Received: 30 December 2023 | Revised: 15 May 2024 | Accepted: 17 June 2024

IET Smart Cities

DOI: 10.1049/smc2.12087

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Predicting the impact of public events and mobility in Smart Cities

Elena Bellodi[1] | **Riccardo Zese**[2] | **Carlo Petrovich**[3] | **Angelo Frascella**[3] | **Francesco Bertasi**[1]

[1]Dipartimento di Ingegneria, University of Ferrara, Ferrara, Italy

[2]Dipartimento di Scienze Chimiche, Farmaceutiche ed Agrarie, University of Ferrara, Ferrara, Italy

[3]CROSS-TEC Laboratory, ENEA, Bologna, Italy

Correspondence

Elena Bellodi.
Email: elena.bellodi@unife.it

Present address

Francesco Bertasi, Exabyte Srl, Rovigo, Italy.

Funding information

Regione Emilia-Romagna, Grant/Award Number: POR FESR 2014-2020 ASSE 1 AZIONE 1.2.2 (CUP E21F18000200007)

Abstract

The ubiquitous presence of smartphones and the ever-expanding Internet of Things are generating a treasure trove of data on human movement. We harness the power of Artificial Intelligence to extract knowledge within this data, in particular for predicting people flows and density in a Smart City. This predictive ability holds immense potential for a multitude of applications, from optimising people flow to streamlining event planning, while offering a powerful tool for pre-emptive identification of situations that may lead to crowd disasters. In this paper, we tackle two crucial aspects of people mobility using data from public events and an Italian mobile phone network: to predict both event attendance and future crowd density in specific areas. The event details (location, time etc.) are automatically gathered and stored in a structured format. Next, we handle these problems are treated in a "supervised learning" setting, and various state-of-art Machine Learning techniques are tested to find the best model for each task. The obtained models will be encapsulated into a Policy Support System contributing to foster planning actions of mobility services.

KEYWORDS

artificial intelligence, data analytics and machine learning, data structures, governance, mobility, planning and policy, policy support system, smart cities, smart cities applications

1 | INTRODUCTION

The ubiquitous integration of Artificial Intelligence (AI) into various sectors traditionally reliant on conventional information and communication technologies (ICT) marks a significant shift. This phenomenon is fuelled by the ever-growing availability of vast datasets ("Big Data") [1], a trend further amplified by the Internet of Things (IoT) proliferation.

The widespread adoption of smartphones has led to an explosion in mobility data volume. This data typically comprises historical records of user visit sequences, enriched with contextual details, such as user subscriptions, timestamps, and location information. Public event information advertised online often includes geographical and temporal details, potentially alongside expected attendee numbers or venue capacity. This diverse and heterogeneous data presents unprecedented opportunities to uncover insights into human mobility patterns triggered by public gatherings and events—insights that local authorities may not fully grasp.

Understanding the flow of mobile users is crucial for the effective operation of numerous applications dependent on such information. Examples include traffic management, location-based advertising, planning access routes, and, more broadly, efficient event management. In fact, unexpected crowd gatherings or traffic congestion can pose significant threats to public safety and stability if not addressed promptly.

Abbreviations: ACE, Aree di CEnsimento in Italian, sub-areas of the Emilia-Romagna Italian region defined by the census; DB, Data base; DT, Decision Tree; ET, Extra Tree; GDBTs, Gradient Boosted Decision Trees; MAE, Mean Absolute Error; ML, Machine Learning; MLP, MultiLayer Perceptron; PSS, Policy Support System; RF, Random Forest; RMSE, Root Mean Square Error; SVM, Support Vector Machine; SVR, Support Vector Regression.

Understanding active people density is particularly critical for public safety in high-traffic areas, such as railway stations, bus terminals, scenic spots, and squares. However, abnormal people aggregations in overlooked areas pose an even greater safety risk. Furthermore, mapping people dynamics holds immense significance for various purposes, including city and transportation planning, public safety warnings, disaster impact assessments, and epidemic modelling. Accurate people density predictions could inform better policy decisions by regional authorities, potentially preventing tragedies, especially at large-scale events or in case of massive crowds. Incidents such as the Hajj crushes in Mecca (2006 and 2015), the Lame Horse fire crush in Perm (2009), the Love Parade disaster in Duisburg (2010), the Kumbh Mela stampede in Allahabad (2013), and the Shanghai New Year's Eve Stampede (2014) exemplify the dangers associated with crowd control. Therefore, ensuring safety requires understanding crowd behaviour and implementing effective crowd control measures.

Smart cities are thought to help solve this type of problems. A Smart City is "a place where traditional networks and services are made more flexible, efficient, and sustainable with the use of information, digital and telecommunication technologies, to improve its operations for the benefit of its inhabitants" [2]. In a Smart City, data about people mobility can be collected, analysed and understood through AI and Machine Learning (ML) tools so that, in the face of a large and unscheduled people influx, decision makers might not only react to the emergencies, but plan them. From this perspective, in this paper we addressed two predictive problems related to people mobility as supervised ML tasks:

- prediction of the increase in the number of people based on the occurrence of a public event;
- prediction of future people density inside delimited regional areas (ACE, in Italian "Aree di CEnsimento", i.e. sub-areas of the Emilia-Romagna Italian region defined by the census), given the density in the previous hours in that ACE and in the nearby ACE.

ML techniques for classification and regression were extensively tested with the goal of building the most accurate predictors based on empirical data, in particular: Decision Trees, Extra Trees, Random Forest, Support Vector Machines/Regressor, MultiLayer Perceptron, Gradient Boosted Decision Trees, AdaBoost. ML techniques have already shown to be one of the most powerful tools available for tourism- or mobility-related tasks [3]. The work presented in this paper was conducted within the framework of the POLIS-EYE[1] project, financed by the Emilia-Romagna (Italy) regional administration. The project aimed to provide managers, policy makers, and companies in the tourism and mobility sectors with advanced ICT solutions from a Smart City perspective, ultimately optimising the management of the Region. The project involved several public and private companies providing a number of data sources describing people mobility. The goal was to develop a Policy Support System (PSS) creating a platform dashboard displaying the results of descriptive, prescriptive, and predictive analysis models. On the one hand, this platform would aid in identifying and managing potentially unforeseen events (e.g. a large and unscheduled influx), enabling efficient emergency response and proactive planning. On the other hand, providing accurate predictions of people density or of the expected number of participants in an event has a significant practical impact both for the decision makers and for the event organisers. In ref. [4] a general overview of the platform is given.

Different data sets were collected, in particular related to mobile phone users and public events. The dataset of public events was built by developing and using a retrieval software of on-line information about events that took place in the Emilia-Romagna region, in a period of 4 months, advertised on several regional websites. According to regional statistics, the number of arrivals in the Region due to public events is remarkable, for example, only for the fair trade activities (business tourism) 2.2 million visitors were estimated in 2019 [5]. The collected information was stored according to a formal event abstract data model. Data was integrated with mobile data obtained from one of the major mobile phone networks in Italy in the same period.

More specifically, the contributions of the paper are as follows:

- an automated procedure for the retrieval of online information about public events;
- an automated procedure for labelling the events also to estimate the capacity of their venue;
- an abstract data model for reconstructing people flows, to be used also for data storage;
- an analysis of the performance of different learning predictors for two main mobility issues: the multi-class prediction of the percentage increase in the number of people due to a public event and the prediction of people density in an ACE in the next 1,2,4,8,12,24 h. We show that Gradient Boosted Decision Trees for the first task reach 86% accuracy, precision and recall in the range 85%–89% for two classes of increase, while for the second task Support Vector Regressors and MLP are able to accurately predict density (the first with mean absolute error and root mean squared error less than 1).

The suggested approaches can be employed to facilitate the development of a Smart City, enhancing the effective distribution of urban resources and traffic flow.

The paper is organised as follows: Section 2 describes related works. Section 3.1 describes data sources and data collection, Section 3.2 the abstract data model for events, Sections 3.3 and 3.4 the procedure for building the event database. Section 3.5 introduces the ML models used for the experiments, whose results are described in Section 4. Section 5 discusses the results, the limitations of the adopted approach and some possible solutions. Section 6 concludes the

[1] https://www.poliseye.it/.

paper. Appendix A reports feature rankings obtained from tree-based models.

2 | RELATED WORK

The study of human mobility is a recognised necessity in Smart Cities, where advances in continuous location tracking devices have enabled large-scale collection and analysis of data regarding people flows, leading to new insights in a wide range of fields [6]. The application of AI to predict human mobility in Smart Cities has already been proposed in the literature. [7] Zhang et al. use a Nokia Mobile Data Challenge dataset for predicting users' future locations by analysing the time series of the movements of the people. The authors characterise the properties of users' visited locations and movement patterns and then extract feature types (temporal, spatial, and system) to quantify the correlation between locations and features. Then, they train a set of models considering Random Forests, Multi Layer Perceptron and Bayesian Networks and apply ensemble methods to predict the future location of people given their previous movements. Our work differs in that we are not concerned with user movement patterns but rather with predicting the actual increase in influx at a specific location of a public event. This approach provides local authorities with a practical tool for managing crowd size. Ensemble models have also been used by Karbovskii et al. [8], who proposed a combined method to forecast crowd movement patterns in short bursts during large-scale events. Their study focuses on the Kumbh Mela, a massive Hindu pilgrimage known as one of the world's largest gatherings [9]. Data for this project includes a 3D model of the venue (a temple in this case) and video recordings of crowd movement. The goal is to predict short-term crowd flow by tracking when and if people pass designated markers placed along corridors and entry points. To achieve short-term crowd flow predictions, the researchers use an ensemble of three approaches: time-shifting, machine learning, and agent-based modelling. Each method has its advantages and limitations: Time Shift assumes a constant number of people and predicts movement based on past patterns. It is limited to situations where people cannot leave the crowd (like a single pathway). Machine Learning uses data analysis to make predictions, but its accuracy depends on the quality of training data. Agent-Based Modelling simulates individual crowd behaviour, but requires large amount of data, processing power, and can be difficult to verify. While this research is a valuable contribution to crowd flow prediction, it differs from our work in its focus on short-term predictions within a single, specific large event. This limitation arises from its reliance on a 3D model of the event venue and time series in form of security camera footage. Consequently, this approach is not applicable to city- or region-scale crowd flow prediction as we do by considering areas such as ACE. Salehi and Granitzer [10] propose to model the social influence of friends on event attendance by using posts to infer users' attendance. The approach is based on graph embedding techniques and neural networks. Another approach is described by Wu et al. [11], who define a model trained to predict people flows within a university campus, considering data on Wi-Fi network device access, vacation periods, and weather conditions. They evaluate different ensemble models created by stacking some or all of the following models: Lasso Regression, Ridge Regression, Decision Tree, Random Forest, Gradient Boosting Tree, and XGBoost. The key difference with our work lies in the scale when predicting future people density. We focus on larger areas than a university campus and therefore cannot rely on Wi-Fi access point records. Instead, we utilise data with lower granularity, such as cell phone tower connections.

While crowd mobility prediction is extremely important in specific locations, such as campuses or event venues, a large number of studies concentrates on a larger perspective, considering people flows on a city or even larger scale. For example, ref. [12] applies Long-Short Term memory (LSTM), a type of recurrent neural network, for tourist flow prediction in China, while ref. [13] predicts the people density of key areas in China in order to reduce the spread risk of Covid-19 and support individuals' travel needs: it proposes a model called Word Embedded Spatial-temporal Graph Convolutional Network (WE-STGCN) to reflect the temporal dependence and spatial correlation of the data. Similar to our work, ref. [14] employs data of mobile phone users provided by the China Mobile Communications Corporation and exploits the interaction between population behaviour and environmental characteristics to predict the spatial–temporal distribution of urban people density using convolutional neural networks. These works apply deep learning techniques without providing a comparison with traditional Machine Learning techniques, as performed in this paper. Similarly, Fan et al. [15] leverage deep learning models based on Gated Recurrent Units (GRU), a simpler variant of LSTM, to predict both routine and irregular citywide human mobility. Their approach involves an ensemble model composed of one pre-trained network (consisting of two GRU layers) for each day of historical data, and a model with a single GRU layer that gathers results from the other models of the ensemble and combines them with data from the current timestamp to predict human mobility. This approach utilises GPS data collected through an app installed on mobile devices. However, there are two main limitations to this data source: presence of a bias towards young people that are more likely to have smartphones with location services enabled, potentially skewing the data; and a significant app adoption requirement, because the app used for data collection needs widespread adoption to be effective. Fan et al. also compared their ensemble model to Random Forest, demonstrating superior performance. The key difference with our approach lies in the fact that, when predicting future people density, we focus on a wider area, leveraging data from the entire Emilia-Romagna region to capture a more comprehensive picture of people flow. Conversely, Fan et al.'s work may suffer from data sparsity in smaller cities or rural areas.

