

The order of the model determines the number of previous states that the current state will depend on. Typically, first-order HMMs are used, which meet two important assumptions: a future state only depends on the current state (Markov assumption), and the probability of generating a certain observation only depends on the current state (output independence) [30]. The goal of the model is to find the optimal state sequence for a given problem. For example, in this case the goal could be to find the sequence of states that best reflect how crowded it is at a certain location throughout the day.

An advantage of these models is their ability to model data that is on a large scale and to cope with noisy or missing data [5]. An important downside of using these models for predicting crowd forming is that the model is being limited by only using the past state to model the next state (the first-order HMM). It is possible to develop an HMM of higher

orders, but often this not efficient from a computational viewpoint [31].

To provide an example, Alvarez-Lozano et al. (2013) applied a HMM to a crowd prediction problem [14]. In this study, they predicted the presence of individual mobile phone users at certain locations 3-5 hours in the future based on GPS location data from a public trajectory data set of a project called GeoLife. The location data consisted of trajectories (timestamps and location coordinates) for individual users ranging over several weeks. They first defined a set of regularly visited locations based on the mobility patterns of individual users. They pre-processed the data to contain 30-minute samples on the current location for each user.

Subsequently, they created a HMM for each day of the week, for each user based on data of past days (of the same day of the week). These models consisted of the following: states (that are in the form of locations), observations (that are in the form of the time and current location), a vector that represents the probability that a user starts the day at a certain location, a transition matrix containing the probabilities of moving from one location to another (or a location that is not listed as regularly visited), and a confusion matrix containing the probabilities that the user is at a certain location at a certain time.

They predicted which locations a user would visit in the near future, using the Viterbi algorithm (an algorithm that finds the optimal state sequence based on the HMM's parameters). The correctness of the predictions was determined based on whether the predicted location at a certain time was equal to the actual location at that time. On average, the models achieved an accuracy of 75% for predicting 3 hours ahead and 72% for predicting 5 hours ahead. By aggregating these results over a group of people, the overall level of crowdedness at certain locations could also be predicted.

2.3.3 Neural network models

The most recent type of models that have been used for predicting crowdedness are neural network models. The simplest type of neural network is a feedforward network [30]. In this network, the input variables (data used for prediction) are represented by artificial units, and together these units form an input layer. Subsequently, the values of these units are being transformed and intermediate values are fed into hidden units (that together form a hidden layer). Finally, these intermediate values are transformed once more, resulting in output values of units in a final output layer. These final output values form the prediction for a target variable (data to predict). As the number of hidden layers, and thus transformation steps, in the network increases, the network becomes deeper. An essential improvement of neural networks over other model types, such as time series models, is their ability to learn

non-linear functions between the input variables and target variable.

In the network, the relationship between the input variables and output variable is learned by training weights (the weights transform the input values to the output value). This is usually done by performing an algorithm called backpropagation. The intuition behind this algorithm is that the prediction error (the difference between the observed and predicted value) is being propagated back into the network, adjusting the weights according to their contribution to the error. In this way, the prediction error of the network is being minimized.

A more advanced type of neural network is a recurrent neural network. Here, the idea is that the information in the network does not flow in one direction, but there are also one or more connections within a single layer [30]. The recurrent connections are usually present in the hidden layer, and these connections make recurrent neural networks suited for processing sequences and thus time series data. In this way, at a certain processing step of the network, the hidden layer can make use of information based on the previous input variables, in addition to information based on the current input variables.

Specifically, this is effective because when determining the values of the current hidden layer, information from the hidden layer values of the previous time step is incorporated as well (which depended on the previous input variables). Importantly, this is not limited to the previous time step, as is true for first-order HMMs, because the information at the previous time step already partly consists of some information from the time step before that, and so on [32].

However, a problem with these simpler types of recurrent neural networks is that they can still only make use of the most recent observations, as information stemming from earlier observations "vanishes" as the number of time steps increases [30]. Crucially, this happens in part because the hidden layer units have two different tasks: determining the prediction for the target variable, and determining what information should be passed on to the next processing step.

2.3.3.1 Long Short-Term Memory

The Long Short-Term Memory (LSTM) neural network takes care of this problem by adding gates to the network [33]. In this network, forget, add and output gates are added, that make sure irrelevant information is forgotten, relevant information is being remembered and information that is relevant specifically for the current time step is passed on, respectively [30]. A visualization of a LSTM network is shown in Figure 5. From left

to right we see the LSTM module over time, where X is the input and h is the output of one LSTM module at each time step. More details on LSTM models and their mathematical derivation can be found in Section 3.3.6.

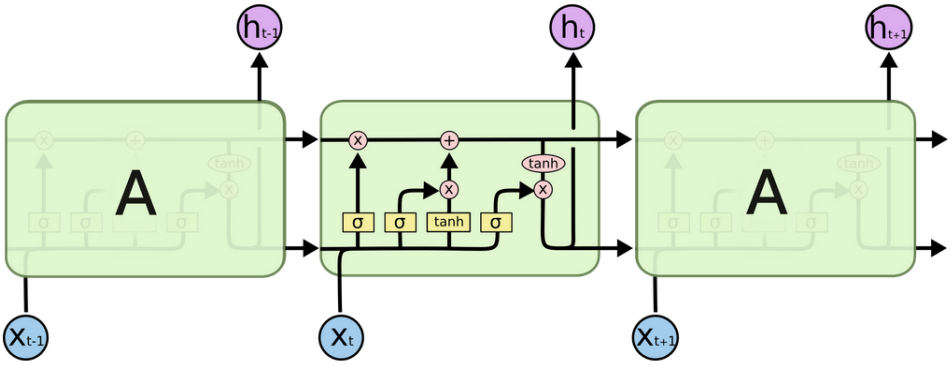


Figure 5. An example of a LSTM network. This image is taken from Olah's (2015) blog post on LSTM models [34].

Because of the network's ability to also process information based on previous observations, including observations from many time steps ago, they are often chosen for time series prediction tasks [35]. Additionally, LSTMs can cope with time series data that are non-stationary, unlike the discussed time series models [15]. However, the improved functionality does come at the price of additional computational load (specifically, additional parameters have to be estimated for the extra layers) [30].

Singh et al. (2020) used a LSTM to predict crowdedness [15]. In their study, the goal was to predict the number of visitors during a large public event in Brussels for the upcoming 30 minutes based on historical data of the same event. WiFi sensors were located throughout two areas that were expected to be most crowded during the event. The total visitor count at a given time was computed by aggregating the number of connected devices to the WiFi sensor over a period of 5 minutes, resulting in total visitor counts for the respective area with an interval of 5 minutes. For each area, there was data for eight non-consecutive days. The days were split into a training and testing set (1-3 days used for training).

They created five versions of a LSTM model, each with a slightly different architecture (e.g. adding bidirectionality), together with a baseline model (Random Walk model, see Section 3.3.1). During training, the input for the LSTM consisted of sets of 12 samples (consecutive samples from a training day) and the output consisted of the 8 subsequent samples, which means that they predicted the next 30 minutes (with 5-minute predictions). Based on the prediction error for the test days, they found that all LSTM models outperformed the baseline model.

Furthermore, out of the LSTM models, the convolutional LSTM performed best in pre-

dicting the number of visitors (improvement of 44% and 48% with respect to the error rate compared to a baseline model, for the two areas respectively). In this model, relevant features are first extracted in convolutional layers based on the historical visitor counts, after which the LSTM is applied to process these features and predict the next visitor count. Moreover, the results showed that as more data was available for training, the prediction accuracy increased.

Another study by Li et al. (2019) also used LSTMs to predict crowdedness [6]. Specifically, their aim was to predict the inflow and outflow of citizens in multiple regions of a city. For this, they used data of bike trajectories in New York and data of taxi trajectories in Beijing, together with weather and holiday data, similar to the study of Zhang et al. (2017) [4]. The bike trajectory data contained of hourly samples of the inflow and outflow per region over a period of several months. The taxi trajectory data contained of 30-minute samples of the inflow and outflow per region over multiple non-consecutive periods (in total over a year worth of data). Non-numerical data on the external factors were encoded as binary variables (e.g. whether there is a holiday at the present time or not).

They divided the city in n regions using a grid map with the size $P \times Q$ where $n = P \times Q$ for both the inflow and outflow in the regions. One observation could then be defined as $X_K \in \mathbb{R}^{2 \times P \times Q}$, where K represents the current time step. Then, the prediction problem was defined as follows: "Given the history of crowd flows data $\{X_t | t = 1, 2, ..., k\}$, predict X_{K+1}." For the bike trajectories data set, they used several months as training data and the last ten days as testing data. For the taxi trajectories data set, the last four weeks were used for testing and the rest of the data was used for training.

For this prediction task, they implemented a spatio-temporal model (called ST-DCCNAL) that consisted of three parts: a spatial part (convolutional neural network), a temporal part (LSTM with an attention mechanism) and external factors. The spatial component models how the crowd inflow and outflow of different regions interact with each other (extracting spatial features), the temporal component models how historical crowd flows can be used as indicators for present crowd flows (extracting temporal features), and the external factors are also being used as predictors for current crowd flows (extracting external factor features). First, the spatial and external factor features are extracted separately, after which these are concatenated and used as input to extract the temporal features and form the prediction for the subsequent time step. During training, they varied the number of samples used as input for the LSTM module for one iteration of training, and found that the optimal length was eight samples.

The proposed spatio-temporal model was compared to a series of baseline models, among

which ARIMA and SARIMA models and other spatio-temporal models. The results showed that the new spatio-temporal model outperforms all baseline models, for both datasets. However, it is important to note that while some external factors were included in the neural network, they were not included in the time series baseline models, while this could have been a possibility (using ARIMAX and SARIMAX models instead).

2.4 Evaluating crowdedness prediction models

For the task of predicting crowdedness, various evaluation methods are being used to evaluate the proposed models. Here, we will mention some methods that are often used for regression and classification prediction problems, respectively.

2.4.1 Error metrics for regression

Typically, the used error metrics are suited for regression problems, since the target variable to predict, the number of visitors, is a continuous variable [36]. In the discussed studies, the Root Mean Squared Error (RMSE) is most often used as the main evaluation metric [4, 5, 6, 7, 22]. In addition other error measures are sometimes reported, such as the Mean Absolute Percentage Error (MAPE), Average Error (AE), or Mean Absolute Error (MAE) [7, 15]. Advantages of the metric MAPE is that it is scale-independent and less sensitive to outliers in the set of errors [15, 37]. In general, these metrics show how far off the predictions are from the observed values. When evaluating models, they are usually not interpreted directly, but relative. For example, a model is compared against a set of baseline models and a decrease in the error metric indicates that this model led to an improvement in performance.

2.4.2 Error metrics for classification

However, when predicting crowdedness, it might not be optimal to frame the prediction task as a regression problem. For example, Singh et al. (2020) mention in their results section that the models are generally underestimating the actual level of crowdedness, which in their case was a good result [15]. Since the goal was to detect crowdedness during a public event, they preferred underestimations so that the chance that crowd management had to act due to a false alarm was relatively small.

In this thesis, the goal was to predict the level of crowdedness, where the focus lies on detecting crowded moments. This means that it is more important to be able to detect a high level of crowdedness, than be able to estimate the number of visitors very precisely.

As a consequence, we can also frame the prediction task as a classification problem, where the aim is to predict the overall level of crowdedness opposed to the exact number of visitors. For this, different levels of crowdedness were defined in terms of classes (see Section 3.1.3.5).

For these type of predictions problems, often used metrics are accuracy (how many predictions are correct when taking all classes together), precision (out of all occasions that class A is predicted, how many times is the actual class A?), recall (out of all occasions that the actual class is A, how many times is class A predicted?), and F_1 (the harmonic mean of precision and recall) [38]. Instead of F_1, the F_β measure can also be used to give more weight to either precision (typically $\beta = 0.5$) or recall (typically $\beta = 2$) [39]. More information on the classification metrics that were used for evaluation in this thesis can be found in Section 3.2.3).

2.5 Contribution to the literature

To conclude, based on the discussed literature, with this research our aim is to contribute to the literature in three ways. First, we included multiple external factors and examined their individual contributions. Second, we expanded the scope of the crowdedness data by using visitor counts based on a combination of pedestrian, bike and car data. Third, we compared a LSTM model with other model types that are also able to incorporate external factors (for example, linear regression and SARIMAX), for a fairer comparison between models.

3. Methodology

In this section, we describe the general approach for all experiments. We elaborate on the different data sources, the procedure for training and evaluating the models, the design of the prediction models, and the methods to examine the contribution of the different external factors on the models' performance. If the design of a specific experiment diverges from the general methodology, details on this can be found in the section that describes the respective experiment.

The goal was to create a model that can accurately predict crowdedness at multiple locations in Amsterdam, two hours ahead in 15-minute time steps. The general procedure for this is depicted in Figure 6. Here, the input to the model are historical observations of the number of visitors based on mobile phone data (the target variable Y), and a set of additional variables based on the external factors (e.g. number of visitors based on camera data). The data is sampled with a 15-minute interval. The output of the model are eight subsequent predictions for the number of visitors, of which the last is selected as the predicted value for the respective time slot.

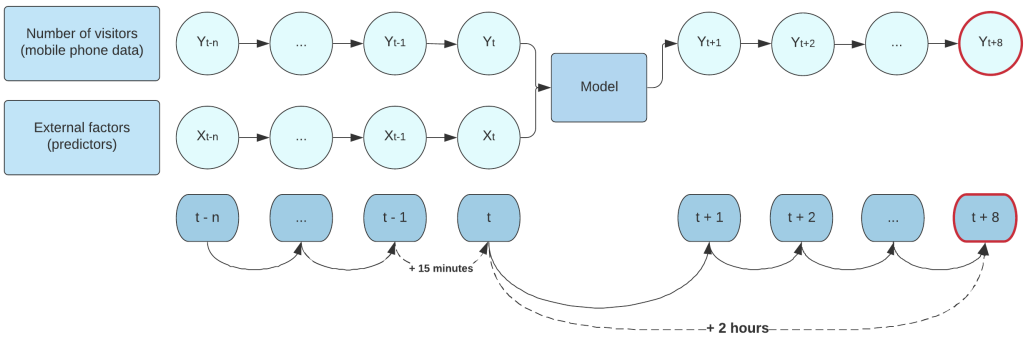


Figure 6. General analysis procedure. This is an example of the general procedure for a prediction window of two hours (the last prediction at time step $t + 8$ is selected as the prediction for this time slot which corresponds to a prediction window of two hours, and this procedure is repeated for each time slot). Here n represents the first time step in the data set.

We can predict crowdedness using the ground truth data in two ways: we use the visitor counts, framing the prediction problem as a regression task, or we use crowdedness levels (visitor counts converted into classes), framing the prediction problem as a classification

task. The selected method depends on the model type used. Details on the conversion to crowdedness levels is discussed in Section 3.1.3.5.

3.1 Data preparation

Complete data sets were created where the visitor counts that serve as the ground truth were combined with the external factors on a 15-minute time scale, for each location. For this, we followed the steps depicted in Figure 7.

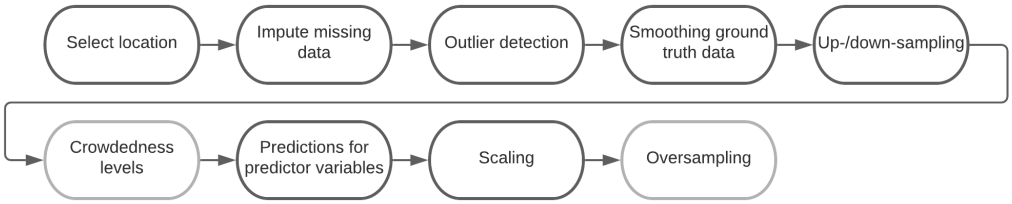


Figure 7. Processing steps taken to prepare the data for modelling. Note that the steps *crowdedness levels* and *oversampling* were optional steps, depending on the model type.

3.1.1 Locations

We predicted the number of visitors at three locations in Amsterdam (see Figure 8). We selected these locations because historically they are known to have crowded moments, and they each represent a different type of location (a square, park and market) with different characteristics.

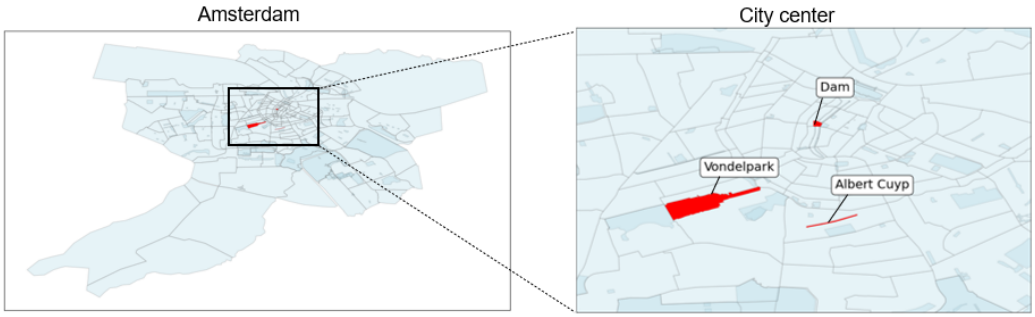


Figure 8. The three target locations.

Dam square. The first location is the Dam square, which is in the center of the city, surrounded by stores and restaurants. Specifically, we focused on the western area of the Dam square (see Figure 9). The exact size of the area for which the number if visitors is recorded is 24688 m^2.

Vondelpark. The second location is Vondelpark, the largest park in the city. Specifically,

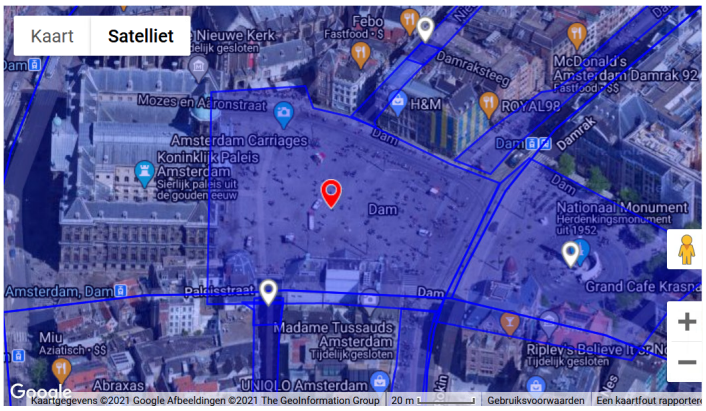


Figure 9. Dam square area [40].

we focus on the eastern area of the park (see Figure 10). The exact size of the area for which the number if visitors is recorded is 634418 m^2.

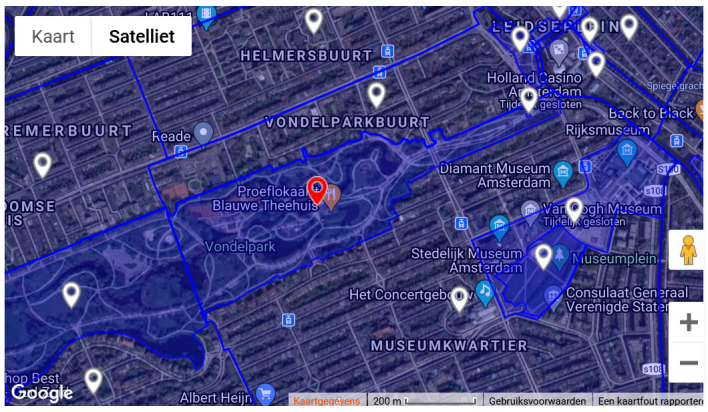


Figure 10. Vondelpark area [40].

Albert Cuyp. The third location is Albert Cuyp, which consists a long street where a daily market takes place (see Figure 11). The exact size of the area for which the number if visitors is recorded is 34535 m^2.

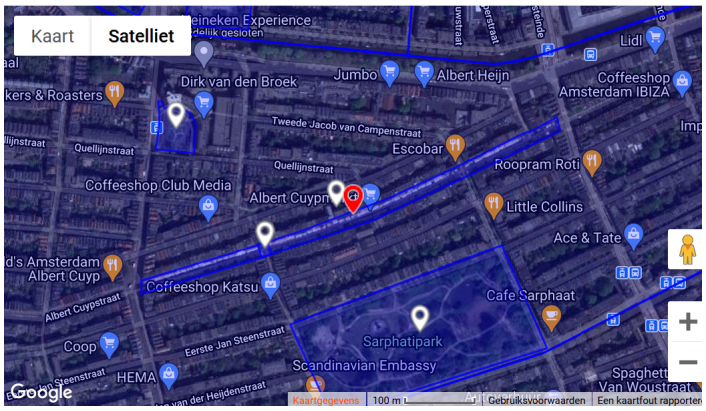


Figure 11. Albert Cuyp area [40].

3.1.2 Data sources

We used several data sources that are either publicly available or property of the municipality of Amsterdam. The exact time period over which the data ranges differs per location and experiment. The target variable to predict are visitor counts based on mobile phone data, and all other variables were used as external factors. They are listed below.

3.1.2.1 Mobile phone data

Unique visitor counts for each location, updated every 15 minutes. This data was gathered by the company Resono [40]. A hyperfencing method is used in which an area is defined, such that as a mobile device enters the area, the device starts to generate location data. Specifically, a statistical model outputs the probability of a device being in the specified area. This only happens for devices for which users have given consent to share location data in certain mobile apps that have a partnership with Resono (e.g. Weeronline). About 6-7% of the Dutch population is actively sharing location data in this manner. Another model is used to scale up the retrieved data (aggregated per area) to provide a representation of the total visitor count at a certain location, and was validated by Resono using road traffic data and public transport data.

The total count of unique visitors consists of stationary, walking and driving visitors (thus including both pedestrian, bike and car data). The data is available since the beginning of October (November for Albert Cuyp), 2020 and is not publicly accessible. An illustration of this data source for Vondelpark is depicted in Figure 12.

Considerations. An important advantage of using this data source as an indicator of crowdedness is its coverage. When gathering data based on mobile phone usage, it is possible to record data for many locations as well as area sizes. For example, the

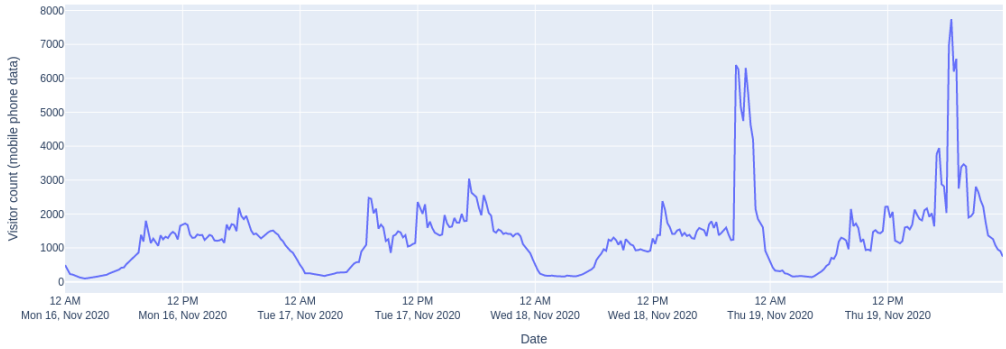


Figure 12. A few days of mobile phone data for Vondelpark.

municipality is in possession of crowdedness data based on mobile phone data for more than a few hundred locations in Amsterdam, covering almost the complete city as a result.

On the other hand, there is an important disadvantage of using this data source as the ground truth for crowdedness. As mentioned above, the raw data is scaled up based on validation data, which means that the data used as the ground truth will always still be an approximation of the real number of visitors. Therefore, we have to interpret the visitor counts in relative terms, rather than absolute. Based on this, it should still be possible to detect the overall level of crowdedness at a location, however we cannot make any claims on the real number of visitors with sufficient certainty.

Additionally, an important downside of using this data source is that occasionally erroneous peaks occur in the data due to measurement errors. This poses the challenge to separate true peaks in the number of visitors from erroneous peaks (i.e. outliers). We tried to deal with this by performing outlier removal, however it remains difficult to be certain that we are removing faulty peaks and keeping true peaks when pre-processing this data source.

3.1.2.2 Crowd Monitoring System Amsterdam (CMSA)

Unique visitor counts for each location, updated every 15 minutes. 27 cameras are placed at different locations in Amsterdam, mostly in the city center (e.g. the shopping street Kalverstraat or the Central Station). These cameras estimate the count, density, speed and direction (north/south) of visitors. Important to note is that this data consists of pedestrians only. In this study, we only used the total visitor count. For each location, the exact camera location overlaps with part of the mobile phone data location (e.g. a camera at the northern entrance of Vondelpark). This data is available since August, 2020 and is not publicly accessible. An illustration of this data source for Vondelpark is depicted in Figure 13.

Considerations. An advantage of using this data source is its reliability. Since the number

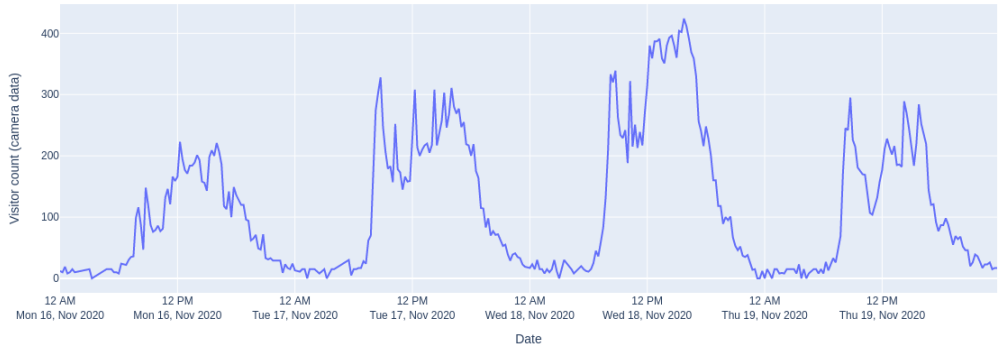


Figure 13. A few days of camera data for Vondelpark.

of visitors is determined based on camera images, this data source is appropriate to use as ground truth. Conversely, a disadvantage of using this data source is that its coverage is limited. The number of locations for which there is data on the level of crowdedness is restricted by the number of cameras and their reach.

3.1.2.3 Weather data

Multiple weather-related variables with measurements for the city as a whole (e.g. temperature, wind speed, and rainfall). The data was gathered by the Dutch weather institute KNMI [41] and is based on the weather station closest to Amsterdam (Schiphol) and consists of historical observations. The sampling rate is hourly. The specific variables that we used in this study are the following: total rainfall in millimeters, duration of sunshine (in units of 10 minutes per hour), average wind speed, temperature in Celsius, and degree of cloudiness from 1 to 8 (where 8 is most clouded). This data is publicly accessible through KNMI's online data platform [42]. An illustration of the temperature variable is depicted in Figure 14.

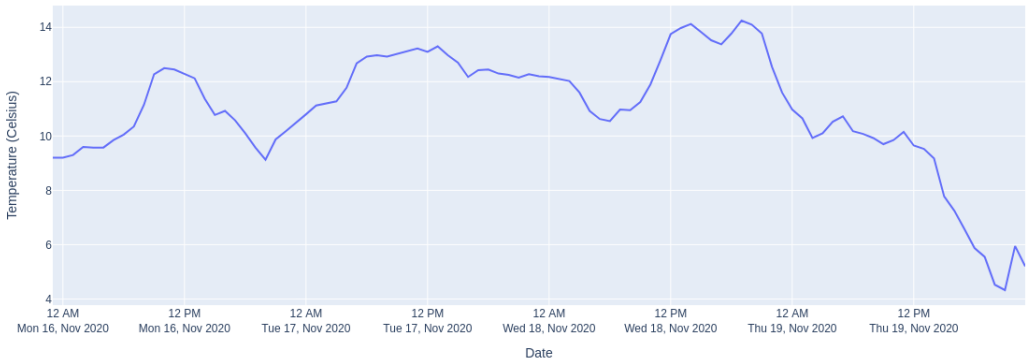


Figure 14. Temperature across a few days.

Considerations. One possible downside of using this data source is that the variation

in the data may be limited because we are only using a couple of months of data per experiment (for example, only covering the autumn/winter period).

3.1.2.4 COVID-19 Government Response Stringency Index

We used the Government Response Stringency Index from the Oxford COVID-19 Government Response Tracker initiative (in collaboration with Oxford University) [43]. This index is determined by means of 18 indicator variables, that represent regulations in different areas (for example containment, healthcare, or economy). Examples are school closures, testing policies and financial support. The index ranges from 1 to 100 and is updated daily. We considered the use of this index in various ways: creating a smoothed version of the index by fitting a third order polynomial to it, and creating two binary variables (whether the index has increased compared to the previous day and whether the index has decreased compared to the previous day). This data is publicly accessible since the beginning of January 2020 [44]. As an example, Table 1 and Table 2 display the duration of the different Stringency Index values that occurred in the selected time periods for the first experiment for the first two and third locations, respectively.

Starting date	End date	Stringency Index
2020-10-17	2020-11-03	62.04
2020-11-04	2020-11-21	65.74
2020-11-22	2020-12-14	56.48
2020-12-15	2020-12-15	76.85
2020-12-16	2020-12-20	84.26

Table 1. Time period of each Stringency Index value that occurred from October 17th, 2020 to December 20th, 2020.

Starting date	End date	Stringency Index
2021-01-09	2021-01-22	78.70
2021-01-23	2021-02-07	82.41
2021-02-08	2020-02-20	78.70

Table 2. Time period of each Stringency Index value that occurred from January 9th, 2021 to February 20th, 2021.

