Forecasting Crowd Counts with Wi-Fi Systems: Univariate, Non-seasonal Models

Jean-François Determe*, Utkarsh Singh*, François Horlin*, and Philippe De Doncker*

***Abstract*—Recently, event organizers and researchers have advocated the development of novel technologies supporting crowd control, notably for public events. This paper presents a crowd monitoring system based on probe requests (PRs), which are Wi-Fi packets smartphones send periodically. By estimating the global rate at which nearby smartphones send PRs, Wi-Fi sensors can estimate crowd counts. The core contribution of this paper is a computationally tractable method that forecasts crowd counts up to thirty minutes in the future, with forecasts becoming available as soon as two hours of data are available. The forecasting method relies on autoregressive integrated moving average (ARIMA) models. Contributions also include two methods that compute prediction intervals associated with the forecasts, one of which is based upon generalized autoregressive conditional heteroskedasticity (GARCH) models. Recent real-world data from Winter Wonders 2018/2019 (an event that took place in Brussels, Belgium) notably demonstrate that the proposed forecasting method outperforms its immediate variations as well as baseline models (i.e., random walk models).**

***Index Terms*—Crowd monitoring and control, forecast, autoregressive time series, ARIMA, GARCH, Box-Cox transformation**

I. Introduction

MONITORING large public events is a key challenge with which event organizers deal. Such a monitoring entails estimating crowd densities in real time to determine whether they exceed acceptable limits [1]. Estimating crowd densities allows law enforcement personnel to close roads, redirect people to less crowded areas and detect abnormal patterns (such as sudden influxes of people). Preventing crowd overcrowding not only entails monitoring but also forecasting, for countermeasures should be executed prior to the occurrence of crowd overcrowding.

This paper presents a crowd monitoring approach based on data acquired by Wi-Fi sensors, which collect probe requests (PRs)—special Wi-Fi packets that are internal to the Wi-Fi protocol and broadcasted by user terminals (e.g., smartphones, computers) to detect nearby access points (APs). Counting such messages provides a proxy for the number of smartphones with Wi-Fi connectivity within the vicinity of the sensor; in turn, this number is a proxy for the number of people within the area covered by the deployed sensors (up to an *extrapolation factor*).

On the basis of a real-time Wi-Fi counting system that we developed, we present forecasting methods for crowd counts. Often, crowd densities need not be directly measured because, for a given available space, a maximum crowd density is easily converted into a maximum crowd count.

We show that integrated autoregressive time series models (such as auto regressive integrated moving average (ARIMA) models) forecast up to 30 minutes in the future. We also discuss methods to derive prediction intervals (PIs) for forecasts in real time, the former quantifying the reliability of the latter.

Although we applied our forecasting method on data stemming from a Wi-Fi counting system, it should perform properly if applied on counts obtained using different (yet comparably precise) technologies (e.g., cameras with unobstructed lines of sight). The reason is that crowd dynamics are unaffected by concealed nonobstructive counting devices and, as a result, any reliable proxy for crowd counts should generate accurate forecasts if fed into our forecasting method.

Recent interviews have revealed event managers wish to use modern technologies to prepare events and monitor them in real time [2, Sec. 7]. Our line of research focuses on these endeavors; it aims to i) help event organizers prepare future editions of events by providing them with crowd count estimates of past editions, ii) enable security teams to monitor events in real time, iii) forecast crowd counts and thus overcrowing. Out of all these use cases, our paper focuses on pure forecasting, which is the foundation for forecasting overcrowding and detecting abnormal counts.

For any event, computing short-term forecasts is of interest, because more sophisticated forecasting models relying on seasonality—the presence of a seasonal pattern emerging every day—or multi-variate approaches require more measurements to be fit than ours. Thus, to get forecasts for the first hours and days of such events, univariate non-seasonal models remain the best options. Moreover, for short-term forecasts, univariate models may be more accurate than seasonal ones. Finally,

*All authors are with the OPERA Wireless Communications Group, Université libre de Bruxelles, 1050 Brussels, Belgium. Corresponding e-mail: jdeterme@ulb.ac.be. Innoviris fully funded Jean-François Determe and Utkarsh Singh.

some events last several days but do not feature a strong seasonal pattern (e.g., music festivals with multiple stages, whose attendances strongly depend on the relative popularity of all the artists performing). As a result, the proposed forecasting methods are the most general ones in that they apply to the widest class of events.

The presented Wi-Fi counting system is, however, especially suited to open and free events because they feature no controlled entrance and exit points, thereby precluding the use of counting methods based on, e.g., barcode scanning or turnstiles. It is also suited to events that cameras cannot easily monitor, such as those with a complex setup that entails line-of-sight obstructions or poor lightning conditions; conventional cameras could also provide imprecise counts for open-air events because of weather effects (e.g., heavy rain and fog).

A. Detailed contributions

This work relies on Wi-Fi counts estimates derived from the rate at which all nearby smartphones emit probe requests. Our paper cursorily discusses this crowd estimation approach and show that its principle makes it unaffected by media access control (MAC) address randomization (see Section II-C).

Our main contribution is a method for computing forecasts (of the conditional mean) using univariate, non-seasonal approaches; forecasts start becoming available after 2 hours of measurements become available (typically prior to most of the attendees reaching the event). We also present and validate two viable ways of deriving PIs, one based upon canonical ARIMA models with Gaussian innovations and the other relying on generalized autoregressive conditional heteroskedasticity (GARCH) models [3], [4]. More precisely, our forecasts of the conditional mean are generated by a "rolling" ARIMA(2,2,1) model whose coefficients and innovation variance are reevaluated whenever a new measurement becomes available (every five minutes).

We validate our methods using real data (collected during *Winter Wonders 2018/2019*, an event in Brussels, see Section III). We show our forecasting method to be consistently superior (or at least comparable) to baseline forecasting models (i.e., the random walk (RW) model, see Section V-B). We also demonstrate that it outperforms its immediate variations, which rely on mainstream concepts in time series analysis.

B. Comparison with state of the art

Occupancy measurements have already been obtained using Wi-Fi PRs; detecting Wi-Fi PRs enabled researchers to i) localize Wi-Fi devices using unmanned aerial vehicles [5]; ii) measure occupancy in indoor environments [6], [7] (including motor shows [8]); iii) monitor occupancy in public places, such as festivals [9], airports [10], public transportation systems [11], [12] and other urban environments [13]. Up to 2016, [8, Sec. *Related work*] presents a good overview of measurement campaigns similar to ours.

Let us now briefly discuss the underlying technology for measurements. In comparison to other solutions—such as cameras or manual counts relying on humans—Wi-Fi sensors preserve privacy to a higher extent [14, Sec. 1] and may also incur lower expenditures [12, Table 1] [15]. From a technical point of view, they also do not suffer from dark lightning conditions or line-of-sight obstruction [14, Sec. 1] [2, Fig. 1].

Other methods based on Wi-Fi rely on the physical layer. The authors of [14] use channel state information (CSI) to count people: their experiments, however, include low number of people (no more than 30 people according to [14, Sec. IV-G and Fig. 15]) and there are, to the best of our knowledge, no experiments validating this approach for hundreds of people around a sensor. Other works based on CSI suffer from the same limitation (see [16, Sec. 7], [17]). The authors of [18] present another system, which is based on the received power between Wi-Fi devices to count people. Again, the system has not been tested on dense crowds [18, Sec. IV]. A recent survey on techniques used for crowd size estimation is [19].

Regarding forecasting crowd counts using Wi-Fi sensors, we found very few works in the literature. In [20], the authors propose a Wi-Fi localization system based on triangulation for indoor environments, which they use to forecast queuing times using methods similar to ours (autoregressive models). Another work is [21], which very succinctly shows how ARIMA models can forecast shopper volume in malls, with counts derived from Wi-Fi messages. The closest work to ours are some slides from 2013 [22], which cursorily show forecasting results for crowd counts obtained using Bluetooth sensors in a public event in *Ghent* (a city in Belgium). The work [23] reviews the different methods used in traffic forecasting as of 2014; these notably include smoothing approaches, Kalman filters and non-parametric modeling (based on non-parametric regression and neural networks). Future work endeavors could compare these methods with ours

This work presents forecasting results based on real measurements that do not frequently appear in the literature. Our measurement scenario involves dense crowds. We also focus on methods evaluating prediction intervals accurately, for quantifying the reliability of forecasts is important for event organizers and automated overcrowding detection algorithms—this is rarely done in works similar to ours.

C. Outline

First of all, Section II introduces Wi-Fi PRs and how to derive crowd counts by counting them. Section III succinctly presents *Winter Wonders 2018/2019* in Brussels, the real events on which we deployed sensors and collected data. Then, Section IV presents—from a theoretical point of view—the forecasting tools on which we rely. Section V describes the metrics for evaluating forecasting and PI accuracy and details the exact forecasting methods we test. Section VI evaluates the accuracy of the methods proposed in Section V on the basis of data from *Winter Wonders 2018/2019*. It evaluates both the accuracy of the forecasts and that of the associated PIs. Finally, Sections VII and VIII are the future work and conclusion, respectively.

II. Crowd counts using Wi-Fi probe requests

We shall now briefly discuss the detection and processing of probe requests to derive counts.

A. Probe requests

As already stated in the introduction, the estimated crowd counts that we use for forecasting purposes are derived from PRs [24, Chapter 4]. PRs are management frames Wi-Fi devices periodically send to ask nearby access points (APs) to make their existence known. This is an active scanning mechanism for discovering APs. Because of their purpose, smartphones transmit PRs even if they are not connected to a Wi-Fi network, which makes them interesting messages to generate crowd counts.

PRs notably include a source address (SA) field [24, Fig. 4-52], which is—in theory—the MAC address of the Wi-Fi controller on the mobile station. Several almost identical PRs are sent in a row, within a time frame lasting less than 10ms [25, Sec. 2.1]; in this paper, we call a set of such PRs a probe request burst (PRB). To make tracking smartphones difficult, modern operating systems randomize the MAC addresses in PRBs [25]–[27]. As discussed later on, MAC address randomization does not affect our counts.