Pulselli et al. [16] introduce a real-time monitoring method for people density in urban areas, leveraging cell-phone activity as evidence of mobile-phone users' presence. It analyses time series data collected from cell-phone antennas within a

100 km^2 area in Milan, Italy. These data are provided by a national mobile telecommunications company in an aggregated and anonymous format. The intensity of cell-phone activity is mapped across 24 h to visualise variations in people density throughout a typical day. By presenting these maps in rapid sequence, researchers gain insights into dynamic social behaviour in the metropolitan area. Practically, this technique can be used to analyse urban dynamics, such as observing changes in patterns before, during, and after events like the construction of new infrastructures (e.g. subway stations or roads), which may alter mobility flows. Unlike our work, this approach is not concerned with prediction but focuses solely on monitoring and analysis, even though it utilises mobile data.

Li et al. [17] use data from mobile network operators for real-time monitoring and forecasting of large-scale active people density to prevent public safety accidents caused by abnormal people aggregation. Four forecasting methods—Simple Exponential Smoothing (SES), Double Exponential Smoothing (DES), Triple Exponential Smoothing (TES), and Autoregressive Integrated Moving Average (ARIMA)—are employed, with experimental results suggesting accurate forecasts for up to 135 min into the future. Their time perspective is significantly shorter than ours. [18] Zhang et al. employ mobile phone data and a deep learning approach to estimate and predict mobile users' dynamic distribution. They utilise a convolutional long short-term memory (ConvLSTM) module to predict the activity of mobile phone users' distribution, comparing it with a traditional time-series prediction model, that is, AutoRegressive Moving Average (ARMA), and a popular deep learning method, that is, LSTM, as baselines. The correlation analysis between the predicted and actual results is conducted every 2 h, revealing that the ConvLSTM method outperforms both ARMA and LSTM in accuracy. Feng et al. [19] introduce a bimodal model that leverages data from mobile networks, including call detail records and mobility management signals, to estimate real-time people distribution at the street block level. The model undergoes evaluation across six cities, and the results demonstrate a correlation between its performance, city size, and data type. Notably, the model performs better in larger cities, and finer temporal granularity enhances its accuracy.

In other cases, the use of people density prediction using mobile phone data has objectives very different from ours. For instance, Anderson et al. [20] employ Random forest on mobile phone data to obtain reliable small area estimates of population density for regions lacking direct samples, such as census data; Lee et al. [21] focus on characterising urban activity and mobility patterns in Korean cities; Gariazzo and Pelliccioni [22] investigate the complexity and dynamics of population mobility over long-term time scales (years) and intra-urban/regional/intra-national spatial scales, analysing population density variations using mobile phone data. Other similar works, such as refs. [23, 24], use deep learning techniques and different ensemble models to predict population density in Australia and New Zealand, and Turkey respectively considering census data. However, all these works share the objective of population density prediction during years instead of in the next hours. Interestingly, the study of forecasting people density in the next 24 h remains relatively unexplored. Our research aims to explore this new direction. We refer to [25] for a more comprehensive review of studies that use geospatial data to analyse urban areas with ML methods.

Another important aspect that sets our work apart from the majority of the literature is that the other studies are confined to the analysis aspects of people flows, while the proposed models have been designed to be integrated in a Policy Support System for mobility and tourism services. This PSS combines a data collection/exchange platform with an open implementation-oriented interoperability, AI-based business analytics services, and an advanced user interface. Also, it is designed to use and combine different external data sources and the possible entry of new variables and sources of information: in fact, two different versions of the database were used here according to the prediction task at hand.

3 | MATERIALS AND METHODS

3.1 | Data collection

Both weather data and events, such as business fairs, big concerts, soccer matches at the stadium, holidays, influence the flow of people and can highly increase the local presence of people in a particular area. An inflow analysis of people in a particular area has therefore to take into account the weather conditions, as well as events organised in that area, together with other external conditions (e.g. Covid-19 lockdown restrictions etc.). This data was collected from open sources and refer to the periods from August to September 2019 (61 days) and 2020 (61 days). We collected (i) daily weather data from www.ilmeteo.it (minimum, maximum and mean temperature, wind speed, humidity, pressure, rainfall, precipitation type); (ii) data relative to public events of different kind downloaded (automatically) from municipal, regional, and private tourist websites. All possible public events of all kinds in the Emilia-Romagna Region have been searched, even if deeper investigations were concentrated in the cities of Bologna, Modena and in the part of the region called Romagna, because of their higher attractiveness. Filters, in order to classify relevant and irrelevant events (according to attendances), have been applied in a second stage as discussed in Subsection 3.3. For August-September 2019 about 3100 events were retrieved, taking also minor recurrent events into account (such as markets); for August-September 2020 about 2000 events in total were found, most of which are small events because of the pandemic, for a total of 5123 records. Detailed information about (ii), together with the retrieval procedure from the web, can be found in the following sections.

We also acquired mobile data from one of the major Italian mobile phone networks, sampled every 15 min, and specific to each ACE of the region. Figure 1 shows the Emilia-Romagna map partitioned into ACE.

Mobile data store age and gender of people connected to the network, nationality (Italians/foreigners), type of contract

FIGURE 1 Partitioning of Emilia-Romagna into ACEs.

(business/regular), type of client (commuter/resident). The mobile phone data was available only for the two periods previously reported. The use of mobile data allows us to track people's movements anonymously by exploiting the connection of mobile phones to different cell towers. Anonymity is ensured because, for each mobile device included in the data, we only have a unique code containing no sensitive information. By analysing the time-stamped list of cell tower connections for each device, we can gain detailed insights into people movement patterns. This level of granularity would be difficult or impossible to achieve using Wi-Fi hot spots or other sources with less widespread territorial coverage. Finally, the specific time-frame and overall data volume available for analysis were determined by the purchase agreements with the mobile service provider, and were subject to restrictions.

The collection of data about organised events in a dedicated database is crucial, but at the same time turns out to be complex for different reasons: the absence of a unique data source for all the events, the variety of the organisers, the different formats and structures—if any—in which the event data are provided by the organisers and by the dissemination centres (e.g. sport authorities, local municipalities, websites for entertainment or tourism etc.). More importantly, the number of people participating in the events is not monitored, making it difficult to quantitatively measure the impact on people flow. Because of those reasons, two activities were performed for building a database (DB) for Emilia-Romagna events, as illustrated in the following sections.

3.2 | An abstract data model for events

In literature, characterisation criteria for events were proposed for example, by Getz [26], which identified 18 classification items, such as the main goal of the event, frequency, duration and timing, number of attendances. For web applications, a categorisation of events has been proposed for example, by AGID (the Agency for Digital Italy), within the OntoPiA project [27], which is a network of ontologies and controlled vocabularies for public administration. In particular, a controlled vocabulary has been proposed for the "Types of public events" [28], together with an ontology for the representation of public events, called CPEV-AP_IT (Core Public Event Vocabulary—Italian Application Profile) [29]. The latter identifies 29 items to characterise an event. A finer modelling of events, with XML elements, was developed within the E015 Digital Ecosystem, promoted by the Lombardy Region in Italy for Expo 2015 [30].

The abstract data model for events developed for this work takes inspiration from the E015 specifications and extends some categories and controlled vocabularies. Events are categorised according to the following features: event ID, event title, description, qualifier (type of event, explained later), category, theme, location (described by an identifier, type, name, environment (indoor, outdoor), address, zip code, town, province, geographic coordinates), start and end date, frequency in case of repeated events, organiser, URL used to download the information, target users and dimension. The *dimension* attribute is an integer representing the expected number of participants according to a scale 1–6, such that 1 corresponds to the interval [0, 50 [people, 2 to [50, 200 [people, 3 to [200, 1000 [people, 4 to [1000, 10,000 [people, 5 to [10,000, 100,000 [people, 6 to ≥ 100,000 people.

We considered three main types of event qualifiers: (1) organised single events (e.g. concerts) or periodical events (e.g. monthly or weekly markets); (2) anniversaries (festivities, local patron saint holidays etc.); (3) context events (lockdown restrictions, scholar calendar etc.). Many fields are not "free text" (e.g. the location type of the event), but predefined: 11 code lists, for the respective features, were prepared (for example the location type can be: city downtown, a hall, a palace, a park etc.). Table 1 shows an example for a specific event only for the main fields. With respect to E015 and other event models, two main issues have been improved: (1) Comprehensive code lists allow us to control the labels and therefore aid the assignment of values to the *dimension* attribute, in particular thanks to the labels referring to "location". This is one of the main challenges of this work, as explained in subsection 3.3. (2) The model is conceived to be consistent with an existing and tested Smart City model [31], allowing a comprehensive approach, including transmission of data in a uniform JSON format. This model includes other phenomena description, such as transit

TABLE 1 Example of the abstract data model used for events.

Event features	Event info	
ID (identifier)	5001	
Title	COSMOPROF	
Description	International hair and nail and beauty salon	
Event qualifier	Event	
Tag (keyword)	Cosmetics	
Category	Fairs	
Theme (scope of the event)	Beauty	
Place	ID (identifier)	37
	Type of place	Exhibition centre
	Name of the place	Fiera di Bologna
	Environment	Indoor
	Address	Viale della fiera, 20 40127 Bologna BO Italy
	Geolocation (WGS84 format)	44.509231, 11.368966
Period		
	Start date	2020-06-11
	End date	2020-06-11
…	…	
Dimension	5	

Note: Dimension = 5 means an inflow between 10,000 and 100,000 people.

data models, weather data models and energy data models, which are defined by an ontology. Moreover, a software layer is able to automatically extract the JSON data format from this ontology. This opens up the possibility of analyses and data exchanges among different sector domains. For a detailed description of the design, the properties and the code lists, see the technical report [32] and the website [33].

3.3 | Assessment of people attendance to the events

People attendance to events can be very different: events can involve a very small number of people, such as 20–30 people (e.g. presentation of a book in a bookshop) or 40,000 people (e.g. a soccer match). We provided an assessment of the amount of people attending events in the generated DB by labelling each one with an integer number (1–6) representing a specific range of people attending the event. This number represents the class of the event and corresponds to the *dimension* feature, so that multi-class classification techniques could be applied for prediction purposes. The procedure for labelling events was split in two steps:

(i) Automatic scan and analysis of the *location type* feature when available in the database. According to the value of the attribute, a dimension is assigned to the event. The approach is based on the idea that the location is chosen by the organisers taking into account the foreseen number of people attending the event. Events classified with 5 or 6 (more than 10,000 people) had taken place in big stadiums, big fairs, big arenas, while small events classified with class 1 or 2 were those happening in libraries, restaurants, small halls etc.

(ii) For events for which the *location type* feature was not available, as could not be determined during the event retrieval process, the class was assigned by analysing together the "category" and "town" attributes of the event, where "category" can be Fairs, Markets, Exhibits, Plays, Musicals, Patron saint's festivals, Competitions, Workshops. For each category (except "Markets") for which at least one event had been assigned a class at step 1, all assigned classes were collected in a list; all lists turned out to contain two classes. Then, if the "town" corresponded to a provincial capital, the greatest value for the class was assigned to the event, otherwise the lowest one. For the case of markets, given that the information about the market type was available in the database ("farmers' market", "weekly market"), class 1 was assigned to farmers' markets, class 2 was assigned to weekly markets in a town that is not a province, and class 3 to weekly markets in a provincial capital. The idea behind this is that farmers' markets are small, while weekly markets can be small or relatively big according to the city population.

3.4 | Automatic event retrieval

A software[2] written in Java and ASP fills a PostgreSQL DB of events through the following steps:

1. It scans more than 30 selected websites designed for dissemination of planned events in Emilia-Romagna, from which complete event information could be retrieved. The websites included both those managed by public institutions (e.g. Bologna municipality, Emilia-Romagna Region) and by private companies (media magazines etc.), providing sport events, exhibition events, food events, cultural and musical events etc. Features were retrieved by querying the html and json data or metadata of the websites;
2. It analyses locations, usually expressed as an address or building name (e.g. 'Stadium of …'), in order to automatically match the different names of the same places with their geographical coordinates (extracted by a REST query towards the https://opencagedata.com/ website);
3. Deletes duplicate events, adjusts formats, identifies recurrent events, and integrates the acquired information, whenever possible;
4. Adds further events, not detected from the web (e.g. festivities or other special kinds of events);
5. Populates a PostgreSQL database with the above data. The data structure of the DB is designed with the ER (Entity Relationship) diagram: all categories (events, locations etc.) are related to each other through well modelled relationships; the generalised use of code lists allows one to represent the information precisely and unambiguously.

The software ran a few times in the years 2020–2021 and retrieved almost 18,000 major events in Emilia-Romagna occurred in the entire years 2019–2020, which become more than 37,000 if also minor recurrent events (e.g. local markets) are taken into account. From this collection events taking place in August-September 2019 and 2020 were extracted.