Considerations. Our expectation is that there is a relationship between the severeness of the COVID-19 regulations and crowdedness: for example, as shops and restaurants are closed, we expected that the visitor count will be lower on average for locations nearby these facilities. On the other hand, locations that are still accessible might become more crowded (e.g. the market of Albert Cuyp). An advantage of using this data source as an indicator of COVID-19 regulations is that the regulations are represented by a continuous measure and are thus made quantitative rather than qualitative. A possible downside of using this data source is that there is not a lot of variation in the index over the relevant

period of time, making it more difficult to capture its relationship with the target variable. The COVID-19 regulations are relatively strict during the full time period that is covered in this study, which makes it more difficult to show the effects of the regulations on crowdedness.

3.1.2.5 Parking data

Percentage of parking spots that are in use at parking garages close to the locations. For the Dam square, these are the parking garages QPark Bijenkorf and Rokin. For Vondelpark this is parking garage Byzantium. For Albert Cuyp, this is the parking garage also named Albert Cuyp. This data was gathered by several parking management systems. The data consists of the number of free parking spots and the total capacity, for both short-term visitors and long-term visitors (e.g. subscription holders). The data is updated every 5 minutes and is not publicly available. An illustration of this data source for Vondelpark is depicted in Figure 15.

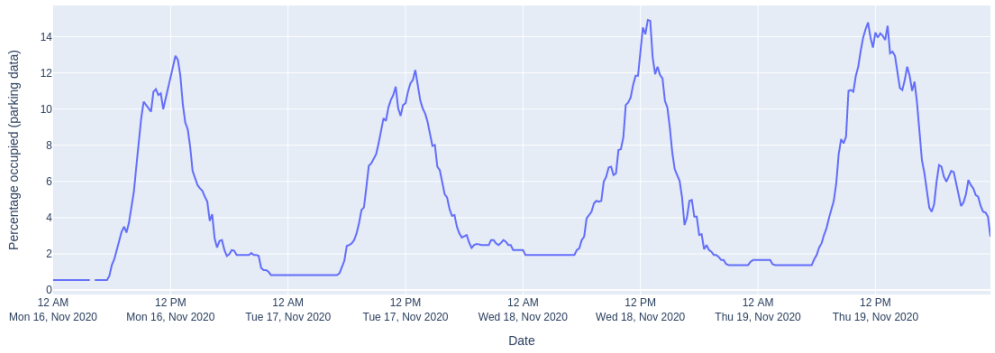


Figure 15. A few days of parking data for Vondelpark.

In the experiments, we focused on the short-term parking spots, since visitors that use these spots are more likely to visit the target locations than long-term visitors. The precise variable that we used is the percentage of parking spots that are in use, calculated using the amount of free short-term spots and the total short-term capacity. To account for the fact that there is a limit to this variable (maximum capacity is reached), we replaced the values for which the maximum capacity has been reached using interpolation.

Considerations. This data source could possibly improve crowdedness predictions because it accounts for visitors arriving by car nearby the target location. We expected that many recent arrivals indicate an increase in the number of visitors at nearby locations. Conversely, if there are very little arrivals we expected that the nearby locations will become less crowded. One disadvantage of this data source is that we do not know what percentage of visitors that arrive at the parking garage will visit the target location. This percentage is probably variable, and therefore it could be difficult to establish a clear

relationship between the two data sources.

3.1.2.6 Holiday data

In some cases we included data on national days by including a variable indicating whether the current day is a national day or not. Examples of holidays are Black Friday or Sinterklaas.

3.1.2.7 Periodic data

The models could benefit from having some additional information on the temporal aspects of the data. Therefore, we included the following variables: weekday or weekend, time of day (morning/afternoon/evening/night), day of the week and hour of the day (transformed into two dimensions using the sine and cosine so that 11:00 PM lies relatively close to 01:00 AM).

3.1.2.8 Historical values ground truth data

We also used some predictor variables that provide information on the historical values of the ground truth. These are lagged versions of the ground truth (value of the previous two hours, or at the same time slot on the previous day), the difference between the value of the previous two hours and the average value, and the difference between the value of the previous two hours and the time step before that.

3.1.3 Pre-processing

Before describing the process of training and testing prediction models, multiple pre-processing steps had to be performed to create complete data sets for each location.

3.1.3.1 Missing data

Since we are using time series data, it is preferable to use interpolation instead of discarding missing data so that we preserve as much data as possible. When interpolating, a continuous function is estimated based on the function of the time series [45]. Missing values can then be replaced by points on this interpolated function at the respective time step. We used cubic spline interpolation for this, as this method seemed to result in realistic interpolated samples (based on training data). The method is similar to polynomial interpolation, but the main difference is that the first and second derivative of the interpolated function are continuous as well [46]. Before performing the interpolation, we made sure that the observations are equally spaced in time by inserting a missing observation if there is none for a certain time slot. Importantly, if for the ground truth complete days were missing, we

discarded these days as interpolation would in this case not give realistic results due to a lack of samples.

3.1.3.2 Outlier detection

We performed outlier removal, in which we replaced outliers using the same interpolation method as for the missing data. Outliers were detected using One-Class Support Vector Machines [47] with a ν parameter of 0.2 (upper-bound on the number of outliers). This an unsupervised machine learning algorithm that learns the probability distribution of a variable and classifies each sample as either an outlier or not an outlier, based on the likeliness that the sample is obtained from this probability distribution. In this way, samples that fall in a low density region will be labelled as outliers. We chose this outlier detection method because it can be applied in a short amount of time, and led to adequate results based on a small validation experiment with data for many locations.

3.1.3.3 Smoothing of ground truth data.

To increase the chance that when a crowdedness threshold is reached it reflects a true crowded moment, we smoothed the target variable by averaging each observation with the observations from the previous two time steps. In this way, it is more often the case that when a threshold is reached the number of visitors remains above the threshold for a longer time period than a single 15-minute observation. Thus, this should reduce some false peaks.

3.1.3.4 Up-/down-sampling data

When combining all variables, it is necessary to either up- or down-sample certain variables so that in the final data set all variables have the same sampling frequency. Only the parking data has to be down-sampled, and for this we took the mean of the observations for each group of 5-minute observations that should be combined. Almost all other data sources had to be up-sampled to a frequency of 15 minutes to match the mobile phone data. For this, either new instances of observations were interpolated, (using cubic spline interpolation) or the most recent value was being used as the new value (forward filling missing values).

3.1.3.5 Crowdedness levels

Depending on the model type, the ground truth data was converted into crowdedness levels. For each location there are fixed thresholds that indicate the level of crowdedness: `not crowded`, `somewhat crowded` and `very crowded`. The thresholds for a specific location are based on the following: location type (e.g. park, square or district), area density and path lengths in the case of parks. The thresholds have been validated using

visualizations of a historical data set with the thresholds applied. It is important to note that the thresholds are being monitored and updated by the municipality iteratively. This results in the thresholds being different for data sets of different time periods. In some cases the thresholds were recalculated with a small correction, either based on information on the true crowdedness (e.g. park closures) or to ensure that some `very crowded` samples were present in the data set.

For all three locations, the largest part of observations fall under the first threshold, and are thus labeled `not crowded` (typically around 75%), followed by observations that fall between the first and second thresholds that are labeled `somewhat crowded` (typically around 20%). Only a small part of observations reaches the second threshold, and are labeled `very crowded` (typically around 5%).

3.1.3.6 Predictions for the external factors

Since in a realistic setting we would not know the values of the external factors during the period we are trying to predict, we decided to replace the observations of the external factors for the prediction window (validation and test data) with predicted observations. For COVID-19 regulations data, future observations are typically known (since the data is updated daily it is safe to use the most recent observation, given that the prediction window is 2 hours). For holiday data, future observations are also known. For the other data sources, future observations are unknown. For these, we used the observations from the same time of either the previous week (camera and parking data) or the previous day (weather data) as predicted observations.

3.1.3.7 Scaling

As a last step, each variable was scaled using z-normalisation. This was necessary because the variables are on different scales, and normalising them so that each variable is on the same scale may improve modelling performance. After the normalisation, all variables had a mean value of 0 and a standard deviation of 1.

3.1.3.8 Synthetic Minority Oversampling Technique

For all locations it holds that most observations can be labeled as `not crowded`, and only a small part of the observations can be labeled as `somewhat crowded` and `very crowded`. However, the focus should be on accurately predicting observations that belong to these minority classes. Because of this, for some model types (the regression models) we oversampled the training data using the Synthetic Minority Oversampling Technique (SMOTE) [48] when training some of the models. The advantage of this technique is that the decision region of the minority class(es) becomes more general. This has the result

that the model has more diverse training examples of the minority class(es) and therefore the predictions might improve.

In this method, an observation of the minority class is selected randomly and based on its k-nearest neighbours a synthetic observation of the minority class is created. First, the values of the predictor variables for one if the k-nearest neighbours is selected randomly. Then, the difference between the values of the predictor variables of the minority class observation and its neighbour is calculated and multiplied by some random number between 0 and 1. After this, these values are added to the values of the predictor variables for the neigbour. This feature vector forms the new synthetic minority class sample. In this study, we used $k = 2$ and an oversampling rate of 50 percent, meaning that if there are 10 observations of the minority class and 50 observations of the majority class, after oversampling there will be $10 + (50 - 10) * 0.5 = 30$ observations of the minority class.

Because we explored different sets of features, there were also cases where the data consisted of nominal features. In these cases, we used a variation of SMOTE called SMOTE-NC (Synthetic Minority Oversampling Technique-Nominal Continuous) [48]. There are two important differences in this method compared to regular SMOTE: 1) when computing the difference between the feature vectors of the minority class observation and its neighbor, for each nominal predictor that is different between the two observations, the median of the standard deviations of all continuous features is added, and 2) when determining the value of a nominal predictor for the new synthetic sample, the value is chosen that occurs most often across all k-nearest neighbors.

When using (non)-linear regression, we used SMOTE in the same fashion with one exception: the target variable for the new synthetic observation was assigned the mean value of the target variable for the respective class. Thus, if a new synthetic observation was created that belonged to the class `very crowded`, its real value (number of visitors) is the average number of visitors for all observations that belong to the `very crowded` class.

3.2 Analysis plan

To test whether the proposed models can adequately predict new unseen data, we split the complete data set into a training, validation and test set. We first selected the best hyper parameter settings (options differ per model type) based on the prediction results for the validation set, and evaluated the resulting models based on the prediction results for the test set.

3.2.1 Data stratification

For each location, the first week(s) of the data set are used as training data, and the subsequent week(s) as validation and test data respectively. The exact dates that correspond to the training, validation and test sets differed for each experiment. The dates were selected so that the distribution of the different crowdedness levels was as similar as possible in the different sets.

3.2.2 Training procedure

Because we are working with time series data, we used a variation of the typical cross-validation procedure that can be applied to time series data that is based on a rolling-origin-recalibration procedure (also termed forward validation, or one-day forward chaining) [17, 37, 49]. In this method, a model is trained up until a certain time step and used to predict the subsequent time step. After this, the model is retrained with the data of this time step included (rolling-origin-recalibration), and used to predict the subsequent time step. This process is repeated until the validation/test period is complete. In this way, the model results will be more robust since an average can be taken over multiple time steps.

The procedure in this study is as follows: for each model type we trained a model for each hyper parameter setting and predicted all time steps for the validation set. As the next time step is predicted, the model is retrained consecutively using the most recent data available. After predictions were made for all time steps, we computed the model performance across all time steps. Based on this result the best model settings were selected for each model type, and new models for each model type were trained and used to predict the unseen test data (this time also using the validation data for training, in the same consecutive manner). An illustration of the rolling-origin-recalibration procedure is shown in Figure 16. In this example we see the data partitioning for the first three test samples.

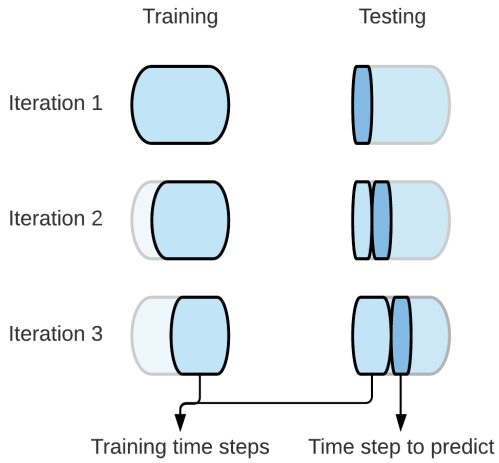


Figure 16. The rolling-origin-recalibration procedure.

3.2.3 Evaluation procedure

To compare the performance across different models, we computed the average of the F_β measure of each class based on the predictions for the validation or test data. This measure is calculated as follows:

$$F_\beta = (1 + \beta^2)\frac{precision * recall}{\beta^2 * precision + recall} \tag{3.1}$$

where β represents the weight [39]. If the weight is higher than 1 recall is considered more important, and if the weight is smaller than 1 precision is considered more important. Recall (also called the true positive rate) is given by:

$$TPR = \frac{TP}{TP + FN} \tag{3.2}$$

where the true positives are cases where class A is both the predicted and true class, and false negatives are cases where any class other than class A is the predicted class while the true class is A. Precision (also called the true negative rate) is given by:

$$TNR = \frac{TN}{FP + TN} \tag{3.3}$$

where the true negatives are cases where any other class than class A is the predicted class and the true class is also any other class than class A, and the false positives are cases where the predicted class is class A, while the true class is any other class than class A.

Here, for the classes `somewhat crowded` and `very crowded`, recall should be given

a higher weight than precision and for the class `not crowded` the opposite, as it is more important that as many crowded moments are being detected as possible at the cost of some false positives. For each model type the model's hyper parameter settings that have the highest averaged F_β score are selected for further testing. If there was a tie, we chose the more parsimonious settings if possible. In addition to the average F_β score, we also examined the overall pattern of results for each class by looking at the confusion matrix, which displays the true versus predicted pairs for each class.

3.3 Model types

We considered the following model types: a Naïve model (baseline model), a linear regression model, an ordinal regression model, a non-linear regression model, a SARIMAX model and a LSTM model (two variants: one for regression and one for classification). The possible hyper parameters of the models differs per model type.

3.3.1 Baseline model

As a baseline model, we used a Naïve model (also termed a Random Walk model) [17]. This model always selects the value of the last observation as the predicted value for the next observation:

$$\hat{y}_{t+h|t} = y_t \tag{3.4}$$

where $\hat{y}_{t+h|t}$ represents the observations for the prediction window ranging from the observation at time step t to observation $t+h$ (h is the prediction window) and y_t represents the observation at time step t.

Here, this means that the model uses the value of the last time slot as the predicted value for the next time slot in the prediction window. In other words, the predictions for this model actually represent a lagged variable of the ground truth (e.g. for a prediction window of 15 minutes, the predictions are equal to the ground truth with a lag of 1). This model does not incorporate the predictor variables, only the target variable is required. Due to its simplicity, the complete training procedure and model selection step were omitted for this model type.

Hyperparameters. This model type has no hyper parameters.

3.3.2 Linear regression model

This is a linear model in which the visitor counts are predicted by means of the predictor variables. This model has the following form:

$$\hat{Y}_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + ... + \beta_m X_{mi} \tag{3.5}$$

where $\hat{Y}_i$ represents the predicted value of the target variable for observation i, X_i represents the observation i of predictor variables 1 to m, and β represents the coefficients for the predictor variables 1 to m (β_0 represents the intercept). The coefficients for the predictor variables are estimated using the least squares method, in which the sum of the squared differences between the predicted and observed values are minimized [50].

Hyperparameters. This model type has no hyper parameters.

3.3.3 Ordinal regression model

This is a linear model in which the crowdedness level is predicted by means of the predictor variables. In this model, the classes are treated as discrete ordinal integers, based on $K-1$ thresholds, $\theta_1 < \theta_2 < ... < \theta_{K-1}$ where $\theta_0 = -\infty$ and $\theta_K = \infty$. The classes can be ordered as follows: [`not crowded` $<$ `somewhat crowded` $<$ `very crowded`]. The model has the same form as in Equation 3.5, with the difference that the coefficients for the predictor variables are estimated using the mean absolute error (MAE) [51] opposed to the least squares method so that a larger distance between the predicted and observed class results in a higher error. The loss function to minimize is as follows:

$$loss(\hat{y}; y) = \sum_{l=1}^{K-1} f(s(l; y)(\theta_l - \hat{y})) \tag{3.6}$$

where

$$s(l; y) = \begin{cases} -1 & \text{if } l < y \\ 1 & \text{if } l \geq y \end{cases} \tag{3.7}$$

Here, where $\hat{y}$ represents the predicted value of the target variable for a specific observation, y represents the actual value of the target variable for the respective observation, l represents the index of the vector of thresholds ranging from 1 to $K-1$, and θ_l represents the threshold given this l. In this model, L2 regularization is used to prevent overfitting, with $\lambda = 1$.

Hyperparameters. This model has no hyperparameters.

3.3.4 Non-linear regression model

This is a non-linear model in which the visitor counts are predicted by means of the predictor variables. In this model, the coefficients for the predictor variables are estimated by optimizing a custom cost function using the Levenberg-Marquardt algorithm. The cost function to optimize is based on the cost per classification error type, where some types of errors have a higher cost than others. The cost function can be depicted as a cost matrix that matches a confusion matrix with true and predicted observations (see Table 3). The coefficients for the predictor variables are estimated using the Levenberg-Marquardt algorithm which is a non-linear iterative optimization technique. It combines gradient descent optimization with Gauss-Newton optimization to find a local minimum of the cost function. This model has the following form:

$$f(\theta) = \sum_{i=1}^{n} f(r_i(\theta)^2) \tag{3.8}$$

where $\theta = (\theta_1, \theta_2, ..., \theta_m$ represents the m coefficients for the set of m predictor variables $X = X_1, X_2, ..., X_r)$, m represents the total number of observations, f represents the cost function and $r_i(\theta)$ represents the i-th element of the vector of residuals. The algorithm finds the best set of coefficients as follows: we start with an initial guess for the coefficients, and at each iteration the coefficients are adjusted using the partial derivatives with respect to the (sum of squares of the) cost function. This procedure is repeated until a stopping criteria has been reached, indicating a that a local minimum of the cost function has been found. The Levenberg-Marquardt method uses a combination of both the gradient descent and Gauss-Newton methods [52]. In gradient descent, coefficients are adjusted using following updating rule:

$$\theta_{t+1} = \theta t - \lambda \nabla f \tag{3.9}$$

where θ_t represents the coefficients at the current iteration, λ represents the learning rate and ∇ represents the partial derivative with respect to the cost function f.

The Gauss-Newton method has the addition of second order partial derivatives. In this method coefficients are adjusted using the following updating rule:

$$\theta_{t+1} = \theta t - (\nabla^2 f(\theta_t))^{-1} \nabla f(\theta_i) \tag{3.10}$$

where ∇^2 represents the second order partial derivative with respect to the cost function f.

Finally, the Levenberg-Marquardt method uses a combination of both the gradient descent and Gauss-Newton methods, which results in the following equation used to iteratively

optimize the coefficients:

$$\theta_{t+1} = \theta t - (H + \lambda diag[H])^{-1} \nabla f(\theta_i) \tag{3.11}$$

where H represents the Hessian matrix evaluated at θt. The Hessian matrix contains the second order partial derivatives of a function.

		Predicted		
		Not crowded	Somewhat crowded	Very crowded
True	Not crowded	0	1	4
	Somewhat crowded	3	0	2
	Very crowded	5	3	0

Table 3. Custom cost matrix based on importance of different kinds of classification errors. An error where the observation is predicted as `not crowded` while the observation actually belongs to the class `very crowded` has the highest cost.

Hyperparameters. This model has no hyperparameters.

3.3.5 SARIMAX

The SARIMAX model is short for Seasonal Auto-Regressive Integrated Moving Average with Exogenous variables. It is a linear model that shows some resemblance to a regression model, but there are some key differences that are discussed next. The SARIMAX model can be perceived as a combination of the following sub models [17]:

AR(p) model. This model predicts future observations by calculating a linear combination of past observations of the target variable. The number of past observations that are used for this is determined by the parameter p. The formula for this model is as follows:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \varepsilon_t \tag{3.12}$$

where y_t represents the target variable at time step t, c represents a constant, ϕ represents the coefficients for the y at the previous time steps 1 to p and ε represents the error.

The value of p can be determined by examining the partial autocorrelation function, which shows the unique autocorrelation at lag p (by controlling for the autocorrelation for preceding lags) [25]. The value of p to select is the highest value for which the autocorrelation is still significant.

I(d). This is not a model, but a differencing computation that is often done to transform the time series into a stationary time series. The trends are removed by replacing the original

observations with the difference between subsequent pairs of observations. The parameter d indicates how many times differencing has to be performed. The formula is given by:

$$y'_t = y_t - y_{t-1} \tag{3.13}$$

The value of d can be determined by examining the decomposed time series [25]. Usually the time series does not have to be differenced more than 1 or 2 times.

MA(q) model. In this model future observations are predicted by calculating a linear combination of past errors. The number of past errors that are used for this is determined by the parameter q. The formula is as follows:

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + ... + \theta_q \varepsilon_{t-q} \tag{3.14}$$

where θ represents the coefficients for the ε at the previous time steps 1 to q.

The value of q can be selected by investigating the autocorrelation function, which shows the autocorrelation at lag q [25]. The highest value of q for which the autocorrelation is still significant should be selected.

Together the models form an ARIMA model. A SARIMA model is an ARIMA model with additional parameters that model the seasonal component of the time series. For this, three extra subparts are added to the model:

AR(P) model. This model is similar to the AR(p) model, with the difference that past observations of the target variable that are used are selected in steps of s instead of 1, where s represents the seasonal period. For example, if the seasonal effect is daily, and one day consists of 96 time steps, s will be 96. The number of times that we take the previous observation by using y_{t-s} is determined by the parameter P. The formula is as follows:

$$y_t = c + \Phi_1 y_{t-s} + \Phi_2 y_{t-2s} + ... + \Phi_{Ps} y_{t-Ps} + \varepsilon_t \tag{3.15}$$

where Φ represents the coefficients for the y at the previous time steps s to Ps. The value of P can be determined in the same manner as for p, only considering time steps of length s instead of 1.

I(D). Differencing can also be performed to remove seasonal effects in addition to trends. Likewise, the parameter D indicates how many times seasonal differencing has to be

performed. The formula is given by:

$$y'_t = y_t - y_{t-s} \tag{3.16}$$

Here is also holds that the value of D can be determined by examining the decomposed time series.

MA(Q) model. As with the AR(P) model, the past errors are modelled in the same way as for the MA(q) model, only using errors at time steps s instead of 1. The number of times that we take the previous error by using y_{t-s} is determined by the parameter Q. The formula is as follows:

$$y_t = c + \varepsilon_t + \Theta_1\varepsilon_{t-s} + \Theta_2\varepsilon_{t-2s} + ... + \Theta_Q\varepsilon_{t-Qs} \tag{3.17}$$

where Θ represents the coefficients for the ε at the previous time steps s to Qs. The value of Q can be determined in the same manner as for q, only considering time steps of length s instead of 1.

All these sub models together form a SARIMA model. The last remaining part of the SARIMAX model is the part that uses the exogeneous variables in a typical linear regression. The formula for this part is very similar to the one in equation 3.5:

$$y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + ... + \beta_n x_{nt} + \varepsilon_t \tag{3.18}$$

where x represents the observations of the predictor variables 1 to n at time step t, and β represents the coefficients for the predictor variables 1 to n (β_0 represents the intercept).

Finally, all sub models can be combined into one SARIMAX model [21]. The complete formula for this is given by:

$$\begin{aligned}
&(1 - \phi_1 L - \phi_2 L^2 - ... - \phi_p L^p) \\
&\times(1 - \Phi_1 L^s - \Phi_2 L^{2s} - ... - \Phi_P L^{Ps}) \\
&\times(1 - L)^d \\
&\times(1 - L^s)^D \\
&\times(y_t - \beta_0 - \beta_1 x_{1t} - \beta_2 x_{2t} - ... - \beta_n x_{nt}) \\
&= (1 + \theta_1 L + \theta_2 L^2 + ... + \theta_q L^q) \\
&\times(1 + \Theta_1 L^s + \Theta_2 L^{2s} + ... + \Theta_Q L^{Qs}) \\
&\times\varepsilon_t
\end{aligned} \tag{3.19}$$

where L represents a lag operator:

$$Ly_t = y_{t-1} \tag{3.20}$$

As a first step the predicted value of y_t is calculated in the regression part, after which this predicted value for y_t is further modelled by the AR and seasonal AR parts. Lastly, the predicted value of y_t is modelled by the MA and seasonal MA part after which only ε at time t remains.

The coefficients are estimated using maximum likelihood estimation [21]. After the model has been fitted, the residuals can be inspected to ensure that they are uncorrelated and have a mean of zero. If this is not the case it is an indication that the model can be further improved [17]. This can be done by performing a Ljung-Box test, which tells us that the residuals are independent if the test statistic is not significant.

Hyper parameters. In some cases it is ambiguous what the best value of the parameters should be. In this case, we searched a set of possible parameter values and selected the model with the settings that lead to highest performance. Due constraints on the time and computational resources we selected the following hyperparameters for the SARIMAX model: $(p = 1, d = 0, q = 1)(P = 0, D = 0, Q = 0, s = 96)$. As we added seasonal parameters to the model, the training time increased to a great extend making it infeasible to test these versions of the model.

3.3.6 LSTM

The LSTM model is a non-linear model and a type of recurrent neural network, with the addition of an add, forget and output gate. These gates regulate what information is discarded and what information is passed on in the network over time [30]. This enables the network to incorporate information over a long range of observations, making this type of model a favorable choice for modelling time series data.

The architecture of a single LSTM module is depicted in Figure 17. The computational steps of a LSTM model are outlined below:

The **forget gate** ensures that the network will not pass through irrelevant information to the hidden layer:

$$f_t = \sigma(U_f H_{t-1} + W_f x_t) \tag{3.21}$$

where f_t is the forget gate at time step t. First, the weight matrix W belonging to the forget gate f in the input layer is multiplied by the observations of the predictor variables x at

time step t. Second, the weight matrix U belonging to the forget gate f is multiplied by the hidden layer representation of the previous time step $t-1$. Third, both components are added and transformed using a sigmoid activation function.

The **add gate** ensures that the network will pass through relevant information to the hidden layer:

$$i_t = \sigma(U_i H_{t-1} + W_i x_t) \tag{3.22}$$

where i_t is the add gate at time step t. Its representation is calculated in the same manner as for the forget gate. However, note that the weight matrices here are different from those used to compute the representation of the forget gate.

The forget and add gate together form a separate **context layer**. First, the context layer of the previous time step is updated using the forget gate:

$$k_t = C_{t-1} \odot f_t \tag{3.23}$$

where C_{t-1} represents the context layer at the previous time step. Then, we compute what information should be passed on to the hidden layer separately from the forget and add gates:

$$g_t = \tanh(U_g H_{t-1} + W_g x_t) \tag{3.24}$$

where g_t is the context update at time step t. Its representation is calculated in the same manner as for the other gates, however also here the weight matrices are separately trained. Also, the components are transformed using a tanh activation function.

Then, the context update and add gate together determine what information should be added to the hidden layer:

$$j_t = g_t \odot i_t \tag{3.25}$$

Finally, the add gate, context update and forget gate together determine the new context layer c at time step t:

$$C_t = j_t + k_t \tag{3.26}$$

The **output gate** ensures that the network will pass through information that is relevant for the prediction of the current time step to the hidden layer:

$$o_t = \sigma(U_o H_{t-1} + W_o x_t) \tag{3.27}$$

where o_t is the output gate at time step t. Its representation is calculated in the same manner as for the other gates, but once again with different weight matrices.

As a last step, the information from the forget, add and output gates together determine the hidden layer representation h at the current time step t:

$$h_t = o_t \odot \tanh(C_t) \tag{3.28}$$

As can be seen in the above formulas, the type of activation functions used in a LSTM are typically sigmoid for the three gates, and tanh for the context update and hidden layer computation. The sigmoid functions is given by:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{3.29}$$

where z represents the vector of values that is being transformed. And the tanh function is computed as follows:

$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{3.30}$$

The LSTM model can have multiple of these modules, and thus multiple hidden layers.

If we perform regression, thus predicting the visitor counts, the last step consists of multiplying the hidden layer representation by another set of weights, resulting in the output layer which contains the prediction for time step t:

$$\hat{y}_t = \sigma(U_a h_t) \tag{3.31}$$

where $\hat{y}_t$ represents the predicted value for the visitor count at time step t, and U_a represents the weight matrix belonging to the output layer a.

If we perform classification, thus predicting the crowdedness level, in the last step the output layer will represent a vector of probabilities equal to the number of classes, opposed to a single number. The probability of the observation belonging to the respective class is calculated using a softmax function:

$$\hat{y}_{it} = \frac{e^{z_i}}{\sum_{k=1}^{k} e^{z_k}} \tag{3.32}$$

where $\hat{y}_{it}$ represents the probability of the observation at time step t belonging to class i, z represents the outcome of $U_a h_t$, and k represents the number of classes. The unit with the highest probability determines the predicted class.

