IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 June 2022, accepted 2 July 2022, date of publication 21 July 2022, date of current version 27 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3192860

RESEARCH ARTICLE

An End-to-End Smart IoT-Driven Navigation for Social Distancing Enforcement

HAMDI FRIJI[1], (Student Member, IEEE), ABDULLAH KHANFOR[2], (Member, IEEE), HAKIM GHAZZAI[3], (Senior Member, IEEE), AND YEHIA MASSOUD[3], (Fellow, IEEE)

[1]School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ 07030, USA
[2]College of Computer Sciences and Information Systems, Najran University, Najran 55461, Saudi Arabia
[3]Innovative Technologies Laboratories (ITL), King Abdullah University of Science and Technology (KAUST), Thuwal 23955, Saudi Arabia

Corresponding author: Yehia Massoud (yehia.massoud@kaust.edu.sa)

This work was supported in part by the King Abdullah University of Science and Technology (KAUST); and in part by Ministry of Education, Saudi Arabia and the Deanship of Scientific Research at Najran University under Grant NU/RC/SERC/11/6.

ABSTRACT The unprecedented global spread of the coronavirus pandemic COVID-19 has significantly promoted novel Internet-of-things (IoT)-based solutions to prevent, combat, monitor, or predict virus spread in the population. The proliferation of these technologies has fostered their utilization for different practical use-cases to offer reinforced control, discipline, and safety. This paper proposes an end-to-end smart navigation framework that uses Social IoT (SIoT) and Artificial Intelligence (AI) techniques to ensure pedestrians' navigation safety through a given geographical area. The aim is to mitigate the risks of exposure to the virus and impose social distancing practices while avoiding high-risk areas identified from the SIoT data. First, we create weighted graphs representing the social relations connecting the different IoT devices in the area of interest. Second, we regroup the devices into communities according to their SIoT relations that consider their locations and owners' friendship levels. Next, we extract CCTV recorded videos to estimate the level of social distancing practice on different roads using a computer vision model. Accordingly, the road segments are assigned weights representing their safety levels based on the extracted data. Afterward, a graph-based routing algorithm is executed to recommend the route to follow while achieving a trade-off between speed and safety. Finally, the proposed framework is generalized to enable multi-user coordinated navigation. The feasibility of the proposed approach on real-world maps and IoT datasets is corroborated in our simulation results showing an ability to balance safety and travel distance, which can be adjusted according to the user's preferences.

INDEX TERMS Navigation, smart city, artificial intelligence, social Internet of Things, COVID-19, graph analytic.

I. INTRODUCTION

Since early 2020, the world has faced an unprecedented pandemic that severely impaired countries and shaped global healthcare and economic crises. Several measures were taken to limit the virus exposure and its rapid spread [1]. Virologists have highly recommended many precautionary practices such as hand cleaning, mask-wearing, social distancing, and close contact avoidance to mitigate the spread of COVID-19 and its variants. Moreover, several technological solutions have been tested and implemented to tackle the COVID-19 crisis [2]–[4] all over the world. One of the most promising approaches is to exploit heterogeneous and ubiquitous systems such as the Internet of Things (IoT) to monitor spread, contact tracing, and crowded gatherings. IoT can provide low-cost and efficient solutions to help practice social distancing and limit the spread of the infection [5], [6]. Thus, the main advantage of using IoT technology in healthcare and crowd monitoring applications is to produce intelligent tools that encourage the practice of social distancing without austerity measures such as strict lock-downs and travel restrictions.

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.

The current capabilities of smartphones and wearable devices and the existing 5G broadband technology can boost the development of IoT-based solutions in a prompt and large-scale manner to combat pandemics [7]. Many examples of IoT-based solutions have been proposed in literature [8], [9]. For instance, the smart disease surveillance systems had demonstrated an efficient degree of control for the pandemic's spread within the city of Wuhan and other major cities in China [10]. Despite the significant issues of privacy, South Korea's exemplary accomplishments for relatively containing COVID-19 are due, in part, to the commissioning of a coherent information system that tracks visitors and confirmed patients with an alerting system of potential infections [11]. The system provides the community with essential information to assess the spread. Furthermore, Taiwan uses various IoT technologies, such as tracking the citizens and travelers through their mobile phone locations. Thus, if citizens are exposed to a high-risk area of getting infected, they will be altered. Also, if a traveler comes from a high-risk area and violates the self-quarantine procedure, the residents in that area will be notified through a text message to alert them [12].

One of the main approaches that can help mitigate the spread of infectious viruses such as COVID-19 and its variants is to encourage the practice of social distancing between people. Most of the existing techniques are designed to monitor the practice of physical distancing in different areas. However, they do not provide concrete guidance for users to maintain sufficient distance and avoid close contact with other people, specifically in private zones where it is easier to impose measures on the residents. On the other hand, the built-in capabilities of connected devices such as GPS, thermometers, and other sensors, can be exploited to approach social distancing adequately. For instance, in constructions or industrial zones, wearables can guide workers to maintain a safe distance from their peers by generating alerts if social distancing is violated. It also allows for tracking virus spread when an infected person has been detected in the working area and helps avoid the complete shutdown [11].

The emergent concept of Social IoT (SIoT) [13] can be a valuable tool to leverage the traditional IoT systems and enable a better understanding of the ubiquitous IoT network [14]. SIoT is developed to help understand the interactions between the devices and their users and model them with different social relations. These relationships can be established between machine-to-machine, human-to-machine, and human-to-human connections [15] and transform the IoT network into a socially connected network of devices that can be effectively analyzed using graph analytic tools such as community detection [16], [17] and machine learning [18], [19]. By leveraging the SIoT, providing new applications to encounter the virus spread can emerge and minimize the pandemic's negative impacts.

In this paper, we propose a smart navigation framework that exploits the SIoT data to provide pedestrians, and even cyclists, with safe routing bypassing areas where the risk of COVID-19 transmission is high. In other words, the framework recommends a walking route that guarantees social distancing and avoiding close contact. The proposed approach includes four phases. The first phase, or the pre-processing phase, identifies the IoT devices in the area of interest and establishes social graphs interconnecting these devices using different SIoT relations. Our approach focuses primarily on two social relations: 1) a distance-based relation that identifies crowded and highly dense IoT devices and 2) a device friendship relation that allows labeling streets where the user may possess a high chance of meeting a close social IoT friend (e.g., potential socially connected co-worker). The second phase applies the Louvain community detection algorithm to the previously generated graphs to determine the different communities of IoT devices that share strong characteristics for each SIoT relation. In addition to the community detection part, this phase exploits CCTV video cameras registered in the IoT database to detect the pedestrian flow in a real-time manner and extract safety-related information about the crowds in the street or walking zones, including the offline pedestrians in the decision-making during the navigation. The output of this phase is to compute two different weights associated with each component of this phase. According to the previously calculated weights, the third phase intends to assign scores to each street or segment of a street in the area of interest, reflecting the safety level of that map segment. Finally, in the last phase, the city map is transformed into a weighted undirected graph to which we apply the Dijkstra algorithm [20] as a routing algorithm to determine a route characterized by a certain level of safety. A weighted objective function balancing between the shortest and safest routes is developed. The framework will then deliver the trajectories to the user, e.g., via a mobile application for the best path to follow to reach a destination. The proposed routing approach considers the mobility of IoT devices and may update the recommended route regularly by repeating the process, as mentioned earlier.

The proposed smart navigation framework is supported by two low complexity algorithms enabling the routing for a multi-user system where the trajectories for different users are simultaneously determined in order to achieve a balance between safety and speed. The proposed framework is designed to promote the preservation of social distancing by proposing safe routes with a low risk to its users. It exploits the collected IoT data and the social relationships among the devices to autonomously determine the status of the area of interest and define routes for the different users interested in safe navigation. Industrial zones, campuses, or indoor areas such as offices and malls can be equipped with such smart services promoting additional safety to its personnel/users. In a nutshell, the contributions of this paper can be summarized as follows:

- We design an end-to-end smart navigation framework that leverages the SIoT features to develop a routing strategy for pedestrians allowing them to navigate safely through a given geographical area.

- We develop interrelated graph analytic and artificial intelligence approaches, including:
  - ✦ a community detection approach to cluster IoT devices according to their social relations,
  - ✦ a computer vision-based pedestrian tracking for crowd counting and social distancing practice measuring,
  - ✦ a Graph Convolutional Weight Completion (GCWC) algorithm to predict the status of roads missing IoT information,
  - ✦ and a graph-based routing strategy to enable single-user navigation in a weighted graph.

  These approaches are applied to the data extracted from a SIoT dataset to identify risky roads and high-density areas to guarantee safe navigation on the map.
- We propose low complexity routing approaches for multi-user systems to guarantee safe navigation for pedestrians sharing a common environment simultaneously.
- We apply the proposed framework to a real-world dataset and offline map, and we validate the performance of the proposed end-to-end navigation framework using two different practical scenarios.