3.5 | Predictive machine learning

In this section, we describe the prediction models we used for addressing two tasks: (1) prediction of the class of expected percentage increase of people in a specific area based on the occurrence of a public event (classification task); (2) prediction of people's future density inside an area, given the density in the previous hours and in the nearby areas (regression task).

We employed both single classifiers or regressors and ensemble models. As for the latter, *boosting* is a type of ensemble model, a paradigm that leverages the collective intelligence of multiple models (possibly using different learning algorithms) to achieve superior performance compared to a single model [34]. These methods work by training a collection of "weak learners" and then combining their predictions for a final output. This approach aims to address the limitations of individual models and enhance the overall accuracy, robustness, and generalisability of the learning process. Besides boosting, another possible approach to ensemble model is *bagging*, introduced by Breiman [35], that focuses on reducing the variance of a model by training multiple models on different subsets of the original data (created with sample with replacement). These models, typically decision trees, can be diverse due to the variations in the training data. Bagging is effective for unstable models prone to high variance, leading to improved prediction stability.

Among the ensemble models available in the ML literature, we propose in the following Gradient Boosted Decision Trees and AdaBoost, which fall into the *boosting* category, and Random Forest, an extension of *bagging* that also randomly selects subsets of features used in each data sample.

3.5.1 | Classification models

Random Forests (RFs) [36] are known for their good computational performance and scalability. Also, they are known for their versatility in handling both classification and regression tasks with high accuracy. A RF is a meta estimator that fits a number of Decision Tree (DT) classifiers on various sub-samples of the data and uses averaging to improve the predictive accuracy and control overfitting, making it robust against noise within the dataset. In RFs each tree is built from a sample drawn with replacement (i.e. a bootstrap sample) from the training set. In contrast to the original publication, the implementation used in the experiments combines DT classifiers by averaging their probabilistic prediction, instead of letting each classifier vote for a single class. Additionally, RFs handle high-dimensional data efficiently and provide useful indicators of feature importance, that provides insights on which features more significantly impact the predictions. Extra Trees (ET) [37] differ from classic decision trees in the way they are built: they choose the split at each node randomly, instead of taking the best one, to add more randomness in the process.

Support Vector Machines (SVM) [38] are powerful methods allowing to construct models that are complex enough (they include radial basis function nets, polynomial classifiers, a large class of neural nets as special cases) for non-linear classification, pattern recognition etc., and they offer strong generalisation guarantees, which are crucial for the predictive stability of our models.

MultiLayer Perceptron (MLP) [39] is a fully connected class of feedforward Artificial Neural Networks which are suitable for classification prediction problems and has the capability to learn non-linear models. It is particularly powerful in capturing complex patterns and interactions between features, making them suitable for tasks where feature relationships are deeply non-linear.

Gradient Boosted Decision Trees (GBDT) [40] are an excellent model for both regression and classification, in

[2] https://www.cross-tec.enea.it/tecnopolo/imple/pgcl.asp?lingua=en&p=757.

particular for tabular data. This algorithm falls into the category of ensemble models and is a generalisation of boosting to arbitrary differentiable loss functions. It builds an additive model in a forward stage-wise fashion: in each stage *n_classes* regression trees (where *n_classes* is the number of classes) are fit on the negative gradient of the loss function, for example, binary or multiclass log loss. GBDT is highly effective in handling non-linear relationships between features due to the use of decision trees as the base learners, and it often provides predictive accuracy that is superior to many other classification algorithms due to its method of combining the output from many weak learners. Similarly to other tree-based methods, GBDT inherently provides estimates of feature importance, which can be useful for understanding the influencing factors in data-driven decisions.

AdaBoost (Adaptive Boosting) is another boosting algorithm introduced by Freund and Schapire [41] which fits a sequence of weak learners (i.e. models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction. The data modifications at each so-called boosting iteration consist of applying weights to each of the training samples. An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. The default base estimator is a DT classifier. AdaBoost can significantly enhance the accuracy of simple models by combining multiple weak learners into a strong learner in a sequential manner, where each subsequent model focuses on the errors of the previous ones. It requires very few parameters to tune (e.g. number of iterations), making it simpler to use compared to more sophisticated machine learning algorithms. It can be used with any learning algorithm that accepts weights on training instances, and can be applied to binary classification problems or extended to handle multi-class targets.

3.5.2 | Regression models

Decision trees, Extra trees and Random Forests can also be used as regressors. For RFs, in the trees of the forest each node chooses a threshold to split the examples assigned to the node in two sets, until a leaf is reached. In this case, a leaf represents a region of the example space, defined by a set of assignments of the nominal variables and upper and lower bounds on numerical variables. The label of the leaves is usually the mean of the observations occurring within that specific region. The output of the RF will be the average of the prediction given by each single tree.

Unlike SVM, Support Vector Regression (SVR) seeks to find a hyperplane that best fits the data points in a continuous space. SVR is typically robust to outliers and has excellent generalisation capability. It was also selected for its effectiveness in high-dimensional spaces—this is particularly useful in the dataset used for the regression task (with about 2100 dimensions).

MLP can also be used for regression: each real value to predict is associated with an output neuron. MLP can be used as a regressor due to its proficiency in approximating any continuous function, which is a pivotal asset in regression tasks where complex, non-linear interactions among variables must be captured. MLPs are particularly advantageous in scenarios where the relationship between the input variables and the output variable is not only non-linear but also non-obvious, requiring a model that can learn these relationships from the data without explicit specification. MLPs are composed of multiple layers of neurons with non-linear activation functions. This architecture enables them to learn non-linear relationships more effectively than linear models or even some other non-linear models. MLPs scale relatively well to larger datasets and more complex models, which is essential given the volume and dimensionality of data used for the second task.

Gradient Boosted Decision Trees for regression build an additive model in the same way done for classification, and many parameters are unchanged from the classifier version: one exception is the max *_iter* parameter that replaces *n_estimators*, and controls the number of iterations of the boosting process. In addition, available losses for regression are squared error, absolute error, which is less sensitive to outliers, and poisson. For classification, log loss is the only option.

An AdaBoost regressor is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction. As such, subsequent regressors focus more on difficult cases. The default base estimator is a DT regressor.

4 | RESULTS

This section presents the experimentation and detailed performance evaluation of the discussed methods applied for solving tasks (1) and (2).

The library scikit-learn [42] was used to train models for classification and regression except for neural networks. Scikit-learn is an open source Python machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection, model evaluation, and many other utilities.

For learning neural networks, AutoKeras[3] and Tensorflow were used. AutoKeras is a Python implementation of AutoML (Automated Machine Learning) for deep learning models based on the Keras API, specifically the API provided by TensorFlow 2. It is a tool for automating the process of model design and hyperparameter tuning in Neural Architecture Search, or NAS for short, in order to identify the combination

[3] https://autokeras.com/.

with the best performance. The use of AutoML tools to handle MLPs offers several significant advantages: one of the primary benefits is the capability to automatically search for the optimal model architecture. For MLPs, this means it can determine the best number of hidden layers and neurons per layer, as well as the appropriate types of activation functions. Alongside architecture, these tools optimise other critical hyperparameters, such as learning rate, batch size, and epochs. This ensures that the MLP is not only structurally optimal but also fine-tuned to perform best with the training data.

4.1 | Predicting the class of expected increase of people due to public events

The target is to learn a predictive model that, given a future event (in a specific ACE) for which the number of visitors is unknown, determines the expected percentage increase of people among five classes of increase. So, Random Forest, Extra Trees, SVMs, MLPs, GBDTs and AdaBoost models for multi-class classification were learnt. From the database we built, we were able to use a total of 3133 events, as a result of some pre-processing steps, which removed, from the initial 5,123, events without position (longitude and latitude), events located in an ACE not related to the mobile phone data, events with mistaken dates (end date before start date); in the end, non-daily events were replicated in order to generate one row for each day.

For every event, the mobile phone data was used to determine the "average peak", the "average peak per day" and "average peak per month". The first one represents the mean of the highest numbers of individuals present each day within the event venue over a 61-day period. The second one is the mean of the highest numbers of people present in the event venue on each specific day of the week over the same 61-day duration. The third one is the mean of the highest numbers of people present in the event venue over the first and the second month. One-hot encoding was used to convert categorical attributes such as category, theme, day of the week, and precipitation type into numerical values. Finally, we obtained a dataset of 3133 events × 14 columns, representing: average peak, average peak per day, average peak per month, category, theme, province, duration, day of the week, target attendance, dimension, minimum, maximum and average temperature, and precipitation type. 'Theme' was dropped as carrying similar information to 'category'; 'province' was dropped as being evaluated as unrelated to the percentage increase of people; 'target attendance' was dropped as converted into the 'Dimension' feature as explained in Section 3.2. According to the attendance scale, 472 events were assigned size 1, 2124 events size 2, 443 events size 3, 1 event size 4, 93 events size 5: the distribution of events shows a significant imbalance of examples in favour of the first dimensions (those related to smaller-sized events). The attribute to be predicted (class) is the percentage increase in the number of visitors calculated as $100 \times \max \; peak/average\, peak\, per\, day$, where max peak is the maximum number of people connected to the mobile network where the event takes place on the day of the event. The percentage increase resulting from the formula above was split into the five classes +[0–25]%, +(25–50]%, +(50–75]%, +(75–100]%, and +≥100% to best fit the data. Overall the dataset contained 15 attributes. The dataset was divided into 2193 training examples (70%) and 940 test examples (30%). The training events are distributed as follows according to the classes: +[0–25]%: 1152 examples, +(25–50]%: 137 examples, +(50–75]%: 26 examples, +(75–100]%: 17 examples, +≥100: 861 examples. The classes in the training set are unbalanced, in particular in the cases +(50–75]% and +(75–100]%, for which the number of examples available is very small. The test set had the following support: 494 examples for class + [0–25]%, 59 examples for class +(25–50]%, 11 examples for class +(50–75]%, 7 examples for class +(75–100]%, 369 examples for the last class.

To tune the hyper-parameters of a RF and an ET a grid search was performed, using values [2, 5, 10, 20] for max_depth, [1, 10, 20] for min_samples_leaf, [10, 50] for n_estimators and [5, 10, 20] for max_features (the number of attributes to consider). The best hyper-parameters found for training the RF and the ET are reported in Table 2. All the other parameters were set to their default values. Grid search and training took less than a minute.

To tune the hyper-parameters of a GBDT a grid search was performed, using values [5, 10, 15, 20] for max_depth, [1, 5, 10, 15] for min_samples_leaf, [100, 105, 110] for n_estimators and [10, 15, 20, 24] for max_features. The best hyper-parameters found for training the GBDT are reported in Table 2. To tune the hyper-parameters of AdaBoost a grid search was performed, using values [*gini*, log *_loss*] for *estimator_criterion*, [5, 10, 15, 20, 25, 30] for *estimator_max_depth*, [1, 5, 10, 15] for *estimator_min_samples_leaf*, [10, 15, 20, 24] for *estimator_max_features*, [170, 180, 190] for *n_estimators*. The estimator used was a Decision Tree classifier, with the SAMME algorithm. The best hyper-parameters found for training the model are reported in Table 2.

With SVMs the best hyper-parameters found were $kernel = rbf$ (radial basis function), $C = 208.79$ (regularisation parameter), $\gamma = 0.00026$ (kernel coefficient).

For learning the MLP all values in the dataset were normalised. AutoKeras was applied on the training data using $max\_trials = 10$ (the maximum number of trained networks), $tuner = Bayesian$, $epochs = 250$, $batch\_size = 64$,

TABLE 2 Best hyper-parameters for the tree-based models for the prediction of expected increase of people due to public events.

Hyper-parameter	RF	ET	GBDT	AdaBoost
max *_depth*	5	20	5	25
min *_samples_leaf*	1	1	5	5
max *_features*	20	20	24	15
n_estimators	10	50	105	180
estimator_criterion				*gini*

validation_split = 0.2, Adam optimiser, *learning_rate* = 0.01, and Dropout. The best architecture found is a shallow network with one hidden layer with 256 neurons, ReLU activation function and Dropout. The output layer has 5 neurons, one for each class, with Softmax activation function.

4.1.1 | Variants of the original experiment

In order to understand the contribution of each feature or group of features to the model performance, we generated a correlation matrix for numeric attributes and one for nominal attributes (Figure 2). Firstly, as the 'target attendance' contains the actual number of people attending an event, we selected the features with a low correlation (<0.1) with this attribute: duration, minimum, maximum and average temperature, to be dropped before re-learning the models. Secondly, we searched for high-correlated attributes (correlation ≥0.95): the average temperature is highly correlated with the minimum and maximum values, so it could be kept in the dataset without the other two attributes; the average peak is highly correlated with the average peak per day (0.98) and per month (0.89), so the average peak per day could be dropped. We generated four variants of the original dataset according to this analysis:

1. Variant 1: the dataset with the 'Duration' feature dropped.
2. Variant 2: the dataset with the 'Duration' + 'MIN_Temp' + 'MAX_Temp' features dropped.
3. Variant 3: the dataset with the 'Duration' + 'MIN_Temp' + 'MAX_Temp' + 'AVG_Temp' features dropped.
4. Variant 4: the dataset with the 'Duration' + 'MIN_Temp' + 'MAX_Temp' + 'AVG_Temp' + 'Average peak per day' features dropped.