B. Detection of probe requests

We developed sensors collecting and sending PRBs to a central server. Each one of our Wi-Fi sensors consists in

- A Raspberry Pi 3 running Raspbian Stretch.
- A Wi-Fi dongle supporting monitor mode (i.e., a mode that allows to capture all the detected Wi-Fi messages, without being connected to a Wi-Fi network). The dongle is an *Alfa AWUS036NHA*, whose chipset is an *Atheros AR9271L*. We use the standard straight antennas shipped with *Alfa AWUS036NHA* units. Sensors are installed with their antennas pointing to the sky or the ground vertically.
- A 4G dongle for providing Internet connectivity, thereby allowing sensors to transmit anonymized probe requests to a central server.

The sniffing program running on each Raspberry Pi is written in C++, is multi-threaded, and uses libpcap for packet capture. For each detected PRB, sensors send to a central server i) an anonymized MAC address of the PRB, ii) the timestamp of detection iii) a received signal strength indicator (RSSI) value, which is a number quantifying the received power.

In practice, when sensors are deployed, their ranges often overlap. This is preferable because it is hard to determine the exact range of a sensor, especially because it depends on the density of people in the area (the human body attenuates Wi-Fi signals). Therefore, a high density of sensors ensures that the area of interest is fully covered. Ranges overlapping implies that several sensors may detect identical PRBs. As a result, the central server jointly processes the measurements of neighboring sensors to derive counts, so that each detected PRB is counted only once. Figure 1 depicts the classical sensing scenario with overlapping sensor ranges.

C. Processing probe requests to compute counts

We deploy $n_{\mathcal{S}}$ sensors $\mathcal{S}_s$ in an event. We shall generate count time series for $n_{\mathcal{A}}$ areas $\mathcal{A}_\alpha$, subsets of sensors; these time series consist of counts every $T = 5$ minutes. The first step consists in computing counts for disjoint, contiguous elementary time frames of duration $T_e = 30$ seconds.

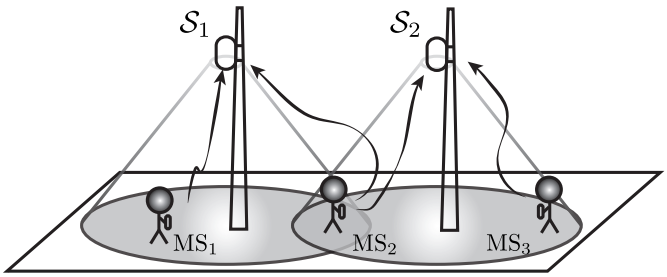

Fig. 1. WiFi sensing scenario. 2 sensors ($\mathcal{S}_1$ and $\mathcal{S}_2$) on poles detect the probe request bursts (PRBs) of 3 mobile stations (MSs). Each cone depicts the detection range of the corresponding sensor. Wavy arrows depict sensors detecting PRBs. The PRB of MS_2 is detected by both sensors whereas those of MS_1 and MS_3 are detected by the nearest sensor only.

For each elementary time frame, we extract from our database all the MAC addresses whose timestamps belong to it, which creates an array `arr_mac` of 3-tuples; the ith 3-tuple is $(\mathcal{S}^{(i)}, \mathrm{aMAC}^{(i)}, \mathrm{RSSI}^{(i)})$—where $\mathcal{S}^{(i)}$ is a sensor ID, $\mathrm{aMAC}^{(i)}$ denotes the ith MAC address and $\mathrm{RSSI}^{(i)}$ is the ith RSSI. For privacy reasons, sensors anonymize MAC addresses, which means $\mathrm{aMAC}^{(i)}$ is not a true MAC address but an anonymized one. Then, we count the number of distinct PRBs within each elementary time frame per sensor. A PRB, if detected multiple times, is uniquely associated with the sensor having measured the highest RSSI. To get a count for any area, it suffices to sum the counts of the sensors this area indexes. For the final time series, we generate counts every $T = 5$ minutes by averaging the counts of the ten corresponding elementary time frames.

Our method relies on the *rate* of PRB transmission on a short time frame (of $T_e = 30$ seconds). It does not try to deanonymize probe requests to track users for a long term (see, e.g., [24]). The number of distinct MAC addresses in an elementary time frame does not depend on MAC randomization because our elementary time frames are sufficiently short in comparison to the rate at consecutive PRBs are emitted. Whether some true MAC addresses are replaced by random ones does not change the number of distinct MAC addresses in an elementary time frame, which prevents MAC randomization from affecting our crowd count estimates.

D. A simple mathematical model of counting

This section formalizes mathematically our counting process. Let us assume n_{ppl} individuals are in a monitored area. During an elementary time frame, each individual has a probability $p_i \leq 1$ ($1 \leq i \leq n_{\mathrm{ppl}}$) to generate a PRB. The probability p_i of PRB generation for individual i may be zero if the individual does not carry a device with Wi-Fi enabled. If it is non-zero, it typically depends on how often the operating system of the smartphone requests PRBs be sent.

Thus, the probabilities p_i are random variables whose common distribution depends i) on how likely it is people turn off Wi-Fi (or have no smartphone) ii) how frequently an average smartphone sends PRBs. The set of all K possible

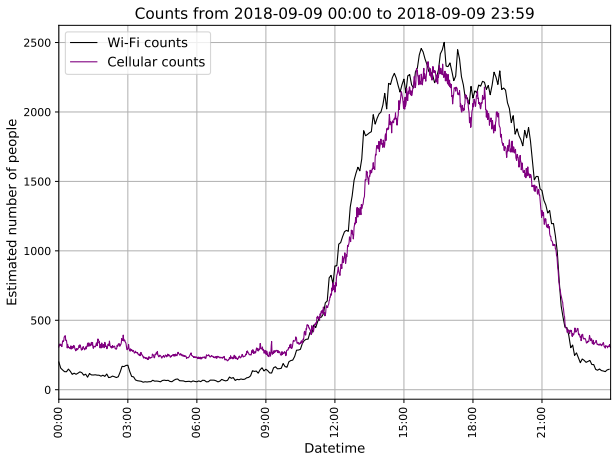

Fig. 2. Comparison of Wi-Fi counts against those from a local cellular cell on September 9, 2018 (for Eat! Brussels). The extrapolation factor for Wi-Fi counts is equal to 3.

probabilities is $\{\alpha_k\}_{1\leq k\leq K}$; for all $i \leq n_{\text{ppl}}$, $\mathbb{P}[p_i = \alpha_k] =: r_k$ $(1 \leq k \leq K)$ and $\sum_{k=1}^{K} r_k = 1$. The case $\alpha_k = 0$ corresponds to an individual without a Wi-Fi-active device.

The number of distinct PRBs in an elementary time frame is $X := \sum_{i=1}^{n_{\text{ppl}}} X_i$ where X_i is equal to 1 if the corresponding individual sends a PRB during an elementary time frame—thereby, we have $\mathbb{P}[X_i = 1|p_i] = p_i$ and $\mathbb{P}[X_i = 0|p_i] = 1 - p_i$. Thus, the marginal distribution of X_i is such that $\mathbb{P}[X_i = 1] = \sum_{k=1}^{K} \mathbb{P}[X_i = 1|p_i = \alpha_k]\mathbb{P}[p_i = \alpha_k] = \sum_{k=1}^{K} \alpha_k r_k =: \mathbb{E}[p_i]$ (law of total probability). As a result, $\mathbb{E}[X_i] := 1\,\mathbb{P}[X_i = 1]+0\,\mathbb{P}[X_i = 0] = \mathbb{E}[p_i]$. Our final (unbiased) estimator of the number of people in the area is $\hat{C} := \beta X$ where $\mathbb{E}[\hat{C}] = n_{\text{ppl}}$ with an extrapolation factor $\beta := 1/\mathbb{E}[p_i]$.

Our main conclusions are that:

- The exact extrapolation factor β is the inverse of the mean probability that an individual sends PRBs during an elementary time frame.
- MAC randomization does not affect our counting method because whether or not it occurs on an elementary time frame does not change the value of our estimator, $\hat{C}$.

E. Calibration of the counting system

We calibrated our system by comparing our data from a previous event with that of a telco operator and found that an extrapolation factor of 3 is adequate. We used data from an event of 2018 (Eat! Brussels), which takes place in the middle of a park in Brussels (including during a Sunday). This particular event setup is ideal because the counts associated with the local telco cell measures almost only people attending our event. Figure 2 plots our Wi-Fi counts against the telco counts and shows a good match.

Our tests also revealed that more than 95% of smartphones anonymize their PRBs (95% of observed PRBs only appear once). It means that the value T should have little importance, in that modifying T can be easily compensated for by adapting the extrapolation factor.

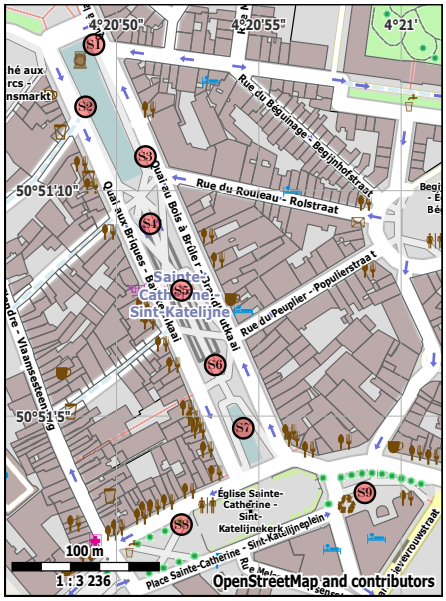

Fig. 3. Sensor arrangement on the area of Sainte-Catherine (St. Cath). Generated using Maperitive with mapping data from OpenStreetMap.

III. Presentation of Winter Wonders 2018/2019

Let us now discuss the event on which we collected data. *Winter Wonders 2018/2019* is an event that took place from November 30 (2018) to January 6 (2019) in Brussels, Belgium. Its two most populated areas are referred to as *Sainte-Catherine* (or *St. Cath*) and *Bourse*, areas on which this paper focuses. Figures 3 and 4 detail the areas of *Sainte-Catherine* and *Bourse*, respectively.