The remainder of this paper is as follows. In Section II, We examined the related work and recent study employing AI and IoT devices to combat pandemics, especially Covid-19. In Section III, we discuss the objectives and architecture of the proposed framework with an overview of the data flow. Section IV describes the used dataset and the creation of various social relations between the IoT devices. Moreover, the computer vision camera collection phase and the updated social relations weights. Section V investigates the multi-user navigation scenario where low complex navigation algorithms are developed. In Section VI, we present the results of the proposed framework. Finally, we conclude our work and give some possible directions and future works in Section VII.

II. RELATED WORK

Generally, the widespread of IoT combined with AI advancement has been exploited in many other applications in different smart cities and health-related applications [21]. For example, Life First Emergency Traffic Control (LiFE) is an AI-based emergency traffic control using the IoT sensors' aggregated data. LiFE serves emergency vehicles to navigate congested intersections during rush hours, opening the traffic signals to provide the fastest way to reach the incident. Not only that, IoT and navigation in the smart city have been utilized for applications such as finding the paths where the cyclist is trying to find clean air areas based on aggregated data of air quality and providing this information to the user [22]. The pedestrians in a smart city can also benefit from the sensors distributed with user-defined functions that aim to reduce the exposure to pollution and get the low average temperature routes for the user [23]. An example that has been widely adopted starting from 2020 is applying biometric sensing machines that utilize computer vision to identify symptomatic people and prevent them from getting to crowded public places and spreading the virus [24]. In general, IoT, as well as AI, can provide cost-efficient and practical solutions to assist in practicing social distancing and hence, limiting the spread of infections [6], [25].

This heterogeneous IoT data leveraged with AI technology is currently used to contend with the outbreak effects and prevent its spread [31]. For instance, computer vision and deep learning can provide real-time social distancing monitoring by measuring people's physical gaps. Some applications have been utilizing the AI and IoT sensors in social distancing, where machine learning algorithms with inputs from live cameras in public spaces assist in monitoring the social distancing practice [32]. Furthermore, other methods are used to identify citizens' locations in smart cities by using the energy consumption before and during the lockdown. Thus, the trained machine-learning algorithms based on the energy consumption profiles provide decision-makers with tools to predict the population behavior based on different social distancing policies [33]. Other studies investigated the population response to COVID-19 and its corresponding interventions by proposing an AI-based early-warning detecting system in time series of visits to points of interest of essential and non-essential services [34]. The authors of [13] proposed a particle swarm optimization algorithm applied to the emerging Social IoT concept to simulate how the coronavirus spreads. This work aims to assist the government in finding practical control strategies to combat the spreading of the virus.

Different applications have been developed to monitor and prevent the spread of COVID-19 by essentially focusing on utilizing IoT devices [35]. The analysis of the generated data from IoT devices is essential while considering trust and relation levels among the connected devices, which results in trustworthy services in SIoT [36]. Device monitoring can be the first stage of providing an actionable IoT system for the users. For instance, recommending routes for IoT users in a real-time manner could be devised based on the collected data. As such, the authors of [22] have developed a path planning strategy to avoid poor air quality and highly polluted areas. They provided a web-based application for the users based on IoT sensors in smart cities. This work can be used to avoid high-risk infection areas using the appropriate IoT sensors to help mitigate the spread of viruses.

In Table 1, we examine recent studies investigating the navigation and route recommendation for users while exploiting the IoT data to help on mitigating the spread of infection, especially COVID-19 [26]–[30]. We found that majority of selected studies focus exclusively on indoor navigation, while the other concern only on out-door navigation. As a result, these studies limit the generalization of the proposed methods to navigate in out-door and in-door environments. Moreover, it is noticeable that these methods utilize a limited number of IoT devices, such as proximity sensors or cameras, without capturing multiple sources of IoT devices to leverage them in

TABLE 1. Qualitative comparison of our proposed framework with recent relevant studies.

Study	Social distancing	Route recommendation	Real-time routing	Environment	IoT devices (information)
[26]	Yes	Single-user	Yes	Indoor only	Camera
[27]	No	Single-user	Yes	Indoor only	Proximity sensors, mobile phones and Beacons
[28]	Yes	No	No	Indoor only	Wi-Fi and mobile networks
[29]	Yes	Single and multiple users	Yes	Indoor only	Online appointments and mobile phones
[30]	Yes	Multiple users	Yes	Outdoor only	Rule-based system
Proposed	Yes	Single and multiple users	Yes	Indoor and outdoor	Diverse devices (features and SIoT relations)

routing recommendations and social distancing. Therefore, in our study, we offer a more generic approach utilizing heterogeneous IoT devices applicable for indoor and out-door environments, exploiting the social relationships among the devices and simultaneously determining routes for multiple users.

III. OBJECTIVES AND SMART NAVIGATION FRAMEWORK ARCHITECTURE

In this section, we define the objectives of the proposed end-to-end smart navigation framework and present its different phases.

The objective is to serve users requesting routes to avoid the areas in which the risk of infection is relatively high. The risk level of the determined routes is based on the data provided by IoT devices connected to the platform, e.g., smartphones, personal computers, CCTVs in the area of interest. The navigation decision is subject to the SIoT social relations connecting the users of interest to other IoT devices and their owners. Additionally, the video frames generated by CCTV cameras are used to also predict risky routes. The framework also considers the traveled distance to balance between the route's safety and navigation speed. Therefore, three inputs are required to determine the safest/fastest route:

- *Offline map:* this input consists of an undirected graph representing the different paths of the area of interest. The vertices and the edges correspond to the intersections and path segments in the map, respectively.
- *User-related information:* it contains real-time information about the IoT devices for all users connected to the IoT database. The information includes the devices' current locations, owners, and types.
- *CCTV camera input:* the framework has access to CCTV cameras connected to the platform on some path segments or intersections that provide real-time videos helping assess the situation in the covered areas.

The different phases of the proposed smart navigation approach and their interconnections are presented in Fig. 1. The framework starts with a pre-processing phase that processes existing information in the IoT database to generate a graph representing the offline map, including the CCTV locations and their coverage, as well as other graphs reflecting the SIoT relations among the connected devices. In this study, we investigate two social relations, precisely the Co-LOcation/co-work-based Relation (CLOR) and the Social Friendship and Ownership Relation (SFOR), which will be introduced later. The SIoT relations will be represented by two different graphs where the vertices represent the IoT devices, and the weighted edges represent the strength of the SIoT relation between the devices.

The second phase is composed of two sub-phases. The first one is the community detection sub-phase. It intends to analyze the graphs devised from the IoT devices' social relations to determine communities for each social relation as follows:

- *CLOR communities:* the devices that belong to the same CLOR community are physically co-located. Indeed, the number of devices in a CLOR community determines the density level of the corresponding geographical region.
- *SFOR communities:* the owners of the devices belonging to the same SFOR community share mutual and high friendship levels. Hence, there is a risk of close contact if these devices get close to each other.

In addition to helping understand the SIoT network, the community detection sub-phase helps reduce the complexity of the problem by avoiding processing the vast and complex IoT network individually. Instead, it allows the framework to deal with communities of socially connected IoT devices.

The second sub-phase of Phase 2 involves a computer vision module to process the video frames generated by the CCTV cameras to identify high-risk areas where social distancing is not perfectly practiced. Indeed, we assume that

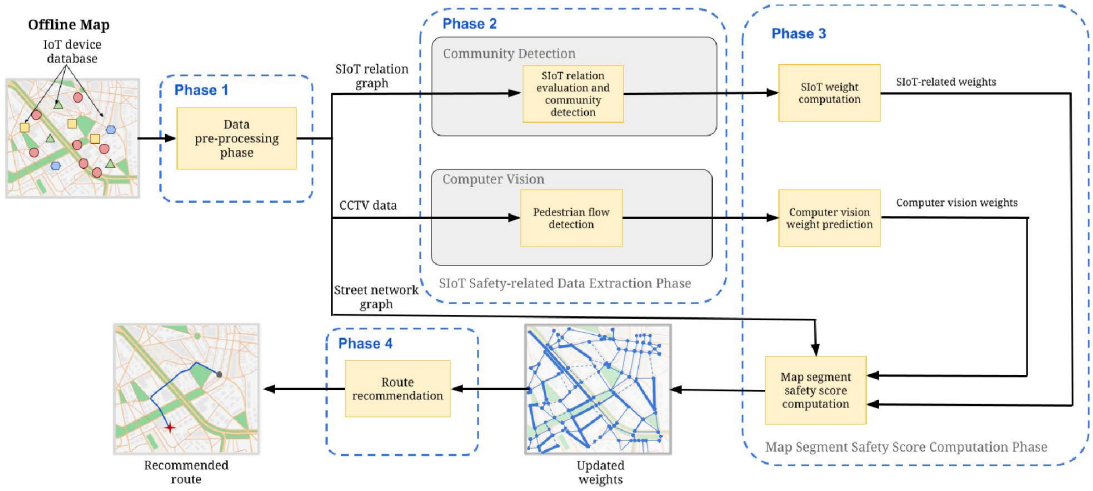


FIGURE 1. The proposed SIoT-based smart navigation framework recommending a safe route to the user while avoiding close contact and maintaining social distancing.

not all users are connected to the IoT platform. Hence, this sub-phase helps better assess the situation in covered areas and identifies any contamination risk due to the crowd behavior. In fact, in many smart cities and industrial zones, CCTV cameras already exist on most roads to provide reliable and high-quality visual information in a real-time manner. The goal of the computer vision module is to estimate the number of pedestrians for each street by counting the pedestrians' entry to the selected street and the social-distancing disobedience rate. Consequently, we can determine the congestion levels of the monitored path segments and assign a weight for that street for navigation purposes.