By removing those features from the dataset one at a time we could trace how the model performance varied.

We performed the hyper-parameter search as explained in the previous section. For the RF, ET and MLP models learnt from the dataset variants we obtained the same best hyper-parameters. For SVM we obtained the same kernel (rbf) and similar values for C and γ. For GBDT and AdaBoost the best values for the hyper-parameters varied for max_depth, min_samples_leaf, max_features, n_estimators, criterion according to the dataset variant.

All dataset variants were subjected to a feature analysis by exploiting the "feature importance" option available when learning a tree-based model (all models except SVMs and MLP). Detailed information about this can be found in the Appendix A.1: all feature rankings agree on assigning the highest importance to temperatures and the average peaks in predicting the class.

Results, in terms of accuracy, precision, recall and F1-score on the test set, are shown in Tables 3–8. Note that, while precision, recall and F1-score are computed for every class, the value of accuracy is unique.

Given information about a future event, the classifiers predict the class of percentage increase in the number of visitors (e.g. +(0–25]%) on the day of the event w.r.t. the average number of people in the same ACE when no event is scheduled.

4.2 | Predicting future people density

The aim of this task was to create a predictive model able to determine people density in an ACE of Emilia-Romagna in the next 24 h, based on the number of people of the previous day in the same ACE and in the neighbouring ones (see Figure 3).

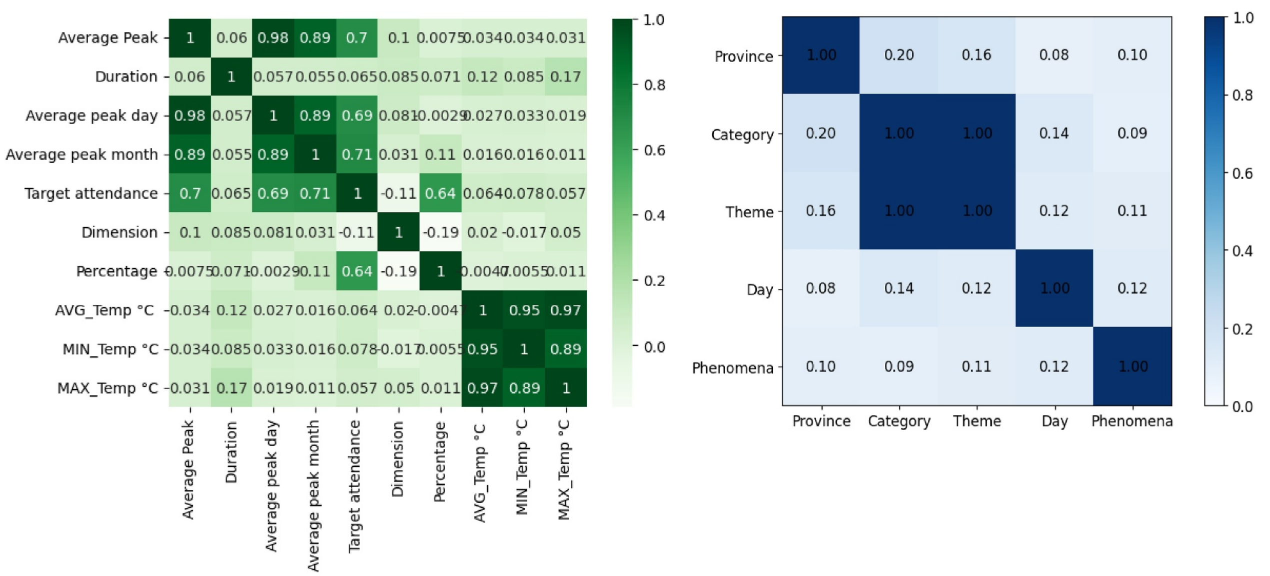

FIGURE 2 Correlation matrices of numeric (left) and nominal (right) attributes.

TABLE 3 Results of the Random Forest classifier on the test set for predicting the class of increase of visitors to a given event in terms of accuracy, precision, recall and F1-score.

Random Forest																				
	Accuracy					Precision					Recall					F1-score				
Class	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4
	0.83	0.83	0.81	0.78	0.79															
[0–25]%						0.84	0.84	0.84	0.81	0.82	0.87	0.87	0.83	0.82	0.84	0.86	0.86	0.84	0.82	0.83
(25–50]%						0.65	0.69	0.62	0.70	0.69	0.47	0.59	0.63	0.53	0.51	0.50	0.64	0.63	0.60	0.59
(50–75]%						0.83	0.83	0.83	0.67	1.00	0.45	0.56	0.56	0.22	0.56	0.59	0.67	0.67	0.33	0.71
(75–100]%						0.00	0.00	0.33	0.25	0.25	0.00	0.00	0.17	0.17	0.17	0.00	0.00	0.22	0.20	0.20
>100%						0.83	0.83	0.80	0.75	0.77	0.85	0.84	0.83	0.79	0.79	0.84	0.84	0.82	0.77	0.78

Note: "Orig." means original dataset, "VI" means the ith variant of the dataset.

TABLE 4 Results of the Extra Trees classifier on the test set for predicting the class of increase of visitors to a given event in terms of accuracy, precision, recall and F1-score.

Extra Trees																				
	Accuracy					Precision					Recall					F1-score				
Class	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4
	0.83	0.85	0.76	0.80	0.76															
[0–25]%						0.86	0.87	0.80	0.84	0.80	0.85	0.87	0.80	0.82	0.80	0.85	0.87	0.80	0.83	0.80
(25–50]%						0.64	0.66	0.63	0.63	0.60	0.59	0.67	0.59	0.65	0.59	0.62	0.67	0.61	0.64	0.60
(50–75]%						0.71	0.62	0.60	0.62	0.60	0.56	0.56	0.33	0.56	0.33	0.63	0.59	0.43	0.59	0.43
(75–100]%						0.00	0.40	0.22	0.20	0.18	0.00	0.33	0.33	0.17	0.33	0.00	0.36	0.27	0.18	0.24
>100%						0.82	0.85	0.74	0.79	0.74	0.85	0.86	0.75	0.81	0.74	0.84	0.86	0.75	0.80	0.74

Note: "Orig." means original dataset, "VI" means the ith variant of the dataset.

TABLE 5 Results of the SVM classifier on the test set for predicting the class of increase of visitors to a given event in terms of accuracy, precision, recall and F1-score.

SVM																				
	Accuracy					Precision					Recall					F1-score				
Class	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4
	0.634	0.74	0.73	0.74	0.73															
[0–25]%						0.55	0.78	0.76	0.74	0.81	0.51	0.81	0.81	0.81	0.83	0.53	0.80	0.78	0.77	0.82
(25–50]%						0.58	0.59	0.56	0.61	0.64	0.51	0.59	0.57	0.61	0.73	0.55	0.59	0.57	0.61	0.69
(50–75]%						0.57	0.71	0.57	0.75	1.00	0.61	0.56	0.44	0.33	0.33	0.59	0.63	0.50	0.46	0.50
(75–100]%						0.60	0.00	0.00	0.00	0.25	0.55	0.00	0.00	0.00	0.17	0.57	0.00	0.00	0.00	0.20
>100%						0.71	0.73	0.73	0.72	0.76	0.80	0.69	0.68	0.64	0.74	0.76	0.71	0.71	0.68	0.75

Note: "Orig." means original dataset, "VI" means the ith variant of the dataset.

In this case we employed only the mobile phone data. The following regression models were tested: Random Forest, Decision trees, Extra Trees, SVR, MLP, GBDT and AdaBoost, and six models were built in each case for predicting density after 1, 2, 4, 8, 12 h and after 24 h.

The dataset underwent a series of preprocessing operations. The result was a matrix 58696 × 432 where each row represents a 24-h time series of people presence in a given census area (506 ACE × 116 available days) and each column the number of people sampled every hour for every attribute (18 attributes × 24 h). The 18 attributes are the ones describing the mobile phone users. Finally, 1728 columns were added representing the average, maximum, minimum and total value of each attribute of the neighbouring ACE in the last 24 h.

TABLE 6 Results of the MLP classifier on the test set for predicting the class of increase of visitors to a given event in terms of accuracy, precision, recall and F1-score.

MLP	Accuracy					Precision					Recall					F1-score				
Class	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4
	0.7	0.52	0.58	0.52	0.56															
[0–25]%						0.74	0.57	0.62	0.60	0.62	0.81	0.96	0.69	0.50	0.64	0.77	0.72	0.65	0.55	0.63
(25–50]%						0.49	0.37	0.32	0.00	0.37	0.42	0.31	0.22	0.00	0.21	0.45	0.34	0.26	0.00	0.26
(50–75]%						0.49	0.31	0.37	0.50	0.46	0.39	0.89	0.43	0.24	0.39	0.43	0.46	0.40	0.33	0.42
(75–100]%						0.67	0.00	0.25	0.00	0.00	0.31	0.00	0.08	0.00	0.00	0.42	0.00	0.12	0.00	0.00
>100%						0.74	0.40	0.63	0.48	0.55	0.73	0.01	0.60	0.72	0.61	0.73	0.01	0.62	0.57	0.58

Note: "Orig." means original dataset, "VI" means the ith variant of the dataset.

TABLE 7 Results of the Gradient Boosted Decision Tree classifier on the test set for predicting the class of increase of visitors to a given event in terms of accuracy, precision, recall and F1-score.

Gradient Boosted Decision Tree	Accuracy					Precision					Recall					F1-score				
Class	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4
	0.85	0.86	0.82	0.75	0.77															
[0–25]%						0.88	0.89	0.86	0.79	0.82	0.86	0.87	0.84	0.84	0.79	0.87	0.88	0.85	0.85	0.80
(25–50]%						0.74	0.69	0.69	0.62	0.62	0.69	0.69	0.73	0.73	0.69	0.72	0.69	0.71	0.71	0.65
(50–75]%						0.60	0.67	0.62	0.29	0.50	0.33	0.44	0.56	0.56	0.44	0.43	0.53	0.59	0.59	0.47
(75–100]%						0.00	0.00	0.33	0.18	0.20	0.00	0.00	0.17	0.17	0.17	0.00	0.00	0.22	0.22	0.18
>100%						0.83	0.85	0.80	0.74	0.74	0.90	0.89	0.83	0.83	0.77	0.86	0.87	0.81	0.81	0.75

Note: "Orig." means original dataset, "VI" means the ith variant of the dataset.

TABLE 8 Results of the AdaBoost classifier on the test set for predicting the class of increase of visitors to a given event in terms of accuracy, precision, recall and F1-score.

AdaBoost	Accuracy					Precision					Recall					F1-score				
Class	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4	Orig.	V1	V2	V3	V4
	0.84	0.83	0.81	0.77	0.76															
[0–25]%						0.86	0.87	0.85	0.81	0.81	0.87	0.85	0.84	0.80	0.81	0.87	0.86	0.85	0.81	0.81
(25–50]%						0.69	0.64	0.63	0.64	0.58	0.63	0.71	0.69	0.61	0.61	0.66	0.67	0.66	0.62	0.59
(50–75]%						0.57	0.67	0.67	0.60	0.50	0.44	0.44	0.44	0.33	0.33	0.50	0.53	0.53	0.43	0.40
(75–100]%						0.00	0.33	0.14	0.14	0.25	0.00	0.17	0.17	0.17	0.33	0.00	0.22	0.15	0.15	0.29
>100%						0.84	0.82	0.80	0.74	0.74	0.85	0.85	0.81	0.77	0.74	0.84	0.83	0.81	0.76	0.74

Note: "Orig." means original dataset, "VI" means the ith variant of the dataset.