The area of *Sainte-Catherine* essentially comprises chalets wherein vendors sell theirs goods. Although Figure 3 suggests that some parts of the area are made of water, a wooden structure has been set above water for this event, which supports both pedestrians and chalets. The area also includes entrances to the local subway station, in between S4 and S5. The setup at *Bourse* is similar to that of *Sainte-Catherine*, except that the geometry of the area is different.

Counts obtained for both areas are available in Figures 5 and 6; these figures also include forecast results to shorten the paper. We report (and shall forecast) counts for the areas that correspond to the aggregation of sensors S1 to S7 and sensors S14 to S19. The days we chose are those for which all sensors were continuously online.

IV. Time series models: theoretical basis

A. Definition of a time series and related concepts

Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and the countable set $\mathbb{Z}$ of (time) indexes, a discrete stochastic process (or time series) x is a function [28, Sec. 1.2] $X : \mathbb{Z} \times \Omega \to \mathbb{R}$, where, for any $t \in \mathbb{Z}$, $X_t : \Omega \to \mathbb{R} : \omega \to x(t, \omega)$ is a random variable. We use the abuse of notation $\{X_t\}_{t\in\mathbb{Z}} = \{X_t\}$. The mathematical expectation operator is denoted by $\mathbb{E}[\cdot]$ and we assume that $\mathbb{E}[X_t^2] < \infty$ (for all $t \in \mathbb{Z}$).

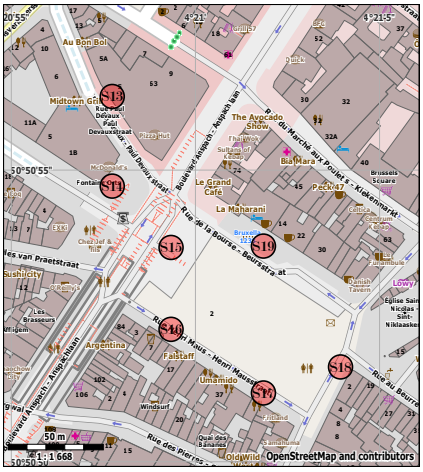

Fig. 4. Sensor arrangement on area of Bourse. Generated using Maperitive with mapping data from OpenStreetMap.

B. Mean function, covariance function and stationarity

The mean function of $\{X_t\}$ is $\mu_X(t) := \mathbb{E}[X_t]$ and the covariance function is $\gamma_X(r,s) := \mathbb{E}[(X_r - \mu_X(r))(X_s - \mu_X(s))]$ [29, Definition 1.4.1]. If, for all $t \in Z$, $\mu_X(t) = \mu_X$ is constant and $\gamma_X(t+h,t) = \gamma_X(h)$ (for all $h > 0$), then $\{X_t\}$ is weakly stationary [29, Definition 1.4.2]. In this case, μ_X is the mean of $\{X_t\}$ and $\gamma_X(h)$ is its autocovariance function (ACVF). The time series with which this paper deals are not weakly stationary. For example, in Figure 5, it is clear that the mean and the variance of counts are lower at 8:00 AM than they are at 10:00 PM. One of our goals will be to transform them into stationary ones. As explained in what follows, these transformations involve differentiation and (optionally) power transformations (such as Box-Cox transformations).

C. Autoregressive moving average models

Autoregressive moving average (ARMA) models model stationary time series. $\{X_t\}$ is an ARMA(p, q) process if it is stationary and if for every t, [29, Adapted from Definition 3.1.1]

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{j=1}^{q} \theta_j w_{t-j} + w_t, \tag{1}$$

where $w_t \sim \mathcal{N}(0, \sigma^2)$ are i.i.d. (independent and identically distributed), and the AR and MA polynomials $\phi(z) := 1 - \sum_{i=1}^{p} \phi_i z^i$ and $\theta(z) := 1 + \sum_{j=1}^{q} \theta_j z^j$ have no common root. Two interesting special cases of ARMA models are pure autoregressive (AR) models and pure moving average (MA) models; the former are such that $q = 0$ whereas the latter are such that $p = 0$. The quantity $w_t \sim \mathcal{N}(0, \sigma^2)$ is called the *innovation* at time t. It is an unpredictable quantity that is responsible for the random nature of the time series.

D. Differentiation

When the AR polynomial $\phi(z)$ of a model like (1) contains a unit root (i.e., $\phi(1) = 0$), it is integrated, which means it is not an ARMA model anymore because it is not stationary.

One of the most common approach to transform a non-stationary time series with a unit root into a stationary one is to use differentiating. We shall differentiate the original time series several times until the differentiated time series looks like an ARMA process without unit roots. We can then fit an ARMA model on the differentiated time series and carry out forecasting; these forecasts can be reverted back to their non-differentiated counterparts by integration.

E. Tests for unit roots

We shall first briefly discuss two statistical tests evaluating the number of differentiation operations required. In practice, we keep differentiating the time series of interest until unit root tests suggest there is no unit root left. The first test is the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test [30]. Its null hypothesis, H_0, is that the tested time series is stationary or trend stationary depending on the version of the test. The basic equations for the test are [30, Eqs. (2) and (3)] and the associated statistic is computed for a given number of lags [30, Eq. (6)]. The null hypothesis of KPSS tests is stationarity. We would prefer the null hypothesis to be that of non-stationarity. Nevertheless, state-of-the-art statistical packages use KPSS tests to compute the differentiation order (see [31, Sec. 3.1]).

Another test is the augmented Dickey-Fuller (ADF) test, which is an extension of the Dickey-Fuller (DF) test [32]. It tests the null hypothesis that a time series possesses a unit root and assumes that the time series of interest can be well represented by an AR(p) process. ADF tests are interesting because they can reject the null hypothesis of non-stationarity. However, they are valid only if the order of the AR process encompasses most of the serial correlation of the time series. An excessive value of p prevents ADF tests from rejecting H_0 and, conversely, a value that is too low makes them biased.

F. Autoregressive integrated moving average models

Autoregressive integrated moving average models (ARIMA) models consist in ARMA models applied on differentiated time series to generate forecasts for the non-differentiated time series. Their order is denoted by the 3-tuple (p, d, q), where d is the order of differentiation and (p, q) is the order of the underlying ARMA model for the differentiated time series. Tests in Section IV-E assist in determining d.

Formally, $\{X_t\}$ is an ARIMA(p, d, q) process if the time series differentiated d times is a causal ARMA(p, q) process [29, Definition 6.1.1]. Using the backshift operator B defined by $B^k X_t := X_{t-k}$, we can write the recurrence equation of an ARIMA model as follows:

$$(1 - B)^d \phi(B) X_t = \theta(B) X_t, \tag{2}$$

where $\phi(B) := 1 - \sum_{i=1}^{p} \phi_i B^i$ and $\theta(B) := 1 + \sum_{j=1}^{q} \theta_j B^j$ are the AR and MA polynomials of the underlying ARMA model expressed in terms of the backshift operator B.

G. Fitting ARIMA models of fixed orders

Let us now discuss about how to practically fit ARIMA models of a given order (p, d, q). Fitting ARIMA models is

theoretically equivalent to fitting an ARMA model on the differentiated time series. Modern fitting techniques rely on a state-space representation of the full ARIMA model, however; we refer the reader to [29, Chapter 8] for an introduction to the subject and to [33] for a thorough discussion of it.

This paper relies on the statistical environment R [34]. For fitting a model of a specific order (p, d, q), we use the `Arima` function from the R package `forecast` [31], [35].

H. Computing appropriate model orders

Fitting an ARIMA model consists in computing its model order (p, d, q), its coefficients (whose number depends on p and q), and the innovation variance σ^2. We have just discussed how to fit coefficients and σ^2 (and previously, how to compute d). The story is more complicated for finding an appropriate ARMA model order (p, q).

A traditional method for finding (p, q) relies on a visual inspection of the sample autocorrelation function and sample partial autocorrelation function (see, e.g., [21]), which tail off after a certain number of lags that corresponds to p and q. Because this method entails a visual inspection from a statistician, it cannot be automated and we shall not use it.

Another set of methods consists in fitting ARMA models for different orders and then retaining the one that optimizes a criterion, which typically include a penalty for the forecast accuracy on the training set and another penalty that increases with the number of parameters the model encompasses, which prevents overfitting.

We assume that n observations are available for training and define vectors $\boldsymbol{\phi} \in \mathbb{R}^p$, $\boldsymbol{\theta} \in \mathbb{R}^q$, which contain the AR and MA coefficients, respectively. The main metrics used for judging the merits of all tested models are the Akaike information criterion with bias correction (AICC) and Bayes information criterion (BIC). These criteria (or metrics) include the log-likelihood function $\ell(\boldsymbol{\phi}, \boldsymbol{\theta}, \sigma)$ (which quantifies to what extent a given ARMA model fits the observed measurements) and a penalty that increases with the number of parameters of the ARMA model. For each candidate order (p, q), the procedure first fits an ARMA(p, q) model and then computes the resulting value of the AICC or BIC. When the AICC or BIC values have been computed for all the tested model orders, the model whose value is the lowest is picked.

We do not delve into the theoretical basis of the expressions of the AICC and BIC. For a number of parameters equal to $n_p = p + q + 1$, the expressions are [36]–[38]

$$\mathrm{AICC}(\boldsymbol{\phi}, \boldsymbol{\theta}, \sigma) = -2\ell(\boldsymbol{\phi}, \boldsymbol{\theta}, \sigma) + 2n_p + 2\frac{n_p^2 + n_p}{n - n_p - 1},$$

$$\mathrm{BIC}(\boldsymbol{\phi}, \boldsymbol{\theta}, \sigma) = -2\ell(\boldsymbol{\phi}, \boldsymbol{\theta}, \sigma) + \log(n) n_p.$$

The procedure above requires us to identify, using maximum likelihood (ML) estimation, ARMA parameters ($\boldsymbol{\phi}^{\mathrm{ML}}$, $\boldsymbol{\theta}^{\mathrm{ML}}$, σ^{ML}) for each candidate order (p, q). The processing power of modern computers makes it a tractable approach—especially for univariate, non-seasonal ARMA models.