In the next phase (i.e., Phase 3), we assign to each edge of the offline map a weight reflecting the traveled distance and the path segment's risk level based on the computations of Phase 2 (i.e., SIoT-related weighs and computer vision-related weights). In other words, the weight of each edge incorporates the travel speed and the risk levels based on the outcome of Phase 2. However, some streets are not monitored by CCTV cameras. Therefore, we propose to estimate their weights using an intelligent graph theory algorithm, namely the Graph Convolutional Weight Completion (GCWC) algorithm, to assess the missing information of segments in which we cannot directly determine the number of pedestrians and measure the social distancing violation level. GCWC will output a graph with different risk levels of virus exposure for each path segment.

Finally, in the last phase of the framework, we develop a navigation algorithm applied to the weighted graph obtained from Phase 3. The navigation algorithm searches routes for either a single user or multiple users simultaneously. The route recommendation strategy will then select routes with a minimum sum of weights balancing between safety and speed. Therefore, it determines the most suitable path to recommend to the users of interest, given their preferences (speed and safety).

IV. DATASET AND SMART NAVIGATION FRAMEWORK COMPONENTS

This section presents in detail the different phases of the proposed SIoT-driven navigation framework. It first introduces the SIoT dataset and the original graph representing the offline map. Then, it delves into the IoT-based methods employed to determine the risky areas in the map. Afterwards, it describes the edges' weights computation method, and finally, it closes the loop by presenting the route recommendation approach for a single user.

A. PHASE 1: DATA PRE-PROCESSING

The data pre-processing phase consists in extracting the information that are useful for the development of the proposed navigation framework. The data source includes the IoT dataset, from which we extract the IoT devices' features and the SIoT relations, and the offline map from which we extract the potential paths for navigation.

1) IoT DATASET

The proposed framework requires the following information as an input to achieve its goals:

- *IoT devices information:* The targeted devices are the environment sensors and the high computational devices such as smartphones, smartwatches, and personal computers. The devices' information includes two types of ownership: public-owned, usually the city authority, and private-owned devices owned by the citizens and organizations in the city or the area of interest. Each IoT device is labeled as static or mobile to indicate its mobility. Other information that are also used includes the geographical position, owner, type, etc.
- *CCTV video cameras:* In addition to the set of information described above, the CCTV cameras, as particular IoT devices for our framework, are used to provide high-quality visual information about the monitored path segments.

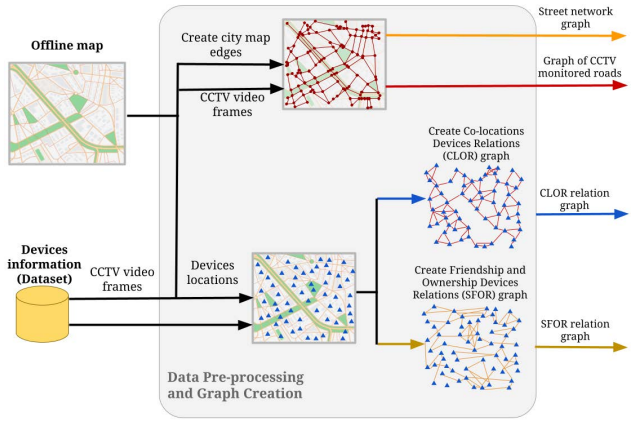


FIGURE 2. Overview of the pre-processing phase.

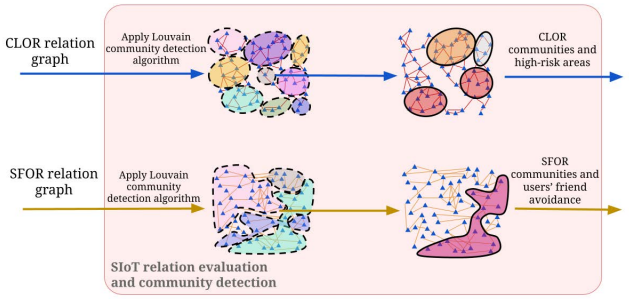


FIGURE 3. Overview of the SIoT relation evaluation and community detection sub-phase.

They are also identified by other information, such as the geographical locations of the cameras, their field of view, and the exact time and date of the recording.

In this study, we apply our framework to a dataset that comprises real-world IoT objects from the city of Santander, Spain, as well as simulated devices like smartphones, tablets, and personal computers by Marche *et al.* [37]. The framework can be applied to any other dataset providing sufficient similar information about the IoT devices.

2) SOCIAL IoT RELATIONS

In this study, two SIoT relations are investigated to generate graphs representing the strength of the relationships between the different devices. Each SIoT relation is used to deduce parameters that may impact the navigation of the user. They are defined and generated as follows:

- **CLOR relation:** By setting a defined threshold for the distance between the devices, we can specify whether the devices belong to a distinct cluster. Hence, a relation is established between them based on the distance separating them. This relation aids in identifying crowded areas where there is a risk of limited social distancing among these devices. Therefore, it is mandatory to avoid passing through these overcrowded areas labeled as high-risk areas.
- **SFOR relation:** To create SFOR relations, we first consider that the same person's two devices are strongly connected. Then, we establish additional SFOR relations using social media networks or other friendship indicators of devices owned by different owners. For example, if two owners are friends in the social network, then the SFOR relation between their devices can be modeled by an edge with a weight reflecting the number of people to reach each owner. Thus, the SFOR relations can be used to avoid users' friends or friends of a friend.

The CLOR and SFOR topologies are undirected and weighted networks. The nodes in these graphs are heterogeneous IoT devices. The edges between these devices represent the SIoT relations stated previously. These graphs do not include self-loop edges to the node.

3) PATH NETWORK GRAPH STRUCTURE

On the other hand, the framework requires knowledge about the offline map, specifically the streets and intersections. In this study, we model the path map network of the area of interest as a graph, denoted by $G = (V, E)$, where each node or vertex, $v \in V$ represents an intersection of two or multiple path segments. An edge $e \in E$ connecting two vertices denotes the path between two intersections. Each edge is given a weight, denoted by ω_e or $\omega_{u,v}$ representing the length of the path corresponding to edge e connecting vertices u and v. We denote by Ω the weighted adjacency matrix associated to graph G where $\Omega_{u,v} = \omega_{u,v}$. If two intersections u and v are not directly linked then $\Omega_{u,v} = 0$. Additionally, we divide long path segments with lengths higher than a threshold length L_{th} into multiple segments where each segment of the road is labeled solely and can be later assigned a different weight. The number of segments for a road of length L_{road} can be expressed as $\lceil \frac{L_{road}}{L_{th}} \rceil$. Note that a path segment can correspond to a street, a walkway, or a corridor in an indoor environment.

In Fig. 2, an illustration of the pre-processing phase converting the raw data of the IoT dataset and the offline map into multiple graphs reflecting different relations among the IoT devices is presented. The output of this phase includes a street network graph, the locations of the CCTV and their corresponding covered path segments (or edges), and the locations of IoT devices and their features. The latter output will be employed in Phase 2 to determine graphs associated to the SIoT relations.

B. PHASE 2: SIoT SAFETY-RELATED DATA EXTRACTION PHASE

This objective of this phase is to extract safety-related information from the IoT dataset and connect it to the offline map. To this end, two sub-phases are proposed. The first one focuses on the IoT features and their social relations to determine risky zones/streets for the user of internet, while the second sub-phase uses the CCTV cameras to identify crowded path segments where social distance obedience is not respected.

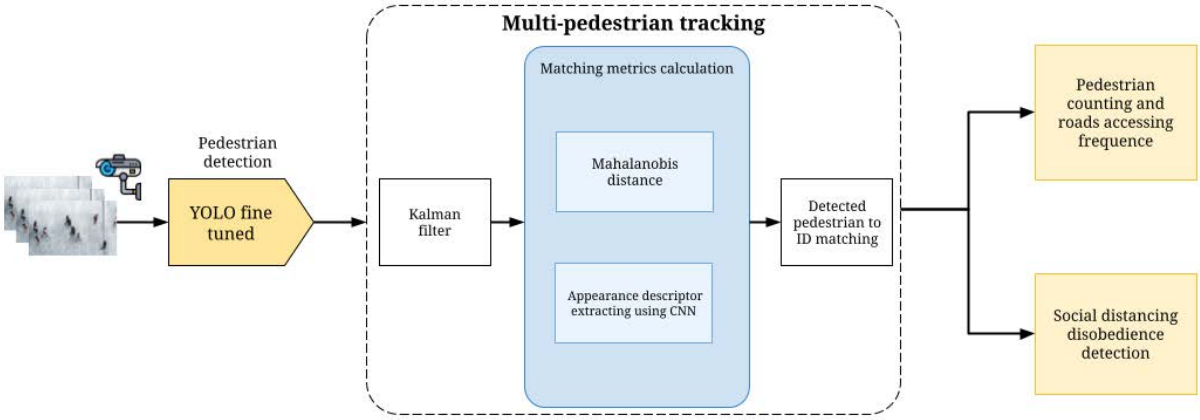


FIGURE 4. Overview of the computer vision tracking system.