Note that predicting density in an hour, 2 h etc. means that we are predicting at the end of the hour (59th minute) or of the 2 hours (119th minute). Moreover, we chose to sample hourly instead of every 15 min, as in the original mobile data because, in spite of a reduction in computational time, there was no significant decrease in custom accuracy. Finally, the length of the input time series (24 h) was chosen after a preliminary test in which the input size of the time series was varied among the

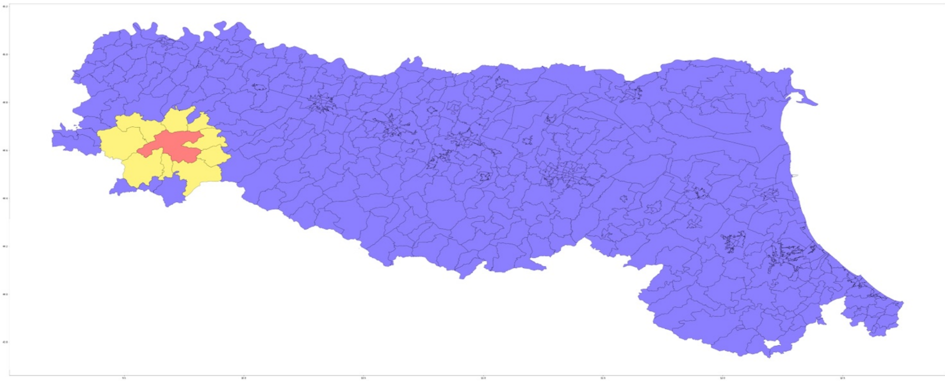

FIGURE 3 Representation of the target ACE (in red) and the adjacent ACEs (in yellow).

following values: 48, 24, 12, 8, 4, 2, 1 h. By calculating the corresponding accuracy on the test set of a Random forest model in predicting the next 1, 2, 4, 8, 12, and 24 h, we found that a 24-h time series represented a fair compromise in predicting the next 24 h.

For testing the performance several error metrics were considered: Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and a custom accuracy. Custom accuracy, once a threshold is set, allows one to determine the total number of examples that the model is able to predict with "sufficient" accuracy. The threshold was set to 0.05, that is, a prediction is considered correct if the predicted value deviates less than 5% from reality. Other values of custom accuracy tested are 1% and 10%, however they showed to be respectively too strict and too loose.

The optimal hyper-parameters for DTs, RFs, and ETs were determined through a grid search method with a 4-fold cross validation. The training process involved using 75% of the dataset, while the remaining 25% was reserved for testing. The best performing model among DT, ET and RF was found to be Random Forest (with hyper-parameters shown in Table 9, where the values marked with (*) are values that depend on the hour of the prediction), so we present here the results only for this case.

The optimal hyper-parameters for SVRs were determined through a randomised search method with a 4-fold cross validation. The best hyper-parameters found were $kernel = rbf$, $C = 33.3$, $\gamma = auto$ (equal to 1/number of features), $epsilon = 0.0332$.

As regards neural networks, AutoKeras was applied on the training data with all the default settings except for max_trials (set to 20), and tuner (set to Bayesian). For each prediction (1–24 h), 20 neural networks were learnt and tested: each one was obtained with $epochs = 300$, $batch\_size = 256$, $validation\_split = 0.25$, Adam optimiser with a learning rate of $1e-5$. Dropout was never used. We found out architectures composed of three hidden layers except for the 24-h prediction, where we obtained a 1-hidden layer network. All the hidden layers' neurons apply the ReLU activation function. All networks have an output layer with one single neuron which returns the real value of the prediction of the density. The six neural networks were then validated on the test data.

TABLE 9 Best hyper-parameters for tree-based models for the prediction of future people density. The values marked with (*) are values that depend on the hour of prediction.

Hyper-parameter	RF	GBDT	AdaBoost
max _*depth*	20	10/20 (*)	20/25 (*)
min _*samples_leaf*	1/10/20 (*)	10/20 (*)	One-fifth (*)
max _*features*	500	100	100
n_estimators	100	100/120 (*)	100/120 (*)
estimator_criterion			*gini*

The optimal hyper-parameters for GBDTs and AdaBoost were determined through a grid search method with a 4-fold cross validation, obtaining the values reported in Table 9.

4.2.1 | Variants of the original experiment

A second version of the dataset was created by exploiting the knowledge of event locations (geographical coordinates) and adding an attribute indicating the presence or not of a public event in the ACE for which we were predicting the density. Experiments were repeated but no model learnt from this data was able to increase the custom accuracy on the test set. This could be due to the specific period under consideration, in which the high influx of people to coastal areas of the Region may not have been due to events but to tourism in general. So this second version was not considered anymore.

In order to understand the contribution of features to the model performance, we generated a correlation matrix[4] and we searched for the features with a low correlation (<0.1) with the target attribute (total people presence in a given ACE every day), but we did not find any. Secondly, we searched for high-correlated attributes (correlation ≥0.95): we selected the number of people with regular contracts and the number of women to be dropped as being highly correlated with the number of Italians, then we selected the number of foreigners

[4]Due to the size of the dataset, the matrix cannot be included in the paper.

to be dropped as we discovered a redundant column for this information. We generated one variant (called 'Variant 1' in the following) of the original dataset according to this analysis, by dropping altogether the above mentioned attributes.

In addition, the original dataset was subjected to a feature analysis by exploiting the "feature importance" option available when learning tree-based models (RFs, GBDTs, AdaBoost). Detailed information can be found in Appendix A.2, where we reported the feature rankings only for prediction in the next hour, in the next fourth hour and in the next 12th hour as an example, as the rankings related to the other predictions were the same. GBDTs and AdaBoost models give more importance to features representing the number of people split according to their age, while RFs use other attributes.

Finally, when predicting the density in the next 8 h with RFs, for instance, the attributes related to the neighbouring ACEs ranked 80th and beyond. This was confirmed by repeating the experiments with RFs and comparing custom accuracies, that were essentially unaffected. For this reason both the original dataset and its variant do not contain those attributes.

We performed the hyper-parameter search as explained in the previous section. For the RF, GBDT and AdaBoost models learnt from the dataset variant we obtained the same best hyper-parameters.

For SVRs, we obtained the same kernel (rbf), tha same value for γ, values for C in the range $[2.5 - 31.6]$ and values for *epsilon* in the range $[0.012 - 0.039]$.

Results in terms of accuracy, MAE and RMSE on the test set are shown in Tables 10 to 12 and they are graphically represented in Figures 4 and 5.

5 | DISCUSSION

5.1 | Quantitative analysis of the results

Results for the classification task show that, thanks to the new amount of data gathered with respect to preliminary tests [3], model accuracy increases from 0.75 to 0.83 with RF, and from 0.70 to 0.85 with ETs; the two boosting models are able to increase accuracy to 0.84 (AdaBoost) and 0.86 (GBDT). Overall, the best performance is obtained by GBDTs on the first variant (V1) of the dataset, being able to classify 86% of increases of people due to events correctly. Note that the percentage increases (classes) are different with respect to the previous work: several configurations were tested here, and the split reported here is the one that achieved the best results. Moreover, w.r.t. [3], we were able to increase the recall for the +(50–75]% class (with all classifiers). Unfortunately the +(75–100]% class is heavily under-represented in the dataset and this justifies the values close to 0. We could successfully analyse the performance of the events with more than 100% increase, too: note that, in almost all cases except MLP, performance for this class are >0.74, so in case of a large increase of people the models' behaviour is good.

MLP are clearly the worst performing model.

By comparing performance between the 4 variants of the dataset, all tree-based models (RF, ET, GBDT and AdaBoost) reach the highest values for all metrics on Variant 1, meaning that the 'duration' feature is not relevant for the prediction, while the temperatures and the average peak per day should be

TABLE 11 Results of MLP on the test set for predicting people density in the next 1, 2, 4, 8, 12, 24 h in terms of accuracy and standard error metrics for regression.

MLP						
	MAE		RMSE		Accuracy	
Next hours	Orig.	V1	Orig.	V1	Orig.	V1
1	100.713	111.411	205.942	226.249	0.948	0.935
2	151.507	180.926	322.559	327.365	0.900	0.833
4	206.101	221.421	431.26	476.657	0.834	0.814
8	268.074	337.508	553.991	621.129	0.748	0.640
12	305.493	319.056	612.572	661.904	0.700	0.684
24	367.449	346.491	860.975	792.307	0.659	0.668

Note: "Orig." means original dataset, "V1" means Variant one of the dataset.

TABLE 10 Results of the Random Forest and SVR models on the test set for predicting people density in the next 1, 2, 4, 8, 12, 24 h in terms of accuracy and standard error metrics for regression.

Random Forest							SVR						
	MAE		RMSE		Accuracy			MAE		RMSE		Accuracy	
Next hours	Orig.	V1	Orig.	V1	Orig.	V1	Next hours	Orig.	V1	Orig.	V1	Orig.	V1
1	239.31	102.89	1596.63	402.72	0.92	0.97	1	0.046	0.090	0.202	0.166	0.825	0.731
2	303.54	176.91	1642.34	541.27	0.83	0.91	2	0.047	0.132	0.202	0.195	0.820	0.737
4	521.81	262.82	2561.84	717.45	0.69	0.81	4	0.050	0.096	0.200	0.185	0.804	0.739
8	573.93	357.18	2627.13	1006.68	0.65	0.73	8	0.055	0.207	0.201	0.275	0.796	0.710
12	589.67	374.20	2727.55	1022.79	0.65	0.71	12	0.057	0.124	0.199	0.202	0.792	0.719
24	712.23	376.02	3183	1392955.44	0.62	0.71	24	0.059	0.145	0.214	0.225	0.779	0.710

Note: "Orig." means original dataset, "V1" means Variant one of the dataset.

TABLE 12 Results of GBDTs and AdaBoost models on the test set for predicting people density in the next 1, 2, 4, 8, 12, 24 h in terms of accuracy and standard error metrics for regression.

GBDTs							AdaBoost						
	MAE		RMSE		Accuracy			MAE		RMSE		Accuracy	
Next hours	Orig.	V1	Orig.	V1	Orig.	V1	Next hours	Orig.	V1	Orig.	V1	Orig.	V1
1	134.40	144.69	389.06	383.70	0.944	0.935	1	138.97	147.98	414.97	418.68	0.944	0.937
2	186.21	198.85	528.82	507.27	0.893	0.867	2	195.00	199.15	563.79	552.33	0.883	0.878
4	258.50	259.97	657.52	642.14	0.795	0.793	4	268.92	273.64	732.13	700.67	0.795	0.797
8	326.88	332.30	883.40	909.70	0.731	0.725	8	346.21	346.70	968.85	969.97	0.735	0.737
12	353.38	353.95	910.32	911.83	0.710	0.707	12	365.58	365.11	1003.85	987.11	0.720	0.720
24	384.30	389.02	1132.93	1132.45	0.680	0.675	24	379.13	380.05	1148.14	1145.73	0.706	0.706

Note: "Orig." means original dataset, "V1" means Variant one of the dataset.

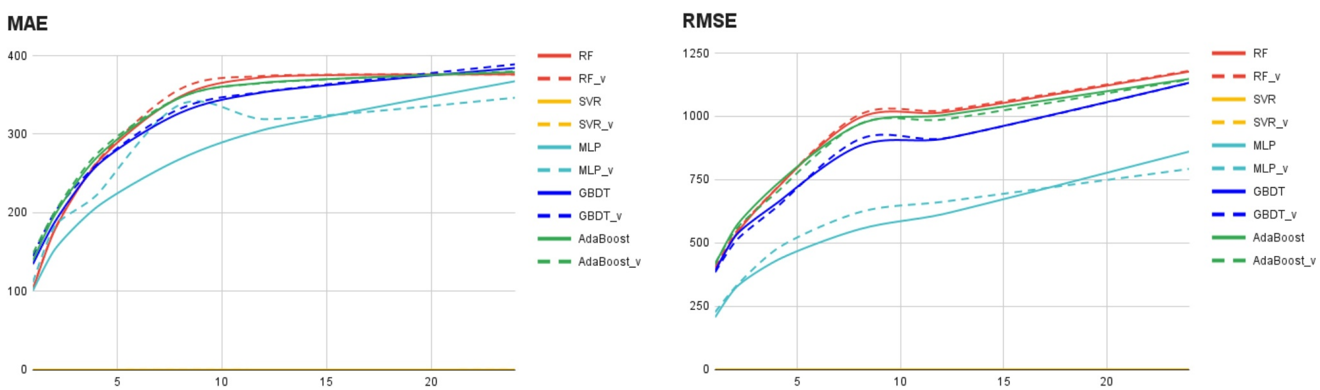

FIGURE 4 Trend of the errors for all regression models. "_v" means the model learnt from Variant one of the dataset.