I. Forecasting with identified ARIMA models

We now discuss forecasting based on a set of observations and a fully identified ARMA model. Let us assume that we observe realizations $\{x_t\}$ of a discrete stochastic process $\{X_t\}$. Given the previous and current observations at time t, $\Omega_t = \{x_s\}_{0 \leq s \leq t}$, we would like to find the h-step estimator of X_{t+h} that minimizes the mean square error (MSE) of forecasts. For a given, arbitrary h-step estimator $\bar{X}_t(h)$ (using observations until time t), the forecast MSE is $\mathrm{MSE}[\bar{X}_t(h)] := \mathbb{E}[\big(X_{t+h} - \bar{X}_t(h)\big)^2]$. As shown in [28, Sec. 2.2.2] and [39, Sec. 3.5], the minimum MSE predictor is the conditional expectation at time t

$$\mathbb{E}_t[X_{t+h}] := \mathbb{E}[X_{t+h} | \Omega_t]. \tag{3}$$

We often say that we forecast the conditional mean when forecasting X_{t+h} given observations of $\{X_t\}$ until time t.

Standard ARIMA forecasting approaches typically implement a forecaster with theoretically optimal MSE (they assume that the fitted ARIMA model is exactly the stochastic process having generated the observations). We shall not thoroughly discuss forecasting methods for ARIMA models. We only point out that—with a known, identified ARIMA model—forecasting is a reliable task and it can be carried out recursively using, e.g., the *innovations algorithm* [29, Sec. 2.5.2] on the underlying ARMA model, with low computational cost. To the best of our knowledge, the canonical functions in R used for forecasting rely on state-space representations, however.

J. Box-Cox transformations for variance stabilization

Let us now discuss variance stabilization, that is the transformation of time series with time-varying variances into ones with constant variances over time. Box-Cox transformations are a class of transformations [40] that stabilize the variance of a time series prior to fitting a model on it [41]. Once fit, the model (e.g., an ARIMA model) operates on the transformed data; forecasts are then reverted back into their non-transformed counterparts by applying the inverse transform and possibly a debiasing coefficient [41, Eq. (10)], which we do not use here because it worsens forecasting accuracy.

Box-Cox transformations depend on a parameter λ_{BC} and transform the original time series into

$$X_t^{(\mathrm{BC})} = \begin{cases} (X_t^{\lambda_{\mathrm{BC}}} - 1)/\lambda_{\mathrm{BC}} & \text{if } \lambda_{\mathrm{BC}} \neq 0 \\ \log(X_t) & \text{if } \lambda_{\mathrm{BC}} = 0 \end{cases}. \tag{4}$$

We can add a constant to X_t prior to computing a Box-Cox transformation if there are zero or negative values of X_t.

Guerrero [41] proposes a computationally simple method to estimate λ_{BC}. Essentially, the method extracts contiguous sub-series of R samples from the available measurements. For each sub-series (indexed by z), the empirical mean $\bar{\mu}_z$ and standard deviation $\bar{\sigma}_z$ are computed. Then, the coefficient of variation (CV) of the set $\{\bar{\sigma}_z / \bar{\mu}_z^{1-\lambda_{\mathrm{BC}}}\}_z$ is computed for a prescribed grid of values of λ_{BC}; the value of λ_{BC} yielding the lowest CV is chosen.

When fitting an ARIMA model, the time series should theoretically exhibit a constant variance, which Box-Cox

transformations may enforce. Consequently, the fit may be improved—which means that forecasts of the conditional mean could be more accurate than those of a model fit on the original time series. Nevertheless, such improvements do not consistently appear in practice [42], [43].

K. Volatility modeling

The previous subsections of Section IV focus on forecasting the conditional mean of a time series, that is the expectation of X_{t+h} conditional on previous observations until time t, $\{x_s\}_{0\leq s\leq t}$. Similarly, this section discusses how to forecast the conditional variance σ^2_{t+h} on the basis of observations until time t, $\{x_s\}_{0\leq s\leq t}$; the variance is now a function of time. This is an important step because it enables us to derive prediction intervals. In the financial literature, the conditional variance often is a measure of the volatility of, e.g., stock prices, hence the name *volatility modeling*.

We describe three strategies for deriving prediction intervals in what follows.

1) Vanilla ARIMA: The first approach is the most basic one and consists in assuming a constant variance for the Gaussian innovations. This approach is that implemented by default in forecasting packages. Once an ARIMA model is fit, it is easy to derive the variance of h-step forecast. With $\sigma_t(h)^2$ being the variance of the h-step forecast made with observations until time t, we have, for $t\to\infty$, [29, Eq. (6.4.6)] $\sigma_t(h)^2 = \sigma(h)^2 := \sum_{j=0}^{h-1}\psi_j^2\sigma^2$, where the $\{\psi_j^2\}$ depend on d, $\phi(z)$ and $\theta(z)$. The quantiles of a zero-mean Gaussian distribution with variance $\sigma(h)^2$ provide the PI of h-step forecasts.

2) ARIMA with Box-Cox transformations: The time series in pre-transformed using Box-Cox transformations. PIs are then derived in the transformed scale, as in Section IV-K1. When reverting back to the original scale, the PI boundaries are also affected by the reverse transformation, which turns constant PIs into varying ones over time.

3) GARCH models: GARCH models, originally derived in [3] on the basis of autoregressive conditional heteroskedasticity (ARCH) models [44], model the conditional variance of a stochastic process. With Ω_t denoting the set of all information until time t, a GARCH(p,q) model is [3, Eq. (1)–(2)]

$$w_t|\Omega_t \sim \sigma_t e_t, \tag{5}$$

$$\sigma_t^2 = \omega + \sum_{j=0}^{q}\alpha_j w_{t-j}^2 + \sum_{i=1}^{p}\beta_i\sigma_{t-i}^2, \tag{6}$$

where all coefficients are equal to or higher than 0 (except for ω, which should be strictly positive) and the $\{e_t\}$ are i.i.d. random variables. The most canonical choice for e_t is a zero-mean normal distribution with variance 1 [3]; Student's t-distributions are another option. In this model, the innovation w_t is directly observed. Such models convey the idea that a high volatility tends to persist over time. In theory, GARCH models can be reliably estimated using quasi maximum likelihood estimators (QMLEs) [4, Theorem 2.1].

GARCH models typically describe time series for which forecasting the conditional mean is impossible (the forecast is always 0) but whose volatility can be. Such time series notably include the residuals of perfectly fitted ARIMA models, which exhibit no serial correlation. We use GARCH models for tracking variances and we are not interested by some of their features that made them popular in econometrics, which include volatility clustering and the leptokurticity of the unconditional variance. Interestingly, GARCH models can generate PIs for other classes of forecasters that do not provide PIs by default—such as neural networks.

V. Metrics and forecasting methods

Now that we introduced all the main theoretical concepts, we shall focus on practical forecasting. First of all, we describe the two main metrics we use to evaluate forecasting accuracy (for the conditional mean); we also present the metric to be used for quantifying the accuracy of PIs. Then, Section V-C presents all the practical forecasting methods that we test.

Clearly, once two or three days of data become available, a seasonal ARIMA model would provide better forecasts than an ARIMA model (given the stability of the daily count pattern). In this paper, however, we decide to investigate the accuracy of univariate forecasting for several days, whereby each day is fit separately. Our approach allows us to determine to what extent forecast accuracy is consistent from one day to another. As our results show, assessing forecasting accuracy on several days independently is necessary, in that not doing so may suggest a forecasting method is reliable even though it fails when applied on another day of the event (in Table II, compare, e.g., *12-01 (Sat)* against *12-15 (Sat)*).

A. Metrics for assessing forecasts

In our results, we always restrict the computation of such metrics to time frames for which forecast accuracy is critical—i.e., the ascending slope of counts before peak time. For *Sainte-Catherine*, the ascending slope time frame ranges from 16:00 to 20:00 from Monday to Thursday, from 14:00 to 21:00 on Friday, and from 14:00 to 20:00 on Saturday. For *Bourse*, it ranges from 10:30 to 16:30.

1) Assessing forecasts of the conditional mean: This paper relies on the root mean square error (RMSE) and mean absolute percentage error (MAPE) to assess the performance of forecasting for the conditional mean; with $\{x_t\}_{0\leq t\leq n-1}$ and $\{\bar{x}_t\}_{0\leq t\leq n-1}$ denoting the set of observed values and the set of forecasts respectively, we have

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=0}^{n-1}(x_t-\bar{x}_t)^2} \tag{7}$$

and

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{t=0}^{n-1}\frac{|x_t-\bar{x}_t|}{|x_t|}. \tag{8}$$

The RMSE does not normalize its value according to the levels (i.e., the values of $|x_k|$); therefore, it may be meaningless to compare RMSE values of time series whose levels are significantly different. It also means that if a time series

has values of different magnitudes, the errors associated with high values generally weight more than those associated with low ones (assuming that the error variance increases with the time series level). Typically, the RMSE is also sensitive to outliers. Conversely, the MAPE is less sensitive to outliers and normalizes each error term by the corresponding observed value. To conclude, the RMSE and MAPE have contrasting ways to account for errors, which is the reason why we chose to use both of them to quantify forecast accuracy.

2) Assessing the accuracy of prediction intervals: Given an $\alpha\%$ PI (whose lower and upper boundaries over time are l_t and u_t, respectively), we want to determine how accurate it is. With n true observations $\{x_t\}_{0\leq t\leq n-1}$, we can compute the empirical proportion of samples within the PI

$$\mathrm{PI}_{\alpha}^{\mathrm{emp}} := \frac{100\%}{n}\mathrm{card}\left(\{x_t \in [l_t, u_t] : 0 \leq t \leq n-1\}\right) \quad (9)$$

and then compare this number against $\alpha\%$ (if the $\alpha\%$ PI is perfect, $\mathrm{PI}_{\alpha}^{\mathrm{emp}}$ should be equal to $\alpha\%$ for $n \to \infty$). Our results report the raw value of $\mathrm{PI}_{\alpha}^{\mathrm{emp}}$ for $\alpha = 90\%$. We choose $\alpha = 90\%$ because our event-organizing partner suggested us that our forecasts should roughly be accurate 90% of the time; thus, properly evaluating the 90% PI is what is most natural.