For regression, the RMSE was used as the loss function when training the network and for

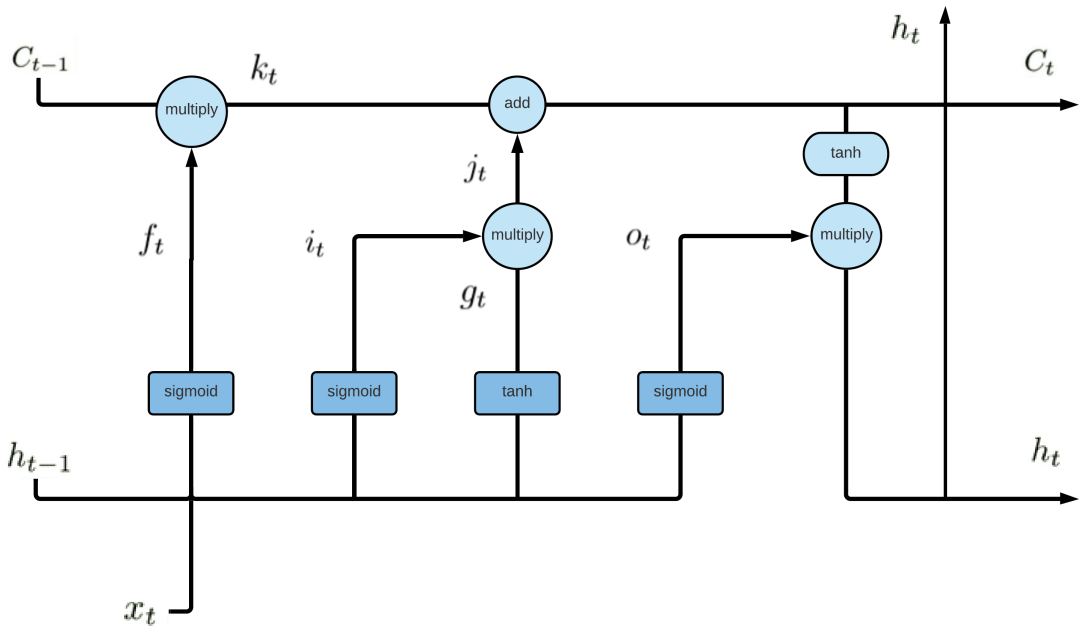


Figure 17. The LSTM architecture. This figure is adapted from Olah (2015) and shows the LSTM module for one time step [34]. Note that the hidden layer at the top of the figure shows that the hidden layer will be used for predicting the target value at the current time step, whereas the hidden layer and context layer at the right of the figure show that they will also be passed on to the LSTM module for the next time step.

classification the cross entropy was used, which is calculated as follows:

$$-\sum_{k=1}^{k} y_{t,k} \log p_{t,k} \tag{3.33}$$

where k represents the number of classes, y represents a binary indicator that is 0 if the observation at time step t does not belong to class k and 1 otherwise, and p is the predicted probability that the observation at time step t belongs to class k.

Following Singh et al. (2020), we used Adam as the optimization method with its default learning rate of 0.001 [15]. Adam performs stochastic gradient-based optimization [53]. Furthermore, we performed dropout regularization by including two dropout layers to prevent overfitting. To prepare the data as input for the LSTM models, we treated the data as separate sequences of 8 time steps (the length of the prediction window), instead of samples consisting of single time steps.

Hyper parameters. For this model, we considered the following hyper parameters: batch size (the number of observations used in one iteration of training the model); the number of epochs (the number of times the complete training data set has been passed through the

network), and the number of units in the hidden layer, using the recommended values of Singh et al. (2020) as a starting point (a batch size of 18, 70 epochs, and 20 units). We varied the hyper parameters incrementally (in both directions) and selected the current hyper parameters as soon as the average F_β score did not further improve by increasing or decreasing the respective hyper parameter. This resulted in the following hyperparameter settings: a batch size of 2, 10 epochs and 20 units.

3.3.7 Technical implementation

All analyses were carried out in Python [54] using the Jupyter Notebook web application [55]. Table 4 shows for each model type the Python library that was used to implement the model.

Model type	**Library**	**Function**
Linear regression	scikit-learn [56]	linear_model.LinearRegression
Ordinal regression	mord [57, 51]	LogisticAT
Non-linear regression	scipy [58]	optimize.least_squares
SARIMAX	statsmodels [59]	tsa.arima.model.ARIMA
LSTM	keras [60]	layers.LSTM

Table 4. Python libraries used to implement the different model types.

3.3.8 Computation time

Table 5 shows the time that it takes for each model type to train the model and use the model to predict the subsequent time slot, for a single 2-hour ahead prediction. We can see that training is fastest for the linear regression model, followed by the ordinal regression model and non-linear regression model. More training time is needed for the LSTM models, with a relatively lower training time for the LSTM model for classification compared to the LSTM model for regression. Furthermore, the SARIMAX model needs a lot more time for training than the other model types. With regard to the time needed for predicting a time step, all models are relatively fast (less than a second). When comparing these times, the LSTM models need most time to make a prediction.

Model type	Training time (s)	Prediction time (s)
Baseline	-	< 0.001
Linear regression	0.113	0.005
Ordinal regression	0.883	< 0.001
Non-linear regression	3.342	< 0.001
SARIMAX	178.729	0.010
LSTM (regression)	88.219	0.076
LSTM (classification)	38.538	0.098

Table 5. Training and prediction time in seconds for the different model types, for a single 2-hour prediction. Here, six weeks of training data were used.

3.4 Predictor variable importance

We analysed the importance of the different predictor variables in three ways: First, for the best performing model(s), we combined a forward feature selection procedure together with the mRMR algorithm [61]. As a first step, for each location a ranking of the predictor variables was computed based on the mRMR algorithm, which provides a mutual information score for each variable (where the highest ranked variable has a strong relationship with the target variable, and a weak relationship with the other predictor variables). On the first iteration, the model is trained using the highest ranked variable, and on each iteration the subsequent variable in the ranking is added to the model (forward feature selection). The advantage of using this method is computational efficiency: we prevent having to train models repeatedly for all possible combinations of predictor variables.

Second, for some of the models we examined the model weights (when using all predictor variables) to see what predictor variables are most important for the model's predictions. For this, we averaged the model weights across all training iterations. Third, we performed some error analyses to see how certain predictor variables might contribute to erroneous predictions (or what information might still be lacking to reach an accurate prediction). For this, we selected some incorrectly labeled samples from the test set and examined the predictor variable values for the respective samples.

4. Preliminary experiment

4.1 Objective

In this section we discuss the experimental results of a first experiment in which we compared the performance of the different models and examined the contribution of the different external factors. First, we briefly mention some additional details on the experimental set-up. Then, we show an exploratory analysis the training data for each location to gain some first insights in crowdedness patterns and the relationship of the number of visitors based on mobile phone data with the other data sources. After this, we provide an overview of the modelling results for all three locations. Subsequently, we further inspected the models' performance and characteristics for a subset of the models by examining the effect of the training set size, the effect of oversampling, the importance of the predictor variables, and an error analysis.

4.2 Experimental set-up

For the locations Dam square and Vondelpark, the selected data ranges from October 17th, 2020 to December 20th, 2020, local time (UTC +1). For Albert Cuyp, the selected data ranges from January 9th 2021 to February 20th, 2021, local time (UTC +1). We considered varying training set sizes, from one to four weeks.

4.3 Exploratory data analysis

4.3.1 Dam square

Figure 18 shows an example of a crowded day at the Dam square. We see that during the afternoon the first crowdedness threshold is reached, and for a short time period between 02:00 and 04:00 PM the second crowdedness threshold is reached[1].

In Figure 19 we see the average number of visitors during the weekdays versus the weekend.

[1] As explained in Section 3.1.2.1, the visitor counts in these graphs should be interpreted as relative counts instead of absolute counts.

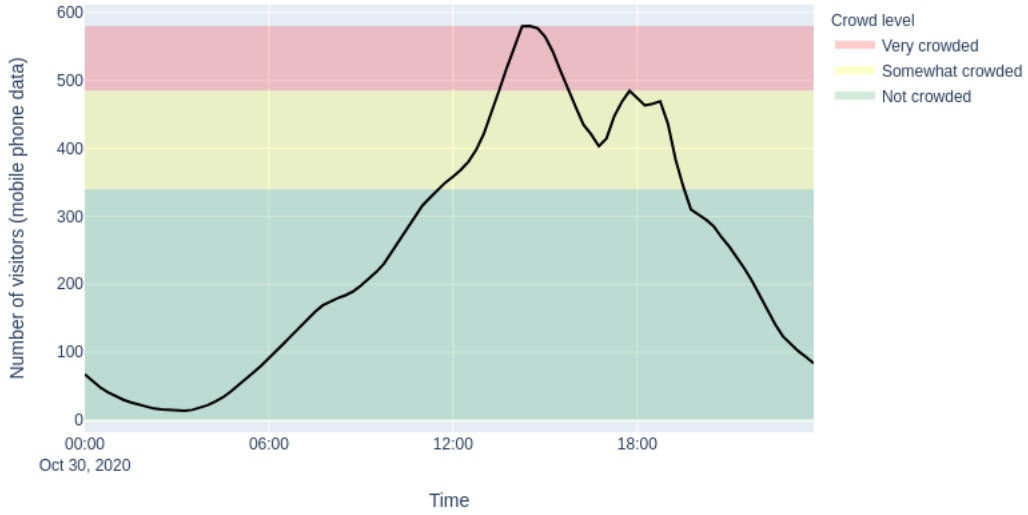


Figure 18. Example of a crowded day.

The difference in daily crowdedness pattern is that there are less visitors in the morning hours during the weekend than during the weekdays. Then, if we look at the average number of visitors for every day of the week, see that especially Saturday is a busy day (with a peak of visitors in the afternoon), whereas Sunday is a relatively quiet day. Thus, for this location the weekend vs. weekday distinction might not be very informative, since the patterns of visitors on Saturdays and Sundays are not very similar. Furthermore, on Thursday evenings we see a peak in visitors, which might be related to shops closing later at night that day.

Figure 21 shows the number of visitors on Black Friday compared to the average number of visitors on Fridays. Compared to the average pattern on a Friday, there is a peak in the number of visitors at the Dam square around 12:00 PM. Interestingly, there are less visitors later in the afternoon than on a regular Friday. Finally, Figure 22 shows the relationship between COVID-19 regulations and the average number of visitors during different times of day. It appears that for the middle value of the Stringency Index there are actually more visitors on the Dam square on average compared to the other two values of the Stringency Index, for all times of day. This is an unexpected finding, however, it is important to note that the lowest value of the index occurred for a shorter time period than the other two index values, making this value more difficult to compare with the others.

In Table 6 we see the correlations between the number of visitors and the predictor variables. There is a moderately positive relationship between the number of visitors

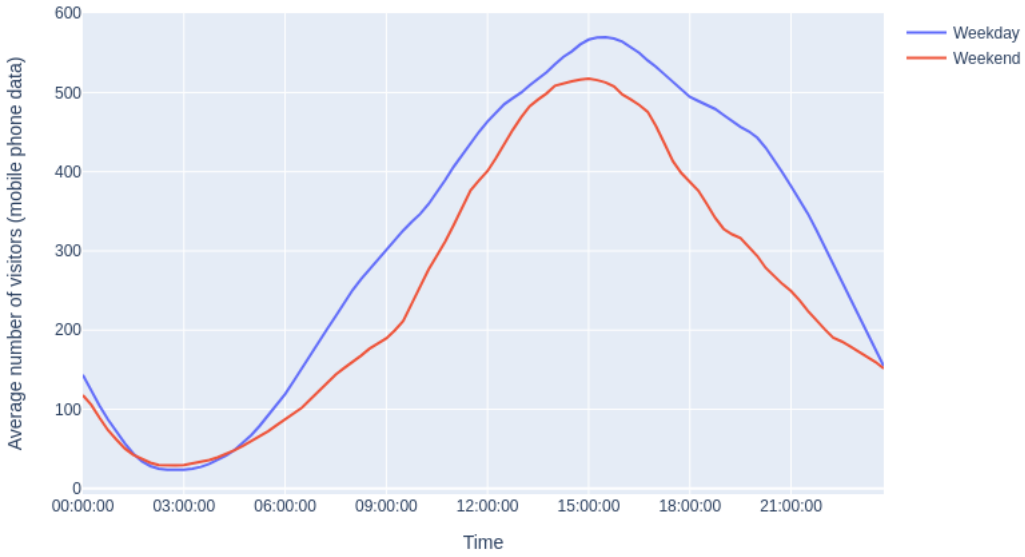


Figure 19. Average number of visitors weekday vs. weekend.

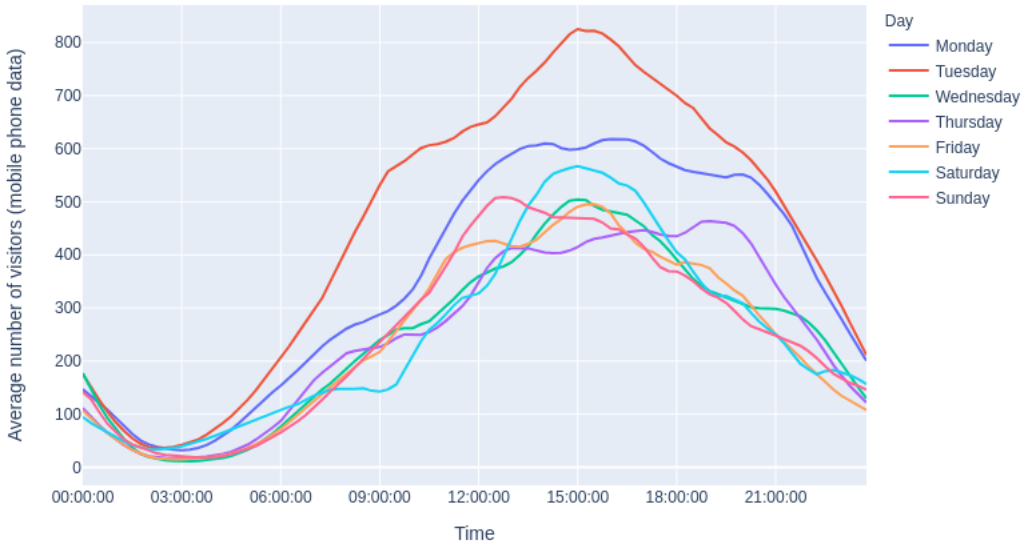


Figure 20. Average number of visitors weekday vs. weekend.

based on mobile data and the number of visitors based on camera and parking data (with a weaker relationship for the Rokin parking garage). Furthermore, there is also a moderate correlation with lagged versions of the number of visitors based on mobile phone data. For the selected time period there is no clear relationship between the COVID-19 regulations and crowdedness based on mobile phone data, as well as for holidays. Lastly, there is a moderate relationship with some of the time indicators: typically there are more

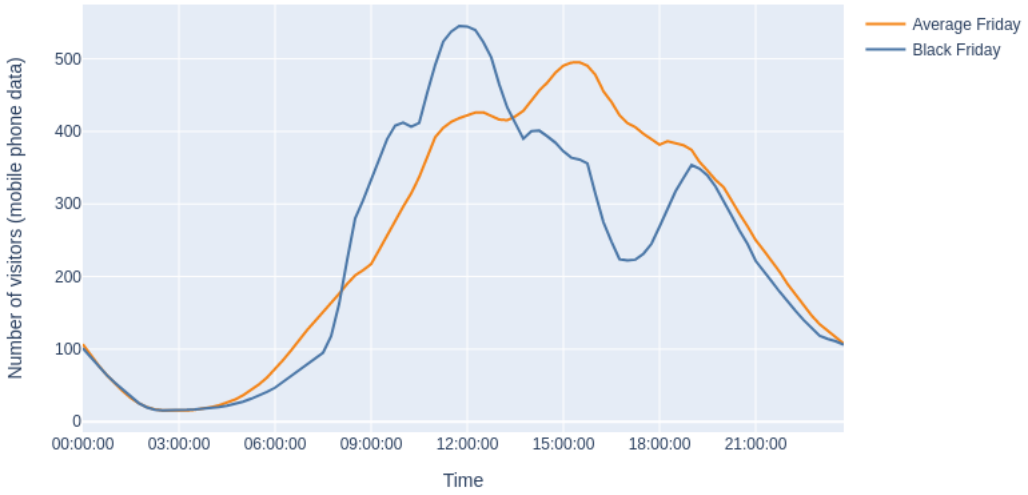


Figure 21. Number of visitors on Black Friday compared to the average Friday.

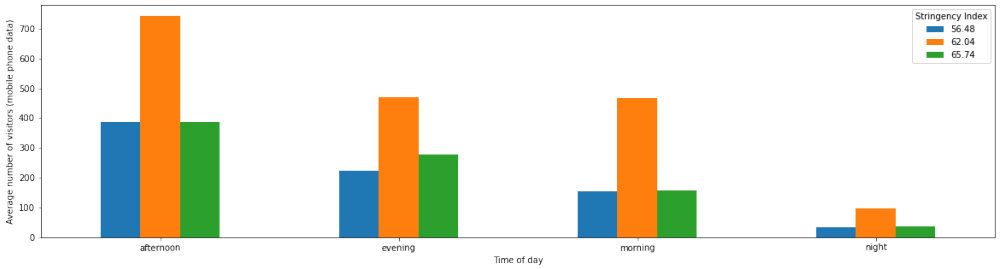


Figure 22. Average number of visitors for each value of the Government Response Stringency Index (severeness COVID-19 regulations).

visitors during the afternoon and less visitors during the night, as expected. The strongest relationships among the predictors are the positive correlations between the camera and parking data (0.49 for Rokin and 0.71 for Bijenkorf parking garage, respectively).

Predictor variable	Data source	Type	Coefficient
number of visitors	camera	continuous	0.63
% occupied spots (Bijenkorf)	parking	continuous	0.46
number of visitors previous 2 hours	mobile phone	continuous	0.76
number of visitors previous day	mobile phone	continuous	0.77
number of visitors previous week	mobile phone	continuous	0.46
afternoon	time	binary	0.60
night	time	binary	-0.61
hour (sine)	time	continuous	-0.62
hour (cosine)	time	continuous	-0.54

Table 6. Correlations between the number of visitors (ground truth) and the predictor variables for which the correlation is higher than 0.4. For continuous predictors, this is the Pearson correlation; for binary predictors, this is the point-biserial correlation.

4.3.2 Vondelpark

Figure 23 shows an example of a crowded day at Vondelpark. We can see that from 08:30 AM onwards the park becomes somewhat crowded, reaching the first crowdedness threshold. Then around 12:00 PM, the number of visitors reaches the second threshold, with a peak in crowdedness around 09:00 PM.

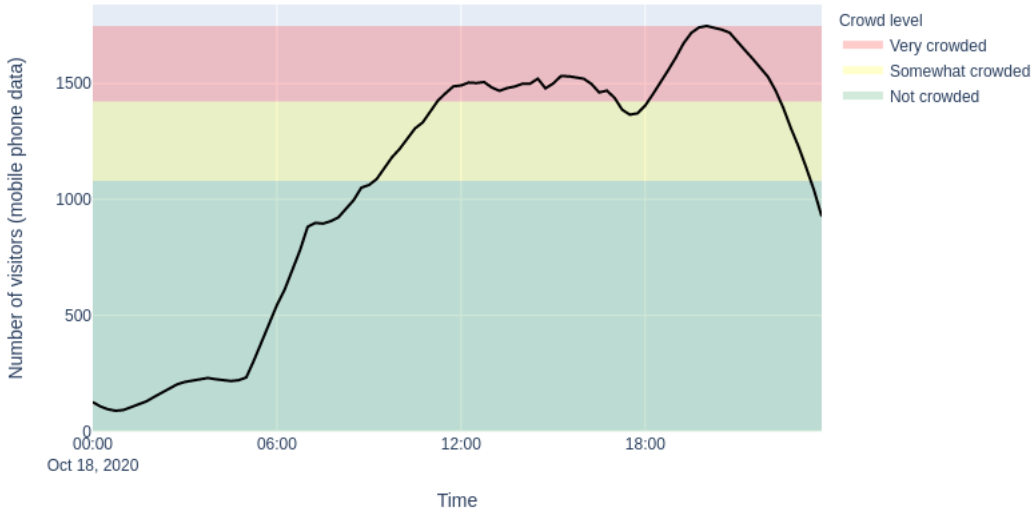


Figure 23. Example of a crowded day.

In Figure 24 we see the average number of visitors during the weekdays versus the weekend. There is a noticeable delay in the increase in visitors starting in the morning in the weekend compared to the regular weekdays. Furthermore, during the beginning and middle of the afternoon the park is somewhat more crowded during the weekends, while during the end

of the afternoon and later in the evening, the park is more crowded during the weekdays. When we consider the average number of visitors for every day of the week, we also see that there is a peak around 09:00 AM on Thursdays and relatively many visitors during the later afternoon on Mondays (see Figure 25).

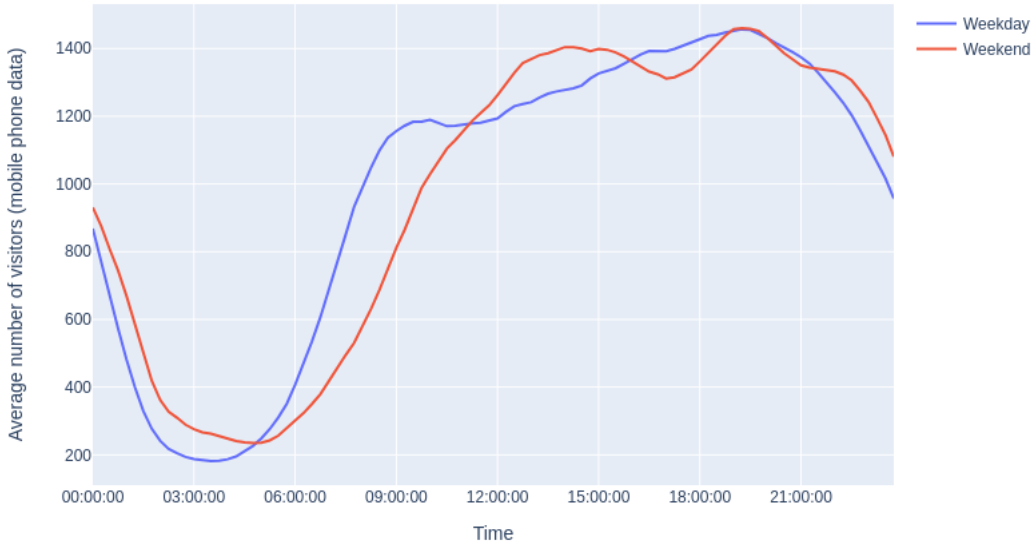


Figure 24. Average number of visitors weekday vs. weekend.

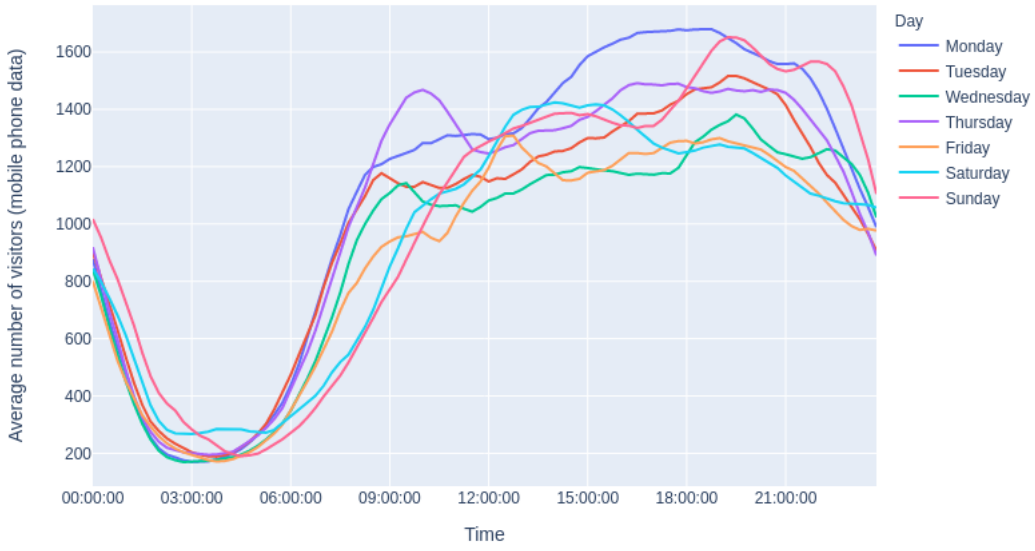


Figure 25. Average number of visitors weekday vs. weekend.

Figure 26 shows the number of visitors on Black Friday compared to the average number of visitors given all Fridays. Overall, the patterns of the number of visitors throughout the day are relatively similar. On Black Friday, there was a peak in the number of visitors in the

beginning of the afternoon. Lastly, Figure 27 shows the relationship between COVID-19 regulations and the average number of visitors during different times of day. It seems that for the afternoon and evening the number of visitors decreases slightly as the Stringency Index increases from 56 to 62, and remains about the same after it increases to 65. Overall, no obvious relationship seems to be present.

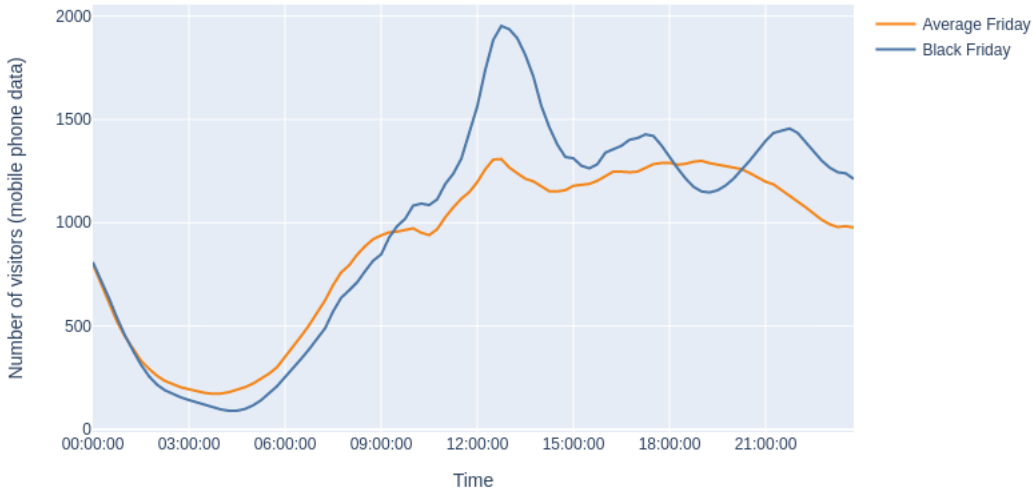


Figure 26. Number of visitors on Black Friday compared to the average Friday.

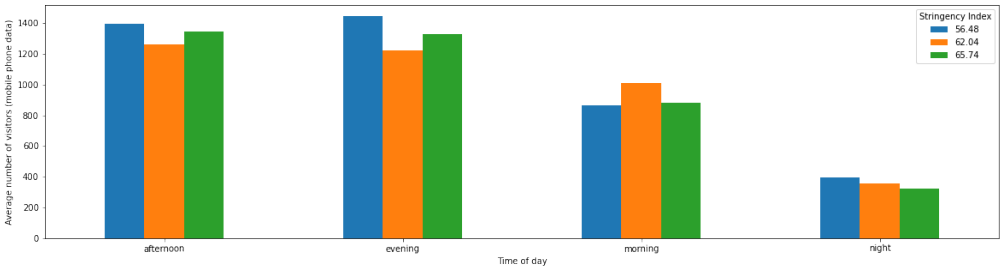


Figure 27. Average number of visitors for each value of the Government Response Stringency Index (severeness COVID-19 regulations).

Table 7 depicts the correlations between the number of visitors and the predictor variables. There is a moderately positive relationship between the number of visitors based on mobile data and the number of visitors based on camera and parking data. There is also a moderate to strong correlation between the number of visitors and lagged versions of the same variable. Furthermore, similar to the Dam square, the positive correlation with `afternoon` and the negative correlations with `night` and `hour (cosine)`/`hour (sine)` indicates that there are relatively more visitors during the afternoon and less visitors during the night. Finally, there are also no clear relationships between the COVID-19 regulations and the holidays. Notable correlations between the predictors themselves are

the moderate to high positive correlation between the number of visitors based on camera and parking data (0.66), and a moderately positive correlation between `sunshine` and the number of visitors based on camera (0.51) and parking data (0.40).

Predictor variable	Data source	Type	Coefficient
number of visitors	camera	continuous	0.45
% occupied spots	parking	continuous	0.43
number of visitors previous 2 hours	mobile phone	continuous	0.75
number of visitors previous day	mobile phone	continuous	0.78
number of visitors previous week	mobile phone	continuous	0.67
night	time	binary	-0.68
hour (sine)	time	continuous	-0.53
hour (cosine)	time	continuous	-0.50

Table 7. Correlations between the number of visitors (ground truth) and the predictor variables for which the correlation is higher than 0.4. For continuous predictors, this is the Pearson correlation; for binary predictors, this is the point-biserial correlation.

4.3.3 Albert Cuyp

Figure 28 shows an example of a crowded day at Albert Cuyp market. In this figure we can see that the that the second crowdedness threshold is reached around 12:00 PM, then briefly drops below the threshold again around 01:30 PM and subsequently reaches the second threshold once more until the start of the evening around 06:00 PM.