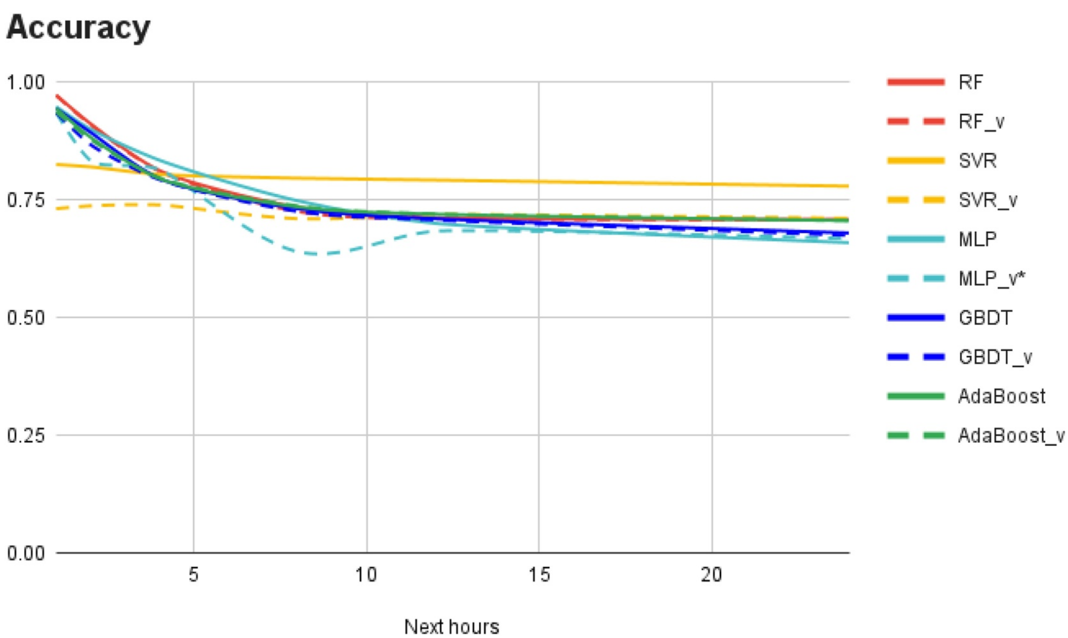

FIGURE 5 Trend of accuracy for all regression models. "_v" means the model learnt from Variant one of the dataset.

kept in the data. This is also confirmed by Tables in Appendix A.1 showing feature rankings for these models.

Results for the regression task show that, as expected, accuracy decreases as the forecast moves further away from the time when data on people density was gathered, hitting its lowest point after 24 h. By considering the overall results (both accuracy and errors), RFs and MLP are most accurate in the next 2 h but RFs suffer from high MAE and RMSE, while SVRs exhibit both a more gradual decline in accuracy and significantly low errors. However, this comes with a higher computational cost, as the implementation of SVR based on the class has a fit time complexity more than quadratic with the number of samples. On an Intel i7-1255U, 1.70 GHz and 16 GB RAM machine, each model took the following learning times: 70 min (next hour = 1); 133 min (next hours = 2); 255 min (next hours = 4); 500 min (next hours = 8); 741 min (next hours = 12); 1446 min (next hours = 24).

Given an input of at least 24 h with hourly sampling, we are able to get a prediction of people attendance in the next 24 h—with SVRs—with accuracy between 78% and 82% and very limited error, or an accuracy even greater than 94% in the next 2 h with AdaBoost (but with higher error). We also demonstrated that only the information coming from the same geographical area (ACE) is relevant to the prediction, with an advantage in terms of reducing learning time.

By comparing performance between the two variants of the dataset, only RFs, MLP and SVR show a clear difference between the two (but they do not agree on the best performing dataset), while the boosting ensemble models do not show a relevant difference. In this case the original dataset could be kept.

5.2 | Qualitative analysis

The results obtained help to identify "when and where" people flux is expected to increase. Therefore the models outcomes can be used for improving transportation planning, such as regulating the number of additional buses and shuttles (e.g. from the airport to international events or from downtown to specific event locations), for modification of the flow of traffic, for automatic messages of traffic alerts to the smartphones of the residents and for the optimisation of commercial services, such as modification of opening times or forecast of necessary additional staff units.

The prediction of the expected increase of people to public events raises some issues related to the generation of a satisfactory database of events. In our opinion some solutions need to be proposed for future developments. In fact, experimental results demonstrated that a more balanced dataset of events ought to be available, with a sufficient number of events collected for the different sizes (1–6). The main critical issues concern:

Completeness. This refers to the goal of retrieving all the big events in the Region. Notwithstanding the high number of the selected websites and of the different web sources, some events can be outside of the official dissemination channels: for example, unofficial, and not authorised events, private events, political events (usually out of the standard circuits), etc. Unofficial websites or manual adjustments can help to complete the missing data. This issue could be solved with a central hub, at least at regional level, where all big events (e.g. involving more than 500 people) are registered and collected, with mandatory communication by the organisers; local administrations could be involved in retrieving data from the responsible authorities (e.g. when an event is authorised by the local municipality).

People attendance estimation. The issue of assigning a dimension to an event remains crucial, since it would be important to select and identify the most relevant events in order to correlate them to the people's flow and mobility. Minor events have to be identified in a better way and excluded from the analysis (e.g. those with ≤200 people), in order to focus on the major events causing a stronger impact on people flows. This issue could be improved using together one or more of the following strategies:

- Top-down approach. The information on the dimension of the event is provided (or estimated) by the organiser. This would involve a system where the information is provided a priori (if mobility and access prediction is foreseen) or a posteriori (if only analyses are provided). At the moment this information is not provided automatically but it would be advisable for the future, at least for the most important events;
- Historical approach: for past big events access should be given to the reported number of participants. This of course can be applied to events that have been organised only in previous years, but the database can increase with time with much more precise attendance values;
- Location capacity. A database of all the locations' capacity in the Region should be implemented (e.g. capacity of a theatre or of a fair). This would need an extensive and very long review of thousands of places. The assignment of the dimensions according to the scale one to six and to the location attributes is not resolutive (e.g. events in the same square of Bologna can be very crowded or not), but can cover most of the events;
- Flow peak analysis: an analysis could be performed on the flow peak data (e.g. peak in traffic or Wi-Fi data), in order to search and identify big events at that time and space. Peaks have to be discriminated against the traffic due, for example, to the presence of highways, accidents, train stations and so on.

Event information. The information retrieved about events is often incomplete and some fields have to be added to partially fulfil the abstract event model (15 entries out of 36 predefined entries have been generally detected). Anyhow, a minimum set of information is necessary (location, date, and full description).

Format of the data source. Every website has a different structure and, moreover, the metadata used by the different websites can be changed by the web owner, compelling updates of the event retrieval software. A universal, accepted event format (common standard) should be fixed, spread, and adopted by the different subjects/organisers, not hindering their business goal, but helping a more complete information and tourist management analysis.

Cancelation of events. Also, the cancelation of the planned events has to be monitored. Events can be disseminated, advertised but cancelled at the last moment (e.g. lockdown restrictions due to Covid-19, weather conditions etc.). This has to be verified by the developed software.

Other limitations of the models of this project concern the *mobile phone network data.* This data was bought from the phone company with budget limitations so only a limited amount of data was available. Accessibility, at affordable costs, of continuous mobile phone network data would be necessary, in order to monitor the correlation with the event attendance and to improve and validate the prediction models. This is true both for the time and spatial variables of network data:

- *Time.* For this project, analysis of people mobility was limited to 4 months. An extension to different years would help to analyse also seasonal effects in mobility;
- *Space.* The areas delimited by the ACE, in which the data are analysed, are probably too big (about 1 km [2] in downtown Bologna, the regional capital) and can still mix areas of higher people density with those of lower people density. Moreover, the ACE might include some transit areas, such as highways, railways, airports, which could be not connected to the local mobility (and therefore to the analysed local events). This would represent noise for our prediction tasks. Mobile data with finer granularity are available but at higher costs.

A broader mobile phone availability would allow to extend the input dataset for predicting the increase of people to public events, obtaining higher performance (accuracy, precision, and recall) for every class of increase.

The last important aspect to consider is that all the systems we used are not inherently explainable. This means the resulting models are black boxes, returning regression/classification results without revealing how they compute those results. This can be a disadvantage, as the user must rely solely on the fact that the models are well-trained. However, this choice was made for two reasons. The first concerns how the results are used. These models have been developed in the context of the development of a PSS able to provide predictions that could be exploited by local administrations for traffic and transportation management, or for a greater logistic efficiency in personnel and visitors management at events for the event organisers. The resulting dashboard of the PSS encapsulates the models presented in this paper to help decision-making regarding, for example, how many law enforcement officers to use, when and where to deploy them in case of expected high flows of people. Therefore, due to the project's initial goals, explanations for individual results are not currently necessary. The second reason is more tightly intertwined with the data we used. Many different explainable systems rely on a set of rules and predicates that allow for constructing the logic behind the system's response. However, this approach can limit the creation of correlations between different data features, as these correlations may not be explicitly encoded in the rules. In our case, numerical data would have been difficult to translate into a format suitable for explainable models. Other explainable systems often have lower expressive power compared to the models we used. Consider, for example, simple models like Decision Trees, which lack the capacity to accurately model the data as discussed in Section 4, where DTs performance on regression are worse than that of RFs. Ensemble methods like RFs, GBDTs and AdaBoost have got higher capacity, perform better but at the cost of explainability. Therefore, since this feature was not explicitly required, we selected ML models by prioritising capacity, accuracy and suitability to our data over explainability.

6 | CONCLUSIONS AND FUTURE WORK

This study has demonstrated the critical role of integrating Artificial Intelligence into the management of public event mobility and safety. Our comprehensive analysis, utilising datasets derived from mobile phone usage and public event information, has underscored the potential of AI and machine learning techniques in transforming urban management and enhancing public safety. Firstly, the successful deployment of predictive models to forecast people density or event attendance in response to public events has proven highly effective. These models facilitate better urban planning and resource allocation, thereby significantly reducing potential public safety risks. The ability to predict increases in human density with high accuracy is pivotal, especially during large-scale events where the risk of incidents is elevated.

The outcomes can be used also for improving transportation planning, for modification of the flow of traffic, for automatic messages of traffic alerts to the residents' smartphones and for the optimisation of commercial services. In general, the aim of this work is for the models to be integrated into a dashboard of a Policy Support System to help decision-making processes regarding, for example, how many law enforcement officers to use, when and where to deploy them in case of expected high flows of people. To do this, the dashboard generates alerts on potential future problems providing to city managers a comprehensive overview of the city's population, and aiding transportation and security management.

However, it should be noted that the results could certainly be improved by providing more extensive data to the algorithms, where "extensive" refers to both temporal and spatial dimensions.

Moreover, our research highlights the advantages of smart cities leveraging digital and telecommunication technologies to improve their responsiveness. The implementation of AI-driven tools enables city planners and policymakers to not only react in real time to unfolding events but also proactively plan for potential scenarios, thus optimising overall city operations. Additionally, the findings from this study contribute to confirming the efficacy of various machine learning algorithms in handling complex urban data.

As cities continue to evolve, the strategic application of AI and ML will be crucial in addressing the dynamic challenges of urbanisation. Moving forward, it will be essential to continue refining these technologies and to foster an environment where innovation in public safety and urban management can thrive, thereby ensuring sustainable and resilient urban futures. In fact, the POLIS-EYE project, which serves as the foundation for this work, has led to the development of a recent new project, financed by the Emilia-Romagna Region, named "Support System for Sustainable Smart Cities" (S4C).[5] This new initiative focuses on providing operational simulation and optimisation tools to enhance planning actions for mobility services, promoting systemic *sustainability*, and reducing the climate change impacts associated with tourist flows.

AUTHOR CONTRIBUTIONS

Elena Bellodi: Conceptualization; Formal analysis; Investigation; Methodology; Supervision; Validation; Visualization; Writing - original draft; Writing - review and editing. **Riccardo Zese**: Conceptualization; Data curation; Formal analysis; Investigation; Methodology; Validation; Visualization; Writing - original draft; Writing - review and editing. **Carlo Petrovich**: Conceptualization; Data curation; Formal analysis; Investigation; Methodology; Resources; Software; Validation; Visualization; Writing - original draft; Writing - review and editing. **Angelo Frascella**: Conceptualization; Data curation; Formal analysis; Investigation; Methodology; Resources; Validation; Visualization; Writing - original draft; Writing review & editing. **Francesco Bertasi**: Conceptualization; Data curation; Investigation; Methodology.

ACKNOWLEDGEMENTS

This research was funded by the Fondo di Sviluppo e Coesione (FSC) of Regione Emilia-Romagna within the context of the POR FESR 2014–2020 ASSE 1 AZIONE 1.2.2 (CUP E21F18000200007) project "POLIcy Support systEm for smart citY data governancE (POLIS-EYE)". A special thanks to Mr. Mattia Buzzoni, Stefano Chiogna, Riccardo Morelli, Sara Varesco and Mattia Cavicchio for providing most of the experimental results.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study were acquired from TIM S.p.A. and are not publicly available.