B. A baseline point of comparison: random walk models

We shall compare the metrics of our forecasts against those obtained using the "naïve" approach, which is often referred to as the "random walk" (RW) or "persistence model" in the literature [45]–[48]. It is an ARIMA$(0, 1, 0)$ model, whose forecasts are equal to the latest observed value, no matter the forecasting horizon. Formally, assuming that the underlying stochastic process generating observations exactly is a RW, $\mathbb{E}_t[X_{t+h}] = X_t$. In practice, the underlying process rarely is a RW process, and using a RW model for forecasting is adequate only if the time series remains approximately constant over the considered forecasting horizon. To be of practical interest, our ARIMA models must outperform RW models.

C. Real-time forecasting methods

We now discuss the main forecasting methods, which can all be implemented in an automated, real-time forecasting system. They first gather 2 hours of data (from 6:00 AM to 8:00 AM) before fitting a first ARIMA model. Every time a new count is available (every 5 minutes), the model is reevaluated. We only use the counts of the current day to build the corresponding ARIMA model; different days are fit independently. When evaluating forecasting accuracy, our method prevents overfitting effects because, when forecasting the count value at time $t+h$, our ARIMA model has not (yet) been trained on the count at time $t + h$.

We now detail the methods for forecasting the conditional mean. As shown in what follows, the best differentiating order is $d = 2$ but our results also cover the case $d = 1$.

1) Rolling ARIMA models with reestimation of the model order (p, q): This procedures reestimates the order of the underlying ARMA model, the AR and MA coefficients, and the innovation variance. The procedure for fitting the ARIMA model relies on R function `auto.arima`, which, for a given ARMA model order (p, q), uses R function `arima` (see Section IV-G for more details). We shall use a brute force approach to search for an optimal ARMA model order (p, q). To limit the computational burden of this procedure, we rely on the step-wise approach described in [31, Sec. 3.2] instead of a true brute force procedure. The step-wise approach initially considers starting model orders and then locally navigates through the model order space.

2) Rolling ARIMA models without reestimation of the model order (p, q): This method is identical to that of Section V-C1 except that the underlying ARMA model order is fixed to $(p, q) = (2, 1)$, a choice that yields good forecasts in practice. A first ARI$(3, d)$ is fitted on the basis of the data from 06:00 AM to 08:00 AM, it is used if fitting the first ARIMA model fails. A pure ML estimator fits this integrated AR model (reliable closed-form formulas exist [28, Sec. 3.4.2]). Whenever we include a new count for fitting, it may fail; in this case, we use the last successfully fitted model. In practice, fitting issues occured for $d = 1$ only. Our results demonstrate that Method V-C2 with $d = 2$ outperforms Method V-C1.

3) Rolling ARIMA models without reestimation of the model order (p, q) and with Box-Cox transform: This method is identical to Method V-C2 except that it uses a Box-Cox transformation. To find the value of λ_{BC} for the Box-Cox transformation, we can either make it fixed or dynamic. The dynamic case reestimates λ_{BC} using Guerrero's method with $R = 4$ samples and with tested values of λ_{BC} being $\{0.0, 0.25, 0.50, 0.75, 1.0\}$. We also do not use debiasing coefficients because they worsen forecasting accuracy. Our results demonstrate that Method V-C2 outperforms Method V-C3.

We now turn to the methods for generating PIs.

4) PIs from rolling ARIMA models with reestimation of the model order (p, q): We use the PIs generated by ARIMA processes with Gaussian innovations, see Section IV-K1.

5) PIs from GARCH models fitted on ARIMA residuals: For each forecasting horizon h, we fit a dedicated rolling GARCH(1,1) model on the residuals of h-step ahead forecasts, which Method V-C2 (with $d = 2$) generates. The GARCH(1,1) model is reestimated whenever a new count becomes available; if fitting fails, the most recent PI is used. For forecasting the variance of the h-step ahead residuals, we use h-step ahead GARCH forecasts. In this paper, we use the `fGarch` package [49] from R to fit GARCH models and forecast the conditional variance. We consider GARCH(1,1) models with conditional variances that are distributed as normals and Student's t-distribution; we denote these two cases by GARCH$(1,1)_{\mathrm{norm}}$ and GARCH$(1,1)_{\mathrm{t-dist}}$, respectively. Our results show that Method V-C5 yields the most accurate PIs but sometimes generates outliers.

VI. Forecasting results and discussions

A. Estimating the differentiation order d

This section relies on KPSS and ADF tests to estimate the order of differentiation d. We run tests on each day independently. With n denoting the number of observations, formula $\mathrm{lags} = 4(n/100)^{0.25}$ [30, Sec. 5] provides a number of lags

TABLE I
RESULTS OF KPSS AND ADF TESTS FOR SAINTE-CATHERINE (SENSORS S1 TO S7 IN FIGURE 3) AND BOURSE (SENSORS S14 TO S19 IN FIGURE 4). THE NUMBER OF OBSERVATIONS PER DAY IS 216 SAMPLES FOR $d = 0$ (COUNTS EVERY 5 MINUTES FROM 06:00 TO MIDNIGHT). THE NUMBER OF LAGS FOR KPSS AND ADF TESTS IS EQUAL TO 5.

Date (Day)	KPSS p-value			ADF p-value		
St.Cath ↓	$d=0$	$d=1$	$d=2$	$d=0$	$d=1$	$d=2$
12-10 (Mon)	< 0.01	0.02	> 0.1	0.55	< 0.01	< 0.01
12-13 (Thu)	< 0.01	0.02	> 0.1	0.52	0.01	< 0.01
12-17 (Mon)	< 0.01	< 0.01	> 0.1	0.56	0.03	< 0.01
12-18 (Tue)	< 0.01	0.02	> 0.1	0.59	< 0.01	< 0.01
12-01 (Sat)	< 0.01	< 0.01	> 0.1	0.65	< 0.01	< 0.01
12-14 (Fri)	< 0.01	< 0.01	> 0.1	0.58	< 0.01	< 0.01
12-15 (Sat)	< 0.01	< 0.01	> 0.1	0.74	< 0.01	< 0.01
12-28 (Fri)	< 0.01	< 0.01	> 0.1	0.70	0.02	< 0.01
Bourse ↓	$d=0$	$d=1$	$d=2$	$d=0$	$d=1$	$d=2$
12-25 (Tue)	< 0.01	< 0.01	> 0.1	0.87	< 0.01	< 0.01
12-26 (Wed)	< 0.01	< 0.01	> 0.1	0.81	< 0.01	< 0.01
12-27 (Thu)	< 0.01	< 0.01	> 0.1	0.67	< 0.01	< 0.01
12-28 (Fri)	< 0.01	< 0.01	> 0.1	0.63	< 0.01	< 0.01
12-29 (Sat)	< 0.01	< 0.01	> 0.1	0.71	< 0.01	< 0.01
12-30 (Sun)	< 0.01	< 0.01	> 0.1	0.65	< 0.01	< 0.01
01-04 (Fri)	< 0.01	< 0.01	> 0.1	0.73	< 0.01	< 0.01
01-05 (Sat)	< 0.01	< 0.01	> 0.1	0.53	< 0.01	< 0.01

for KPSS tests. All tests assume no linear trend is present. Typically, a p-value above 0.1 makes tests inconclusive and a p-value below 0.01 makes them conclusive; anything in between those two critical values is to be discussed.

Table I reports the results. Both tests suggest that $d = 2$, because KPSS tests then fail to reject non-stationarity and ADF tests reject non-stationarity; nevertheless, $d = 1$ is also a good candidate according to ADF tests (especially for Bourse). Of course, the number of lags for tests may make them biased. We shall test both differentiating orders and keep that providing the best forecasts of the conditional mean.

A real-time system deployed in future events would typically enforce d before any measurement is available; for future events, we recommend to keep the value $d = 2$ obtained for *Winter Wonders*. We could also use KPSS or ADF tests online as soon as n becomes sufficiently high.

B. Accuracy of canonical ARIMA for the conditional mean

This section demonstrates which forecasting method (Methods V-C1 to V-C3) delivers the lowest RMSEs and MAPEs consistently; consistency is of paramount importance: a model that occasionally severely misforecast is to be rejected.

1) Rolling canonical ARIMA models with full reestimation (Method V-C1): Table II reports detailed results for *Sainte Catherine*. We remind the reader that all days are fit independently. Sometimes, ARIMA$(p, 1, q)$ models perform similarly to RW models; in these cases, most of the estimated ARIMA$(p, 1, q)$ models are actually RW models, i.e., ARIMA(0,1,0) models. This is true especially when using the BIC, which usually generates sparser models than a fitting based on the AICC; this explains why metrics for $d = 1$ and for the RW model occasionally are strikingly similar. Rolling ARIMA$(p, 2, q)$ models perform generally better than RW models but not consistently. Fixing the model order will help because it constrains how forecasting is carried out.

TABLE II
RESULTS OF ROLLING FORECASTS FOR SAINTE-CATHERINE (SENSORS S1 TO S7 IN FIGURE 3). THE FULL ARIMA MODEL (INCLUDING ITS UNDERLYING ARMA ORDER (p, q)) IS REESTIMATED WHENEVER A NEW COUNT BECOMES AVAILABLE, SEE METHOD V-C1. METRICS ARE DERIVED FOR A FORECASTING HORIZON OF 30 MINUTES. METRICS ARE EVALUATED FOR TIME FRAMES CORRESPONDING TO THE ASCENDING SLOPE (SEE SECTION V-A FOR DETAILS). BOLD NUMBERS REPRESENT THE BEST PERFORMANCE AND UNDERLINED NUMBERS INDICATE THEY ARE HIGHER THAN THOSE OF THE RANDOM WALK (RW) MODEL.