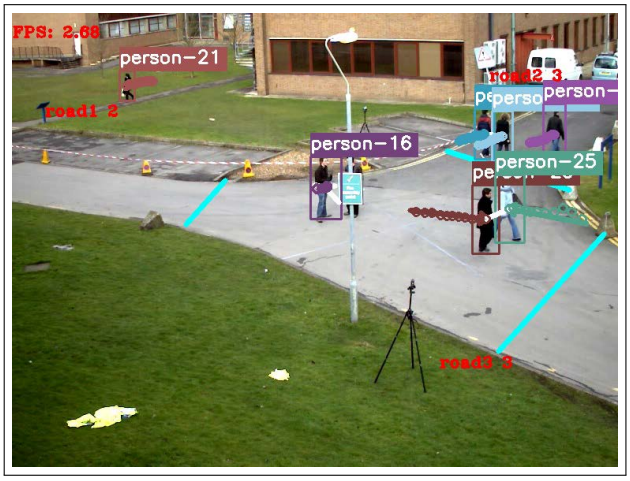


FIGURE 5. Illustrative example showing the extraction of the Pedestrians' behaviors parameters from a CCTV recording in a three-way junction.

1) SIoT RELATION EVALUATION AND COMMUNITY DETECTION

This sub-phase exploits the SIoT paradigm by clustering the IoT devices, in the investigated area, into communities based on their social relationships. In this part, the devices are grouped into communities based on the primary relation (CLOR and SFOR) graphs obtained from the pre-processing data phase (Phase 1).

By creating social IoT relations, we propose to reduce the problem's complexity by employing a community detection algorithm, namely Louvain, that can be used to calculate the effects of the different SIoT communities on the routing recommendation method. A community detection algorithm, in general, converts the complex graph into distinguished clusters of nodes that share strong relations. To this end, we apply the Louvain method [38], which is a greedy optimization method that attempts to maximize the ''modularity'' of a partition of the network [38]. The main advantage of using the Louvain is the running time of $O(n \log n)$, which is considerably faster compared to similar methods [14], [39], [40]. The community detection outcomes in our framework will be used in *Phase 3*.

The Louvain method applied to the CLOR graph will assign the co-located devices into a set of communities. In SFOR communities, the devices are not necessarily geographically co-located. Contrary to CLOR, they are sparsely distributed in the geographical area. However, the owners of these devices may know each other and can meet each other. Therefore, it is recommended that a given pedestrian does not pass by devices belonging to the same SFOR community that his/her device is assigned to for safety reasons and better social distancing practice. Fig. 3 depicts an illustration of the community detection reflecting the SIoT relations of the network.

The performance of the community detection algorithm is investigated using three metrics:

- *Modularity score:* it is a metric designed to measure the strength of the division of a network into modules.
- *Coverage score:* The coverage of clustering is given as the fraction of the weight of all intra-cluster edges with respect to the total weight of all edges in the whole graph [41].
- *Performance score:* The performance of a partition is the ratio of the number of intra-community edges added to inter-community non-edges with the total number of potential edges.

2) COMPUTER VISION PHASE

This sub-phase is dedicated to identifying safety information from the existing CCTV cameras available in the area and connected to the IoT server. CCTV cameras are employed to extract reliable information about pedestrians' density and flow for each covered road. To this end, we employ computer vision algorithms to extract the information from CCTV input frames. The algorithm will predict several features, such as the disobedience of the social distancing rule, the average number of people on that road at an exact timestamp, and the frequency of people entering and leaving the road.

The employed computer vision framework is presented in Fig. 4 where the collected frames are fed into YOu Look Once version 5 (YOLOv5) [42], [43], which is a Convolutional Neural Network (CNN) based object detector. The last layers of the YOLOv5 are fine-tuned to detect only pedestrians and neglect other environmental objects. YOLOv5 is considered one of the most efficient algorithms for object detection due to its fast running time, accuracy, and ability to detect small objects in very crowded areas.

The detection of objects is not enough to ensure accurate results. Therefore, we propose to track every object to avoid counting a pedestrian more than once. The tracking process consists in the association of two detected objects in two adjacent frames. After detecting pedestrians, the estimated bounding box of each person is fed to the Kalman filter to update the target state. The Kalman filter is an optimal recursive Bayesian stochastic method that propagates conditional probability density of the system state based on the actual state [44]. The next step is the matching process, which consists in assigning every new detection to exist targets [45]. Consider the following example: two pedestrians are detected in the first frame and have been assigned the tags ID1 and ID2, respectively. When the video evolves, the object detector, YOLOv5, detects two pedestrians, and during the matching step, the framework recognizes each pedestrian's ID in relation to the previous frame. The framework matching process uses the Hungarian algorithm. The algorithm exploits two metrics to ensure the matching:

- *The Mahalanobis distance:* calculated between the predicted Kalman states and the newly arrived measurements and aims to incorporate motion information in the matching process.
- *Deep appearance descriptors:* those are the semantic descriptors of an image, extracted by a specific convolutional neural network that is intent on extracting the low-level image features.

Finally, the bounding box associated with the target IDs is used to estimate the number of people going on each road, the frequency of accessing a road and detecting any disobedience of the social distancing rule.

In Fig. 5, we illustrate a snapshot of a video regarding the movement of pedestrians in a three-road intersection. The pedestrians are detected using YoloV5 in bounding boxes and tracked using the Kalman filter. The recorded trajectory of each pedestrian is highlighted with the same color used in the bounding boxes. Cyan lines are placed at the edges of the road segments to count the number of pedestrians entering or exiting that road. Finally, the distance separating the centers of two bounding boxes (highlighted by the white color) helps determine whether social distancing is practiced or not.

Fig. 6 displays four different outcomes employed in the smart navigation algorithms. Fig. 6a illustrates offline map. Fig. 6b and Fig. 6c show the communities obtained from the CLOR and SFOR graphs by applying the Louvain method, respectively. Finally, Fig. 6d displays the roads of a street network map that are monitored with CCTV cameras denoted by the red color.

C. PHASE 3: MAP SEGMENT SAFETY SCORE COMPUTATION

In this phase, we propose computing the road weights given their surrounding SIoT communities' statuses and the associated computer vision predicted parameters (pedestrian counting and social distancing disobedience detection). To do so, we assign to each road segment e a weight that is balanced by a coefficient α ($\in [0, 1]$) representing the level of safety set by the pedestrian of interest. Setting a value of $\alpha \rightarrow 0$, the pedestrian intends to determine the shortest path to reach the destination with low consideration of risks. However, if $\alpha \rightarrow 1$, the pedestrian is looking to follow the safest trajectory independently of the expected traveled distance. Values of $\alpha \in]0, 1[$ achieve a trade-off between both routing strategies. The edge's weight of the road network, denoted by ω_e, can be expressed as follows:

$$\omega_e = (1 - \alpha)\, \omega_e^{dist} + \alpha\, \omega_e^{sft}, \tag{1}$$

where ω_e^{dist} is the weight reflecting the expected traveled distance when crossing edge e, while $\omega_e^{sft} = \frac{\omega_e^{SIoT} + \omega^{CV}}{2}$ is the weight reflecting the safety level of the edge which is measured using the weights deduced from the SIoT community detection part, ω_e^{SIoT}, and the ones estimated from the computer vision part ω_e^{CV}. The weights ω_e^{dist} are a normalized value of the segment e's length.