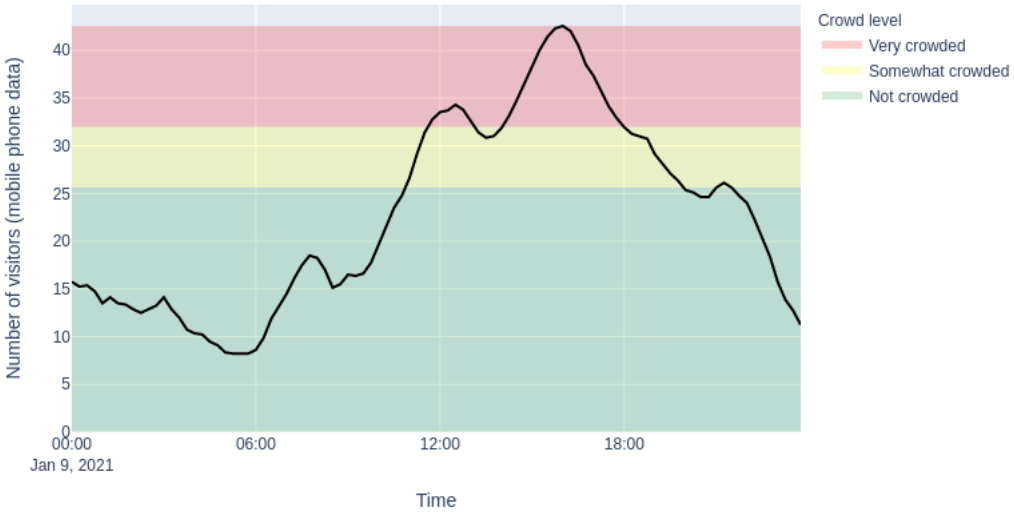


Figure 28. Example of a crowded day.

In Figure 29 we see the average number of visitors during the weekdays versus the weekend. There is a large difference in the number of visitors during the afternoon, where the visitor

count is higher during weekends than weekdays. Then, if we consider the average number of visitors for every day of the week, we see that the difference in weekend vs. weekdays is caused by the Saturday late morning to late afternoon being more crowded than all other days (see Figure 30). Additionally, the number of visitor is actually slightly lower on Sundays.

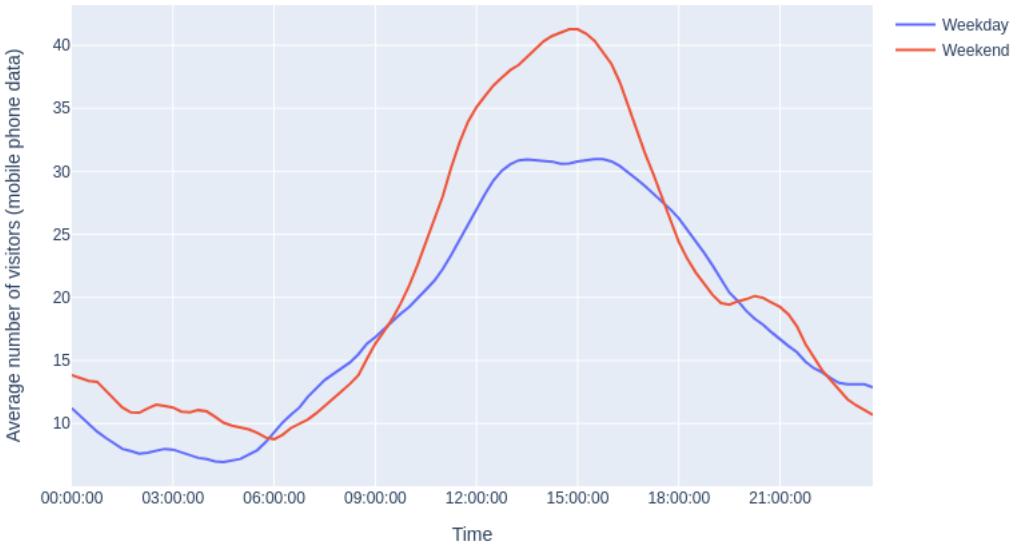


Figure 29. Average number of visitors weekday vs. weekend.

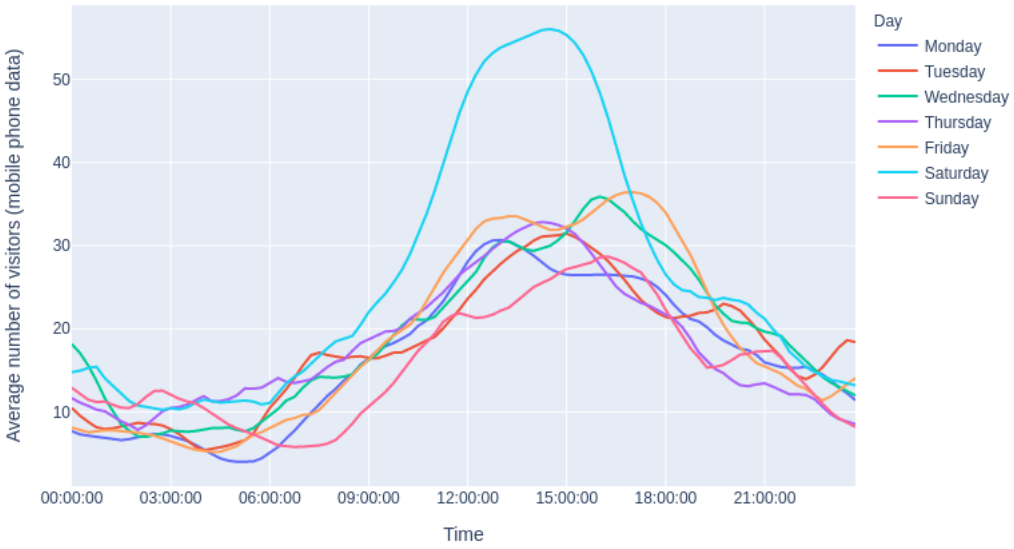


Figure 30. Average number of visitors weekday vs. weekend.

Figure 31 shows the relationship between COVID-19 regulations and the average number of visitors during different times of day. In this figure we can see that only for the afternoon

there is a difference between the two Stringency Index values, where the visitor count is somewhat higher as the COVID-19 regulations are stricter. A possible reason for this could be that the market was still open, whereas other locations were closed down, resulting in an increase of visitors at this location.

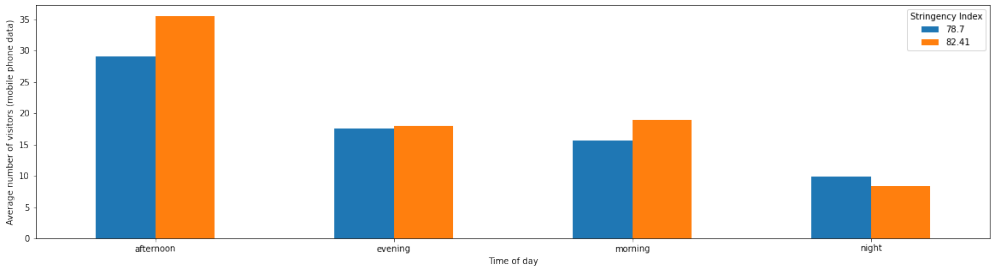


Figure 31. Average number of visitors for each value of the Government Response Stringency Index (severeness COVID-19 regulations).

In Table 8 we see the correlations between the number of visitors and the predictor variables. In general, the correlations are relatively similar to those for the other two locations. There is a moderately positive relationship between the number of visitors based on mobile data and the number of visitors based on camera data. Interestingly, for this location this relationship is not found for the parking data. Furthermore, there is a moderate correlation between the number of visitors and lagged versions of itself. The positive correlation with `afternoon` and the negative correlations with `night` indicate that there are relatively more visitors during the afternoon and less visitors during the night. Lastly, there are also no clear relationships between the COVID-19 regulations and the number of visitors. Likewise, there is a weak negative correlation between the camera data and parking data (-0.30).

Predictor variable	**Data source**	**Type**	**Coefficient**
number of visitors	camera	continuous	0.71
number of visitors 2 hours ago	mobile phone	continuous	0.76
number of visitors previous day	mobile phone	continuous	0.63
number of visitors previous week	mobile phone	continuous	0.75
afternoon	time	binary	0.62
night	time	binary	-0.47
hour (sine)	time	continuous	-0.45
hour (cosine)	time	continuous	-0.75

Table 8. Correlations between the number of visitors (ground truth) and the predictor variables for which the correlation is higher than 0.4. For continuous predictors, this is the Pearson correlation; for binary predictors, this is the point-biserial correlation.

4.4 Results

4.4.1 Model evaluation

Tables 9, 10 and 11 show the performance of the different model types for Dam square, Vondelpark and Albert Cuyp, respectively.

Model	Training	Validation	Testing	Training weeks
Baseline	0.46	0.46	**0.48**	4
Linear regression	0.48	0.48	0.48	2
Ordinal regression	0.43	0.47	0.42	4
Non-linear regression	0.44	0.48	0.38	1
SARIMAX[2]	0.33	0.44	0.46	2
LSTM (regression)	0.40	**0.49**	0.44	3
LSTM (classification)	0.41	0.45	0.48	1

Table 9. Average F_β scores for the different model types on the training, validation and test data for Dam square. The number of training weeks that resulted in the highest F_β score on the validation set is also given.

Model	Training	Validation	Testing	Training weeks
Baseline	0.53	0.61	**0.58**	4
Linear regression	0.61	0.61	0.47	4
Ordinal regression	0.60	0.62	0.46	4
Non-linear regression	0.53	0.58	0.38	4
SARIMAX	0.64	0.58	0.52	2
LSTM (regression)	0.68	0.64	0.48	1
LSTM (classification)	0.63	**0.65**	0.47	1

Table 10. Average F_β scores for the different model types on the training, validation and test data for Vondelpark. The number of training weeks that resulted in the highest F_β score on the validation set is also given.

The results show that overall, the average F_β scores on the training and validation sets are in a similar range for all model types, which indicates that the different crowdedness levels are difficult to distinguish given the available data. If we consider the results on the validation set, for Dam square most model types slightly outperform the baseline model (except for SARIMAX and LSTM for classification), for Vondelpark only the ordinal regression and LSTM model types perform better than the baseline model, and finally for Albert Cuyp the ordinal regression, SARIMAX and LSTM for classification performed better than the baseline model.

For Dam square, the LSTM model for regression has the highest score, although the

[2]Due to constraints on the time and computational resources we only considered a training set size of either 1 or 2 weeks.

Model	Training	Validation	Testing	Training weeks
Baseline	0.57	0.61	0.58	4
Linear regression	0.61	0.58	**0.83**	2
Ordinal regression	0.63	**0.73**	0.57	3
Non-linear regression	0.46	0.33	0.54	1
SARIMAX	0.59	0.63	0.62	1
LSTM (regression)	0.60	0.57	0.57	3
LSTM (classification)	0.64	0.65	0.69	2

Table 11. Average F_β scores for the different model types on the training, validation and test data for Albert Cuyp. The number of training weeks that resulted in the highest F_β score on the validation set is also given.

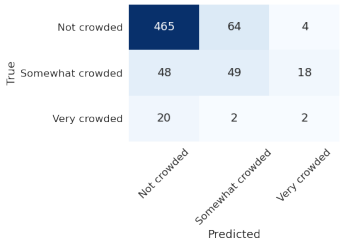
difference in scores is very small. For Vondelpark, the LSTM model for classification obtained the highest score, however, again the difference in scores is small. For Albert Cuyp, the score of the ordinal regression model stands out compared to the other model types.

A more detailed comparison between the ground truth and predictions on the validation set is shown in Figure 32. Based on these confusion matrices we can see that for when we compare the models to the baseline model, on the one hand more `very crowded` cases are being detected, while on the other hand sometimes `somewhat crowded` cases are predicted to be `very crowded` (thus, an increase in false alarms). Importantly, this is not true for the `not crowded` cases. For Dam square, the predictions versus the ground truth are also shown over time in Figure 33.

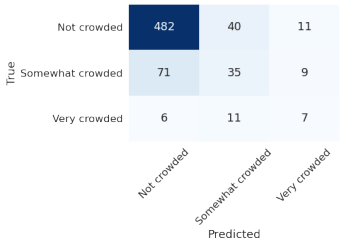
Then, if we look at the results on the test set, for Dam square and Vondelpark, none of the models outperform the baseline model. For Albert Cuyp, only the linear regression model and LSTM model for classification perform better than the baseline model. Thus, the findings are very different for the test set compared to the validation set. This means that overall, the models do not generalize well to new unseen data. It should be taken into account however that there are less crowded moments in the test set than in the validation (and train) set, which means that errors in the test set for crowded samples have a larger influence on the score than for the validation set.

When taking the results on the validation and test sets together, the best model type for Dam square seems to be either the linear regression model or the LSTM model for regression. For Vondelpark this seems to be either the LSTM model for regression or for classification. Lastly, for Albert Cuyp this is the LSTM model for classification (the results are more consistent than for the linear and ordinal regression models). More details on the predictions versus the ground truth for these models on the test set is shown in Figure 34.

Dam square

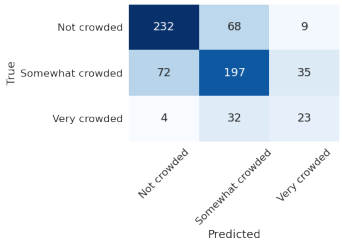


(a) Baseline model.

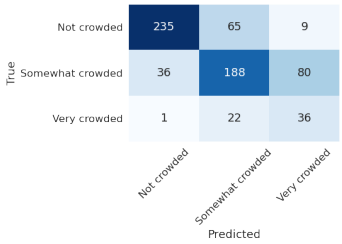


(b) LSTM model (regression).

Vondelpark

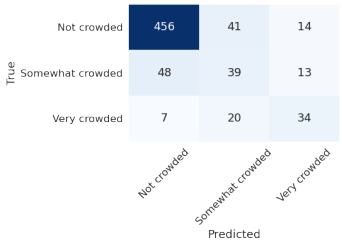


(c) Baseline model.

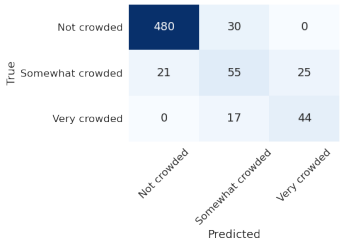


(d) LSTM model (classification).

Albert Cuyp



(e) Baseline model.



(f) Ordinal regression model.

Figure 32. Confusion matrices of the ground truth and predictions for the validation set.

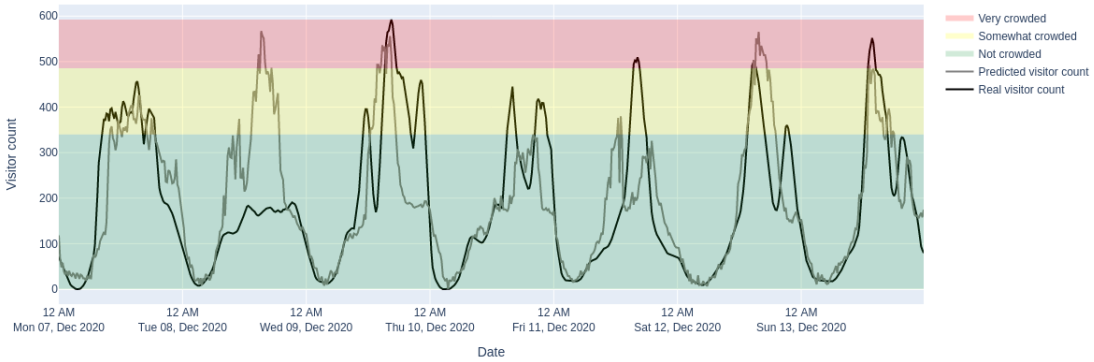
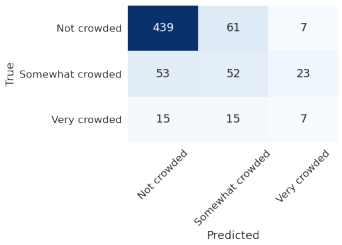


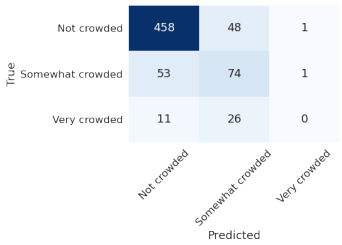
Figure 33. Predictions vs. ground truth for the validation set for the LSTM model (regression) for Dam square.

Here we can see that the crowded moments are not being detected, except for the Albert Cuyp location. Figures 35 and 36 show the predictions versus the ground truth on the test set for Dam square and Vondelpark, respectively.

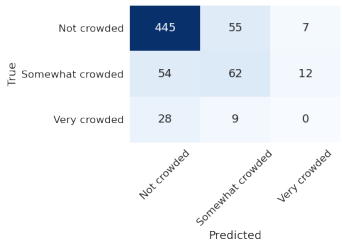
Dam square



(a) Baseline model.

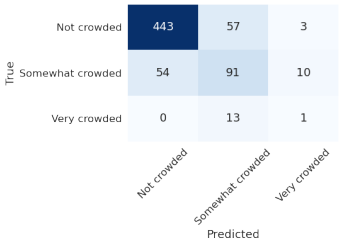


(b) Linear regression model.

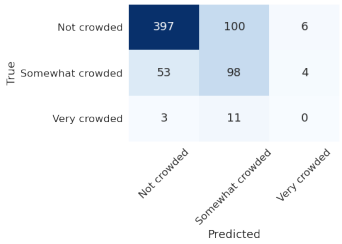


(c) LSTM model (regression).

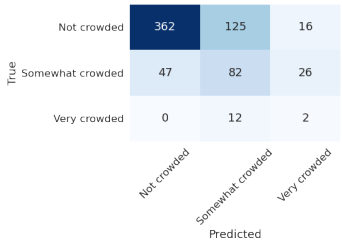
Vondelpark



(d) Baseline model.

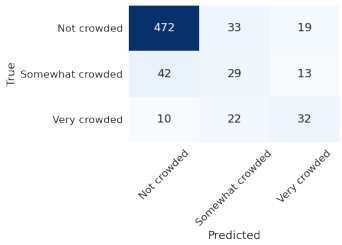


(e) LSTM model (regression).

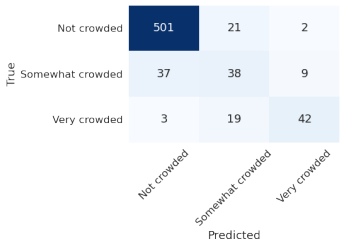


(f) LSTM model (classification).

Albert Cuyp



(g) Baseline model.

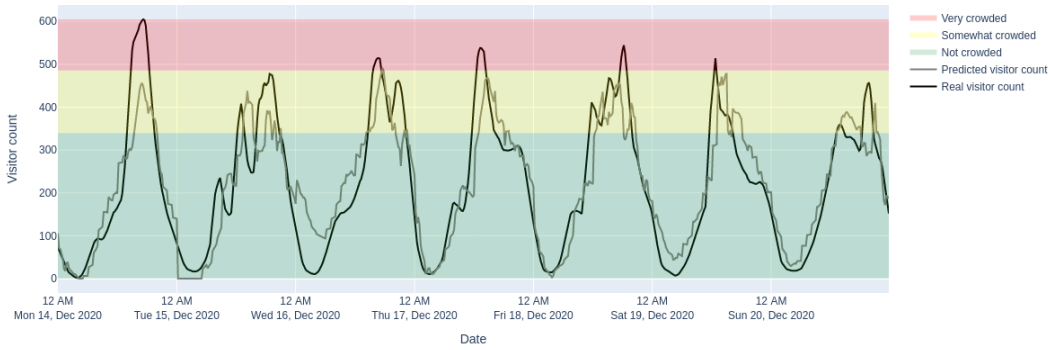


(h) LSTM model (classification).

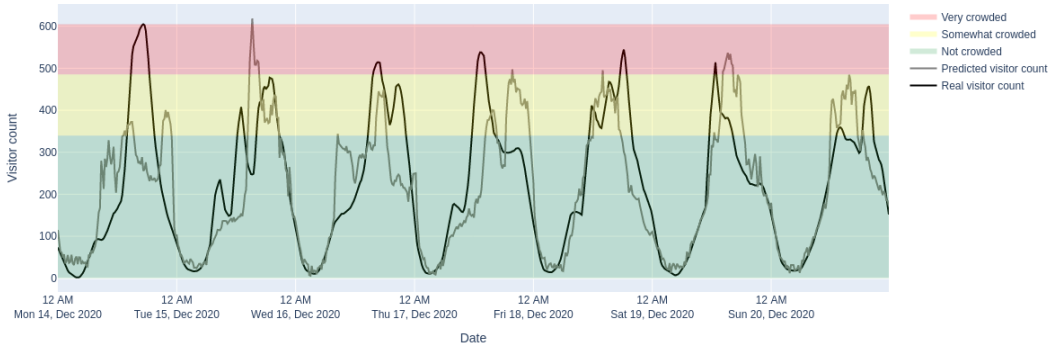
Figure 34. Confusion matrices of the ground truth and predictions for the test set.

Lastly, the selected number of training weeks varies across the model types and locations. Overall, it seems that the model benefits from having at least two weeks of training data. However, this is not true for the LSTM models for Vondelpark, here only one training week was selected. To examine whether the lack of training data might explain the relatively low score on the test set for these models, we predicted the test set once more for Vondelpark, this time with a larger training set of 4 weeks (see Table 12). From the results we can see that the models do not (or only slightly) perform better when provided with more training data.

To conclude, in most cases the LSTM models seem to perform best, although for the Dam



(a) Linear regression model.



(b) LSTM model (regression).

Figure 35. Predictions vs. ground truth for the test set for Dam square.

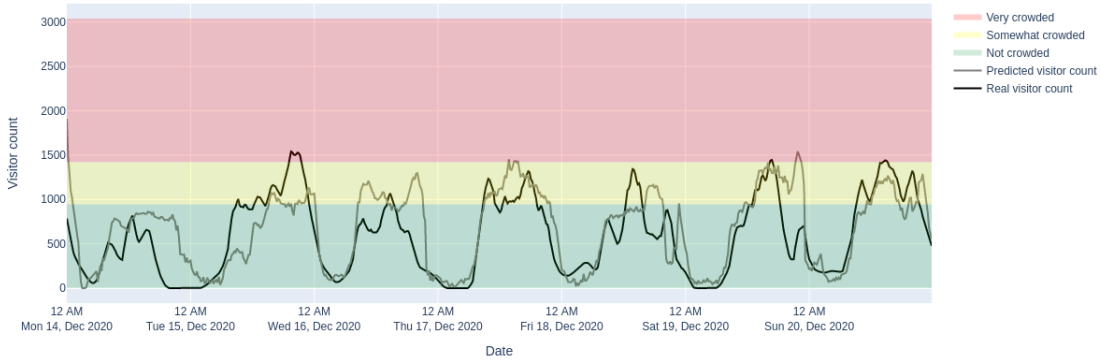


Figure 36. Predictions vs. ground truth for the test set for the LSTM model (regression) for Vondelpark.

square and Vondelpark the models do not outperform the baseline model on the test set. Furthermore, the results do not clearly show that either the regression or the classification approach leads to more accurate predictions. Based on these results, this seems to depend

Model	Testing	Training weeks
LSTM (regression)	0.48	1
LSTM (regression)	0.50	4
LSTM (classification)	0.47	1
LSTM (classification)	0.45	4

Table 12. Average F_β scores for the LSTM models on the test set for Vondelpark, using either 1 or 4 weeks of training data.

on the specific location.

4.4.2 Effect of varying the training set size

In Figures 37, 38 and 39 we can see the effect of varying the training set size on the performance of the LSTM model on the validation set for Dam square (regression), Vondelpark (classification) and Albert Cuyp (classification), respectively. From these figures we can see that there is not a large difference in the average F_β score between the different training set sizes. Moreover, for Dam square, the results are more strongly affected by adding another week of training data than for the other two locations. Interestingly, for both Vondelpark and Albert Cuyp, the average F_β score actually decreases after a training set size of two weeks.

Figure 37. F1 (or F_β) scores for the different classes and averaged over classes for the validation set across the different training set sizes for the LSTM model (regression) for Dam square.

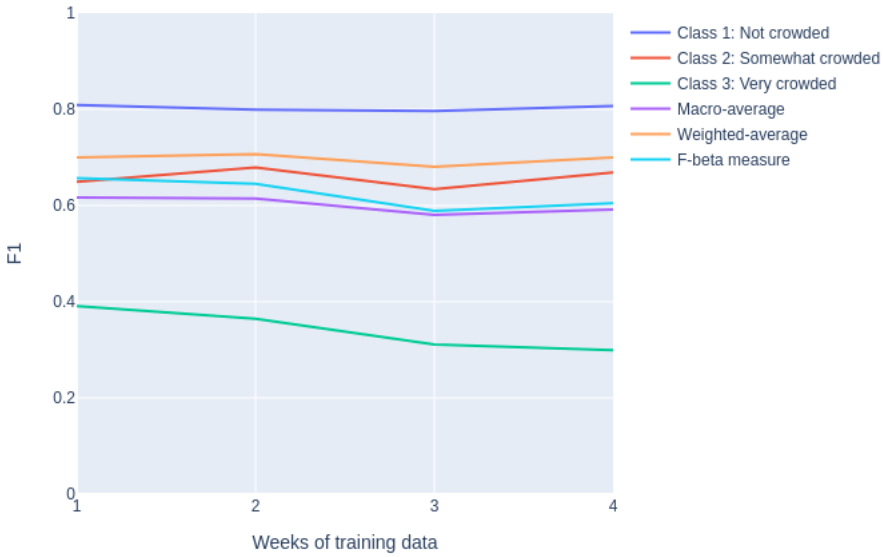


Figure 38. F1 (or F_β) scores for the different classes and averaged over classes for the validation set across the different training set sizes for the LSTM model (classification) for Vondelpark.

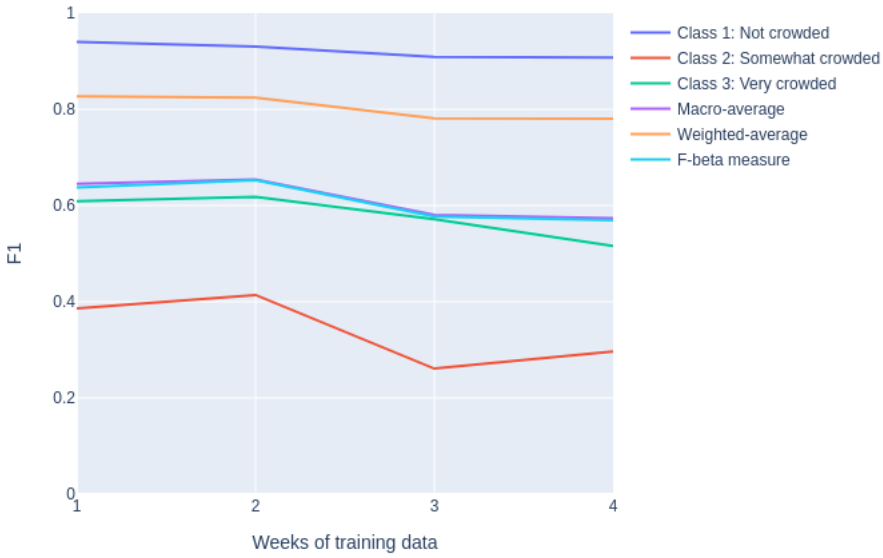


Figure 39. F1 (or F_β) scores for the different classes and averaged over classes for the validation set across the different training set sizes for the LSTM model (classification) for Albert Cuyp.

4.4.3 Effect of oversampling crowded moments

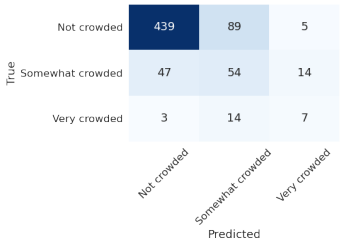
For each location, we examined the effect of oversampling (as discussed in Section 3.1.3.8) to see whether model performance would improve. For this we used the regression model with the highest score on the validation set. The results can be found in Table 13, and Figure 40 shows the predictions versus the ground truth for these models.

For Dam square, the average F_β score improved on both the validation and test set, and is higher than the score of the baseline model in both cases. For Vondelpark, the score improved on both the validation and test sets, but for the test set the model still did not perform better than the baseline model. Lastly, for Albert Cuyp, on the validation set the score decreased, however, the score is still higher that the score of the baseline model. Additionally, the score on the test set increased to a great extent. Thus, the effect of oversampling has mixed effects for the different locations, although the overall effect seems to be positive.

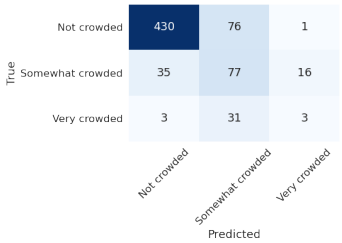
Location	**Model**	**Validation**	**Testing**
Dam square	Linear regression	0.53	0.51
Vondelpark	Ordinal regression	0.64	0.50
Albert Cuyp	Ordinal regression	0.65	0.77

Table 13. Average F_β scores for the regression models (linear or ordinal) with oversampling on the validation set for each location. The training set size corresponds to the selected training set size as in Tables 9, 10 and 11.

Dam square

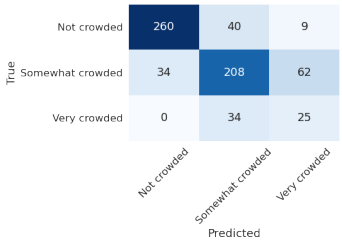


(a) Linear regression model with oversampling (validation set).

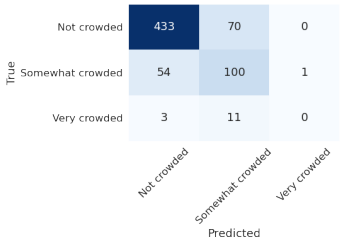


(b) Linear regression model with oversampling (test set).