Date (Day)	RMSE			MAPE (in %)		
AICC ↓	$d=1$	$d=2$	RW	$d=1$	$d=2$	RW
12-10 (Mon)	**125.8**	128.5	144.6	**5.88**	6.22	7.01
12-13 (Thu)	276.1	**271.8**	276.1	9.19	9.68	9.19
12-17 (Mon)	**187.3**	208.6	196.3	**6.58**	7.23	6.97
12-18 (Tue)	266.7	**254.7**	260.7	9.57	9.32	**8.91**
12-01 (Sat)	183.4	**151.2**	187.7	7.14	**6.24**	7.34
12-14 (Fri)	206.5	**199.1**	211.3	6.96	**6.89**	7.27
12-15 (Sat)	**247.5**	266.4	**257.4**	**6.72**	7.44	6.94
12-28 (Fri)	**172.3**	178.2	184.4	**5.31**	5.62	6.01
BIC ↓	$d=1$	$d=2$	RW	$d=1$	$d=2$	RW
12-10 (Mon)	146.8	**129.6**	144.6	7.07	**6.52**	7.01
12-13 (Thu)	276.1	**267.5**	276.1	9.19	**8.98**	9.19
12-17 (Mon)	186.9	215.5	196.3	6.57	7.11	6.97
12-18 (Tue)	264.1	**238.6**	260.7	9.20	**8.78**	8.91
12-01 (Sat)	188.2	**151.5**	187.7	7.37	**6.21**	7.34
12-14 (Fri)	213.8	**182.2**	211.3	7.31	**6.55**	7.27
12-15 (Sat)	**246.8**	262.9	257.4	**6.70**	7.24	6.94
12-28 (Fri)	190.9	**178.1**	184.4	6.10	**5.65**	6.01

2) Rolling, fixed-order canonical ARIMA models (Method V-C2): We fix the order (p, d, q) of the ARIMA model beforehand. In practice, an adequate model order prevents the model from relying on highly local tendencies and prevents overfitting effects (i.e., it includes a sufficiently low number of coefficients in comparison to the number of available measurements). Our experiments revealed that ARIMA$(2, 1, 1)$ and ARIMA$(2, 2, 1)$ models are good options.

Table III shows that Method V-C2 consistently outperforms (or at least compares similarly to) RW models. In average, the accuracy is worse at *Bourse* than it is at *Sainte-Catherine*; this happens because some areas are intrinsically less predictable than others. Comparing Table III against Table II reveals that fixing the model order improves forecasting accuracy in comparison to periodically reestimating the order. Figures 5 and 6 plot the rolling ARIMA(2,2,1) forecasts for *Sainte-Catherine* and *Bourse*, respectively.

3) Rolling, fixed-order ARIMA models with Box-Cox transformations (Method V-C3): Table IV reports the results for *Bourse*. In Table IV, *Roll.* corresponds to the rolling reestimation of λ_{BC} for every newly available count; the table also includes results for fixed values of λ_{BC}. Box-Cox transformations have a detrimental influence on the accuracy of rolling ARIMA(2,2,1) models. In particular, the truly self-adjusting procedure (referred to as *Roll.* in Table IV) is always less accurate than Method V-C2 (which is equivalent to Method V-C3 with $\lambda_{\mathrm{BC}} = 1$). The forecast accuracy improves as λ_{BC} approaches 1, which shows that using Box-Cox transformations to improve forecasting accuracy is pointless.

The conclusion is that, among the three methods, Method V-C2 is the best option for forecasting the conditional mean. Our average MAPE values range from 4.75% to 11%, depending on the day and the area. We report these metrics

TABLE III
RESULTS OF ROLLING FORECASTS FOR SAINTE-CATHERINE (SENSORS S1 TO S7 IN FIGURE 3) AND BOURSE (SENSORS S14 TO S19 IN FIGURE 4). THE ARIMA MODEL (EXCLUDING ITS UNDERLYING ARMA ORDER (p, q)) IS REESTIMATED WHENEVER A NEW COUNT BECOMES AVAILABLE, SEE METHOD V-C2. METRICS ARE DERIVED FOR A FORECASTING HORIZON OF 30 MINUTES. METRICS ARE EVALUATED FOR TIME FRAMES CORRESPONDING TO THE ASCENDING SLOPE (SEE SECTION V-A FOR DETAILS). BOLD NUMBERS REPRESENT THE BEST PERFORMANCE AND UNDERLINED NUMBERS INDICATE THEY ARE HIGHER THAN THOSE OF THE RANDOM WALK (RW) MODEL. FOR THE RMSE, "AVERAGE" IS THE ROOT SQUARE OF THE MSE AVERAGE.

Date (Day)	RMSE			MAPE (in %)		
St. Cath ↓	$d=1$	$d=2$	RW	$d=1$	$d=2$	RW
12-10 (Mon)	125.0	**121.1**	144.6	5.97	**5.78**	7.01
12-13 (Thu)	278.7	**261.2**	276.1	9.33	9.39	**9.19**
12-17 (Mon)	190.5	**142.0**	196.3	6.57	**4.75**	6.97
12-18 (Tue)	237.6	**239.4**	260.7	**8.44**	8.78	8.91
12-01 (Sat)	173.6	**156.5**	187.7	6.87	**6.48**	7.34
12-14 (Fri)	204.8	**177.9**	211.3	7.01	**6.38**	7.27
12-15 (Sat)	**254.0**	261.0	257.4	7.03	7.17	6.94
12-28 (Fri)	**166.7**	177.7	184.4	**5.27**	5.61	6.01
Average	209.3	**198.8**	219.0	7.06	**6.79**	7.46
+ % wrt. best	5.3	0.0	10.2	4.0	0.0	9.9
Bourse ↓	$d=1$	$d=2$	RW	$d=1$	$d=2$	RW
12-25 (Tue)	183.1	**145.6**	186.1	9.04	**6.82**	9.33
12-26 (Wed)	206.8	**193.9**	227.1	10.36	**9.21**	11.57
12-27 (Thu)	244.3	**182.4**	245.4	12.69	**8.37**	12.69
12-28 (Fri)	245.2	**210.1**	240.1	11.53	**9.55**	11.24
12-29 (Sat)	218.0	**184.8**	238.2	10.10	**7.92**	11.01
12-30 (Sun)	219.9	**188.6**	223.4	11.11	**8.78**	11.45
01-04 (Fri)	137.9	**133.1**	142.0	9.15	**8.30**	9.23
01-05 (Sat)	159.0	**156.7**	163.4	**10.93**	10.94	11.12
Average	205.0	**176.1**	211.4	10.61	**8.74**	10.96
+ % wrt. best	16.4	0.0	20.1	21.4	0.0	25.4

for a forecast horizon of 30 minutes. According to our partner organizing events, however, forecast horizons ranging from 10 to 15 minutes are already of interest and the corresponding metrics are lower than those of 6-step ahead forecasts.

C. The accuracy of prediction intervals

We consider 90% PIs generated by three different methods: i) rolling ARIMA(2,2,1) models with Gaussian innovations (see Method V-C4), ii) rolling GARCH(1,1) models with a Gaussian conditional variance distribution, iii) rolling GARCH(1,1) models with a conditional variance that has a Student's t-distribution. The last two methods are particular cases of Method V-C5. We always use rolling ARIMA(2,2,1) models for forecasting the conditional mean (see Method V-C2).

Table V reports the final results. Method V-C4 makes computing PIs computationally easy and it prevents PI boundary outliers from occurring (see Figures 5 and 6); according to Table V, however, the corresponding 90% PI ranges are underestimated. Conversely, GARCH(1,1) PIs are in average more accurate but, as shown in Figures 5 and 6, the PI boundaries generated by GARCH(1,1) models are sensitive to high forecast errors (see, e.g., December 25 and 26 on Figure 6). GARCH models are also more computationally intensive to fit and forecast with than rolling ARIMA models.

As a result, if computational resources are no issue, we recommend to use GARCH(1,1) models and to filter the outliers. Otherwise, the PIs stemming from a rolling ARIMA(2,2,1) model may be sufficient for reporting purposes.

VII. FUTURE WORK

Univariate, non-seasonal forecasting methods tend to over or undershoot whenever a turning point appears (a point at which the first derivative suddenly changes), see Figure 5 at 12:30 and around 20:00). To deal with such points, other avenues of information could be leveraged. For events featuring a strong seasonal pattern, seasonal models are appropriate. Jointly forecasting count time series at different locations could also foresee turning points based on the data from other nearby areas through which attendees go. ARIMA models with exogenous variables (the exogenous variables being counts at other locations) could be a solution, just as multivariate AR(I)MA models (e.g., vector error correction models). Multi-fidelity approaches could also utilize counts from telco operators associated with nearby cells to detect incoming people prior to their arrival on the event. More generally, the accuracy of other classes of forecasters could be investigated (see [23, Table 1] for an overview of such methods).

Finally, the forecasting methods we proposed—although already tested on several days—could be validated using data from other venues (possibly indoor venues as well). The value of the extrapolation factor could also be more precisely estimated by deploying Wi-Fi sensors on events endowed with another precise counting system (e.g., one based on cameras or turnstiles at controlled entrances and exits).

VIII. CONCLUSION

This paper cursorily discusses a passive crowd counting system based on the Wi-Fi probe requests that is unaffected by MAC address randomization. We proposed a computationally tractable forecasting method for public events; it consists in a rolling ARIMA(2,2,1) model that is reestimated every five minutes (see Method V-C2). Using real-world data obtained in late 2018 and early 2019, we validated our method, which starts forecasting after only two hours of data are available—thereby making it a practical algorithm for forecasting counts on one-day events. For a forecasting horizon of 30 minutes, our method outperforms its main variations (using, e.g., Box-Cox transformations) and random walk models. Depending on the area, the average mean absolute percentage error ranges from 6.79% to 8.74%. Finally, we proposed two methods for generating viable 90% prediction intervals. The one relying on ARIMA models generates 90% PIs within which 74.5% to 79.1% of true counts fall, whereas, for PIs stemming from GARCH models with Student's t-distributions, the figure ranges from 89.2% to 89.5% depending on the area.