1) SIoT WEIGHTS COMPUTATION

The values of Social IoT weights ω_e^{SIoT} are measured by combining the impact of the surrounding CLOR communities and devices belonging to the same SFOR community of the pedestrian's device as follows:

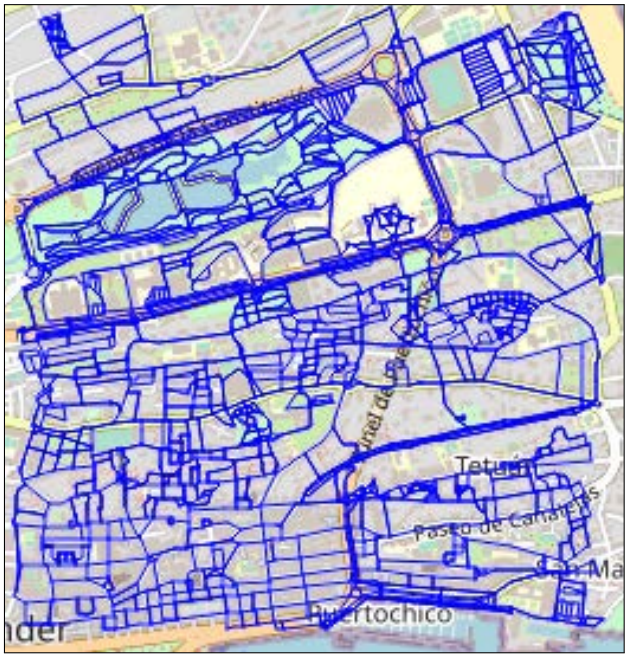
$$\omega_e^{SIoT} = \frac{\omega_e^{CLOR} + \omega_e^{SFOR}}{2}, \tag{2}$$

where the CLOR weights ω_e^{CLOR} are calculated using the following expression:

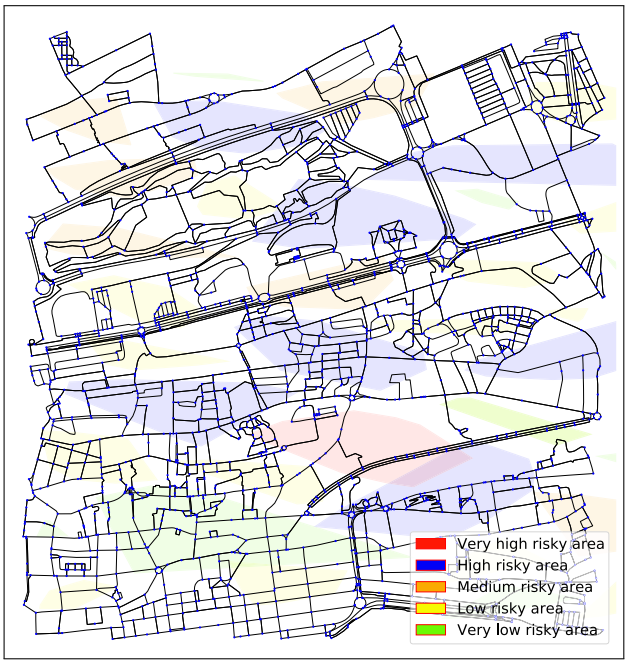
$$\omega_e^{CLOR} = \sum_{c \in \mathcal{C}_e^{CLOR}} \frac{\gamma_c}{|\mathcal{C}_e^{CLOR}|}, \tag{3}$$

where $\mathcal{C}_e^{CLOR}$ is the set of CLOR communities that intersect with the edge e. The cardinality of this set is denoted by $|\mathcal{C}_e^{CLOR}|$. The CLOR communities are modeled as polygons that circumscribe devices belonging to each community with an outer offset ρ. We denote these polygons by $P_c, \forall c \in \mathcal{C}^{CLOR}$ where $\mathcal{C}^{CLOR}$ denotes the set of all CLOR communities obtained using the Louvain method. Thus, the set $\mathcal{C}_e^{CLOR}$ can be defined as: $\mathcal{C}_e^{CLOR} = \{c \in \mathcal{C}^{CLOR} \,|\, P_c \cap \{e\} \neq \emptyset\}$ where $\emptyset$ is the empty set. The offset parameter ρ is added to all polygons associated with the CLOR communities to ensure a safe social distance for the navigating user. Finally, γ_c denotes the density of community c and is calculated as follows:

$$\gamma_c = \frac{N_c}{A_c}, \tag{4}$$



(a) Extracted roads of a selected area of the city of Santander, Spain.



(b) CLOR communities (The colors of each CLOR community indicates the density level of the community).

(c) SFOR communities (IoT devices belonging to the same community are labeled by the same marker).

(d) Graph network of roads equipped with CCTV cameras as described in IV-A3

FIGURE 6. **Outcomes of Phase 2 of the proposed smart navigation framework.**

where N_c is the number of devices in community c and A_c is the surface of the area of P_c.

Similarly, the SFOR weights can be computed as follows:

$$\omega_e^{SFOR} = \sum_{u \in \mathcal{C}_{u^*,e}} \frac{\Omega_{u,u^*}^{SFOR}}{|\mathcal{C}_{u^*,e}|}. \tag{5}$$

The $\mathcal{C}_{u^*,e}$ is a set of devices that are in SFOR relation with the devices u^* of the user pedestrian and their $d(u,e) \leq d_{th}$. It corresponds to the SFOR community to which the user device u^* belongs. Moreover, the user device distance is denoted as d_{th} that measures the edge e distance. The coefficient Ω_{u,u^*}^{SFOR} is obtained from the SFOR graph, and it measures the strength of the SFOR relation between

device u and u^*. The parameter d_{th} denotes the allowed distance in which the user owning the IoT devices will try to avoid any device within the same SFOR community.

In general, the SIoT weight of an edge e significantly increases if it is surrounded by high-density CLOR communities and/or many devices belonging to the same SFOR community of the user of interest. Therefore, in the navigation phase, we aim to select the edges with lower risk, i.e., the trajectory that minimizes the sum of ω_e^{SIoT} for a user looking for a safe walk.

2) COMPUTER VISION WEIGHTS COMPUTATION

For each road, we assign a weight based on the previously discussed features in the computer vision sub-phase of Phase 2. Those weights are calculated as follows:

$$\omega_e^{CV} = \frac{\omega_e^{PF} + \omega_e^{SD}}{2}, \tag{6}$$

where ω_e^{PF} is the number of pedestrians accessing the investigated road per second (i.e., the frequency of pedestrians accessing the road) and ω_e^{SD} is the ratio of people who disrespect the social distancing rule in that road. The latter weight will be assigned to each edge monitored by a CCTV (i.e., the edges colored in red in Fig. 6d). Those weights are calculated at each timestamp as the average of the weights calculated for each frame during the timestamp. So far, the CV weights are only available on the roads or intersections monitored using CCTV cameras registered in the IoT database. Other non-monitored roads lack this data and are not assigned any CV weights. Consequently, we propose an AI-based solution to impute the missing features extracted in the computer vision part and hence, predict CV weights for non-monitored roads based on the monitored roads' features. To assign predicted weights to the edges not supported by CCTV cameras, we exploit the Graph Convolutional Neural Network (GCNN) approach [46]. This graph-based artificial intelligence exploits the topology of the road network graph and the correlations of weights among adjacent edges to estimate stochastic weights for all edges.

This process adopts the intuition of the auto-encoder. Indeed, the street roads graph is provided as input to be fed into a GCNN, followed by a pooling layer that will embed the graph into a lower-dimensional space, which can be regarded as the encoding process. Finally, we decode the output of the encoder using a fully connected layer and thus obtain an estimated fully weighted graph. During the embedding phase, a new weighted embedded graph is created. The edges of the embedded graph are weighted and the problem of the missing weights is solved. The decoding phase will restore the initial graph but with different weights for all edges with no exception. Hence, a full graph containing the predicted w_e^{CV} weights is obtained.

D. PHASE 4: ROUTE RECOMMENDATION

At this stage, a weighted graph is generated. The weights, computed in the previous Phase and specifically using (1), a balance between speed and safety. One method to determine a road based on the safety preference of the user is using Dijkstra's algorithm. The parameter α will decide the safety level of the determined road. Our framework can dynamically recommend new paths to the user based on his/her current location while considering the mobility of other IoT devices. Hence, it needs to update the CLOR and SFOR communities after a specific time period as well as the computer vision features to provide real-time factors. Consequently, the selected path is updated from one-time slot to another. In other words, Phases 2 to 4 will be repeated for each time slot until the user reaches his/her destination.

V. MULTI-USER NAVIGATION

A multi-user functioning is required to obtain an efficient navigation framework for a group of people sharing a common space and aiming to practice safety measures or navigation restrictions. Generally, the multi-agent concept is omnipresent in the navigation realm, such as organizing the cities' traffic flow and multi vehicles' shortest path selection. In this section, we propose to make the model more practical by considering the multi-agent scenarios. For example, it can be used in industrial areas or campuses to assist workers/students in navigating and enhancing pedestrians' protection and decreasing the probability of getting infected in the investigated regions.

A centralized architecture characterizes the framework where every user is identified independently. Each user is defined with a unique ID and has his/her attributes such as the navigation speed, estimated from his/her navigation history, and the exact instant of his/her last connection. In multi-user navigation framework, we aim to synchronize the users' trajectories in a real-time manner to avoid the presence of many users at the same place simultaneously and hence, push towards a better social distancing practice within the area of interest. At each timestamp, the framework must validate the following constraint for all users:

$$|AT_u - AT_v| < \Delta_t, \tag{7}$$

where AT_u is the arrival time of user u to intersection u and Δ_t is the necessary period of time to avoid conflicts between users. Next, we present two Dijkstra-based heuristic algorithms that adopt different strategies for choosing the optimal path. The first algorithm, referred to as Priority Based Iterative Navigation Algorithm (PINA), determines a trajectory for a prioritized user then adjusts on the following user. The second algorithm, namely Step-by-step Iterative Navigation Algorithm (SINA), discovers the simultaneous steps taken by each pedestrian at each iteration to make the subsequent decision.