ORCID

Elena Bellodi https://orcid.org/0000-0002-3717-3779

REFERENCES

1. Acharjya, D.P., Ahmed, P.K.: A survey on big data analytics: challenges, open ResearFch issues and tools. Int. J. Adv. Comput. Sci. Appl. 7(2) (2016). https://doi.org/10.14569/IJACSA.2016.070267
2. Mohanty, S., Choppali, U., Kougianos, E.: Everything you wanted to know about smart cities. IEEE Consumer Elect. Magazine 5(3), 60–70 (2016). https://doi.org/10.1109/MCE.2016.2556879
3. Bellodi, E., Zese, R., Bertasi, F.: Machine Learning in a Policy Support System for Smart Tourism Management, pp. 457–472. Springer-Verlag, Berlin (2021)
4. Seravalli, A., et al.: Towards smart cities for tourism: the POLIS-EYE project. In: 2022 IEEE International Smart Cities Conference (ISC2), pp. 1–7 (2022)
5. .https://imprese.regione.emilia-romagna.it/internazionalizzazione/temi/sistema-fieristico-regionale/rapporto-fiere-e-r-2019.pdf/@@download/file/Rapporto%20Fiere%20E-R%202019.pdf;
6. Lamsfus, C., et al.: Smart tourism destinations: an extended conception of smart cities focusing on human mobility. In: Information and Communication Technologies in Tourism 2015, pp. 363–375. Springer (2015)
7. Zhao, Z., et al.: Mobile crowd location prediction with hybrid features using ensemble learning. Future Generat. Comput. Syst. 110, 556–571 (2020). https://doi.org/10.1016/j.future.2018.06.025
8. Karbovskii, V., et al.: Ensemble learning for large-scale crowd flow prediction. Eng. Appl. Artif. Intell. 106, 104469 (2021). https://doi.org/10.1016/j.engappai.2021.104469
9. Baranwal, A., et al.: Managing the earth's biggest mass gathering event and wash conditions: maha kumbh mela (India). PLoS currents 7 (2015). https://doi.org/10.1371/currents.dis.e8b3053f40e774e7e3fdbe1bb50a130d
10. Salehi Rizi, F., Granitzer, M.: Predicting event attendance exploring social influence. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 2131–2134 (2019)
11. Wu, C., et al.: Ensemble learning for crowd flows prediction on campus. In: Qiu, M. (ed.) Smart Computing and Communication, pp. 103–113. Springer International Publishing (2018)
12. Li, Y., Cao, H.: Prediction for tourism flow based on LSTM neural network. Procedia Comput. Sci. 129, 277–283 (2018). https://doi.org/10.1016/j.procs.2018.03.076
13. Xu, Z., et al.: A graph spatial-temporal model for predicting population density of key areas. Comput. Electr. Eng. 93, 107235 (2021). https://doi.org/10.1016/j.compeleceng.2021.107235
14. Yang, J., et al.: The spatiotemporal prediction method of urban population density distribution through behaviour environment interaction agent model. Sci. Rep. 13(1), 5821 (2023). https://doi.org/10.1038/s41598-023-32529-0
15. Fan, Z., et al.: Online deep ensemble learning for predicting citywide human mobility. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 2(3), 1–21 (2018). https://doi.org/10.1145/3264915
16. Pulselli, R.M., et al.: Computing urban mobile landscapes through monitoring population density based on cell-phone chatting. Int. J. Des. Nat. Ecodyn. 3(2), 121–134 (2008). https://doi.org/10.2495/DNE-V3-N2-121-134
17. Li, Q., et al.: Real-time monitoring and forecast of active population density using mobile phone data. In: Proceedings of National Co Nference on Big Data Technology and Applications, pp. 116–129 (2016)
18. Zhang, G., et al.: Fine-grained, spatial, and temporal analysis, and prediction of mobile phone users' distributions based upon a convolution long short-term model. Sensors 19(9), 2156 (2019). https://doi.org/10.3390/s19092156
19. Feng, J., et al.: A bimodal model to estimate dynamic metropolitan population by mobile phone data. Sensors 18(10), 3431 (2018). https://doi.org/10.3390/s18103431
20. Anderson, W., et al.: Methods for estimating population density in data-limited areas: evaluating regression and tree-based models in Peru. PLoS One 9(7), e100037 (2014). https://doi.org/10.1371/journal.pone.0100037
21. Lee, K.S., et al.: Urban spatiotemporal analysis using mobile phone data: case study of medium- and large-sized Korean cities. Habitat Int. 73, 6–15 (2018). https://doi.org/10.1016/j.habitatint.2017.12.010
22. Gariazzo, A., Pelliccioni, A.: A multi-city urban population mobility study using mobile phone traffic data. Appl. Spatial Analysis. 12(4), 753–771 (2019). https://doi.org/10.1007/s12061-018-9268-4
23. Grossman, I., et al.: Can machine learning improve small area population forecasts? A forecast combination approach. Comput. Environ. Urban Syst. 95, 101806 (2022). https://doi.org/10.1016/j.compenvurbsys.2022.101806

[5] https://www.s4c-project.it/.

24. Şahinarslan, F.V., Tekin, A.T., Çebi, F.: Application of machine learning algorithms for population forecasting. Int. J. Data Sci. 6(4), 257–270 (2021). https://doi.org/10.1504/IJDS.2021.122770
25. Casali, Y., Aydin, N.Y., Comes, T.: Machine learning for spatial analyses in urban areas: a scoping review. Sustain. Cities Soc. 85, 104050 (2022). https://doi.org/10.1016/j.scs.2022.104050
26. Getz, D.: Event tourism: definition, evolution, and research. Tourism Manag. 29(3), 403–428 (2008). https://doi.org/10.1016/j.tourman.2007.07.017
27. .https://github.com/italia/daf-ontologie-vocabolari-controllati;
28. .http://ontopa.opencontent.it/API-Vocabolari-controllati/Tipologie-di-even%20ti-pubblici;
29. .https://content-classes.readthedocs.io/it/latest/docs/Eventi%200(CPEV-AP_IT).html;
30. Zuccalà, M., Celino, I.: Fostering innovation through coopetition: the E015 digital Ecosystem. In: Cimiano, P., et al. (eds.) Engineering The Web In the Big Data EraSpringer, pp. 625–628. International Publishing (2015)
31. Brutti, A., et al.: Interoperability in the smart city: a semantic approach for merging flexibility with strictness. In: Workshop SSC 2018: Fourth IEEE International Workshop on Sensors and Smart Cities, Special Session in Proceedings 2018 IEEE International Conference on Smart Computing 18-20 June 2018 Taormina, Sicily, Italy (2018). IEEE Catalog Number: CFP1816Z-ART. IEEE
32. Brutti, A., et al.: Technical Report | DELIVERABLE M5.1 | O5.2 Step 5 - Data Flow Modeling (2021)
33. https://smartcityplatform.enea.it/UDWebLibrary/en/urbandataset?name=Event%20s;
34. Hastie, T., et al.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, vol. 2. Springer (2009)
35. Breiman, L.: Bagging predictors. Mach. Learn. 24(2), 123–140 (1996). https://doi.org/10.1007/bf00058655
36. Breiman, L.: Random forests. Mach. Learn. 45(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
37. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. 63(1), 3–42 (2006). https://doi.org/10.1007/s10994-006-6226-1
38. Cortes, C., Vapnik, V.: Support vector networks. Mach. Learn. 20(3), 273–297 (1995). https://doi.org/10.1007/bf00994018
39. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. 65(6), 386–408 (1958). https://doi.org/10.1037/h0042519
40. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. 29(5), 1189–1232 (2001). https://doi.org/10.1214/aos/1013203451
41. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55(1), 119–139 (1997). https://doi.org/10.1006/jcss.1997.1504
42. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12(Oct), 2825–2830 (2011)

How to cite this article: Bellodi, E., et al.: Predicting the impact of public events and mobility in Smart Cities. IET Smart Cities. 6(4), 253–275 (2024). https://doi.org/10.1049/smc2.12087

APPENDIX

A.1 | Feature ranking by tree-based models (classification task)

Tables A1–A5 report feature rankings obtained by tree-based models for each variant of the dataset used for the classification task.

1. Original dataset: the dataset as described in Section 4.1 (data about public events).
2. Variant 1: the dataset with the 'Duration' feature dropped.
3. Variant 2: the dataset with the 'Duration' + 'MIN_Temp' + 'MAX_Temp' features dropped.
4. Variant 3: the dataset with the 'Duration' + 'MIN_Temp' + 'MAX_Temp' + 'AVG_Temp' features dropped.
5. Variant 4: the dataset with the 'Duration' + 'MIN_Temp' + 'MAX_Temp' + 'AVG_Temp' + 'Average peak per day' features dropped.

TABLE A1 Feature ranking obtained by Random Forest models for classification. The first column reports the feature ranking from the original dataset, the second column from the first variant.

Feature #	Random Forest (original dataset)	Random Forest (variant 1)
1	**Average peak per month** (0.161793)	**Average peak per month** (0.179325)
2	**Average peak per day** (0.158238)	**Average peak** (0.13867)
3	**MIN_Temp °C** (0.108987)	**Average peak per day** (0.125818)
4	**MAX_Temp °C** (0.102923)	**MIN_Temp °C** (0.114425)
5	**AVG_Temp °C (0.097363)**	**MAX_Temp °C** (0.096583)
6	Duration (0.095935)	AVG_Temp °C (0.092099)
7	Dimension (0.058701)	Dimension (0.050423)
8	Category_FAIRS, EXHIBITION (0.020782)	Day_Saturday (0.016593)
9	Category_SHOWS and SCREENINGS (0.019662)	PHENOMENA_no rain (0.016308)
10	PHENOMENA_no rain (0.018250)	Category_MEETING, DEBATE, ROUNDTABLE (0.016256)
11	Category_MARKETS, EXHIBITIONS MARKET (0.017908)	PHENOMENA_rain (0.016227)

(Continues)

TABLE A1 (Continued)

Feature #	Random Forest (original dataset)	Random Forest (variant 1)
12	Day_Saturday (0.016589)	Category_FAIRS, EXHIBITION (0.016165)
13	Day_Tuesday (0.016023)	Category_MARKETS, EXHIBITIONS MARKET (0.015988)
14	PHENOMENA_rain (0.015512)	Category_SHOWS and SCREENINGS (0.015909)
15	Day_Friday (0.015129)	Day_Tuesday (0.015831)
16	Day_Wednesday (0.012261)	Day_Friday (0.014455)
17	Day_Sunday (0.012045)	Day_Wednesday (0.010791)
18	Day_Thursday (0.011866)	Day_Monday (0.010669)
19	Day_Monday (0.011520)	Day_Sunday (0.010378)
20	Category_MEETING, DEBATE, ROUNDTABLE (0.011468)	Day_Thursday (0.010328)
21	PHENOMENA_rain thunderstorm (0.009437)	PHENOMENA_rain thunderstorm (0.009596)
22	PHENOMENA_rain fog (0.004347)	PHENOMENA_rain fog (0.004547)
23	PHENOMENA_fog (0.002997)	PHENOMENA_fog (0.00257)
24	Category_COMPETITION, RACE, TOURNAMENT, CONTEST and AWARDS (0.000264)	Category_COMPETITION, RACE, TOURNAMENT, CONTEST and AWARDS (0.000046)

Note: The first five features are highlighted in bold.

TABLE A2 Feature ranking obtained by Random Forest models with variants 2, 3, and 4 of the dataset.

Feature #	Random Forest (variant 2)	Random Forest (variant 3)	Random Forest (variant 4)
1	**Average peak per month** (0.186974)	**Average peak** (0.295074)	**Average peak** (0.379781)
2	**AVG_Temp °C** (0.184568)	**Average peak per month** (0.228086)	**Average peak per month** (0.268487)
3	**Average peak** (0.180587)	**Average peak per day** (0.157854)	**Dimension** (0.099092)
4	**Average peak per day** (0.160239)	**Dimension** (0.09281)	**Category_FAIRS, EXHIBITION** (0.030862)
5	**Dimension** (0.056575)	**Category_FAIRS, EXHIBITION** (0.028636)	**Category_MARKETS, EXHIBITIONS MARKET** (0.027207)
6	PHENOMENA_no rain (0.021727)	Category_MARKETS, EXHIBITIONS MARKET (0.026048)	Day_Sunday (0.020364)
7	Category_FAIRS, EXHIBITION (0.021532)	Category_SHOWS and SCREENINGS (0.020554)	Day_Saturday (0.019138)
8	Category_MARKETS, EXHIBITIONS MARKET (0.019106)	Category_MEETING, DEBATE, ROUNDTABLE (0.018821)	Category_MEETING, DEBATE, ROUNDTABLE (0.018141)
9	PHENOMENA_rain (0.017867)	PHENOMENA_no rain (0.015147)	Category_SHOWS and SCREENINGS (0.017669)
10	Category_SHOWS and SCREENINGS (0.016828)	PHENOMENA_rain (0.014655)	PHENOMENA_no rain (0.017631)
11	Day_Friday (0.015607)	Day_Sunday (0.0136)	Day_Tuesday (0.015201)
12	Day_Saturday (0.015187)	Day_Saturday (0.013204)	PHENOMENA_rain (0.014599)
13	Day_Tuesday (0.01456)	PHENOMENA_rain thunderstorm (0.013103)	Day_Wednesday (0.014232)
14	PHENOMENA_rain thunderstorm (0.014175)	Day_Friday (0.011741)	Day_Friday (0.013612)
15	Day_Sunday (0.014138)	Day_Tuesday (0.011702)	PHENOMENA_rain thunderstorm (0.012683)

TABLE A2 (Continued)

Feature #	Random Forest (variant 2)	Random Forest (variant 3)	Random Forest (variant 4)
16	Day_Thursday (0.014014)	Day_Thursday (0.011561)	Day_Thursday (0.012677)
17	Category_MEETING, DEBATE, ROUNDTABLE (0.013863)	Day_Wednesday (0.010975)	Day_Monday (0.008662)
18	Day_Wednesday (0.012594)	Day_Monday (0.006846)	PHENOMENA_fog (0.004767)
19	Day_Monday (0.011663)	PHENOMENA_fog (0.004451)	PHENOMENA_rain fog (0.004225)
20	PHENOMENA_rain fog (0.004061)	PHENOMENA_rain fog (0.003929)	Category_COMP. (0.000972)
21	PHENOMENA_fog (0.003889)	Category_COMP. (0.001205)	
22	Category_COMP. (0.000247)		

TABLE A3 Feature ranking obtained by Extra Trees models for classification. Only the first five features are shown.