Vondelpark

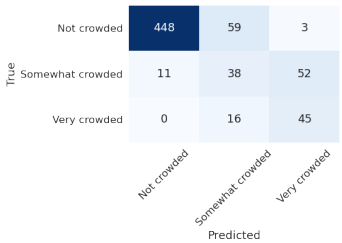


(c) Ordinal regression model with oversampling (validation set).

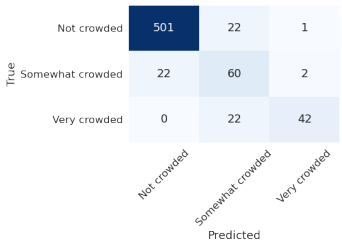


(d) Ordinal regression model with oversampling (test set).

Albert Cuyp



(e) Ordinal regression model with oversampling (validation set).



(f) Ordinal regression model with oversampling (test set).

Figure 40. Confusion matrices of the ground truth and predictions.

4.4.4 Predictor variable importance

To have a closer look at what predictor variables are important in predicting crowdedness, we further examined the predictor variable importance for the best performing model for each location. First, we looked at the effect of adding predictor variables step-wise to the model (following the mRMR ranking as explained in Section 3.4) on model performance. Figure 41 shows this effect for Dam square. The first predictor variable in the ranking is `afternoon`. The average F_β score increases slightly as more predictor variables are added to the model. The highest average F_β score is achieved when the first 16 predictor variables of the ranking are included in the model, however, the difference in score with the full set of predictor variables is very small.

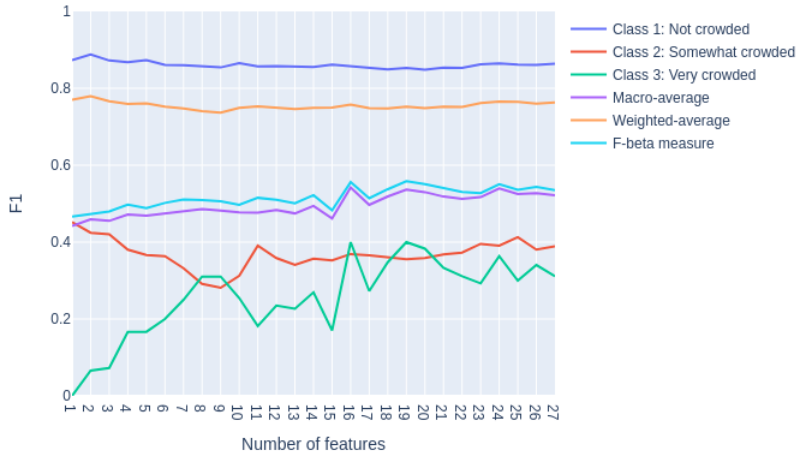


Figure 41. F1 (or F_β) scores for the different classes and averaged over classes for the validation set across the addition of different predictor variables for the linear regression model with oversampling for Dam square.

Figure 42 shows the effect of adding the predictor variables step-by-step on model performance for Vondelpark. The average F_β score increases quite a lot as the fourth predictor variable is added to the model. At this step the model consists of the following variables: `n_visitors_mobile_1day_ago`, `Monday`, `n_visitors_mobile_2hours_ago_diff_avg` (the difference between the average number of visitors and the number of visitors at the previous time step) and `night`. After this, the score increases a bit more until the maximum is reached at 20 variables. However, here the score for the first set of 10 predictor variables is very similar to the score for 20 predictor variables.

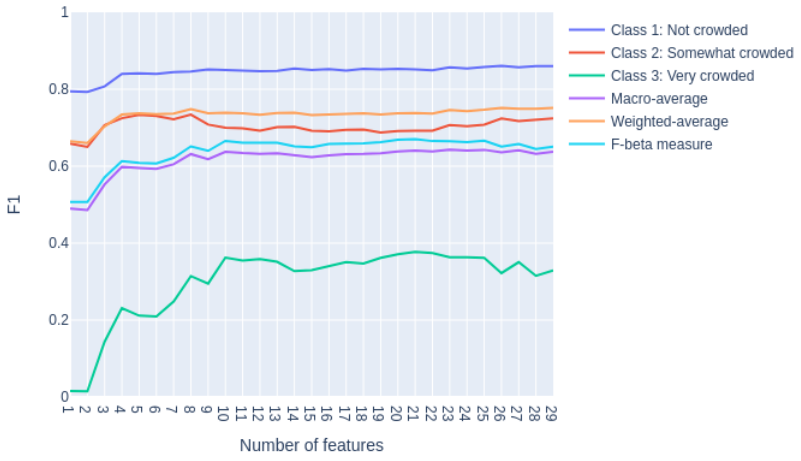


Figure 42. F1 (or F_β) scores for the different classes and averaged over classes for the validation set across the addition of different predictor variables for the linear regression model with oversampling for Vondelpark.

Figure 43 shows this effect for Albert Cuyp. The average F_β score shows a large increase as the COVID-19 data (Stringency Index) is added to the model (third predictor variable in the figure). The variables already present in the model at that point are `afternoon` and `sunshine`. Then as `night` is added to the model the performance actually drops, and increases once more after `n_visitors_camera` is being added. After this, the F_β score only changes slightly as more predictor variables are included in the model. The maximum score is reached when 23 predictor variables are added, however, this score is very close to the score when only 8 predictor variables are in the model. Again, the difference in score between the full set of variables and the number of variables for the maximum score is relatively small.

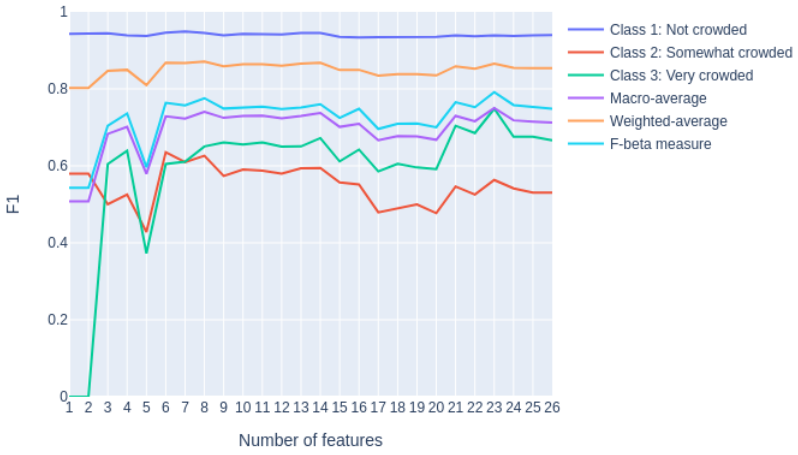


Figure 43. F1 (or F_β) scores for the different classes and averaged over classes for the validation set across the addition of different predictor variables for the ordinal regression model with oversampling for Albert Cuyp.

Next, we examined the weights of the predictor variables to see which variables are most important in the model. Figure 55 shows the predictor variable weights for the best performing model for each location. For all three locations, historical values of the ground truth data seem to be important predictors (for example, the predictor variable `n_visitors_mobile_2hours_ago_diff_size` tells us how much the visitor count has increased or decreased during the previous two time steps). If there was a large increase in the number of visitors recently, the model predicts that the number of visitors will continue to increase further. Then, for all locations, information on the time of day also seem to be important indicators for crowdedness (the `hour_cos`, `hour_sin` and `night` variables).

Then, for each location specific day indicators seem important as well. For example, for Albert Cuyp, if the day to predict is a Saturday, this indicates that on average, the visitor count will be higher than on other days, while on Tuesdays, on average the visitor count

will be lower than on other days. Furthermore, for Vondelpark and Albert Cuyp some COVID-19 related variables (`NL_GRSI_dec` and `NL_GRSI_inc`) have a relatively high importance in the model. Finally, for Vondelpark and Dam square the predictor variable `national_day` is also an important predictor. Interestingly, for Albert Cuyp, the weight for `parking_occupation` is negative. In the exploratory data analysis (Section 4.3.3) we found a positive correlation between this variable and the ground truth, which adds to this finding not being in line with the expectations.

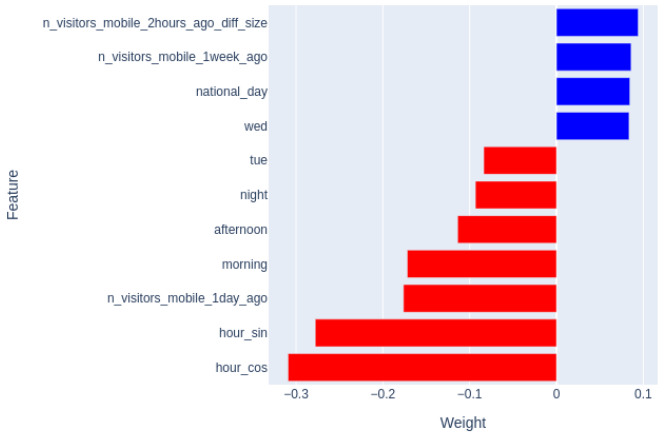
4.4.5 Error analysis

Finally, we examined a couple of miss-classified cases to gather some insight on why certain errors were made by the models. For this, we looked at a crowded moment for each location that was not predicted as such by any of the models.

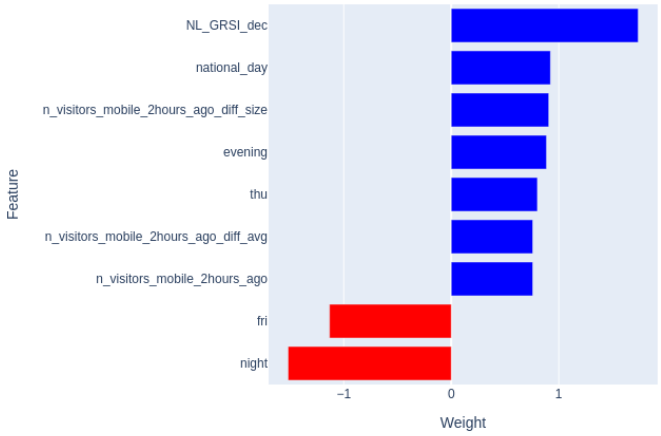
First, for Dam square this is in the afternoon on December 14th, 2020. The estimated number of visitors reaches the second threshold, while the predicted number of visitors remains below the second threshold. In Figure 45 we can see a selection of the predictor variable for these time slots. For this specific case, it seems that the predictor variables do not provide sufficient information to accurately detect this crowded moment. Furthermore, based on the comparison with the subsequent day, the error could possibly be caused by the relatively low visitor count based on camera data, but this remains uncertain.

Then, for Vondelpark Figure 46 shows a missed crowded moment (most models predicted `somewhat crowded`) on the 15th on December, 2020 in the evening. Interestingly, the visitor counts of the ground truth of the previous week show a similar pattern, however, the crowded moment was still miss-classified. Important to note is that the number of visitors according to the ground truth only crosses the second threshold slightly, which makes this moment more difficult to correctly predict.

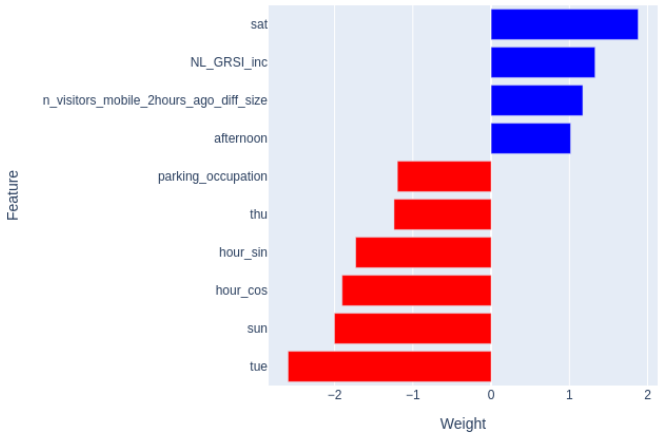
Finally, for Albert Cuyp, Figure 47 shows a miss-classified crowded moment, which took place in the afternoon on the 18th of February, 2021. Interestingly, the predictor variables displayed in the figure all show a similar pattern to that of the ground truth. However, as for the miss-classified crowded moment for Vondelpark, the second threshold is only barely reached, which could make it more difficult for the models to distinguish `somewhat crowded` cases from `very crowded` cases.



(a) Dam square (linear regression with oversampling).



(b) Vondelpark (ordinal regression with oversampling).



(c) Albert Cuyp (ordinal regression with oversampling).

Figure 44. Predictor variables with the highest weights for the best performing model for each location. The weights are averaged over all training iterations.

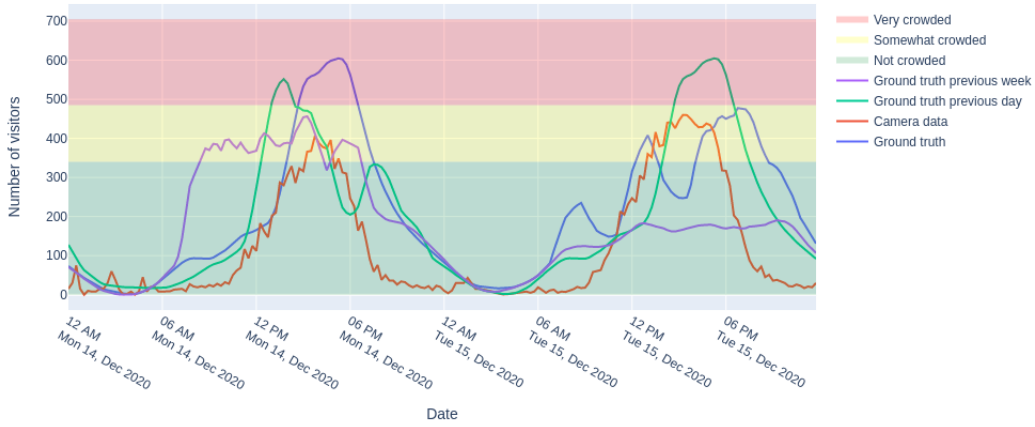


Figure 45. Ground truth and a selection of the predictor variables for Dam square. The miss-classified crowded moment occurred around 04:00 PM on the 14th.

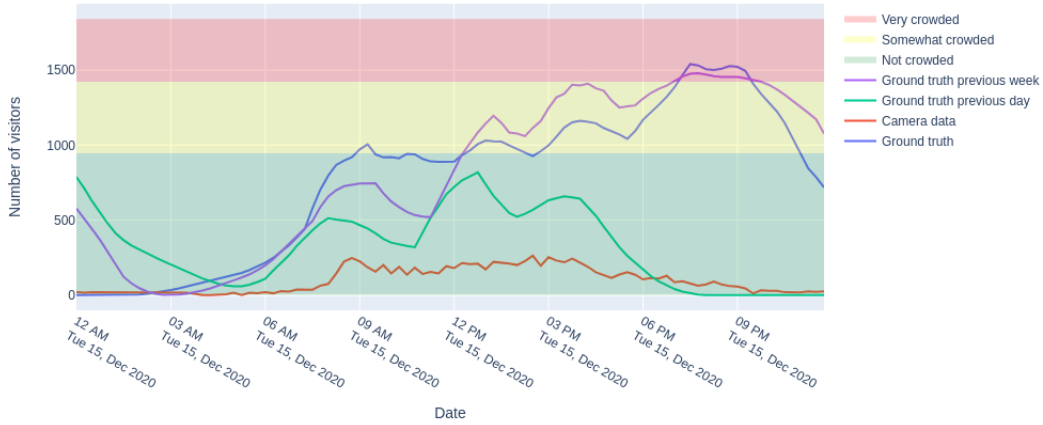


Figure 46. Ground truth and a selection of the predictor variables for Vondelpark. The miss-classified crowded moment occurred around 09:00 PM.

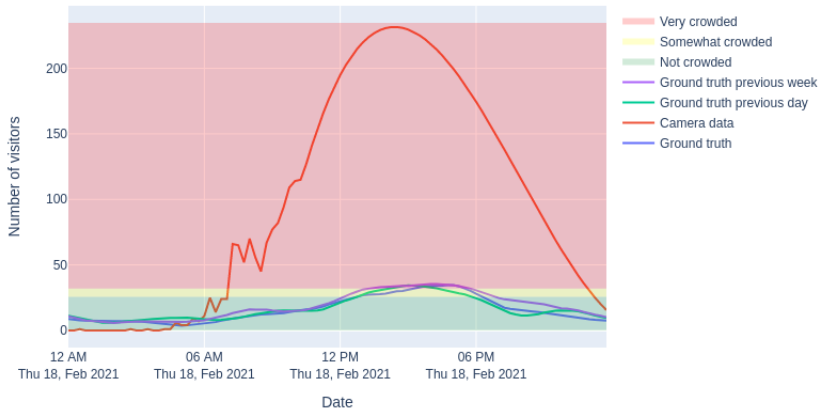


Figure 47. Ground truth and a selection of the predictor variables for Albert Cuyp. The miss-classified crowded moment occurred around 03:00 PM.

5. Experiment with the addition of Twitter and public transport data

5.1 Objective

In the previous subsection (4.4.5), we saw that in some cases it seems that the model does not have sufficient data available to learn from to be able to accurately predict all crowded moments. As an attempt to improve on this, we performed an additional experiment using two new data sources: Twitter data and public transport data. For this experiment, we focused on the Vondelpark location. Before we discuss the findings, we briefly explain the outline of the experiment and provide information on these new data sources.

5.2 Experimental set-up

For this experiment, we used data for Vondelpark ranging from February 05th, 2021 to April 1st, 2021, local time (UTC +1). We used four weeks of training data[1], and the subsequent three days as test data (the 30th of March, 31st of March and the 1st of April). The focus lies on the 31st of March, since on this day the Vondelpark became too crowded, and actions had to be taken by the municipality. However, we also included the day before and the day after in the test set to be able to distinguish model performance for both crowded and not-crowded days.

We compared a subset of the different model types that achieved the highest average F_β scores in the previous experiment (ordinal regression using oversampling and LSTM for classification). Furthermore, we did not perform the outlier removal steps on this data set, because this would have resulted in the incorrect removal of crowded moments. Additionally, we included public transport data and Twitter data. More information on these data sources is provided next.

[1] Since for this time period there were some days without any ground truth data, we extended the training set period to February 15th, 2021, to still have training data consisting of about four weeks.

5.2.1 Public transport data

This data source consists of the number of visitors that enter the area nearby the target location based on checking out of public transport. This data was gathered by the public transport provider of Amsterdam, GVB [62] and covers bus, metro and tram check out counts. The data is updated hourly. If the total count is below ten, the data is not be displayed due to privacy concerns. This data is not publicly available. Figure 48 displays a few days of public transport data. Specifically, this data is for the public transport stop at Leidseplein. The public transport data has a correlation of 0.42 with the ground truth data.

Figure 48. Figure is not displayed because this data source cannot be made public.

Considerations. This data source could improve crowdedness predictions because it accounts for visitors travelling nearby the target location. Our expectation was that many recent arrivals indicate an increase in the number of visitors at nearby locations. Conversely, if there are very little arrivals we would expect that the nearby locations will be less crowded. One disadvantage of this data source is that we do not know what percentage of visitors that arrive at such a public transport stop will visit the target location (as is the case for the parking data). This percentage is probably variable, and therefore it could be difficult to establish a clear relationship between the two data sources.

5.2.2 Twitter data

This data source consists of tweets filtered based on a set of keywords that are related to Vondelpark. The data was retrieved using the platform Trollrensics [63], which is not publicly accessible. Using the functionalities of this platform, we filtered tweets for the selected time period that contained at least one of the keywords shown in Table 14. This resulted in a set of 10,446 tweets. The tweets were not filtered based on location data, since this data was often missing, and not standardized (e.g. all tweets originating from Amsterdam did not necessarily have the same location token). For this experiment, we did not use the content of the tweets in the models, we only considered the amount of tweets across time. However, we inspected some tweets at random to get an idea of what type of content the tweets consist of (see Table 15).

The data was re-sampled so that each 15-minute time slot consists of the sum of tweets that occurred during this time slot. For many time slots, no Vondelpark-related tweets were present (60%), followed by a single tweet (20%), 2-4 tweets (10%) and finally, 5 or more tweets (10%). The maximum amount of tweets that occurred over a period of 15 minutes is 136. We created a set of both continuous and binary predictor variables using

the amount of tweets over time that provide information on for example the amount of tweets at the previous time step, or how much the number of tweets increased or decreased between the previous two time steps.

Considerations. Twitter data could provide additional information on the future level of crowdedness, as we expect that when a location tends to become crowded, more people will mention the location on social media, such as Twitter. This would make Twitter data specifically useful for short-term predictions. However, an important disadvantage of this data source is that it is not feasible to filter out uninformative tweets by hand, which means that there will always be tweets present that are not informative for this prediction problem. Additionally, since the tweets are filtered by a set of keywords, it could also be the case that some tweets are not being detected that do contain information on the crowdedness at the respective location. Despite this, since we are using information on changes in the overall amount of Vondelpark-related tweets, the effect of some misplaced or undetected tweets should not have a large impact on the results.

Keyword
Vondelpark
Vondel
Vondeltje
Wondelpark
Blauwe Theehuis
Blauwe teahouse
Proeflokaal 't Blauwe Theehuis
Groot Melkhuis
#Vondelpark
#Vondel
#Vondeltje
Vondelpak
Vondlpark
Vondpark

Table 14. The list of keywords used to filter the Twitter data (not case-sensitive).

	Tweet content
Original	'Dit krijg je als je jongeren opsluit, #Vondelpark vandaag #Rutte #Ruttedoctrine'
Translation	This is what you get when you lock up young adults, #Vondelpark today #Rutte #Ruttedoctrine'
Original	'Vondel voelt als mini festival atm'
Translation	'Vondel feels like a mini festival atm'
Original	'De " ik ik ik" generatie heeft het Vondelpark verlaten. #Vondelpark'
Translation	'The " me me me" generation has left Vondelpark. #Vondelpark'

Table 15. A few anonymous example tweets.

5.3 Exploratory data analysis

Figure 49 shows the relationship between the number of tweets and the number of visitors (based on mobile phone data). Based on this figure, there does seem to be a trend where a high amount of tweets is related to a high visitor count.

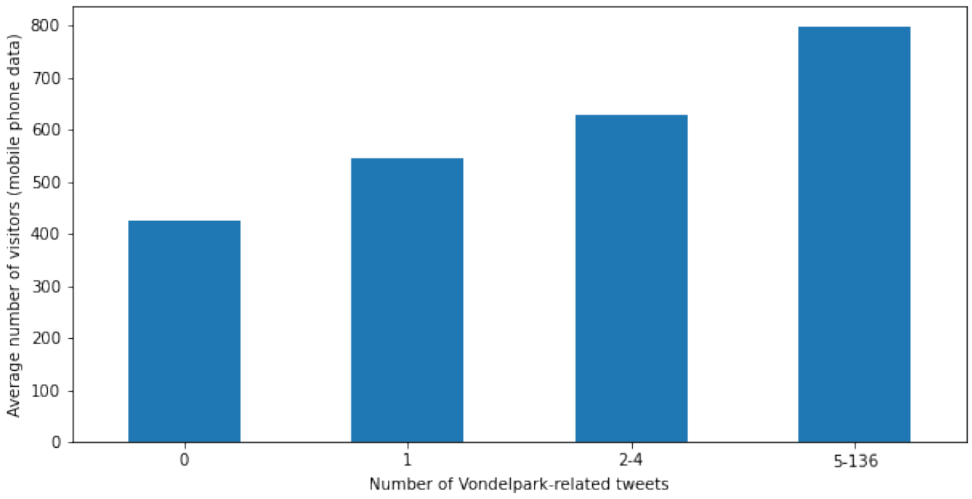


Figure 49. Average number of visitors for different amounts of tweets.

However, it is important to show that there are also crowded moments when no Vondelpark-related tweets are present, and vice versa (see Figure 50). The amount of tweets over time has a correlation with the ground truth of 0.28. When we consider the relationship between the ground truth and lagged versions of the Twitter data (the amount of tweets at the previous two hours or at the previous day), the correlations are 0.26 and 0.13, respectively. If we look at the relationship between the ground truth and forward-lagged versions of the Twitter data (the amount of tweets at the next two hours or at the next day), the correlations are 0.26 and 0.29, respectively. Thus, the relationship with the ground truth is not stronger when we compare the visitor count with a (forward-)lagged version of the amount of

tweets.

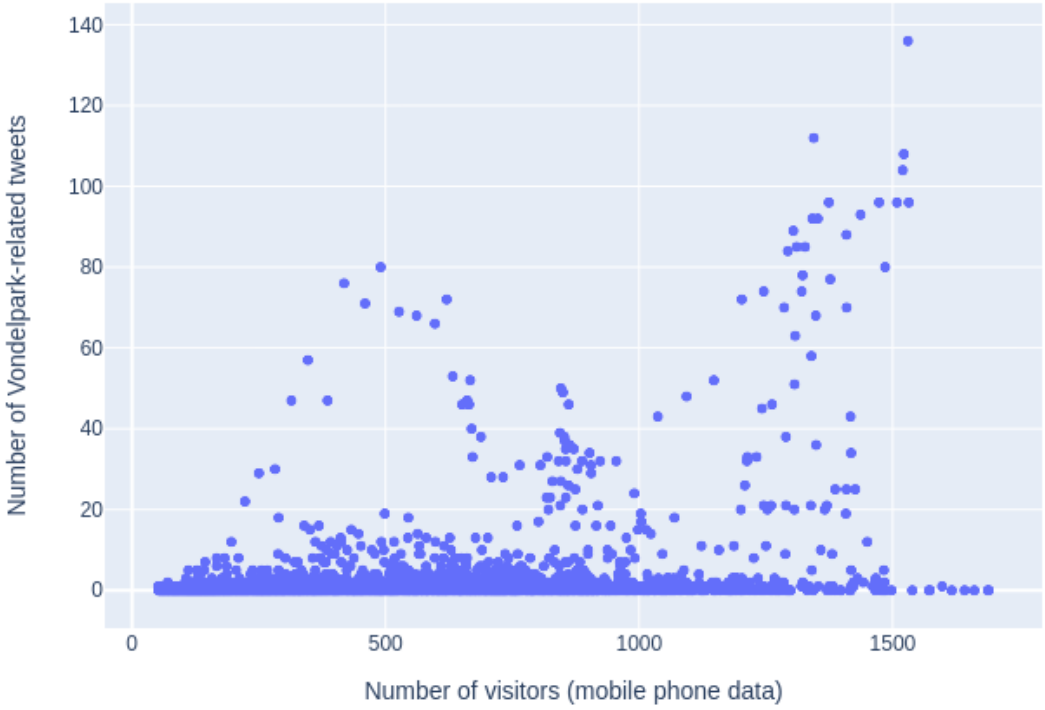


Figure 50. Average number of visitors vs. the amount of tweets.

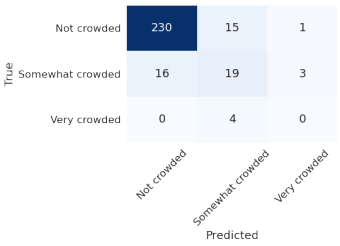
5.4 Results

Table 16 shows the F_β scores for the two model types with and without public transport and Twitter data. In this table we can see that the score either remains the same or even decreases as public transport data, Twitter data, or both data sources are added to the model. Figure 51 shows the accompanying confusion matrices. For the ordinal regression (and baseline) models, the time slots that belong to the `very crowded` class on March 31 are not predicted as such, regardless of whether the new data sources were added. For the LSTM model for classification, when no additional data sources were included, some of the `very crowded` time slots were predicted correctly, whereas this was not the case if the new data sources were included.

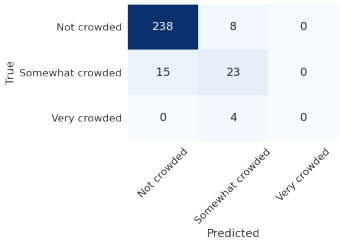
Finally, Figure 52 shows the weights of the most influential predictor variables of the ordinal regression model in which public transport data and Twitter data were added. It seems that two of the new predictor variables have some influence in the model: `n_tweets_2hours_ago_diff_avg_high` (a binary variable indicating whether the difference in number of tweets during the previous time step and the average number

Model	Predictor variables	Training	Testing
Baseline	None	0.73	0.47
Ordinal regression	No Twitter and public transport data	0.51	0.52
Ordinal regression	Twitter data	0.50	0.50
Ordinal regression	Public transport data	0.51	0.52
Ordinal regression	Twitter and public transport data	0.51	0.46
LSTM (classification)	No Twitter and public transport data	0.49	0.59
LSTM (classification)	Twitter data	0.52	0.45
LSTM (classification)	Public transport data	0.48	0.40
LSTM (classification)	Twitter and public transport data	0.49	0.38

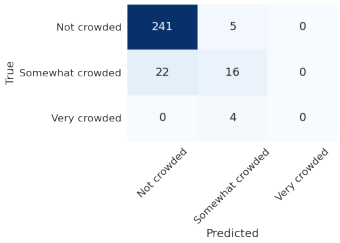
Table 16. Average F_β scores for a selection of the models and different predictor variable sets (Twitter and public transport data either included or excluded). In the ordinal regression model oversampling was used.



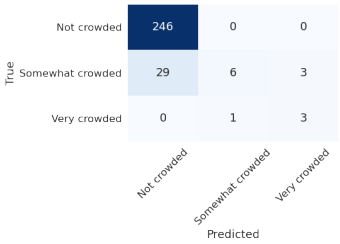
(a) Baseline model.



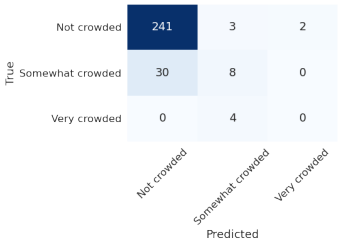
(b) Ordinal regression without public transport and Twitter data.