ACKNOWLEDGMENTS

We thank Innoviris for funding this research through the MUFINS project. We also thank Brussels Major Events for their active collaboration. We thank the anonymous reviewers for their careful review and constructive comments.

TABLE IV
RESULTS OF ROLLING ARIMA(2,2,1) FORECASTS FOR BOURSE WITH BOX-COX TRANSFORMATIONS, SEE METHOD V-C3. BOX-COX PARAMETER λ_{BC} IS EITHER ESTIMATED ONLINE (ROLL.) OR FIXED BEFOREHAND. NO DEBIASING COEFFICIENT IS APPLIED. SEE THE CAPTION OF TABLE III FOR ADDITIONAL DETAILS.

Date (Day)	RMSE					MAPE (in %)				
	Roll.	Fixed λ_{BC}				Roll.	Fixed λ_{BC}			
		0	0.5	0.75	1		0	0.5	0.75	1
12-25 (Tue)	153.6	153.6	146.2	**145.3**	145.6	7.10	7.10	6.84	**6.81**	6.82
12-26 (Wed)	217.4	218.9	198.4	195.2	**193.9**	10.38	10.48	9.57	9.35	**9.21**
12-27 (Thu)	247.6	286.6	197.8	187.3	**182.4**	10.95	12.25	9.04	8.56	**8.37**
12-28 (Fri)	253.7	258.2	221.1	214.2	**210.1**	11.31	11.50	10.16	9.81	**9.55**
12-29 (Sat)	212.3	214.1	193.2	188.1	**184.8**	8.72	8.80	8.18	8.04	**7.92**
12-30 (Sun)	216.1	220.4	197.9	192.2	**188.6**	10.09	10.18	9.18	8.93	**8.78**
01-04 (Fri)	147.4	148.0	138.1	135.2	**133.1**	8.97	9.01	8.54	8.39	**8.30**
01-05 (Sat)	185.2	190.2	165.7	159.6	**156.7**	12.35	12.80	11.50	11.12	**10.94**
Average	207.5	215.9	184.3	179.0	**176.1**	9.98	10.27	9.13	8.88	**8.73**
+ % wrt. best	17.8	22.6	4.7	1.7	0.0	14.3	17.6	4.6	1.7	0.0

TABLE V
EMPIRICAL PERCENTAGES OF TRUE COUNTS FALLING WITHIN 90% PIS. SECTION V-C DESCRIBES THE METHODS FOR GENERATING PIS. METHOD V-C2 GENERATES THE FORECASTS OF THE CONDITIONAL MEAN. METRICS ARE DERIVED FOR A FORECASTING HORIZON OF 30 MINUTES (6-STEP AHEAD). METRICS ARE EVALUATED FOR TIME FRAMES CORRESPONDING TO THE ASCENDING SLOPE (SEE SECTION V-A).

Date (Day)	$\mathrm{PI}^{\mathrm{emp}}_{90\%}$ (in percents)		
St. Cath ↓	ARIMA	$\mathrm{GARCH}(1,1)_{\mathrm{norm}}$	$\mathrm{GARCH}(1,1)_{\mathrm{t-dist}}$
12-10 (Mon)	87.5	83.3	87.5
12-13 (Thu)	68.8	81.2	87.5
12-17 (Mon)	81.2	81.2	83.3
12-18 (Tue)	64.6	79.2	79.2
12-01 (Sat)	90.3	95.8	95.8
12-14 (Fri)	89.3	88.1	88.1
12-15 (Sat)	76.4	97.2	97.2
12-28 (Fri)	75.0	95.2	97.6
Average	79.1	87.7	**89.5**
Bourse ↓	ARIMA	$\mathrm{GARCH}(1,1)_{\mathrm{norm}}$	$\mathrm{GARCH}(1,1)_{\mathrm{t-dist}}$
12-25 (Tue)	70.8	79.2	72.2
12-26 (Wed)	68.1	81.9	84.7
12-27 (Thu)	86.1	87.5	87.5
12-28 (Fri)	68.1	81.9	97.2
12-29 (Sat)	66.7	88.9	90.3
12-30 (Sun)	79.2	88.9	88.9
01-04 (Fri)	79.2	83.3	93.1
01-05 (Sat)	77.8	83.3	100.0
Average	74.5	84.4	**89.2**

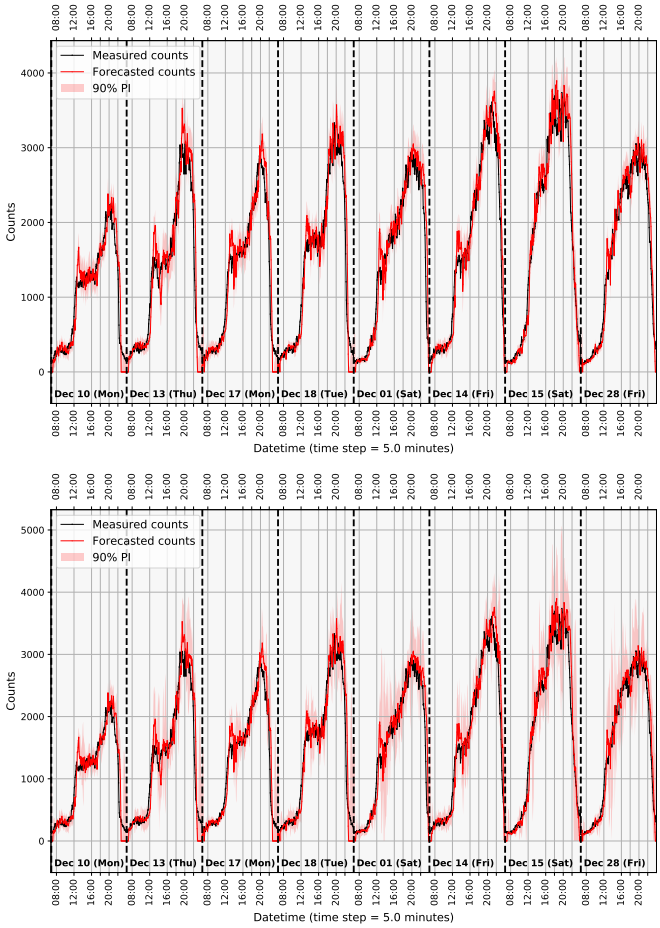

Fig. 5. Raw counts, forecasts and PIs for area *Sainte-Catherine*, which corresponds to Sensors S1 to S7 (see Figure 3). A rolling ARIMA(2,2,1) model generates the forecasts, see Method V-C2. The forecasting horizon is 30 minutes (6-step ahead). All days are fitted independently from one another. Markers on the x axis without label appear at 18:00 and 22:00. The order of the days groups them according to the similarity of their patterns. The extrapolation factor is equal to 3. Some artificial artifacts appear at the beginning of each day (except the first one), they are linked to the concatenation from several days and to forecast counts reaching zero. Top: the ARIMA model generates PIs, see Method V-C4. Bottom: a rolling GARCH(1,1) model (with Gaussian conditional variances) generates PIs, see Method V-C5.

REFERENCES

[1] G. K. Still, *Introduction to crowd science*. CRC Press, 2014.

[2] C. Martella, J. Li, C. Conrado, and A. Vermeeren, "On current crowd management practices and the need for increased situation awareness, prediction, and intervention," *Safety science*, vol. 91, pp. 381–393, 2017.

[3] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[4] C. Francq, J.-M. Zakoian *et al.*, "Maximum likelihood estimation of pure GARCH and ARMA-GARCH processes," *Bernoulli*, vol. 10, no. 4, pp. 605–637, 2004.

[5] V. Acuna, A. Kumbhar, E. Vattapparamban, F. Rajabli, and I. Guvenc, "Localization of WiFi devices using probe requests captured at unmanned aerial vehicles," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2017, pp. 1–6.

[6] B. S. Çiftler, S. Dikmese, I. Güvenç, K. Akkaya, and A. Kadri, "Occupancy counting with burst and intermittent signals in smart buildings," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 724–735, 2018.

[7] E. Vattapparamban, B. S. Çiftler, . Güvenç, K. Akkaya, and A. Kadri, "Indoor occupancy tracking in smart buildings using passive sniffing of probe requests," in *2016 IEEE International Conference on Communications Workshops (ICC)*, May 2016, pp. 38–44.

[8] J. Weppner, B. Bischke, and P. Lukowicz, "Monitoring crowd condition in public spaces by tracking mobile consumer devices with WiFi interface," in *Proceedings of the 2016 ACM International Joint Conference*

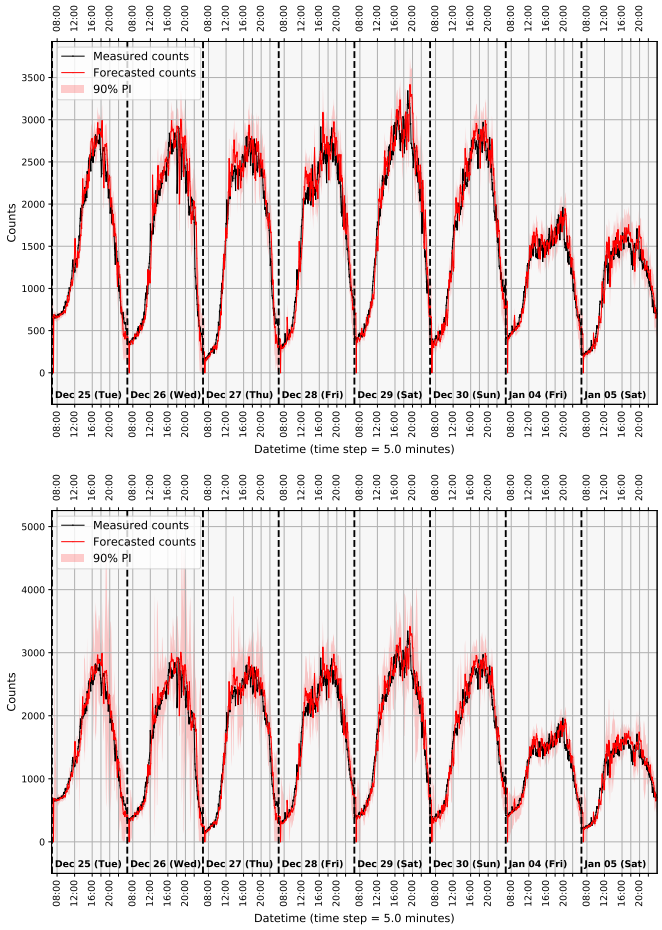

Fig. 6. Raw counts, forecasts and PIs for area *Bourse*, which corresponds to Sensors S14 to S19 (see Figure 4). A rolling ARIMA(2,2,1) model generates the forecasts, see Method V-C2. The forecasting horizon is 30 minutes (6-step ahead). All days are fitted independently from one another. Markers on the x axis without label appear at 18:00 and 22:00. The extrapolation factor is equal to 3. Some artificial artifacts appear at the beginning of each day (except the first one), they are linked to the concatenation of data from several days. Top: the ARIMA model generates PIs, see Method V-C4. Bottom: a rolling GARCH(1,1) model (with Gaussian conditional variances) generates PIs, see Method V-C5.

on Pervasive and Ubiquitous Computing: Adjunct. ACM, 2016, pp. 1363–1371.