A. PRIORITY BASED ITERATIVE NAVIGATION ALGORITHM (PINA)

The PINA algorithm avoids any conflict by redirecting the user to the second selected path. To this end, a priority between users is mandatory for this algorithm to avoid

Algorithm 1 Priority Based Iterative Navigation Algorithm (PINA)

Inputs: Initial point and destination $(\{St_j, De_j\}, j \in \{1, \cdots, N\})$ where N total number of the users.
Initialization Phase:
Initial path for each user j; PA_j = Dijkstra(St_j,De_j).
Set $SP_j = St_j$ and $DP_j = De_j$.
for u in $\mathcal{U}$ **do**
  Calculate the arrival time AT_u for higher priority users $\in \{1, \ldots, u-1\}$.
  Determine the list of segments $\mathcal{L}$ where conflicts with higher priority users are detected.
  Block those segments during the arrival instants plus/minus gap time Δ_t to prevent collision for the rest of users.
  Update path for user u using Dijkstra(SP_u,DP_u).
  Navigate the user u using the updated path.
end for

Algorithm 2 Step-by-Step Iterative Navigation Algorithm (SINA)

Inputs: Initial point and destination $\{St_j, De_j\}, j \in \{1, \cdots, N\}$ where N total number of the users.
Initialization Phase:
Set $SP_j = St_j$ and $DP_j = De_j$.
Find initial path for each user j using Dijkstra(SP_j, DP_j).
for $T = 1, \ldots, \bar{T}$ **do**
  Determine the list $\mathcal{L}$ of intersections at which the condition (7) is not verified.
  for l in $\mathcal{L}$ **do**
    Determine the list $\mathcal{U}$ of users heading to the intersection l.
    Calculate the arrival time AT_u of each user $u \in \mathcal{U}$ to the intersection l.
    Find $\hat{u} = argmin_{u \in \mathcal{U}}(AT_u)$.
    Freeze the navigation of users u in $\mathcal{U} \setminus \{\hat{u}\}$ for a period of Δ_t and update their starting point $SP_j = l$.
    Update path for user u using Dijkstra(SP_u,DP_u).
  end for
end for

the problem of conflicting with other users while avoiding another one. The users will be ordered, e.g., according to the First-In-First-Out (FIFO) concept. After determining the priority levels of all users, the safest trajectory is determined for the user with the highest priority in the examined area. To do so, Dijkstra's algorithm was applied to the weighted street network graph as described in Section IV-D. Then, the trajectory of the lower priority user will be calculated based on the trajectories of the users with higher priority to avoid conflicts. Hence, arriving at an intersection of the map must respect condition (7). Hence, delays are counted in these intersections. In case of conflict, all roads leading to the targeted locations are closed, and a second trajectory is calculated. The intersections are not permanently closed but denoted inaccessible until the previous user passes, considering the time gap Δ_t. The algorithm describing this process is given in Algorithm 1. The selected path for each user needs to consider the potential risk of collision with all previous users. Thus, some roads are blocked for the current user during a specific period; consequently, the user will select a second optimized path that avoids crossing with the other users with higher priority. Then, the user moves one phase ahead and updates its current location till reaching its destination.

B. STEP-BY-STEP ITERATIVE NAVIGATION ALGORITHM (SINA)

SINA recognizes the next step to be chosen by each user simultaneously. At every iteration, each user will take an action and move towards its destination. The framework will assist the user to reach its destination by traveling through the safest possible route by choosing between two actions: wait or navigate. The SINA aims to satisfy all the users and provide them with the safest trajectory without any specific pre-ordering. Indeed, When simultaneously reaching the same intersection, the FIFO principle is applied between the users. Consequently, the user arriving first to the intersection enters the road segment, while the second user waits if necessary. The algorithm starts with an initialization phase where it determines the fastest paths of each user using Dijkstra's algorithm assuming solo trips. Finally, the algorithm converges when all the users reach their destinations. The pseudo-code of the SINA algorithm is given in Algorithm 2.

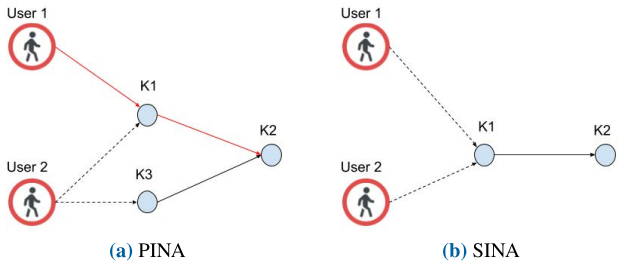
In Fig. 7, we highlight the differences between the PINA and SINA algorithms. With PINA, the trajectory of the highest priority user, namely user 1, is already determined, and the second user with lower priority, user 2, will need to determine its trajectory accordingly. Hence, two options are possible at intersection $K1$ either user 2 will wait to enter $K1$ after a time gap Δ_t or choose another path. With SINA, user 2 arriving first to the intersection $K1$ will have the priority to enter to the segment $(K1, K2)$. User 1 will then be forced to wait.

VI. RESULTS & DISCUSSIONS

A. DATASET AND MAP DESCRIPTION

In our simulations, we use a 16 RAM laptop occupied with an RTX 2070 graphic card. The laptop development environment is based on Python 3.7 under the anaconda platform (i.e., responsible for managing python packages). To preprocess the dataset and the offline map (Phase 1), we first select a 6×6 km^2 area in the city of Santander, Spain, that we identify as the "downtown". Afterward, we extract the city map using the OpenStreetMap project,[1] and we convert its road into a road graph using the OSMnx method [47], and Networkx[2] libraries. The PSMnx is responsible for communicating with OpenStreetMap and extracting all the information

[1] https://www.openstreetmap.org
[2] https://networkx.org/



(a) PINA (b) SINA

FIGURE 7. Comparison between the two proposed heuristic navigation approaches a) PINA and b) SINA. With PINA, user 2 needs to wait before entering K1 or go through K3 instead since priority is given to user 1. With SINA, there is no priority among the users. User 1 is arriving later, hence he needs to yield to user 2 to enter K1 safely.

needed to create the graph. On the other side, the Networkx library is used to manipulate the graph in each section of the framework using its flexibility. Afterward, we project the devices in the selected area from a real-world IoT dataset provided in [37]. The dataset includes 16216 devices covering the whole city. The devices vary from simple sensors such as street lights, environmental sensors, and highly computational devices like smartphones and personal computers. Private and public entities own the devices. The local authorities usually own public devices. For the private-owned devices, there are static and mobile devices. In our simulations, we select mobile devices owned by private entities that are most likely owned by human beings, such as smartphones, smartwatches, tablets, personal computers, etc. The remaining devices from the previous selection process result in 1312 personal IoT devices. We have also performed the same process on another smaller part of the city, specifically the city's harbor, and identified it as the ''Harbor area'', to mimic an industrial zone. In this case, we extract the data of 917 IoT devices. In Fig. 8, we illustrate the two investigated zones in our study.

B. CLUSTERING AND COMMUNITY DETECTION

In this Section, we evaluate the performance of sub-phase 1 of Phase 2. The selected devices will have SFOR and CLOR SIoT relations. We create a fully connected network for CLOR relations. After that, we set a threshold of $d_{th} = 1$ km between the devices. Thus, if devices are higher than the threshold value, we drop the edges connecting two devices; otherwise, they will get the inverse value of the devices separated by less than 1 km. The community detection algorithm applied to the CLOR will return a set of relatively high-density communities to determine high-risk infection areas with the limited practice of social distancing. For the SFOR relations, we apply the social network of the owners of the IoT devices to determine the edges between the devices. Because we lack information about the owners' social network, we employ the Watts–Strogatz generator [48] to create a graph that can exhibit a social network between the owners. In the SFOR, the relations of devices with the same owner are assigned an edge of 1, while the direct

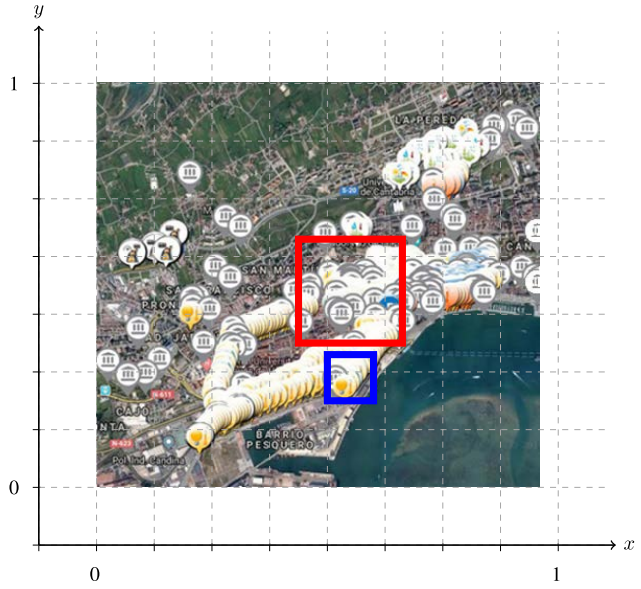


FIGURE 8. The city of Santander, Spain with normalized geographical coordinates where IoT devices are deployed according to the Iot dataset of [37]. The largest area delimited with the red box is the downtown while the smallest area delimited with the blue box is the Harbor area.

friends-owned devices will have an edge of 0.5. Other devices are given weights computed while considering the minimum number of hops needed for one of the vertices (owners) to reach the other vertices. We restricted the relations to three friends of friends since they are socially far away from each other.