(c) Ordinal regression with public transport and Twitter data.



(d) LSTM (classification) without public transport and Twitter data.



(e) LSTM (classification) with public transport and Twitter data.

Figure 51. Confusion matrices of the ground truth and predictions for the test set.

of tweets is larger than some fixed value) and `n_checkouts` (the number of checkouts at the same time slot during the previous week).

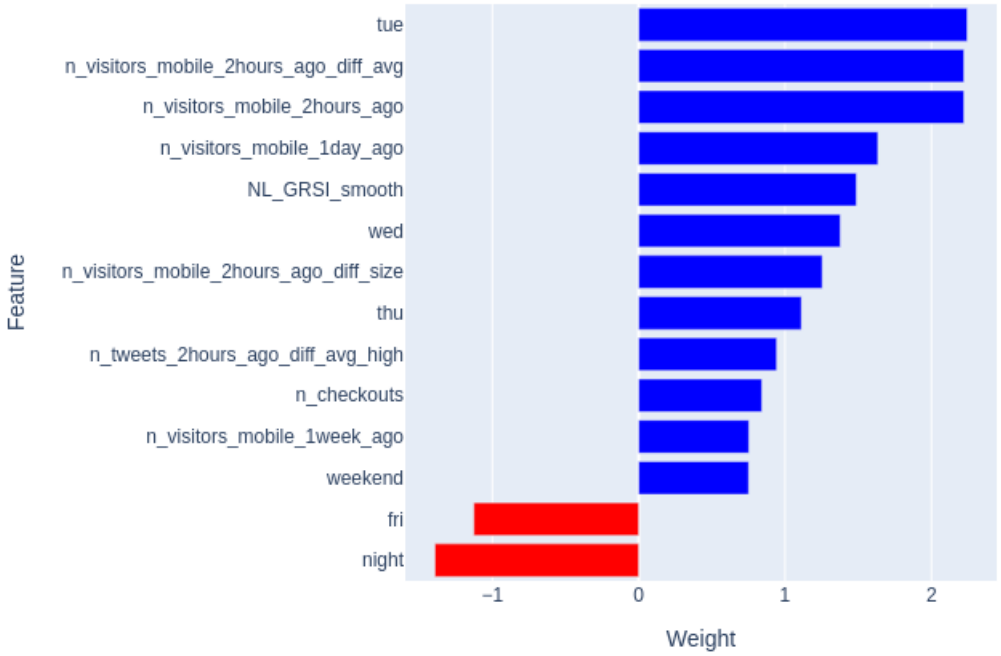


Figure 52. Predictor variables with the highest weights for the ordinal regression model with public transport and Twitter data. The weights are averaged over all training iterations.

6. Experiment on the full data set

6.1 Objective

One important limitation of the previous two experiments is the shortage of validation/test data. Because model performance is determined based on predictions for one week of data (or two when combining the validation and test results), the results are not very reliable. It could be that the crowdedness patterns of that specific week were relatively easy or hard to predict. Therefore, we deemed it necessary to perform an additional experiment in which we extended the existing set of results with predictions on additional data. In doing so, some of the decisions made in this experiment were based on the results of the first experiment (4). First, we provide some details on the experimental set-up and then we show the results for this experiment.

6.2 Experimental set-up

For the Dam square and Albert Cuyp locations, the selected data ranges from February 5th, 2021 to April 10th, 2021, local time (UTC +1). For Vondelpark, the selected data ranges from February 13th, 2021 to April 30st, 2021 local time (UTC +1). For all locations, we used the final two weeks as test data and the six weeks previous to those as training data. Based on the results from the first experiment, we made a selection of the models that provided the most accurate predictions overall. As such, we compared the following model types: the baseline model, linear and ordinal regression models with oversampling, and LSTM models for regression and classification.

We used the same predictor variables as in the first experiment, with the addition of public transport data and Twitter data for the Vondelpark data set. Additionally, for Dam square and Vondelpark we did not perform outlier removal (as in Experiment 2), as outliers either did not seem to be present during the selected time period (Dam square) or the process would lead to the incorrect removal of crowded moments (Vondelpark). Finally, for the linear and ordinal regression models we used L2 regularization to prevent overfitting (with $\lambda = 1$).

6.3 Results

6.3.1 Model evaluation

Tables 17, 18 and 19 show the results for each of the locations. Here we see that the average F_β scores are relatively low for the training data. This is probably caused by the overall daily crowdedness being a lot higher during February compared to March and April, resulting in a high amount of prediction errors for these first few weeks of training data. For all three locations, the selected models outperform the baseline model on the test data. For Dam square, the highest score is achieved by the ordinal regression model, followed closely by the linear regression model. For Vondelpark, the best performing model is the linear regression model, and for Albert Cuyp, both the ordinal regression model and LSTM model for regression result in the most accurate predictions.

For each location, the scores for the different model types (except the baseline model) are in the same range. Overall, the predictions are most accurate for Dam square, however, it is important to note that for this location the test set consisted of no `very crowded` data points. The predictions are the least accurate for Vondelpark (similar to the findings in Experiment 1), since there was only a single set of `very crowded` data points that was not predicted correctly by any of the models.

Model	**Training**	**Testing**
Baseline	0.72	0.49
Linear regression	0.43	0.88
Ordinal regression	0.33	**0.90**
LSTM (regression)	0.40	0.84
LSTM (classification)	0.33	0.78

Table 17. Average F_β scores for a selection of the different model types for the train and test data for Dam square.

Model	**Training**	**Testing**
Baseline	0.66	0.44
Linear regression	0.39	**0.52**
Ordinal regression	0.39	0.45
LSTM (regression)	0.39	0.39
LSTM (classification)	0.39	0.37

Table 18. Average F_β scores for a selection of the different model types for the train and test data for Vondelpark.

Then, to compare the performance of the different models for crowdedness predictions in general, Table 20 shows the performance of the different model types averaged over the three locations.

Model	Training	Testing
Baseline	0.60	0.57
Linear regression	0.53	0.71
Ordinal regression	0.55	**0.75**
LSTM (regression)	0.46	**0.75**
LSTM (classification)	0.51	0.72

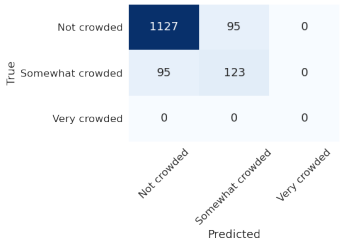
Table 19. Average F_β scores for a selection of the different model types for the train and test data for Albert Cuyp.

Model	Training	Testing
Baseline	0.66	0.50
Linear regression	0.45	**0.70**
Ordinal regression	0.42	**0.70**
LSTM (regression)	0.41	0.66
LSTM (classification)	0.41	0.62

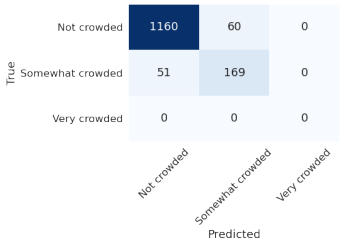
Table 20. The average F_β scores for the train and test data averaged across the three locations for each model type.

Figure 53 shows the confusion matrices based on the test set results for the three locations. Here we can compare the predictions by the baseline model with the predictions of the best performing model. For all locations, the predictions are more accurate for the best performing model compared to the baseline model. However, as mentioned before, the `very crowded` moment for Vondelpark is not detected by the model. Interestingly, in Figure 54 we see that the model does predict a peak in crowdedness (mostly aligning with the peak in the ground truth data), however, the predicted peak does not reach the `very crowded` threshold.

Dam square

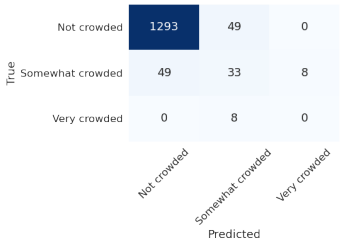


(a) Baseline model.

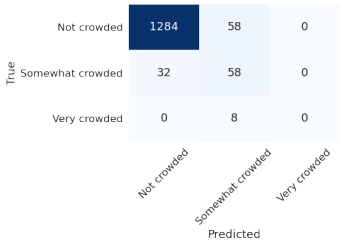


(b) Ordinal regression model.

Vondelpark

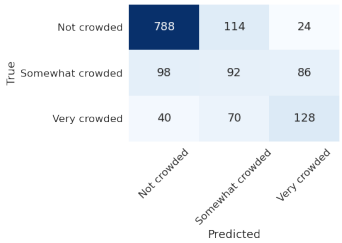


(c) Baseline model.

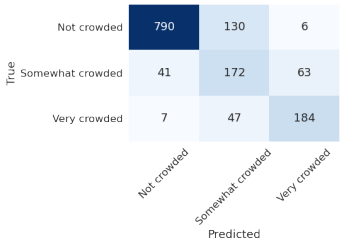


(d) Linear regression model.

Albert Cuyp



(e) Baseline model.



(f) Ordinal regression model.

Figure 53. Confusion matrices of the ground truth and predictions for the test set.

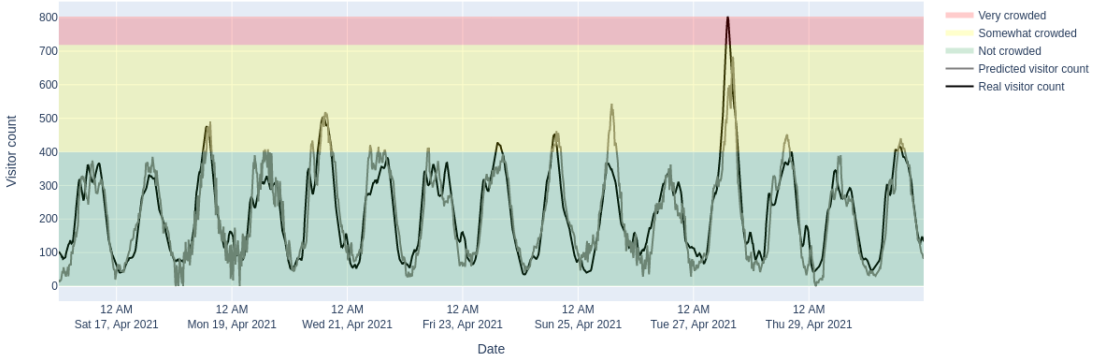


Figure 54. Predictions vs. ground truth for the test set for the linear regression model for Vondelpark.

Additionally, we performed McNemar's Test in which we compared the results based on the linear and ordinal regression models with the results based on the other model types. For this, we combined the results for all three locations and made pairwise comparisons. McNemar's Test is a non-parametric test that, for this specific prediction problem, can indicate whether two models have the same proportion of prediction errors [64]. It can be used when only a single test set is available, and does not have the assumption that the samples have to be independent. Here, the samples are not independent because samples that are used for testing are afterwards also used for training (see Section 3.2.2).

The null-hypothesis is that the proportion of errors on the test set are similar, and the alternative hypothesis is that the proportion of errors on the test set are not similar, with $\alpha = 0.05$. Note that this test does not specify the direction of differences in errors (i.e. which model performs better). The results are shown in Table 21. For all comparisons, except for the pairing of the ordinal regression model and LSTM model for regression, the proportion of errors are different between all tested model pairs based on the test sets. If we look at the comparison between the linear regression and ordinal regression models, it seems that the linear regression model has a larger proportion of errors than the ordinal regression model, based on the values of the respective confusion matrices. However, this test does not distinguish between different types of errors (e.g. a false positive for the class `somewhat crowded` versus a false positive for the class `very crowded`).

Model pair	χ^2	ρ
Linear regression & Baseline	282	0.001
Linear regression & Ordinal regression	93	< 0.001
Linear regression & LSTM (regression)	198	< 0.001
Linear regression & LSTM (classification)	248	0.001
Ordinal regression & Baseline	169	< 0.001
Ordinal regression & LSTM (regression)	209	0.532
Ordinal regression & LSTM (classification)	170	< 0.001

Table 21. McNemar's Test statistic and p-value on the test sets for all locations combined, for different model pairs.

6.3.2 Predictor variable importance

Figure 55 shows the most important predictor variables for the best performing model for each location. Here we see that for all three locations, historical values of the ground truth data are the most important predictors (e.g. `n_visitors_mobile_2hours_ago_diff_size`, which indicates the visitor count at the same time slot of the previous week), followed by information on the time of day (e.g. `night` indicating whether it is day or night) or day of the week (e.g. `sat` indicating whether it is Saturday).

Furthermore, for Dam square we see a positive weight for the COVID-19 regulations variable `NL_GRSI_smooth` suggesting that at the Dam square there are actually more visitors when there are more COVID-19 regulations. For Vondelpark, the variable `n_checkouts` has a relatively large weight, meaning that a high amount of arrived travellers in the area nearby indicates crowdedness.

6.3.3 Error analysis

Finally, we had a closer look at some miss-classified (and also some correctly classified) cases for the three locations. For this, we looked at the predictor variable weights in combination with the predictor variables values for some specific predictions of the best performing model. This gives us some insight in how incorrect predictions arised.

6.3.3.1 Dam square

Figure 56a shows a time slot for which the true class is `not crowded` and the predicted class is `somewhat crowded`. The predictor variables that seem to contribute to this false positive is information on the time of day (`hour_cos`) and the parking data of a nearby parking garage (`parking_occupation_Bijenkorf`).

Figure 56b shows a time slot for which the true class is `somewhat crowded` and the predicted class is `not crowded`. Here it is noticable that the predictor variables related to the parking data are contributing to a higher predicted crowdedness level, but because the predictor variables related to the historical values of the ground truth have a negative contribution a lower crowdedness level is predicted.

Finally, Figure 56c shows a time slot for which both the true class and predicted class is `somewhat crowded`. The resulting variable values are comparable to those of the false positive case, with the difference that here the variables `weekend` and `parking_occupation_Rokin` also contribute to a higher predicted crowdedness level.

6.3.3.2 Vondelpark

Figure 57a shows a time slot for which the true class is `not crowded` and the predicted class is `somewhat crowded`. Here we see that the model incorrectly predicted a `somewhat crowded` moment because the variable `n_checkouts` is relatively high, indicating that many visitors were expected to enter the nearby area using public transport. Other contributing factors were it being a weekend day, a Sunday, and the visitor count at the same time step of the previous week was relatively high.

Figure 57b shows a time slot for which the true class is very crowded, while the predicted class is somewhat crowded. The variable n_checkouts again has a large contribution in the prediction for this time slot. However, this variable combined with information on the visitor count at the previous two hours was not sufficient to accurately predict the very crowded class. On the other hand, the lack of tweets (reflected in the negative values for n_tweets_2hours_ago_diff_avg and n_tweets_2hours_ago contributed to a lower predicted crowdedness level.

Lasty, in Figure 57c we see a time slot for which both the true and predicted class is somewhat crowded. It seems that once more the variable n_checkouts contributed to the, in this case, correct prediction. Apart from this, some information on the time of day (hour_sin) and on the historical ground truth values (n_visitors_mobile_2hours
_ago_diff_size and n_visitors_mobile_1week_ago) also contributed to correctly predicted crowdedness level.

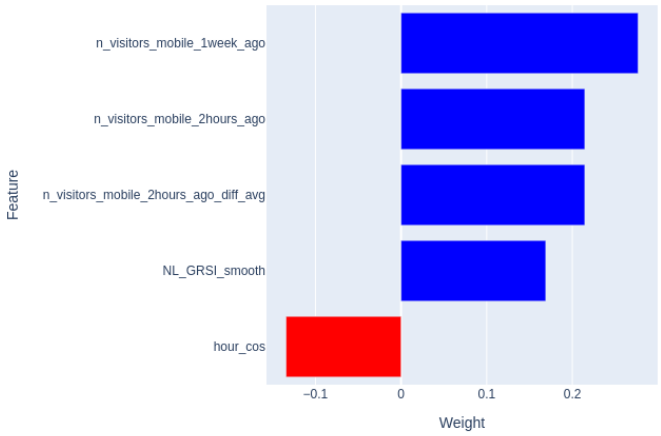
6.3.3.3 Albert Cuyp

For Albert Cuyp, Figure 58a displays a time slot for which the true class is somewhat crowded and the predicted class is very crowded. Here we see that the false positive occurred because the visitor count at the previous two hours was relatively high (reflected by n_visitors_mobile_2hours_ago_diff_avg and n_visitors_mobile_2hours_ago) and it was a Saturday. The historical visitor count variables have high values because there was a very crowded moment shortly before. The variable n_visitors_mobile_2hours_ago_diff_size indicates that the peak has passed and the visitor count is decreasing, however, the decrease is not sufficiently high for the model to predict a lower crowdedness level.

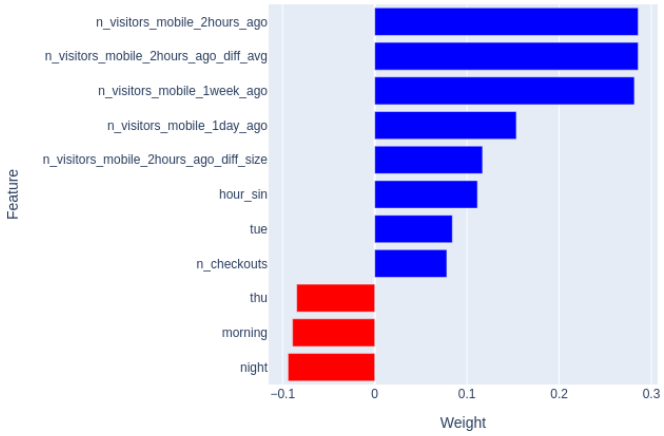
Figure 58b shows a time slot for which the true class is very crowded and the predicted class is somewhat crowded. We can see that historical values of the visitor count and some time information (it is a Saturday and it is morning) contribute to the somewhat crowded prediction. However, the values for the historical ground truth variables are not high enough to predict that the time slot is very crowded. Lastly, the variable weekend contributes to the predicted crowdedness level being lower.

Finally, in Figure 58c we see a time slot with very crowded as both the true and predicted class. As seen before, the historical values of the ground truth (reflected by multiple predictor variables) contribute to the correct prediction, together with the day of the week variable (sat). Again, the variable weekend lowers the predicted crowdedness

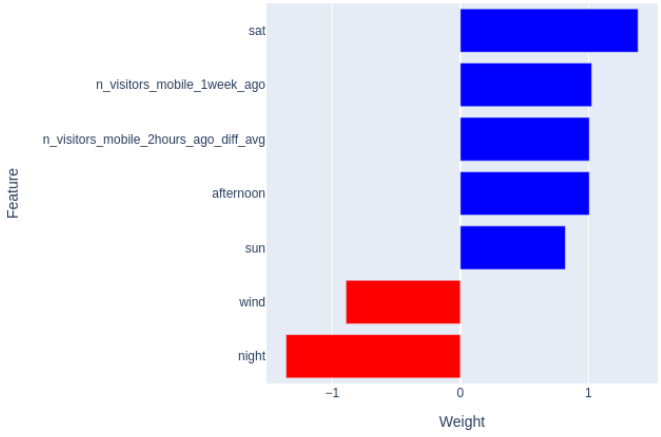
level, but not by enough to result in a false negative.



(a) Dam square (ordinal regression).

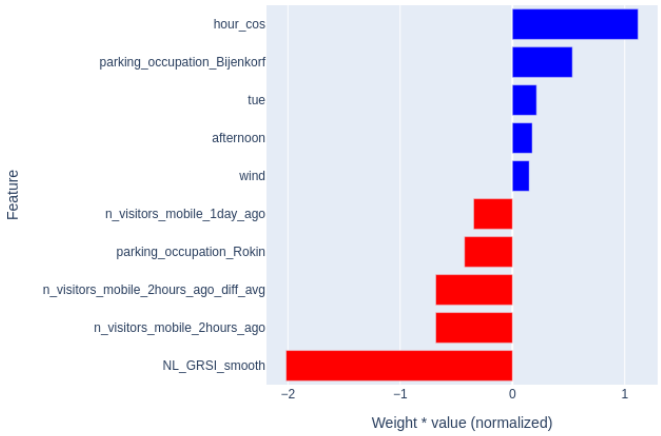


(b) Vondelpark (linear regression).

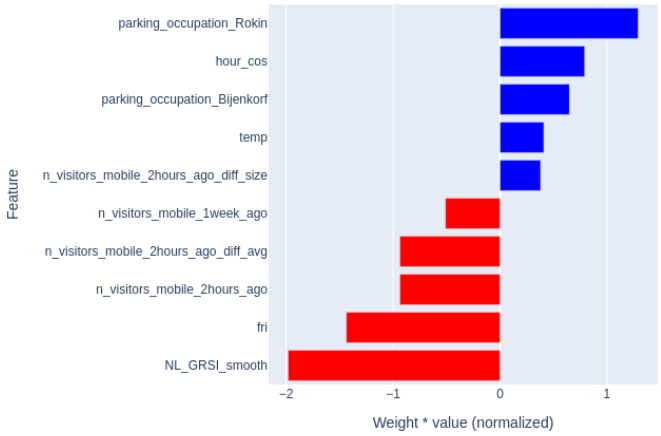


(c) Albert Cuyp (ordinal regression).

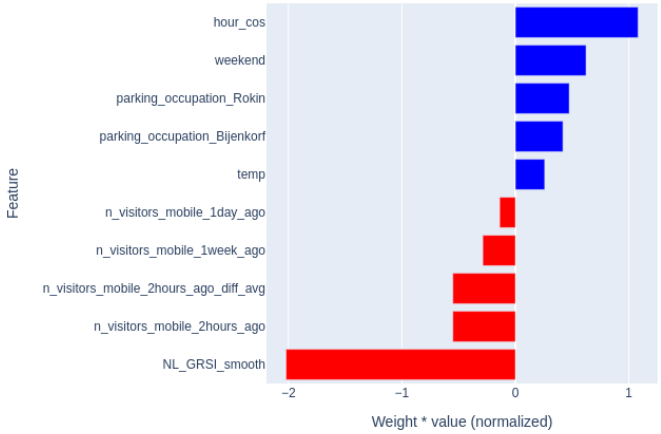
Figure 55. Predictor variables with the highest weights for the best performing model for each location. The weights are averaged over all training iterations.



(a) False positive case (12:00 PM, 6th of April).

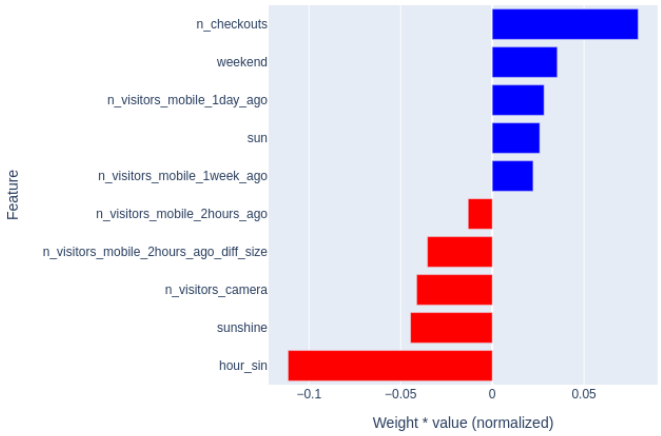


(b) False negative case (09:30 AM, 9th of April).

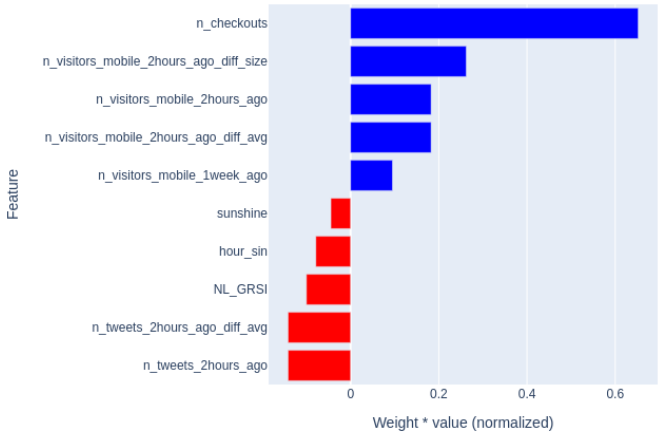


(c) True positive case (01:00 PM, 4th of April).

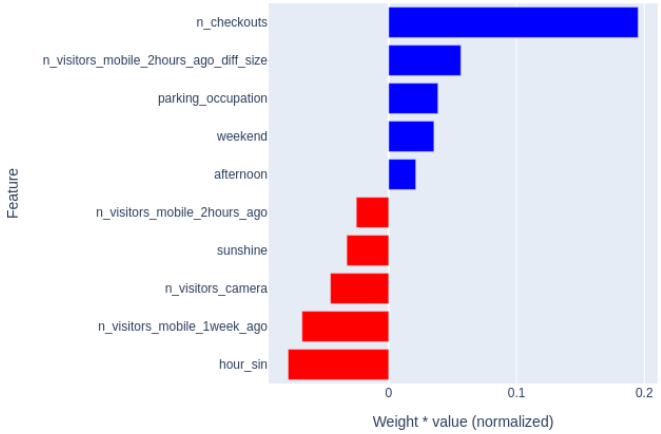
Figure 56. Predictor variable weights multiplied by the values of the predictor variables for specific example cases for Dam square.



(a) False positive case (03:00 PM, 25th of April).

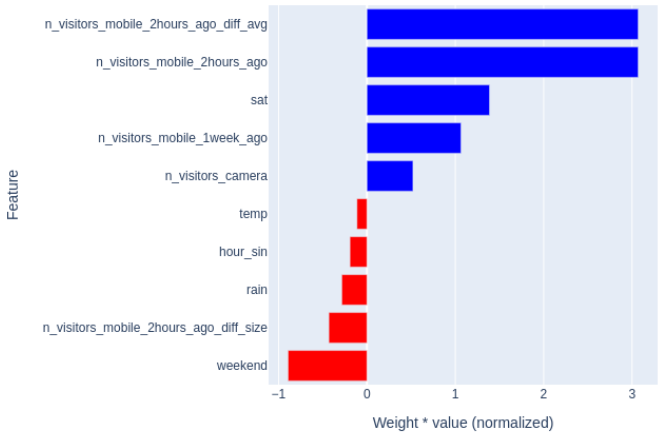


(b) False negative case (02:30 PM, 27th of April).

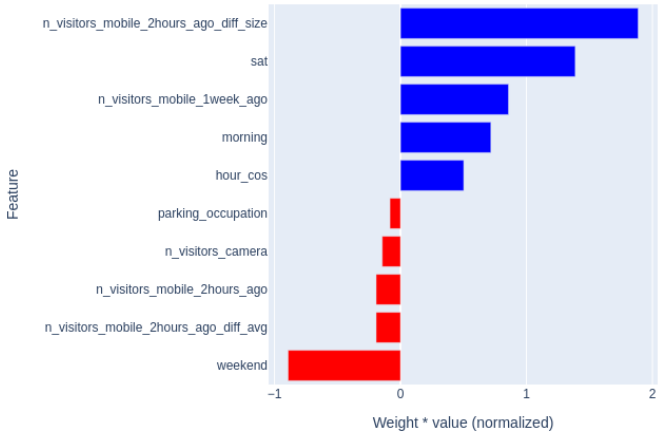


(c) True positive case (02:00 PM, 24th of April).

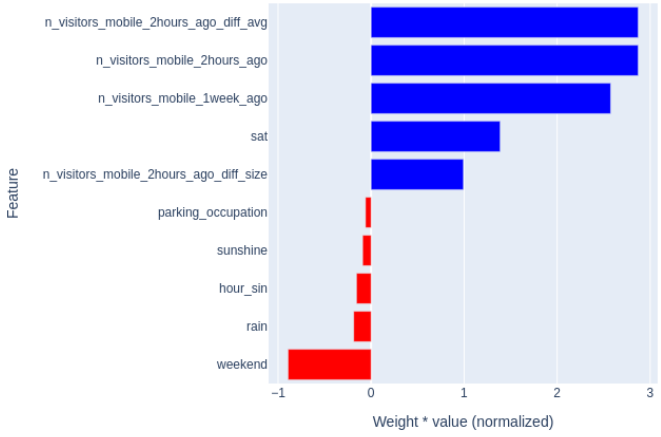
Figure 57. Predictor variable weights multiplied by the values of the predictor variables for specific example cases for Vondelpark.



(a) False positive case (04:45 PM, 10th of April).



(b) False negative case (10:30 AM, 27th of March).



(c) True positive case (03:30 PM, 10th of April).

Figure 58. Predictor variable weights multiplied by the values of the predictor variables for specific example cases for Albert Cuyp.

7. Discussion

In this section, we first evaluate the research questions posed in Section 1.2 based on the findings of the experiments that were performed. Next, we discuss some challenges related to the research carried out in this thesis and finally we provide some suggestions for future research on the topic of crowdedness predictions.

7.1 Influence external factors

One of our main research questions was about what external factors are important when predicting crowdedness on the short-term. For this, we used different methods to evaluate the effect of multiple external factors on prediction accuracy. One method that we have applied is step wise feature selection combined with the mRMR algorithm (see Section 3.4). Here, we found that the model performance was best when almost all external factors were added to the model. However, we also saw that the largest increase in model performance was achieved when only the first couple of predictor variables in the mRMR ranking were included in the model (which typically consisted of the historical values of the ground truth data and periodic data).