[9] C. Chilipirea, A. Petre, C. Dobre, and M. van Steen, "Presumably simple: Monitoring crowds using WiFi," in *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, vol. 1, June 2016, pp. 220–225.

[10] L. Schauer, M. Werner, and P. Marcus, "Estimating crowd densities and pedestrian flows using Wi-Fi and Bluetooth," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ICST (Institute for Computer Sciences, Social-Informatics and ..., 2014, pp. 171–177.

[11] M. Handte, M. U. Iqbal, S. Wagner, W. Apolinarski, P. J. Marrón, E. M. M. Navarro, S. Martinez, S. I. Barthelemy, and M. G. Fernández, "Crowd density estimation for public transport vehicles." in *EDBT/ICDT Workshops*, 2014, pp. 315–322.

[12] A. Kurkcu and K. Ozbay, "Estimating pedestrian densities, wait times, and flows with Wi-Fi and Bluetooth sensors," *Transportation Research Record*, vol. 2644, no. 1, pp. 72–82, 2017.

[13] A. Guillén-Pérez and M. D. C. Baños, "A WiFi-based method to count and locate pedestrians in urban traffic scenarios," in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2018, pp. 123–130.

[14] W. Xi, J. Zhao, X.-Y. Li, K. Zhao, S. Tang, X. Liu, and Z. Jiang, "Electronic frog eye: Counting crowd using WiFi," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 361–369.

[15] "Traffic monitoring guide," U.S. Department of Transportation, Federal Highway Administration, Tech. Rep. FHWA-PL-13-015, September 2013. [Online]. Available: https://www.fhwa.dot.gov/policyinformation/tmguide/

[16] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: device-free location-oriented activity identification using fine-grained WiFi signatures," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 617–628.

[17] S. Liu, Y. Zhao, F. Xue, B. Chen, and X. Chen, "DeepCount: Crowd counting with WiFi via deep learning," *arXiv preprint arXiv:1903.05316*, 2019.

[18] S. Depatla, A. Muralidharan, and Y. Mostofi, "Occupancy estimation using only WiFi power measurements," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 7, pp. 1381–1393, 2015.

[19] M. W. Aziz, F. Naeem, M. H. Alizai, and K. B. Khan, "Automated solutions for crowd size estimation," *Social Science Computer Review*, vol. 36, no. 5, pp. 610–631, 2018.

[20] H. Shu, C. Song, T. Pei, L. Xu, Y. Ou, L. Zhang, and T. Li, "Queuing time prediction using WiFi positioning data in an indoor scenario," *Sensors*, vol. 16, no. 11, p. 1958, 2016.

[21] I. K. Tan, O. B. Yaik, and O. B. Sheng, "Predicting shopper volume using ARIMA on public Wi-Fi signals," *International Information Institute (Tokyo). Information*, vol. 19, no. 8A, p. 3295, 2016.

[22] M. Claeys Bouuaert, "Modeling crowds at mass-events: learning large-scale crowd dynamics from Bluetooth tracking data," http://cartogis.ugent.be/mobileghent/sites/default/files/slides/Modeling%20crowds%20at%20mass-events%20-%20MG2013%20-%20MCB.pdf, 2013.

[23] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transport reviews*, vol. 24, no. 5, pp. 533–557, 2004.

[24] M. Gast, *802.11 wireless networks: the definitive guide*. " O'Reilly Media, Inc.", 2005.

[25] C. Matte, M. Cunche, F. Rousseau, and M. Vanhoef, "Defeating MAC address randomization through timing attacks," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2016, pp. 15–20.

[26] J. Freudiger, "How talkative is your mobile device?: an experimental study of Wi-Fi probe requests," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2015, p. 8.

[27] M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, and F. Piessens, "Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 2016, pp. 413–424.

[28] H. Lütkepohl, *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

[29] P. J. Brockwell, R. A. Davis, and M. V. Calder, *Introduction to time series and forecasting*. Springer, 2002, vol. 2.

[30] D. Kwiatkowski, P. C. Phillips, P. Schmidt, and Y. Shin, "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?" *Journal of econometrics*, vol. 54, no. 1-3, pp. 159–178, 1992.

[31] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: the forecast package for R," *Journal of Statistical Software*, vol. 26, no. 3, pp. 1–22, 2008. [Online]. Available: http://www.jstatsoft.org/article/view/v027i03

[32] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American statistical association*, vol. 74, no. 366a, pp. 427–431, 1979.

[33] J. Durbin and S. J. Koopman, *Time series analysis by state space methods*. Oxford university press, 2012.

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

[35] R. Hyndman, G. Athanasopoulos, C. Bergmeir, G. Caceres, L. Chhay, M. O'Hara-Wild, F. Petropoulos, S. Razbash, E. Wang, and F. Yasmeen, *forecast: Forecasting functions for time series and linear models*, 2018, r package version 8.4. [Online]. Available: http://pkg.robjhyndman.com/forecast

[36] H. Akaike, "A new look at the statistical model identification," in *Selected Papers of Hirotugu Akaike*. Springer, 1974, pp. 215–222.

[37] J. E. Cavanaugh, "Unifying the derivations for the Akaike and corrected Akaike information criteria," *Statistics & Probability Letters*, vol. 33, no. 2, pp. 201–208, 1997.

[38] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[39] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications: With R Examples*. Springer Science & Business Media, 2006.

[40] G. E. Box and D. R. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 26, no. 2, pp. 211–243, 1964.

[41] V. M. Guerrero, "Time-series analysis supported by power transformations," *Journal of Forecasting*, vol. 12, no. 1, pp. 37–48, 1993.

[42] H. L. Nelson Jr and C. Granger, "Experience with using the Box-Cox transformation when forecasting economic time series," *Journal of Econometrics*, vol. 10, no. 1, pp. 57–69, 1979.

[43] H. Lütkepohl and F. Xu, "The role of the log transformation in forecasting economic variables," *Empirical Economics*, vol. 42, no. 3, pp. 619–638, 2012.

[44] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica: Journal of the Econometric Society*, pp. 987–1007, 1982.

[45] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[46] J. L. Torres, A. Garcia, M. De Blas, and A. De Francisco, "Forecast of hourly average wind speed with ARMA models in navarre (Spain)," *Solar Energy*, vol. 79, no. 1, pp. 65–77, 2005.

[47] P. Cortez, M. Rio, M. Rocha, and P. Sousa, "Multi-scale internet traffic forecasting using neural networks and time series methods," *Expert Systems*, vol. 29, no. 2, pp. 143–155, 2012.

[48] P. Torres, P. Marques, H. Marques, R. Dionísio, T. Alves, L. Pereira, and J. Ribeiro, "Data analytics for forecasting cell congestion on LTE networks," in *2017 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 2017, pp. 1–6.

[49] D. Wuertz, T. Setz, Y. Chalabi, C. Boudt, P. Chausse, and M. Miklovac, *fGarch: Rmetrics - Autoregressive Conditional Heteroskedastic Modelling*, 2019, r package version 3042.83.1. [Online]. Available: https://CRAN.R-project.org/package=fGarch

Jean-François Determe received the electrical engineering degree (Master en ingénieur civil électricien) from Université libre de Bruxelles (ULB) in 2013. He also jointly received the PhD in Engineering from ULB and Université catholique de Louvain in 2018. From 2013 to 2017, he was an FNRS research fellow, funded by the Belgian FRS-FNRS. Since 2018, he has been a postdoctoral researcher with the OPERA-WCG department at ULB and is currently funded by Innoviris. His research interests focus on applied time series analysis and on sparse recovery algorithms.

Utkarsh Singh (S'17, M'18) graduated in electrical and electronics engineering from Uttar Pradesh Technical University, India, in 2012. He received the master's degree in Power Systems from Thapar University, India, in 2014. He completed his PhD in Power Quality from the Indian Institute of Technology Roorkee in 2018. Since June 2018, he is working as a postdoc in OPERA-Wireless Communications Group at Université libre de Bruxelles, Belgium. His research interests include artificial intelligence, data analysis, optimization, power systems and signal processing.

François Horlin received the Ph.D. degree from the Université catholique de Louvain (UCL) in 2002. He specialised in the field of signal processing for digital communications. After his Ph.D., he joined the Inter-university Micro-Electronics Center (IMEC). He led the project aiming at developing a 4G cellular communication system in collaboration with Samsung Korea. In 2007, Francçois Horlin became professor at the Université libre de Bruxelles (ULB). He is currently supervising a research team working on next generation communication systems. Localization based on 5G signals, filterbank-based modulations, massive MIMO and dynamic spectrum access are examples of currently investigated research topics.

Philippe De Doncker received the M.Sc. degree in physics engineering and the Ph.D. degree in science engineering from the Université Libre de Bruxelles (ULB), Brussels, Belgium, in 1996 and 2001, respectively. He is currently a Professor with the ULB, where he leads the research activities on wireless channel modeling and electromagnetics.