As shown earlier, Fig. 6b and Fig. 6c show real-world communities obtained from the SIoT graphs by applying the Louvain method. It results in 56 CLOR communities drawn by colored polygons and having different density levels, as illustrated in Fig. 6b. The CLOR communities are classified, based on their densities, into five classes. A high-density community is located approximately at the center of the map. Other blue-colored CLOR communities with a lower density are considered a risky area and require to be bypassed by the user for safety. The medium and low-risk areas might be avoided, but they will be recommended to the user if there are no other paths. The SFOR-based relation clustering results in 10 communities with diverse devices spread over the map. Each community has different symbols and colors as shown in Fig. 6c. The user of interest will belong to one of these communities and needs to avoid close contact with them. To investigate the efficiency of the Louvain algorithm, we calculate the modularity, coverage, and performance score, defined in Section IV-B1. Indeed, the calculated validation scores of communities in two investigated areas, namely the downtown of the City of Santander and the harbor area, are presented in Table 2 and Table 3 and show high-quality measures for CLOR and SFOR. It is clear that all metrics are presenting results very close to 1, which validates the efficiency of the obtained communities.

TABLE 2. Validation results of the CLOR communities.

	Downtown	Harbor area
Modularity score	0.934	0.892
Coverage score	0.977	0.926
Performance score	0.961	0.918

TABLE 3. Validation results of the SFOR communities.

	Downtown	Harbor area
Modularity score	0.901	0.814
Coverage score	0.953	0.902
Performance score	0.923	0.884

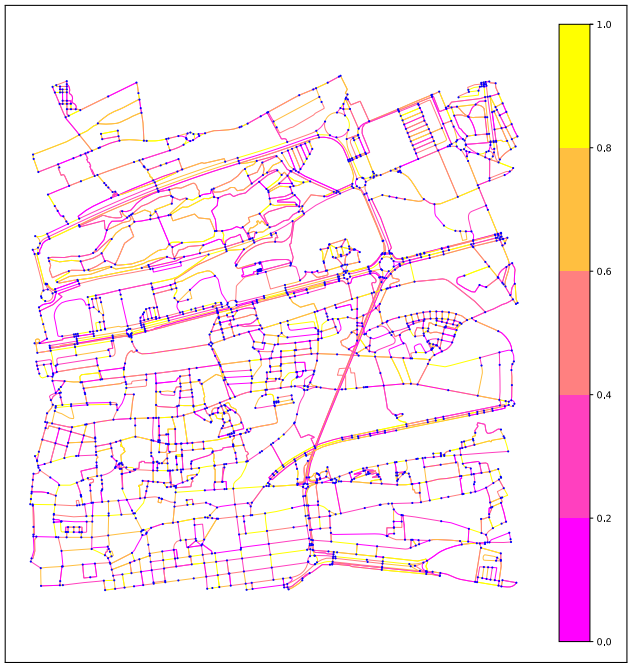


FIGURE 9. Street network after assigning to each road a computer vision weight.

C. COMPUTER VISION & WEIGHT LINK PREDICTION

In this section, we depict the output of sub-phase 2 of Phase, including the predicted weights using the GCNN method in Phase 3. Recall in Fig. 6d, we have plotted the network of roads that are monitored with CCTV cameras, colored in red, where we collected information about the behavior of pedestrians. Nevertheless, the streets unequipped with CCTV cameras are colored in black. We proceed by computing the computer vision weights, as explained in Section IV-C2, and update the graph edge weights by following the mathematical formulation as explained in Section IV-C for the CCTV monitored roads. The navigation algorithm requires all edges to be weighed. Therefore, we estimate the weight of the edges that represents the roads with no cameras using the weight link prediction on the graph as described in Section IV-C2. The obtained graph network after the imputation process is presented in Fig. 9 with a color bar that classifies the roads into five classes based on the computer vision weights.

D. SINGLE-USER NAVIGATION

This section analyzes the outcomes of the proposed framework after applying Phase 4 for a single-user navigation. Fig. 10 illustrates two examples of the recommended routes for the user given same starting points and destinations for three values of α after applying Dijkstra's algorithm using the computed weights. If $\alpha = 0$, the framework will recommend the shortest path (the red route), otherwise, if $\alpha = 1$, the safest path with minimum exposure to the virus is recommended (the green). However, for $\alpha = 0.5$, a trade-off between both metrics is provided (the blue route). In Fig. 10, it is clear that along the green route, the user is avoiding most of the high-density areas surrounding them. It just crosses some of the low-density regions. Furthermore, it avoids getting closer to other SFOR-related devices unless it is forced to do it. This leads to a long route of 2.77 km. With the red route, the user neglects the risk of contamination and crosses all the high-density areas. The corresponding traveled distance is equal to 2.1 km. Finally, the algorithm avoids the red zone for the blue route and tolerates passing by some blue areas. The result of this case is a traveled distance of 2.4 km.

Our framework can be adapted to a dynamic scenario by considering the real-time mobility of the IoT network. In Fig. 11, we plot the routes for three consecutive time slots. In each time slot, IoT devices may change their locations or quit the area. Other devices may appear in the system. Consequently, the CLOR relations are affected as well as their corresponding communities. The positions of devices in SFOR are also subject to modifications. In the dynamic scenario, the shortest path will remain intact. The algorithm is only aware of the distance to be crossed. On the other hand, in the proposed framework that considers the safety weights, the trajectory is regularly updated given the locations of the devices at each time slot. Accordingly, the user starting at the left bottom corner of the map will notice that its trajectory is partially updated at a time slot (t_1) since several SFOR devices left the area. The user can cross in the middle of the map to reach its destination. The navigation algorithm is executed again in the next time slot (t_2). Notice that the user is forced to go around it to reach its destination. As long as he/she is moving, the user is getting closer to the destination, especially if a correct value of α is chosen. Choosing α close to 1 may lead to non-practical results as the user will always try to maintain safe trajectories and may never reach the targeted destination. Therefore, a balanced choice of α is required to achieve a tradeoff between safety and rapidity. Accordingly, our proposed algorithm exploits the Dijkstra algorithm to choose the best path locally. Still, it gives different results from the one obtained by applying Dijkstra globally (from initial departure to the final destination).

In Fig. 12, we conduct a Monte Carlo simulation and plot the Pareto curve investigating the impact of the safety coefficient α defined in (1), while analyzing the effect of the

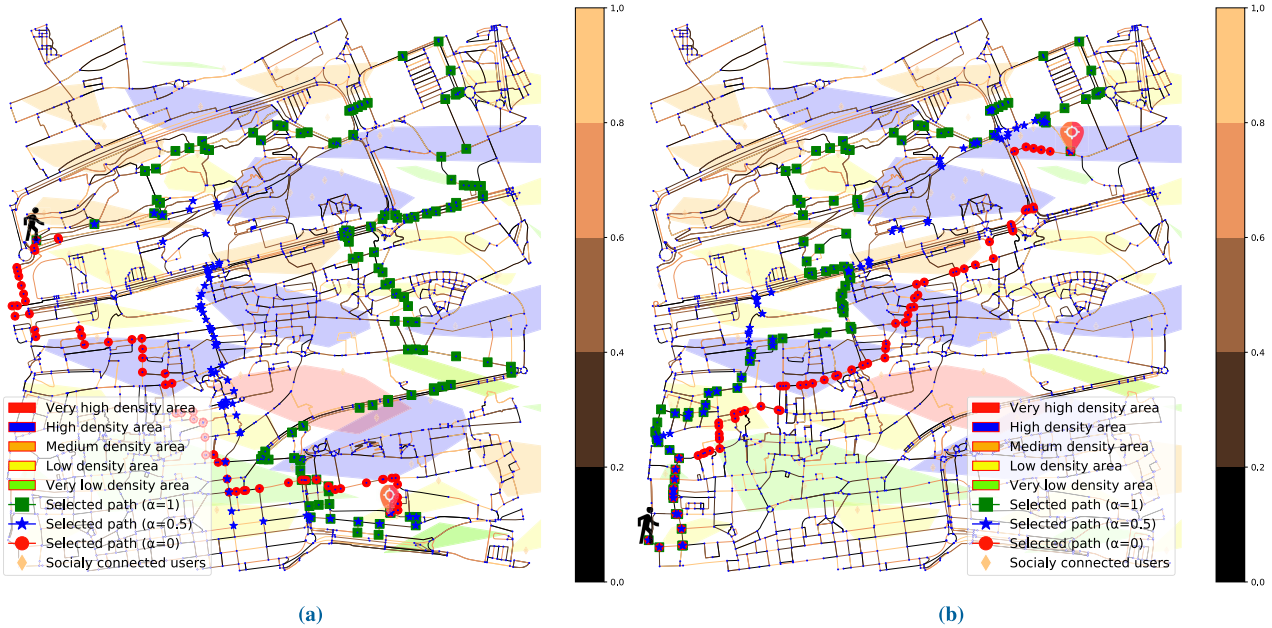


FIGURE 10. Two examples showing the different paths recommended to the user for different values of α.