Another method that we applied was to inspect the model weights of the best performing models. In addition to this, we examined the influence of certain external factors by considering some examples of miss-classified cases. When taking all results together, we found that firstly, historical values of the ground truth data seem to be most predictive of the current crowdedness level. Specifically, very recent information on the past crowdedness levels (based on the ground truth data) seems to be very important for the model's predictions (for example, the visitor count of the past two hours, or by how much the visitor count was increasing or decreasing during the past two hours). This finding is in line with the studies by Fan et al. (2015) and Mu et al. (2020) who also found that the most recent data has a lot of impact on the predictive performance of crowd prediction models [9, 10]. In addition to this, information on the crowdedness levels on the previous day and previous week were also important indicators.

Then, another external factor that was found to be important is information on the time of day or day of the week. The time of day is very indicative of the crowdedness level in

general, as on typical days the number of visitors is very low during the night, relatively low during the early morning and late evening, and relatively high during the late morning and afternoon. However, there are some differences between the locations. For example, Vondelpark was found to be crowded more often during the evenings, compared to Albert Cuyp, while Albert Cuyp was found to be crowded more often during the mornings. Furthermore, in some cases we found different crowdedness patterns for different weekdays. For example, for Albert Cuyp we found a clear effect where the location is more crowded on Saturdays compared to other days. Interestingly, this shows an issue with the use of a variable that indicates whether it is weekend or not. If the pattern is different for Saturdays and Sundays it is not helpful for the model if we group them together (see Figure 58a).

Moreover, there are a few external factors that were also important in the models, but their effect was less clear: these factors seemed to be important in only some models, or only for one of the locations. These are COVID-19 regulations, holidays and public transport data. For Dam square and Vondelpark, in some cases we found a positive effect between the severeness of the COVID-19 regulations and crowdedness levels. While at first this seemed unexpected, this effect might be present because as the regulations become stricter, these two locations remain one of the few locations in Amsterdam that are still accessible for visitors. For Vondelpark, in the best performing model the public transport data is somewhat indicative of the crowdedness levels, where many checkouts in the nearby area relate to higher crowdedness levels[1]. One possible reason as to why these effects were not that clear could be linked to the size of the prediction window. It could be that if we for example were to predict the total number of visitor per day, a month ahead, the effect of COVID-19 regulations and holidays would be more pronounced.

This also means that we did not find a clear contribution of the camera data, parking data and weather data to the crowdedness predictions. One possible reason for this is that the camera and parking data do not add new information to the model that is very distinctive from the information that the model receives based on the historical values of the ground truth data. This is supported by the exploratory data analysis of the first experiment (Section 4.3) in which we saw that the ground truth data, camera data and parking data all correlate (except for the parking data for Albert Cuyp). Another possible reason is that since we use historical values for these data sources as well (because the future values are also unknown) it might be that the relationship between historical values of these data sources and the current crowdedness level is not that strong.

Lastly, for the weather data a possible cause for this finding is that the period for which we used data is too short. One group of data sets covered part of the autumn/winter period,

[1]Note that this data source was not included for the other two locations.

and the other group of data sets covered the part of the winter/spring period. As a result, for some of the weather-related variables there might not have been a lot of variation in the training data (e.g. the temperature was fairly similar across time). This limitation also holds for some of the other external factors. For example, the effect of different holidays might also have been more clear if for example a full year of training data was used.

Finally, for Vondelpark we also investigated Twitter data as an external factor for crowdedness predictions (see Experiment 5. We found a relationship between the two data sources where peaks in the amount of tweets related to peaks in crowdedness. However, the relationship was not very clear, as there were also peaks in tweets with the absence of crowdedness and peaks in crowdedness with the absence of tweets. This result was corroborated by the fact that in the best performing model, variables related to this data source did not have a large influence on the predictions. Interestingly, we actually saw in Figure 57b that including Twitter data contributed to a very crowded moment not being detected by the model. Based on this, we would suggest that this external factor can best be included in a way in which it can only be indicative of crowded moments, and not the other way around.

7.2 Effectiveness of different model types

Our other main research question was about what model types are effective when predicting crowdedness on the short-term. In the first experiment (Section 4) we saw that the non-linear regression model and the SARIMAX model turned out not be suitable for this specific prediction problem. Furthermore, we found that the LSTM models performed best on the test set, although the regression models performed slightly better when using oversampling. However, with the exception of Albert Cuyp, none of the models outperformed the baseline model on the test set, which we saw as an indication that insufficient data was being used to perform a reliable experiment. Still, the first experiment provided us with some important insights on which models could be ruled out immediately, the effect of the use of oversampling, and the importance of sufficient train and test data.

In the final experiment (Section 6) we used new data sets with more train and test data. The result showed that the linear regression model and ordinal regression model led to the most accurate crowdedness predictions (outperforming the LSTM models on most occasions, and the baseline model on all occasions). However, there are some differences across locations: for Dam square and Albert Cuyp the ordinal regression model resulted in a slightly better score than the linear regression model, while for Vondelpark we found the opposite result.

In the studies discussed in Section 2, typically the LSTM model outperformed other model types, however, in these studies the LSTM model was either compared to other types of RNNs [15] or in addition to this also to ARIMA-like models [4, 5, 6, 7, 22]. This makes it difficult to compare the findings of this thesis to the results from these other studies. Still, it could be the case that there are hyperparameter settings for the LSTM model that could have led to an improvement of the model's performance that we did not consider.

7.2.1 Advantages and disadvantages of different model types

All model types that we compared in this thesis have different advantages and disadvantages. Based on our findings, we highlight the most important differences between the model types.

First, an advantage of the regression models (linear and ordinal regression) is that the model is easy to interpret. For example, for these models it is relatively straightforward to obtain insights in what external factors in the model are important, or why a certain time step is assigned a certain crowdedness level by the model. Another advantage that was found to be very important is the possibility to use oversampling for this model type. This was possible because this is not a time series model, and thus the model does not have to account for temporal aspects of the ground truth data. Because there were only few very crowded moments in the data sets, using an oversampling technique improved the model's ability to distinguish somewhat crowded moments from very crowded moments.

On the other hand, a disadvantage of the regression models is tied to their ease of interpretation: these models cannot learn very complex relationships. To illustrate, a regression model might learn to predict a low visitor count for Vondelpark if the temperature is low and a high visitor count if the temperature is high. However, it could be the case that as the temperature becomes very high, the relationship between temperature and visitor count is no longer positively but negatively correlated. A regression model would not be able to learn this kind of relationship. Although, a more complex relationship might be incorporated in the model by using multiple predictor variables that together form the more complex relationship (e.g. an additional variable that indicates whether the temperature is above a certain threshold with a high negative weight).

Second, for the LSTM models the reverse holds: this model is able to learn complex relationships between the external factors and ground truth data, but is less straightforward in its interpretation. One method one could use to explain predictions using a LSTM model is by altering the values of predictor variables for a specific time slot and subsequently examining what the effect is on the prediction. Another disadvantage of this model type is

the increase in training time compared to for example the regression models, as shown in Section 3.3.8.

Third, it could be argued that the SARIMAX model lies more in the middle when considering the interpretability/complexity trade-off. On the one hand, the influence of the external factors could be examined in a similar way as with the regression models, and on the other hand the model can incorporate the temporal component of the data by means of the auto-regressive parameters. This gave us high expectations when applying this model type to this prediction problem. However, when working with this model we found out that to be able to estimate the model's parameters, too much time and computational resources were needed which resulted in the model not being suitable for this prediction problem (see Section 3.3.8). One possible reason for this issue has to do with the high frequency of the time series data (the data consisted of 15-minute samples, whereas ARIMA-like models are usually applied to time series data with lower frequencies, such as daily samples).

Lastly, an advantage of the non-linear regression model is that a specific cost function can be defined that can be optimized when the model is being trained. In relation to this, a downside of using this model type is that the chosen cost function should be a good match for the specific prediction problem. In this thesis we found that the defined cost function resulted in either underestimating of overestimating the crowdedness levels (by assigning either too much or too little cost to incorrect predictions). Thus, we found that the traditional cost functions were actually more suited for this problem (for example ordinary least squares or cross entropy loss).

7.2.2 Regression vs. classification approach

When looking at the results, we did not find an obvious difference in model performance between the models that use regression (predicting visitor counts) or the models that use classification (predicting crowdedness levels). For both the regression models and LSTM models, there are some differences in model performance across locations and data sets (data from the first experiment versus data from the final experiment) but there is no consistent pattern of results in which either the regression or classification approach leads to higher prediction accuracy. Thus, these findings suggest that crowdedness can be predicted equally well using either approach.

7.3 Challenges and limitations

Next, we discuss some important challenges and limitations in relation to the experiments carried out in this thesis. These consist of challenges related to the sparseness of crowded moments, characteristics of the ground truth data, the definition of crowdedness levels, and the amount of data used.

First, a challenge that is specific to the topic of predicting crowdedness is the low occurrence of very crowded moments (known as the problem of class imbalance). This is expressed in this thesis in two ways: since the data consists of relatively few crowded moments, it is difficult for the models to distinguish the very crowded moments from the other crowdedness levels based on training data, and, it is difficult to gather a sufficiently large amount of very crowded moments in a test set to validate the models with. For example, for Dam square we saw that for the whole month of April no peaks in crowdedness occurred, which makes it impossible to evaluate the model based on its ability to predict very crowded moments. We tried to tackle this issue by introducing an oversampling technique for the regression models, and where possible, to select test data so that at least a few crowded moments were included in the set.

Second, there are several challenges that arose while working with the mobile phone data as ground truth data for crowdedness. This data is difficult to validate as ground truth data, as only a subgroup of the true amount of visitors is included in the measured visitor count. Because of this, the estimated visitor count is scaled up with the use of other external data sources (such as camera data, road data or public transport data). Thus, errors can occur when trying to estimate the real visitor count. Additionally, there is a bias in this data source with regard to certain groups of people (e.g. children without phones are not recorded; tourists are less likely to be recorded).

Another challenge related to the mobile phone data is the presence of noise and variation in noise levels over time. Occassionally, false peaks occur in the data, and to deal with this we tried to label these as outliers and alter them. However, it often remained difficult to be certain of whether a peak in crowdedness reflects a true crowded moment or noise. One method we used to test this was to examine news articles on reported crowdedness or closures of the locations (e.g. on some days Vondelpark had to be closed because it became too crowded).

Additionally, we saw that in the end of 2020 more noise seemed to be present in the data compared to the first quarter of 2021. On the one hand, this of course means that the quality of the ground truth data has improved over time, but on the other hand this made it difficult

to combine longer periods of data into a single data set. These changes across time led to changes in the thresholds as well, with the result that if we were to use newer thresholds on a data set of a larger time period, the older data would consist of many crowded moments incorrectly labeled as such.

Furthermore, a challenge related to the mobile phone data and the definition of crowdedess levels is that the visitor counts should be treated as relative counts. While this is not necessarily a problem for this prediction task, as relative counts can be converted into crowdedness levels, it does make it difficult to interpret crowdedness patterns and define appropriate crowdedness levels. For example, say we know that for a certain location 100 visitors means that it is too crowded, and the mobile phone data estimates 100 visitors, how can we know whether this value should be the threshold value if it can only be interpreted in the relative sense? Because of this, the thresholds are defined by using a combination of absolute (e.g. what number of visitors indicates that the location is crowded?) and relative visitor counts (e.g. what number of visitors belongs to the top 10 percent of data points with the highest values?). As a result of this, for some locations the absolute values and thresholds seem unrealistically low (e.g. Albert Cuyp).

Third, a limitation of the experiments performed is the amount of data that has been used, both for training and testing the models. As mentioned in the previous subsection, for some of the external factors we do not have a complete picture of their influence on crowdedness predictions since the used data sets do not cover very long periods (e.g. multiple seasons) which leads to less variation within the data for some of the predictor variables.

Lastly, another limitation of the present work is that for most of the external factors we are restricted to work with expected data instead of real-time data. This limitation was already briefly discussed in the previous section. This is an important limitation, since information on the external factors with respect to the moment that one is trying to predict is more informative than information on a past, though comparable, moment. One possible way to improve on this is by investigating whether an additional modelling step can be implemented, in which predictions are made for the external factors, which then in turn will be used for the crowdedness predictions.

7.4 Future work

Based on the findings and challenges that we encountered in this research, we provide some ideas for future work on this topic.

There are several other data sources that might be beneficial for crowdedness prediction

models. For example, as more COVID-19 regulations are being revoked, more events will start taking place again. These can be planned events, that can be incorporated by for example using information on whether an event takes place on a certain day, or perhaps even the expected amount of visitors for a certain event. These can also be unplanned events, which are more difficult to incorporate in the model, but could be signaled indirectly by for example using social media data, such as Twitter data (e.g. tweets related to a demonstration).

Another data source that might be valuable to add is data on road works. For example, for Dam square we saw that for some period in January there was a construction site at one of the roads beside Dam square (of which part is included in the area for which the ground truth data is being measured), which caused an apparent drop in crowdedness for multiple weeks. Finally, data on tourism, especially in post COVID-19 times, might also be predictive of crowdedness. Examples of useful data sources related to tourism could be public transport data with respect to trains, or data on incoming flights from a nearby airport.

Another suggestion is that the external factors could be used for crowdedness predictions in a different way altogether. In this thesis, we concatenated the external factors into one set of predictors and used this combined data to predict crowdedness, which can be referred to as early integration. However, Alpaydin (2018) discusses two alternative ways to combine multi-modal data: intermediate integration and late recognition [65]. In intermediate integration, the data of the different external factors is first pre-processed to form more abstract representations, after which they are combined similar as with early integration. In late recognition, separate models are trained for data of each modality, and subsequently the predictions of the different models are combined into a single prediction.

The motivation to use either intermediate integration or late recognition when dealing with multi-modal data is that with early integration, as more external factors are added the model becomes more complex. Also, the external factors may be on different scales, making it difficult for a model to process them together. For this specific prediction task, there is variety in the type of external factors (for example, continuous visitor counts are very different from a binary holiday indicator). Therefore the predictions might improve if the external factors were combined in a different manner.

Related to this suggestion, another idea is to investigate the use of two separate models. One model can be viewed as a baseline model that predicts the typical daily crowdedness patterns, only using information on the historical values of the ground truth data and periodic information (for example using the linear or ordinal regression model). The

second model can be viewed as a crowdedness detection model that predicts whether it is likely to become very crowded or not at a given moment in time, using information on other factors such as holidays, events and social media data. The idea is then that in the first place the baseline model's prediction is used, but if the second model has sufficient reason to believe that it will become very crowded (for example by reaching a certainty threshold) the prediction will be overwritten by this model's prediction.

Finally, another recommendation for future work is to investigate the use of a model that can be applied to multiple locations throughout the city. In this thesis, we created models for individual locations, but another option is to use a model that can have multiple inputs and multiple outputs, predicting crowdedness for multiple locations at once. In this way, spatial information could also be incorporated in the model. For example, we could then examine whether crowdedness at one location affects crowdedness at other locations. Furthermore, it would then also be interesting to use additional information on the characteristics of the locations (e.g. park versus square, shopping area vs. residential area). This has been done by multiple studies discussed in Section 2 [4, 6, 7], where crowd flows were modelled across multiple regions of a city.

8. Conclusion

In this thesis, we examined the use of various external factors and model types for short-term crowdedness predictions. In doing so, we trained and tested a set of models using data of multiple external factors for three different locations in Amsterdam. We have found that the typical daily crowdedness patterns can be predicted accurately, and the challenge lies in predicting (unexpected) peaks in crowdedness. Specifically, very crowded moments were often not predicted as such by the models. Furthermore, for some locations the crowdedness patterns were more difficult to predict than for other locations. Specifically, for Albert Cuyp model performance was better overall compared to Dam square and especially Vondelpark. However, important to note here is that the Albert Cuyp data consisted of more crowded moments than the data of the other two locations, which could have resulted in higher prediction accuracy.

To summarize, based on the results we found that the predictors that have the largest influence on the model predictions are information on the historical values of the ground truth data and periodic data. Specifically, the most important predictors provided information on the following: the crowdedness of the previous time step, whether it was more or less crowded at the previous time step compared to the time step before that, the day of the week, and the time of day. Moreover, the linear regression and ordinal regression models achieved the best results overall, possibly due to the use of oversampling. Finally, there are no clear differences in prediction accuracy of crowdedness predictions when framing the prediction problem as either a regression or classification problem.

However, we also found that even when using a more complex prediction model, the LSTM, or when using oversampling techniques, crowded moments remained difficult to predict. Additionally, it could be the case that some external factors (such as the weather, or holidays) did not have a clear effect on the predictions because there might not have been sufficient variation in the data, or the different data sources might not have been combined in the most effective manner. Moreover, as was described in the previous section, there were several challenges related to the use of mobile phone data as ground truth data, such as the presence of noise and the interpretability of the visitor counts, that made this a challenging prediction problem.

All in all, we showed that when combining a regression model with oversampling, accurate predictions for crowdedness on the short-term can be generated for a set of example locations in the city of Amsterdam, each representing a different type of location (a square, park and market). These predictions can be used to support crowd management by indicating which locations are likely to become crowded in the next few hours.

9. Appendix

9.1 Preliminary experiment: Confusion matrices for all model types

9.1.1 Dam square

Baseline model

(a) Validation set results.

(b) Test set results.

Figure 59. Confusion matrices.

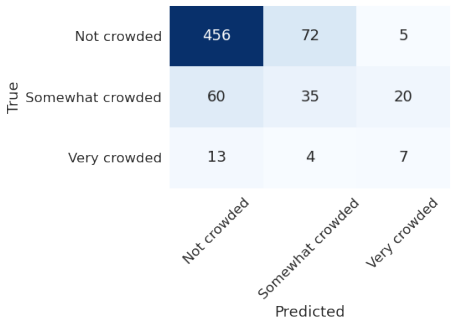
Linear regression model

(a) Validation set results.

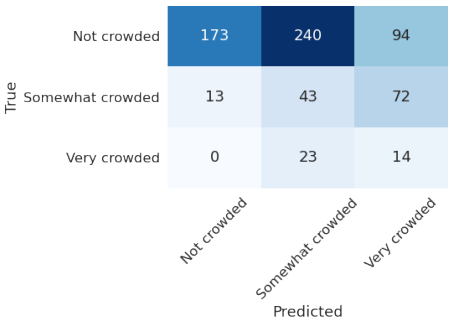
(b) Test set results.

Figure 60. Confusion matrices.

Non-linear regression model



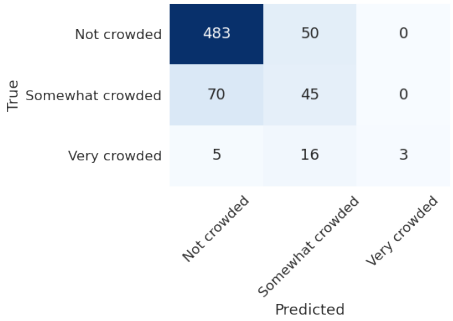
(a) Validation set results.



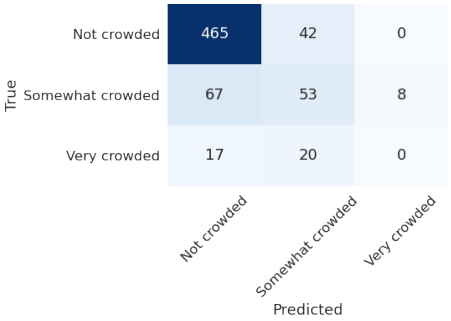
(b) Test set results.

Figure 61. Confusion matrices.

Ordinal regression model



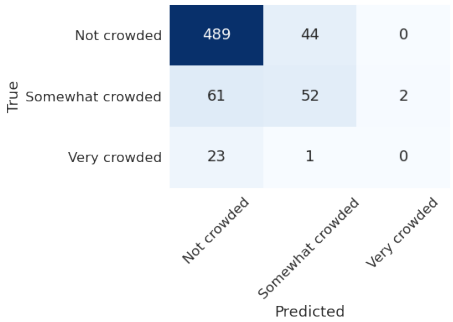
(a) Validation set results.



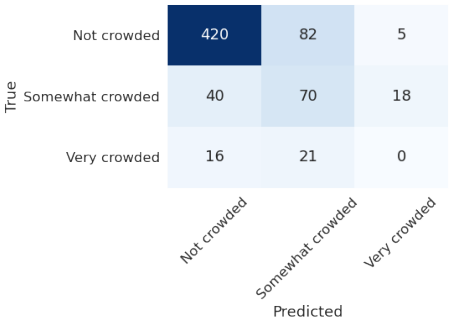
(b) Test set results.

Figure 62. Confusion matrices.

SARIMAX model



(a) Validation set results.



(b) Test set results.

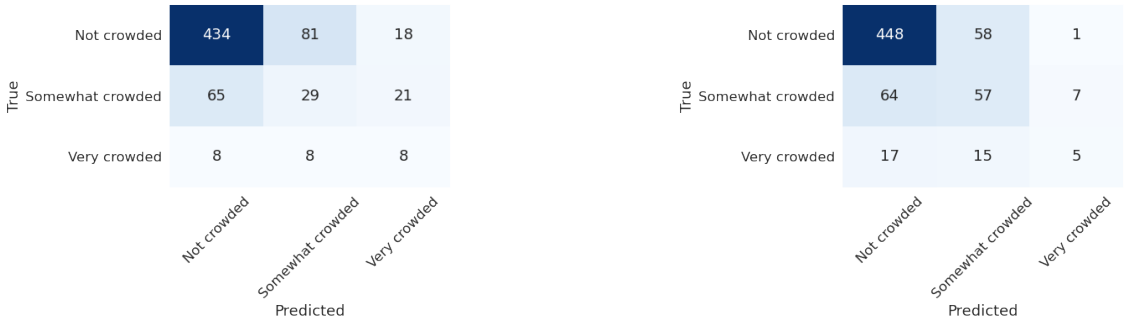
Figure 63. Confusion matrices.

LSTM model (regression)

(a) Validation set results. (b) Test set results.

Figure 64. Confusion matrices.

LSTM model (classification)

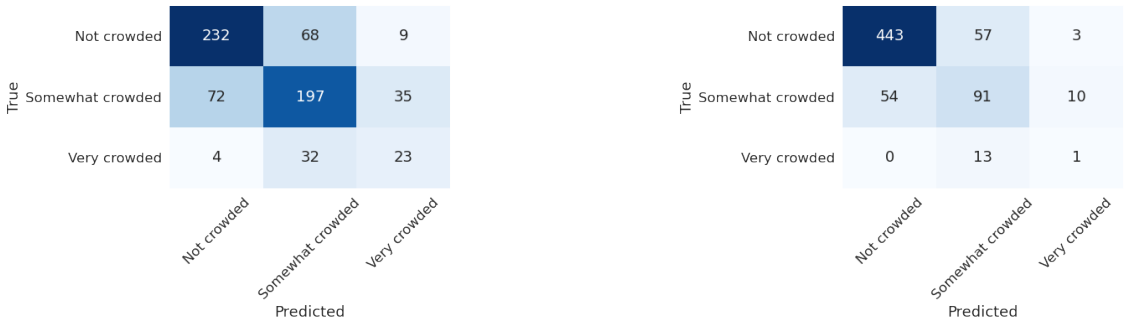


(a) Validation set results. (b) Test set results.

Figure 65. Confusion matrices.

9.1.2 Vondelpark

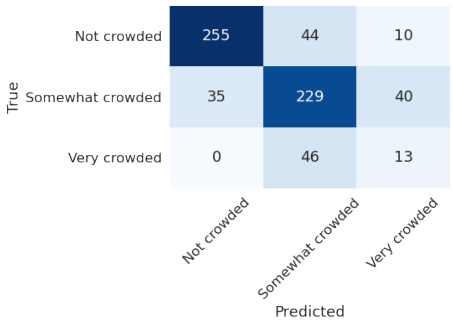
Baseline model



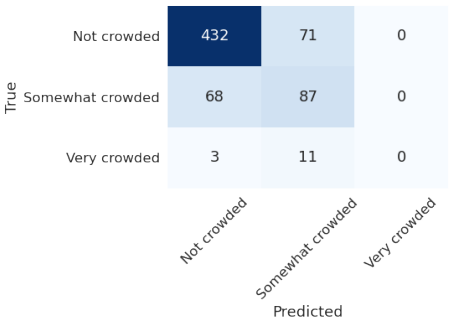
(a) Validation set results. (b) Test set results.

Figure 66. Confusion matrices.

Linear regression model



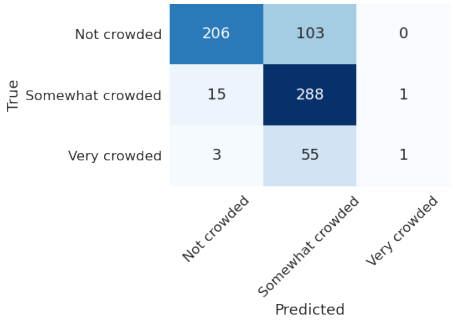
(a) Validation set results.



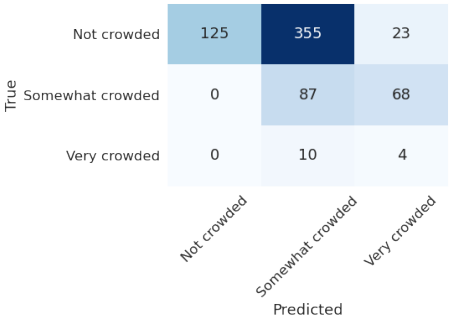
(b) Test set results.

Figure 67. Confusion matrices.

Non-linear regression model



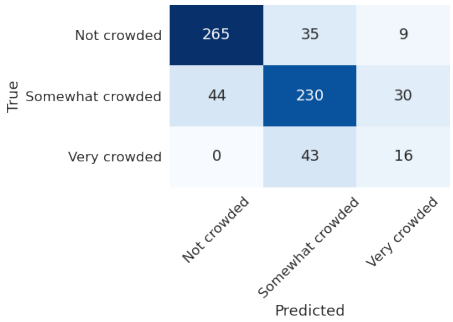
(a) Validation set results.



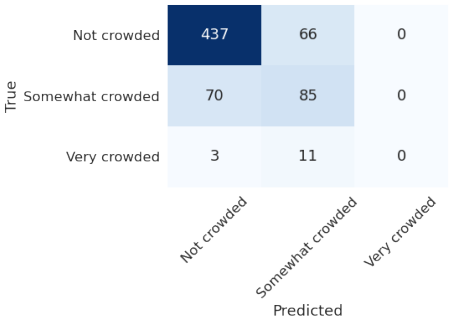
(b) Test set results.

Figure 68. Confusion matrices.

Ordinal regression model



(a) Validation set results.



(b) Test set results.

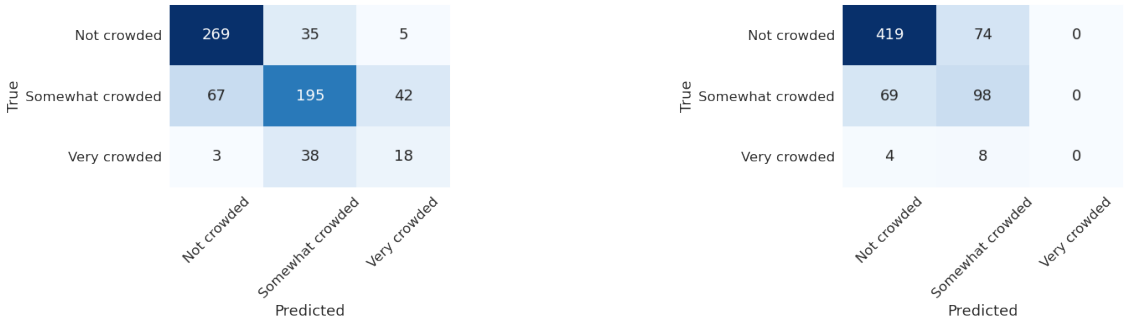
Figure 69. Confusion matrices.

SARIMAX model

(a) Validation set results. (b) Test set results.

Figure 70. Confusion matrices.

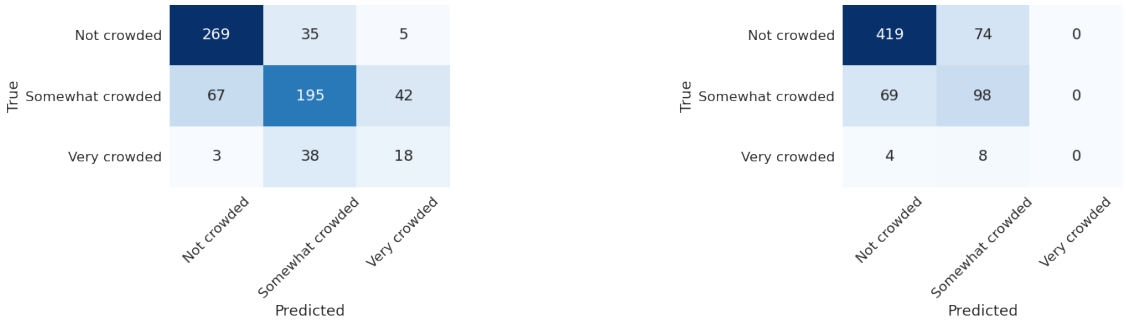
LSTM model (regression



(a) Validation set results. (b) Test set results.

Figure 71. Confusion matrices.

LSTM model (classification)

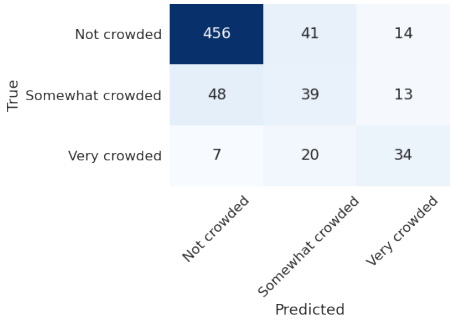


(a) Validation set results. (b) Test set results.