#	Original Dataset	Variant 1	Variant 2	Variant 3	Variant 4
1	Avg peak (0.1509)	Avg peak month (0.1628)	Avg per peak (0.2668)	Avg peak (0.2048)	Avg peak (0.3797)
2	Duration (0.1223)	Avg peak (0.1412)	Avg peak month (0.1961)	AVG_Temp °C (0.1874)	Avg peak month (0.2444)
3	Avg peak day (0.1218)	Avg peak day (0.1056)	Avg peak day (0.1611)	Avg peak month (0.1567)	Dimension (0.0658)
4	MIN_Temp (0.1015)	AVG_Temp (0.0953)	Dimension (0.0720)	Avg peak day (0.1263)	Day_Saturday (0.0316)
5	AVG_Temp (0.0982)	MIN_Temp (0.0944)	PHENOMENA_no rain (0.0307)	Dimension (0.0646)	PHENOMENA_no rain (0.0310)

TABLE A4 Feature ranking obtained by GBDT models for classification. Only the first five features are shown.

#	Original Dataset	Variant 1	Variant 2	Variant 3	Variant 4
1	Avg peak (0.2442)	Avg peak month (0.2206)	Avg peak (0.2438)	Avg peak (0.2366)	Avg peak (0.4098)
2	Avg peak day (0.1739)	Avg peak (0.1355)	Avg peak month (0.2421)	Avg peak day (0.2172)	Avg peak month (0.3243)
3	Duration (0.1001)	MIN_Temp (0.1268)	Avg peak day (0.1688)	Avg peak month (0.2110)	Dimension (0.0906)
4	MIN_Temp (0.0983)	Avg peak day (0.1206)	AVG_Temp (0.1403)	Dimension (0.0614)	Category_FAIRS, EXHIBITIONS (0.0285)
5	MAX_Temp (0.0763)	MAX_Temp (0.0981)	Dimension (0.0633)	Day_Friday (0.0513)	Category_SHOWS, SCREENINGS (0.0278)

TABLE A5 Feature ranking obtained by AdaBoost models for classification. Only the first five features are shown.

#	Original Dataset	Variant 1	Variant 2	Variant 3	Variant 4
1	Avg peak day (0.1623)	Avg peak month (0.1570)	AVG_Temp (0.1886)	Avg peak month (0.2678)	Avg peak month (0.3420)
2	Avg peak (0.1277)	Avg peak day (0.1463)	Avg peak month (0.1829)	Avg peak day (0.2189)	Avg peak (0.2740)
3	MIN_Temp (0.1101)	Avg peak (0.1084)	Avg peak day (0.1724)	Avg peak (0.1787)	Dimension (0.0722)
4	MAX_Temp (0.1030)	MIN_Temp (0.1060)	Avg peak (0.1245)	Dimension (0.0654)	Category_SHOWS, SCREEN. (0.0376)
5	AVG_Temp (0.0826)	MAX_Temp (0.0925)	Dimension (0.0606)	PHENOMENA_no rain (0.0346)	PHENOMENA_no rain (0.0374)

A.2 | Feature ranking by tree-based models (regression task)

Tables A6–A8 report feature rankings obtained by tree-based models for each variant of the dataset used for the regression task.

1. Original dataset: the dataset as described in Section 4.2 (data about people density).
2. Variant 1: the dataset with the number of people with regular contracts, the number of women and the number of foreigners features dropped.

TABLE A6 Feature ranking obtained by Random Forest models for regression. Only the first 10 features are shown and for 3 different instants of future prediction (first hour, fourth hour, 12th hour). "Ni" = number of Italians; "Gm" = number of males, "P" = total number of people; "T-X" (with X ∈ [0 − 23] representing the hour) is the sampling time.

#	Original Dataset—1 h	Original Dataset—4 h	Original Dataset—12 h	Variant 1–1 h	Variant 1–4 h	Variant 1–12 h
1	(P, T-0) (0.069724)	(P, T-0) (0.040247)	(Gf, T-11) (0.034901)	(Ni, T-4) (0.041201)	(Gm, T-0) (0.050774)	(Ni, T-12) (0.030415)
2	(Ni, T-1) (0.062184)	(Ni, T-1) (0.039437)	(P, T-9) (0.030836)	(Gm, T-0) (0.034608)	(Ni, T-19) (0.049414)	(Gm, T-9) (0.029352)
3	(Tc, T-0) (0.057684)	(Gm, T-0) (0.033508)	(Tc, T-12) (0.029503)	(Gm, T-1) (0.033814)	(P, T-17) (0.045190)	(Gm, T-12) (0.026223)
4	(Gm, T-0) (0.051491)	(Tc, T-0) (0.032361)	(P, T-8) (0.027825)	(Ni, T-5) (0.028900)	(P, T-19) (0.042646)	(Ni, T-14) (0.022757)
5	(Ni, T-0) (0.049329)	(Ni, T-0) (0.030298)	(Gf, T-10) (0.026336)	(Ni, T-3) (0.027165)	(P, T-18) (0.039632)	(Ni, T-5) (0.021442)
6	(Gm, T-1) (0.045182)	(Gm, T-20) (0.028061)	(P, T-11) (0.023321)	(Gm, T-6) (0.026596)	(P, T-0) (0.039261)	(Ni, T-23) (0.021426)
7	(P, T-1) (0.039286)	(Gm, T-19) (0.027594)	(Gm, T-10) (0.023284)	(P, T-4) (0.025604)	(Ni, T-2) (0.034615)	(Gm, T-15) (0.021079)
8	(P, T-2) (0.033456)	(P, T-19) (0.025991)	(Gm, T-13) (0.023169)	(Ni, T-2) (0.025290)	(P, T-2) (0.032496)	(Ni, T-15) (0.019321)
9	(Gm, T-3) (0.029255)	(Gm, T-2) (0.025846)	(Tc, T-10) (0.023070)	(Gm, T-2) (0.025088)	(Gm, T-1) (0.029948)	(Ni, T-10) (0.018359)
10	(Gf, T-2) (0.028160)	(Gf, T-22) (0.020201)	(Gf, T-8) (0.021928)	(Ni, T-22) (0.024569)	(P, T-20) (0.028642)	(P, T-12) (0.018080)

TABLE A7 Feature ranking obtained by GBDT models for regression. Only the first 10 features are shown and for 3 different instants of future prediction (first hour, fourth hour, 12th hour). "F4" = number of people aged [41 − 50]; "F5" = number of people aged [51 − 60]; "F6" = number of people aged >60; "Gf" = number of females; "T-X" (with X ∈ [0 − 23] representing the hour) is the sampling time.

#	Original Dataset—1 h	Original Dataset—4 h	Original Dataset—12 h	Variant 1–1 h	Variant 1–4 h	Variant 1–12 h
1	(Gf, T-25) (0.176057)	(Gf, T-25) (0.191806)	(F5, T-11) (0.167195)	(F4, T-25) (0.140817)	(F4, T-25) (0.171519)	(F4, T-35) (0.124901)
2	(F5, T-2) (0.122141)	(F4, T-20) (0.140498)	(F6, T-31) (0.123955)	(F5, T-26) (0.122105)	(F4, T-42) (0.118472)	(F5, T-10) (0.107977)
3	(F4, T-24) (0.090190)	(Gf, T-44) (0.106242)	(F5, T-12) (0.066520)	(F4, T-43) (0.074296)	(F5, T-26) (0.093398)	(F5, T-36) (0.085262)
4	(F5, T-5) (0.080913)	(F4, T-24) (0.063905)	(F4, T-8) (0.048408)	(F5, T-1) (0.071058)	(F4, T-43) (0.086672)	(F5, T-9) (0.084192)
5	(F4, T-20) (0.048113)	(F6, T-24) (0.042821)	(F6, T-34) (0.042143)	(F5, T-28) (0.068031)	(F5, T-20) (0.061580)	(F5, T-34) (0.079828)
6	(Gf, T-28) (0.039845)	(F4, T-21) (0.031745)	(F5, T-9) (0.038706)	(F5, T-4) (0.049261)	(F5, T-1) (0.054594)	(F4, T-33) (0.049449)
7	(F4, T-2) (0.037483)	(F4, T-1) (0.028512)	(F4, T-9) (0.033076)	(F5, T-0) (0.049246)	(F5, T-0) (0.051777)	(F5, T-37) (0.047544)
8	(F6, T-30) (0.034064)	(F6, T-41) (0.027404)	(F5, T-13) (0.032787)	(F4, T-32) (0.041652)	(F5, T-21) (0.038264)	(F5, T-5) (0.036453)
9	(F4, T-45) (0.032964)	(F5, T-2) (0.026497)	(F6, T-38) (0.032142)	(F5, T-5) (0.036880)	(F5, T-19) (0.033493)	(F5, T-0) (0.032585)
10	(F4, T-0) (0.032269)	(F4, T-0) (0.021232)	(Gf, T-37) (0.030076)	(F5, T-25) (0.034633)	(F5, T-22) (0.033160)	(F4, T-32) (0.026337)

TABLE A8 Feature ranking obtained by AdaBoost models for regression. Only the first 10 features are shown and for 3 different instants of future prediction (first hour, fourth hour, 12th hour). "F4" = number of people aged [41 − 50]; "F5" = number of people aged [51 − 60]; "F6" = number of people aged >60; "Gf" = number of females; "T-X" (with X ∈ [0 − 23] representing the hour) is the sampling time.

#	Original Dataset—1 h	Original Dataset—4 h	Original Dataset—12 h	Variant 1–1 h	Variant 1–4 h	Variant 1–12 h
1	(F5, T-1) (0.047257)	(F6, T-25) (0.040474)	(F4, T-13) (0.055517)	(F5, T-0) 0.057564	(F5, T-0) 0.050806	(F4, T-38) 0.045402
2	(F6, T-25) (0.045097)	(F5, T-0) (0.035306)	(Gf, T-36) (0.043332)	(F5, T-2) 0.049718	(F5, T-2) 0.044648	(F4, T-36) 0.042101
3	(F6, T-27) (0.043816)	(F6, T-24) (0.034700)	(F4, T-14) (0.041053)	(F5, T-24) 0.048184	(F5, T-24) 0.044015	(F4, T-32) 0.038673
4	(F5, T-0) (0.040556)	(F5, T-1) (0.033067)	(F5, T-10) (0.040434)	(F5, T-1) 0.042997	(F5, T-1) 0.040046	(F4, T-33) 0.035966
5	(F6, T-26) (0.037173)	(F4, T-0) (0.032987)	(F4, T-37) (0.032521)	(F5, T-27) 0.037891	(F4, T-47) 0.038206	(F5, T-13) 0.032435
6	(F6, T-24) (0.034775)	(F4, T-2) (0.031974)	(F4, T-10) (0.030231)	(F5, T-25) 0.035139	(F4, T-41) 0.037150	(F4, T-37) 0.031744
7	(F4, T-0) (0.031604)	(Gf, T-24) (0.031566)	(F4, T-8) (0.029629)	(F4, T-47) 0.033083	(F5, T-27) 0.035928	(F4, T-34) 0.030509
8	(F4, T-2) (0.031207)	(F4, T-43) (0.030037)	(Gf, T-32) (0.029594)	(F5, T-28) 0.033011	(F4, T-44) 0.034855	(F5, T-33) 0.029317
9	(F5, T-2) (0.029181)	(F6, T-26) (0.027169)	(F5, T-12) (0.026573)	(F5, T-6) 0.029317	(F4, T-25) 0.032319	(F4, T-35) 0.029029
10	(F6, T-29) (0.028084)	(Gf, T-27) (0.024984)	(F5, T-14) (0.026513)	(F4, T-46) 0.028032	(F5, T-42) 0.027296	(F4, T-40) 0.028433