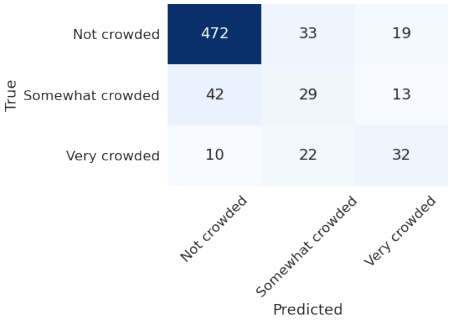
Figure 72. Confusion matrices.

9.1.3 Albert Cuyp

Baseline model



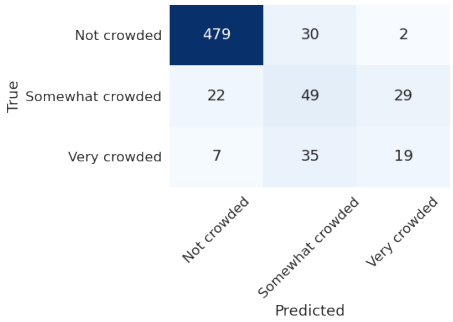
(a) Validation set results.



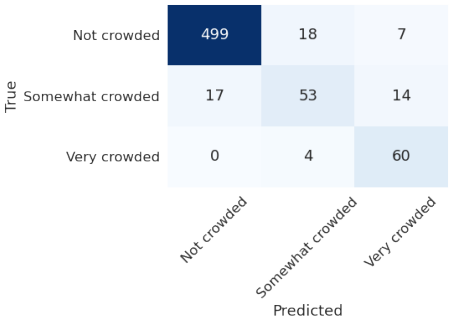
(b) Test set results.

Figure 73. Confusion matrices.

Linear regression model



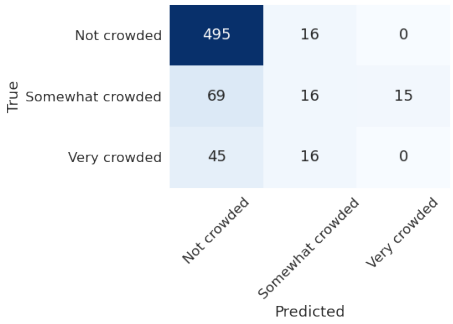
(a) Validation set results.



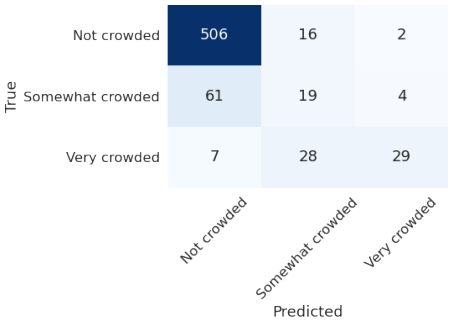
(b) Test set results.

Figure 74. Confusion matrices.

Non-linear regression model



(a) Validation set results.



(b) Test set results.

Figure 75. Confusion matrices.

Ordinal regression model

(a) Validation set results.

(b) Test set results.

Figure 76. Confusion matrices.

SARIMAX model

(a) Validation set results.

(b) Test set results.

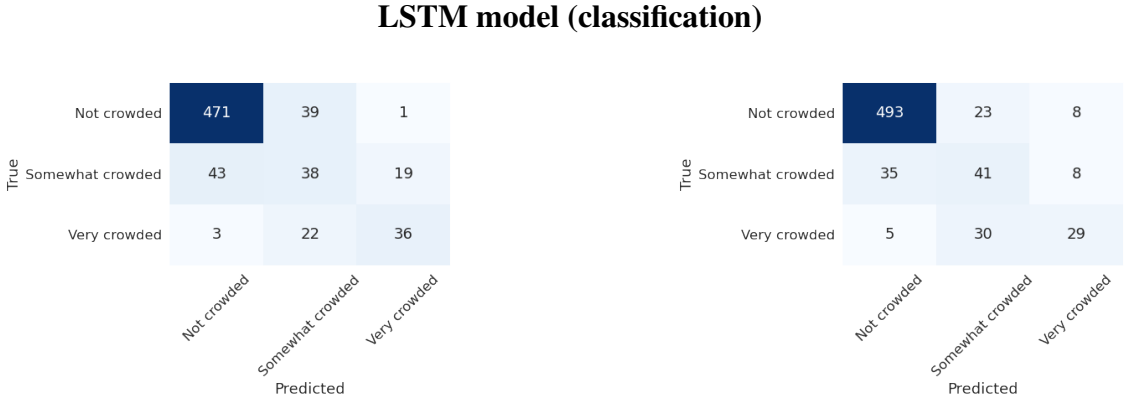
Figure 77. Confusion matrices.

LSTM model (regression

(a) Validation set results.

(b) Test set results.

Figure 78. Confusion matrices.



(a) Validation set results.

(b) Test set results.

Figure 79. Confusion matrices.

9.2 Experiment on the full data set: Confusion matrices for all selected model types

9.2.1 Dam square

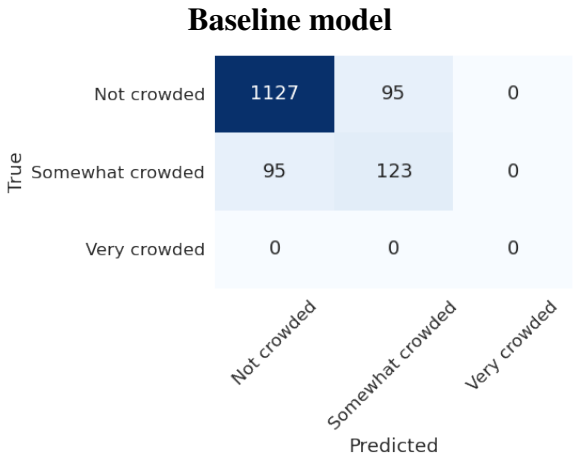


Figure 80. Confusion matrix based on the test set results.

Linear regression model

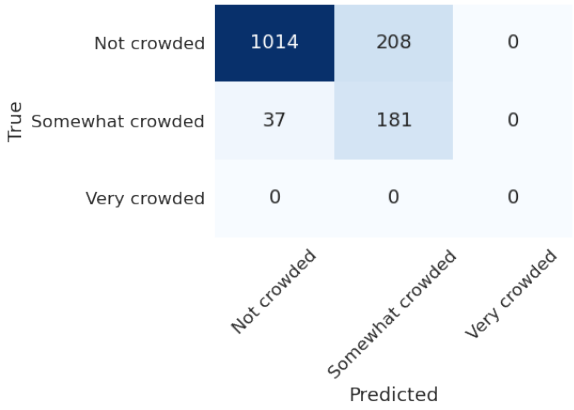


Figure 81. Confusion matrix based on the test set results.

Ordinal regression model

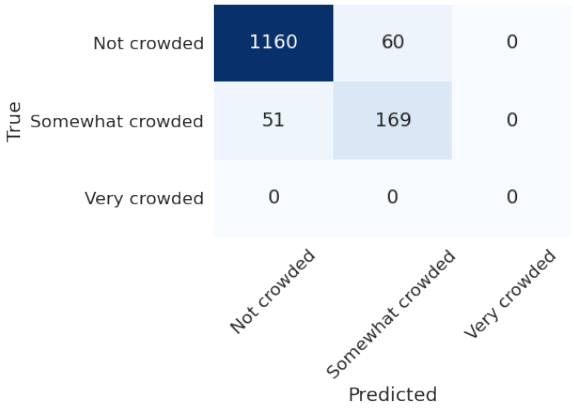


Figure 82. Confusion matrix based on the test set results.

LSTM model (regression)

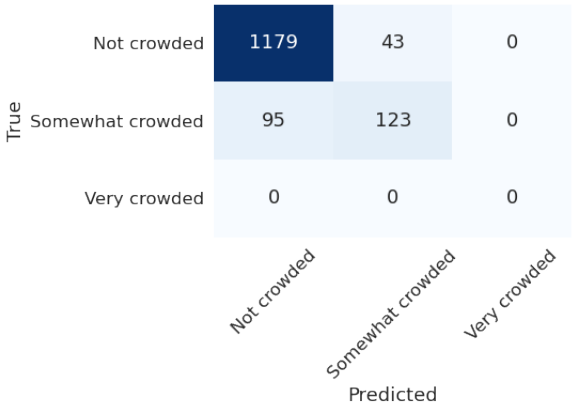


Figure 83. Confusion matrix based on the test set results.

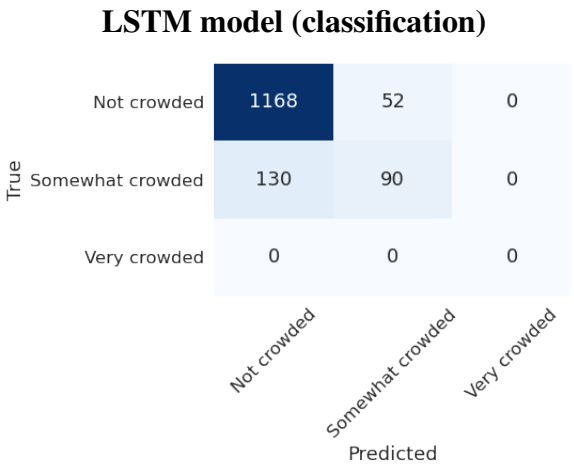


Figure 84. Confusion matrix based on the test set results.

9.2.2 Vondelpark

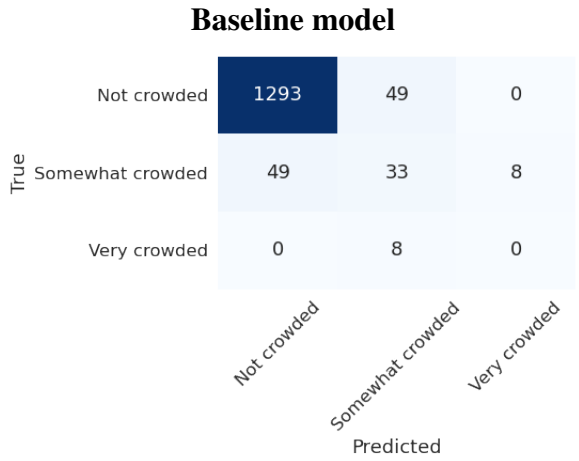


Figure 85. Confusion matrix based on the test set results.

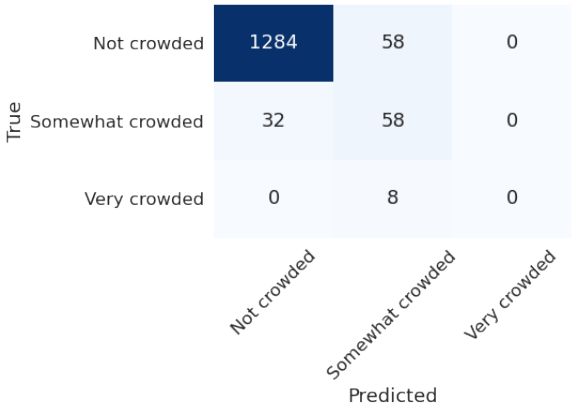


Figure 86. Confusion matrix based on the test set results.

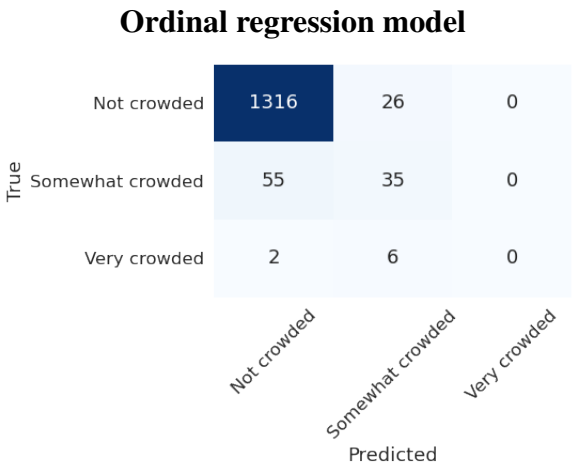


Figure 87. Confusion matrix based on the test set results.

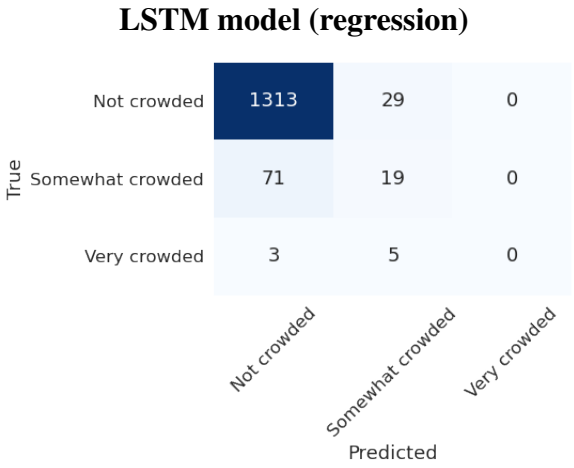


Figure 88. Confusion matrix based on the test set results.

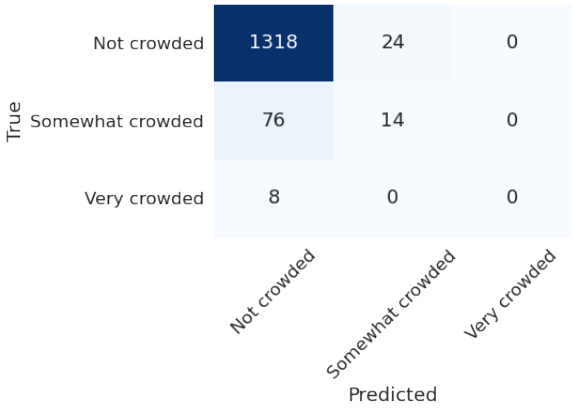


Figure 89. Confusion matrix based on the test set results.

9.2.3 Albert Cuyp

Baseline model

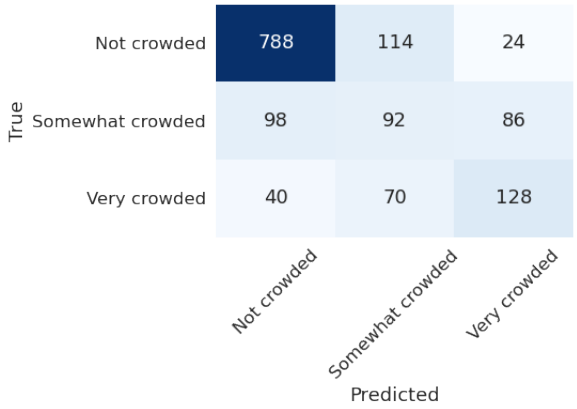


Figure 90. Confusion matrix based on the test set results.

Linear regression model

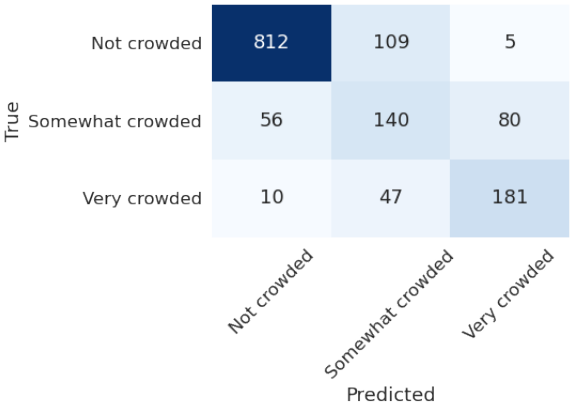


Figure 91. Confusion matrix based on the test set results.

Ordinal regression model

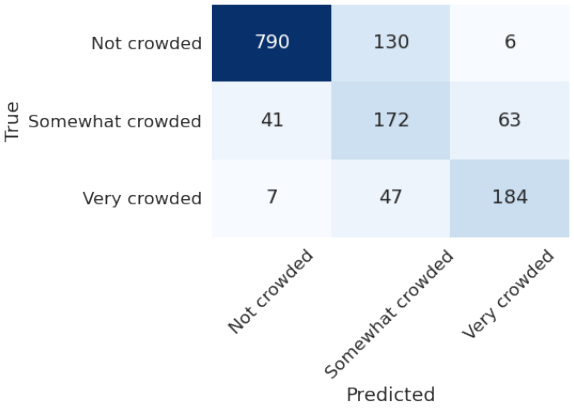


Figure 92. Confusion matrix based on the test set results.

LSTM model (regression)

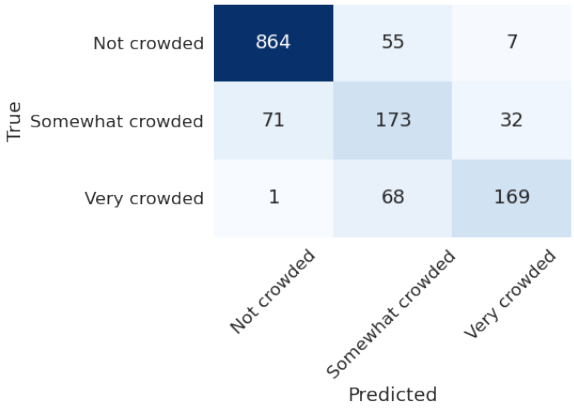


Figure 93. Confusion matrix based on the test set results.

LSTM model (classification)

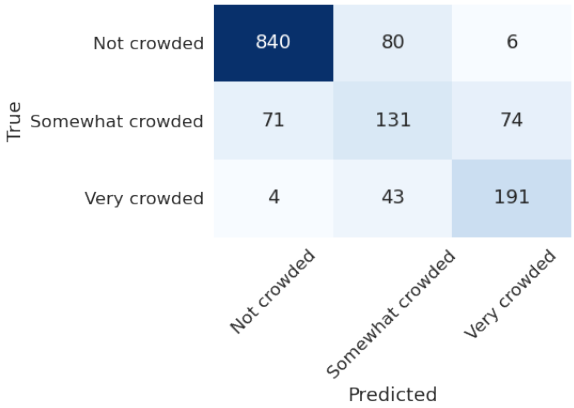


Figure 94. Confusion matrix based on the test set results.

Acronyms

AE Average Error (Section 2.4.1).

ARIMA Auto-Regressive Integrated Moving Average (Section 2.3.1, 3.3.5).

ARIMAX Auto-Regressive Integrated Moving Average with Exogenous variables (Section 2.3.1).

CDR Call Detail Records (Section 2.2.2, 2.1.2).

CMSA Crowd Monitoring System Amsterdam (Section 3.1.2.2).

FN False Negative (Section 3.2.3).

FP False Positive (Section 3.2.3).

GPS Global Positioning System (Section 2.2.1, 2.2.2, 2.1.2).

HMM Hidden Markov model (Section 2.3.2).

KNMI Koninklijk Nederlands Meteorologisch Instituut (Section 3.1.2.3).

LSTM Long Short-Term Memory (Section 2.3.3.1, 3.3.6).

MAE Mean Absolute Error (Section 2.4.1, 3.3.3).

MAPE Mean Absolute Percentage Error (Section 2.4.1).

mRMR Max-Relevance Min-Redundancy (Section 3.4, 4.4.4, 7.1).

RMSE Root Mean Squared Error (Section 2.4.1, 3.3.6).

SARIMA Seasonal Auto-Regressive Integrated Moving Average (Section 2.3.1, 3.3.5).

SARIMAX Seasonal Auto-Regressive Integrated Moving Average with Exogenous variables (Section 2.3.1, 3.3.5).

SMOTE Synthetic Minority Oversampling Technique (Section 3.1.3.8).

SMOTE-NC Synthetic Minority Oversampling Technique-Nomial Continuous (Section 3.1.3.8).

ST-DCCNAL Spatio-Temporal Densely Connected Convolutional Networks and Attention LSTM (Section 2.3.3.1).

ST-ResNet Seasonal Spatio-Temporal Residual Network (Section 2.2.1).

TN True Negative (Section 3.2.3).

TNR True Negative Rate (Section 3.2.3).

TP True Positive (Section 3.2.3).

TPR True Positive Rate (Section 3.2.3).

Bibliography

[1] Juan Ramón Santana et al. “A Privacy-Aware Crowd Management System for Smart Cities and Smart Buildings”. In: *IEEE Access* 8 (2020), pp. 135394–135405.

[2] Nanda Wijermans et al. “A landscape of crowd-management support: An integrative approach”. In: *Safety science* 86 (2016), pp. 142–164.

[3] Caspar AS Pouw et al. “Monitoring physical distancing for crowd management: Real-time trajectory and group analysis”. In: *PloS one* 15.10 (2020), e0240963.

[4] Junbo Zhang, Yu Zheng, and Dekang Qi. “Deep spatio-temporal residual networks for citywide crowd flows prediction”. In: *arXiv preprint arXiv:1610.00081* (2016).

[5] Minh X Hoang, Yu Zheng, and Ambuj K Singh. “FCCF: forecasting citywide crowd flows based on big data”. In: *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 2016, pp. 1–10.

[6] Wei Li et al. “Densely Connected Convolutional Networks With Attention LSTM for Crowd Flows Prediction”. In: *IEEE Access* 7 (2019), pp. 140488–140498.

[7] Hao Yuan et al. “Deep multi-view residual attention network for crowd flows prediction”. In: *Neurocomputing* 404 (2020), pp. 198–212.

[8] Cao Lijun and Huang Kaiqi. “Video-based crowd density estimation and prediction system for wide-area surveillance”. In: *China Communications* 10.5 (2013), pp. 79–88.

[9] Mu Mu. “WiFi-based Crowd Monitoring and Workspace Planning for COVID-19 Recovery”. In: *arXiv preprint arXiv:2007.12250* (2020).

[10] Zipei Fan et al. “CityMomentum: an online approach for crowd behavior prediction at a citywide level”. In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2015, pp. 559–569.

[11] Dorine C Duives, Guangxing Wang, and Jiwon Kim. “Forecasting pedestrian movements using recurrent neural networks: An application of crowd monitoring data”. In: *Sensors* 19.2 (2019), p. 382.

[12] Deepak Sharma et al. “A review on technological advancements in crowd management”. In: *Journal of Ambient Intelligence and Humanized Computing* 9.3 (2018), pp. 485–495.

[13] Adarsh Jagan Sathyamoorthy et al. “COVID-Robot: Monitoring social distancing constraints in crowded scenarios”. In: *arXiv preprint arXiv:2008.06585* (2020).

[14] Jorge Alvarez-Lozano, J Antonio Garcıa-Macıas, and Edgar Chávez. “Crowd location forecasting at points of interest”. In: *International Journal of Ad Hoc and Ubiquitous Computing* 18.4 (2015), pp. 191–204.

[15] Utkarsh Singh et al. “Crowd Forecasting based on WiFi Sensors and LSTM Neural Networks”. In: *IEEE Transactions on Instrumentation and Measurement* (2020).

[16] Barbara Furletti et al. “Discovering and understanding city events with big data: the case of rome”. In: *Information* 8.3 (2017), p. 74.

[17] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[18] NOS. “Volle Dam bij racismeprotest, geen anderhalvemeterboetes uitgedeeld”. In: *NOS* (June 1, 2020). URL: https://nos.nl/collectie/13842/artikel/2335843-volle-dam-bij-racismeprotest-geen-anderhalvemeterboetes-uitgedeeld.

[19] Government of the Netherlands. “Temporary tightening of partial lockdown”. In: *www.government.nl* (Nov. 3, 2020). URL: https://www.government.nl/latest/news/2020/11/03/temporary-tightening-of-partial-lockdown.

[20] Philippe Esling and Carlos Agon. “Time-series data mining”. In: *ACM Computing Surveys (CSUR)* 45.1 (2012), pp. 1–34.

[21] Mario Cools, Elke Moons, and Geert Wets. “Investigating the variability in daily traffic counts through use of ARIMAX and SARIMAX models: assessing the effect of holidays on two site locations”. In: *Transportation research record* 2136.1 (2009), pp. 57–66.

[22] Junbo Zhang et al. “DNN-based prediction model for spatio-temporal data”. In: *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 2016, pp. 1–4.

[23] Ari Arango Jake Esprabens. *Time Series for Beginners*. June 2020. URL: https://bookdown.org/JakeEsprabens/431-Time-Series/.

[24] A. Malali J. R. Maat and P. Protopapas. *TimeSynth: A Multipurpose Library for Synthetic Time Series in Python*. 2017. URL: http://github.com/TimeSynth/TimeSynth.

[25] Andrew T Jebb et al. “Time series analysis for psychological research: examining and forecasting change”. In: *Frontiers in psychology* 6 (2015), p. 727.

[26] M Claeys Bouuaert. *Modeling crowds at mass-events: learning large-scale crowd dynamics from Bluetooth tracking data*. 2013.

[27] Danish A Alvi. "Application of probabilistic graphical models in forecasting crude oil price". In: *arXiv preprint arXiv:1804.10869* (2018).

[28] Yan Qi and Sherif Ishak. "A Hidden Markov Model for short term prediction of traffic conditions on freeways". In: *Transportation Research Part C: Emerging Technologies* 43 (2014), pp. 95–111.

[29] Lawrence R Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[30] Dan Jurafsky and James H Martin. *Speech and Language Processing*. 3rd. 2019.

[31] Sotirios P Chatzis and Yiannis Demiris. "A reservoir-driven non-stationary hidden Markov model". In: *Pattern recognition* 45.11 (2012), pp. 3985–3996.

[32] Zachary C Lipton, John Berkowitz, and Charles Elkan. "A critical review of recurrent neural networks for sequence learning". In: *arXiv preprint arXiv:1506.00019* (2015).

[33] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[34] Christopher Olah. *Understanding LSTM Networks*. Aug. 2015. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[35] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. "A comparison of ARIMA and LSTM in forecasting time series". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2018, pp. 1394–1401.

[36] B. Aydoğdu and A. A. Salah. "Machine Learning for Urban Computing". In: *S. Carta (ed.), Machine Learning, Artificial Intelligence and Urban Assemblages: Applications in architecture and urban design, Wiley,* (forthcoming).

[37] Christoph Bergmeir and José M Benıtez. "On the use of cross-validation for time series predictor evaluation". In: *Information Sciences* 191 (2012), pp. 192–213.

[38] Mohammad Hossin and MN Sulaiman. "A review on evaluation metrics for data classification evaluations". In: *International Journal of Data Mining & Knowledge Management Process* 5.2 (2015), p. 1.

[39] Alaa Tharwat. "Classification assessment methods". In: *Applied Computing and Informatics* (2020).

[40] *Betere beslissingen dankzij betrouwbare data en inzichten*. Oct. 2020. URL: https://reso.no/.

[41] *About KNMI*. URL: https://www.knmi.nl/over-het-knmi/about.

[42] *KNMI Data Platform*. URL: https://dataplatform.knmi.nl/open-data-info/index.html.

[43] *Coronavirus Government Response Tracker*. URL: https://www.bsg.ox.ac.uk/research/research-projects/coronavirus-government-response-tracker.

[44] *Coronavirus Government Response Tracker*. URL: https://github.com/OxCGRT/covid-policy-tracker.

[45] Pushpendra Singh et al. "Some studies on nonpolynomial interpolation and error analysis". In: *Applied Mathematics and Computation* 244 (2014), pp. 809–821.

[46] Mathieu Lepot, Jean-Baptiste Aubin, and François HLR Clemens. "Interpolation in time series: An introductive overview of existing methods, their performance criteria and uncertainty assessment". In: *Water* 9.10 (2017), p. 796.

[47] Bernhard Schölkopf et al. "Support vector method for novelty detection." In: *NIPS*. Vol. 12. 1999, pp. 582–588.

[48] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[49] Leonard J Tashman. "Out-of-sample tests of forecasting accuracy: an analysis and review". In: *International journal of forecasting* 16.4 (2000), pp. 437–450.

[50] Andy P Field, Jeremy Miles, and Zoë Field. *Discovering statistics using R/Andy Field, Jeremy Miles, Zoë Field*. 2012.

[51] Jason DM Rennie and Nathan Srebro. "Loss functions for preference levels: Regression with discrete ordered labels". In: *Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling*. Vol. 1. 2005.

[52] Ananth Ranganathan. "The levenberg-marquardt algorithm". In: *Tutoral on LM algorithm* 11.1 (2004), pp. 101–110.

[53] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[54] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[55] Thomas Kluyver et al. "Jupyter Notebooks – a publishing format for reproducible computational workflows". In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Ed. by F. Loizides and B. Schmidt. IOS Press. 2016, pp. 87–90.

[56] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830.

[57] Fabian Pedregosa-Izquierdo. “Feature extraction and supervised learning on fMRI: from practice to theory”. PhD thesis. Université Pierre et Marie Curie-Paris VI, 2015.

[58] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[59] Skipper Seabold and Josef Perktold. “statsmodels: Econometric and statistical modeling with python”. In: *9th Python in Science Conference*. 2010.

[60] Francois Chollet et al. *Keras*. 2015. URL: https://github.com/fchollet/keras.

[61] Hanchuan Peng, Fuhui Long, and Chris Ding. “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy”. In: *IEEE Transactions on pattern analysis and machine intelligence* 27.8 (2005), pp. 1226–1238.

[62] *Welcome to GVB*. URL: https://en.gvb.nl/.

[63] *Find Propaganda Networks*. URL: https://www.trollrensics.com/.

[64] Thomas G Dietterich. “Approximate statistical tests for comparing supervised classification learning algorithms”. In: *Neural computation* 10.7 (1998), pp. 1895–1923.

[65] Ethem Alpaydin. “Classifying multimodal data”. In: *The Handbook of Multimodal-Multisensor Interfaces: Signal Processing, Architectures, and Detection of Emotion and Cognition-Volume 2*. 2018, pp. 49–69.