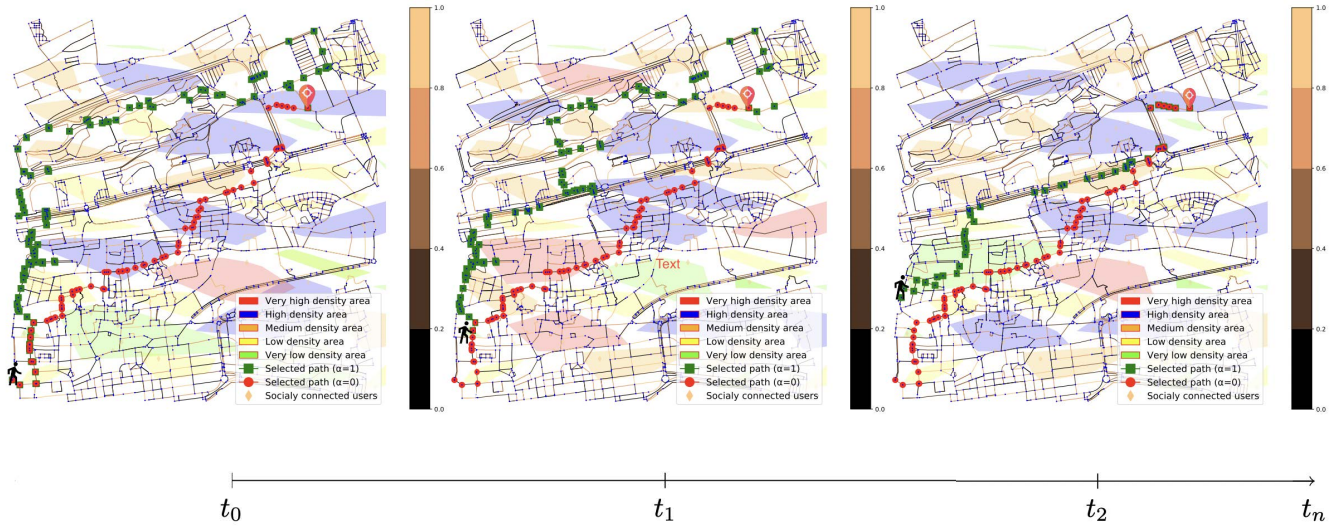


FIGURE 11. Street network after assigning to each road a computer vision weight.

social distancing factor ρ (introduced in Section IV-C) on the selected path. Fig. 12 plots the average travel distance achieved over 100 scenarios versus different values of α for two choices of ρ (the social distancing outer offset). A higher value of ρ indicates an increasing preventive navigation strategy leading to trajectories a little bit far away from the CLOR communities. The figure shows that the travel distance increases if α increases, moving from 1.6 km to over 2.6 km, while the cumulative safety score is almost linearly decreasing. A compromise between safety and speed can be achieved for α around 0.44. By increasing the outer offset ρ, a more strict social distancing is applied, and hence, the traveled distance increases even for the exact value of α. For instance, for $\alpha = 0.44$, the distance changes from 1.9 km to 2.3 km, slightly improving the safety score.

E. MULTI-USER NAVIGATION

This section investigates and compares the performance of the two proposed multi-user navigation algorithms, PINA and SINA, in the harbor area of 1×1 km^2 area. This new map assumes that six workers navigate from different starting points to the same destination. The trajectories of all workers are shown in Fig. 13 for the PINA and SINA algorithms, respectively. The proposed approach avoids users' simultaneous presence at the same intersection or any situation where

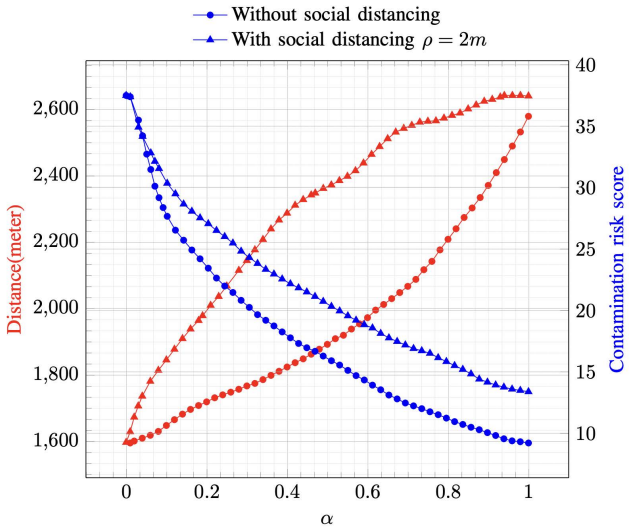


FIGURE 12. A trade-off between the safety factor and destination amount for the proposed framework.

the inter-user distance is less than two meters (e.g., small roundabouts).

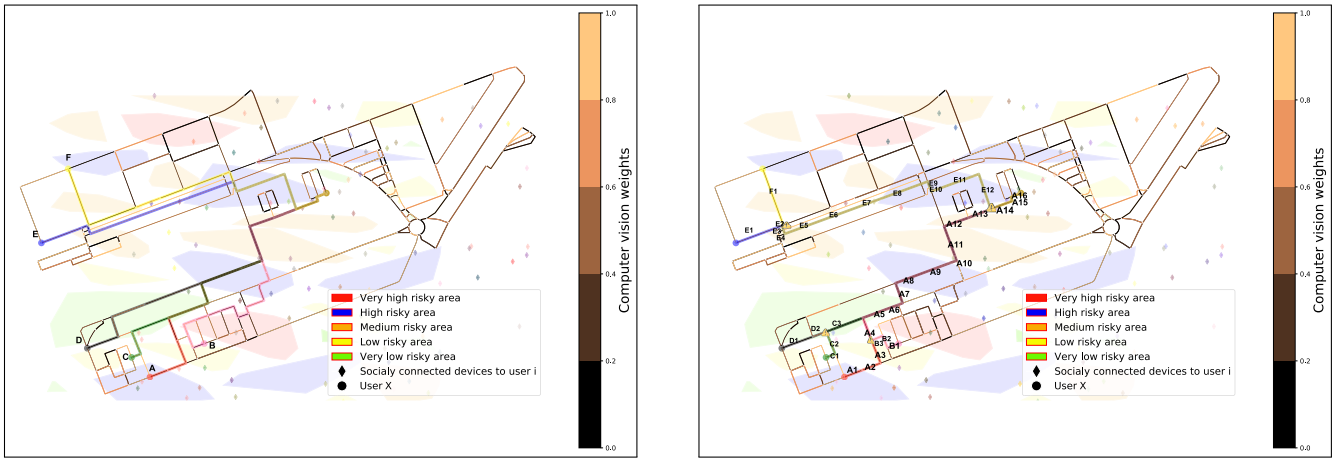
The PINA results shown in Fig. 13(a) indicate that the framework changes the path of the user at each conflicting intersection and minimizes the risk by avoiding high-risk areas deduced from the CLOR communities and keeping a safe distance from socially connected users identified from the SFOR communities. Note that the workers' priority for the PINA is ordered according to their alphabetical indices. In other words, ID A has the highest priority and ID E has the lowest one. On the other hand, in Fig. 13(b), we plot the trajectories of the users obtained with the SINA algorithm. We also highlight the four conflicting intersections where the FIFO principle is applied between users using the warning sign symbols. With the SINA navigation, some workers will need to wait to allow other passes to the following road segments.

TABLE 4. Comparison between the PINA and SINA algorithms: Navigation time in seconds and safety score.

	PINA		SINA	
	Navigation time	Safety Score	Navigation time	Safety Score
User A	484	12.35	474	12.35
User B	436	10.13	436	13.96
User C	498	9.45	468	9.45
User D	461	9.85	461	9.91
User E	474	11.76	474	11.76
User F	528	11.91	498	12.09

To further investigate the SINA-based navigation, we present, in Fig. 14, the cumulative navigation time of each user. The y-axis denotes different users' IDs, while the x-axis measures the duration to reach the destination in Seconds and the time spent to cross each road segment. For each user, the rhombus symbolizes the beginning and end of a crossed road segment or the beginning and end of waiting events. The length of the lines between the rhombus corresponds to the navigation time spent per road segment or the waiting time. The alphabetic letters above each line, e.g., A1, A2, B1, B2, specify the different road segments illustrated in Fig. 13(b). Additionally, the letter W-i denotes awaiting event due to conflict with user ID X where $X \in \{A, \ldots, F\}$. Fig. 14 shows that some users reach their destination with no waiting time (ID B, ID D, and ID E). Those users are prioritized users using the FIFO principle because they reached the intersections earlier. However, other users have waiting times, and the algorithm imposes this to avoid conflicts and enforce social distancing.

Table 4 compares the risk score and the navigation time in seconds of both algorithms for each user. We can notice



(a) PINA (b) SINA

FIGURE 13. Navigation paths for six users A, B, C, D, E, and F using the multi-user navigation algorithm.

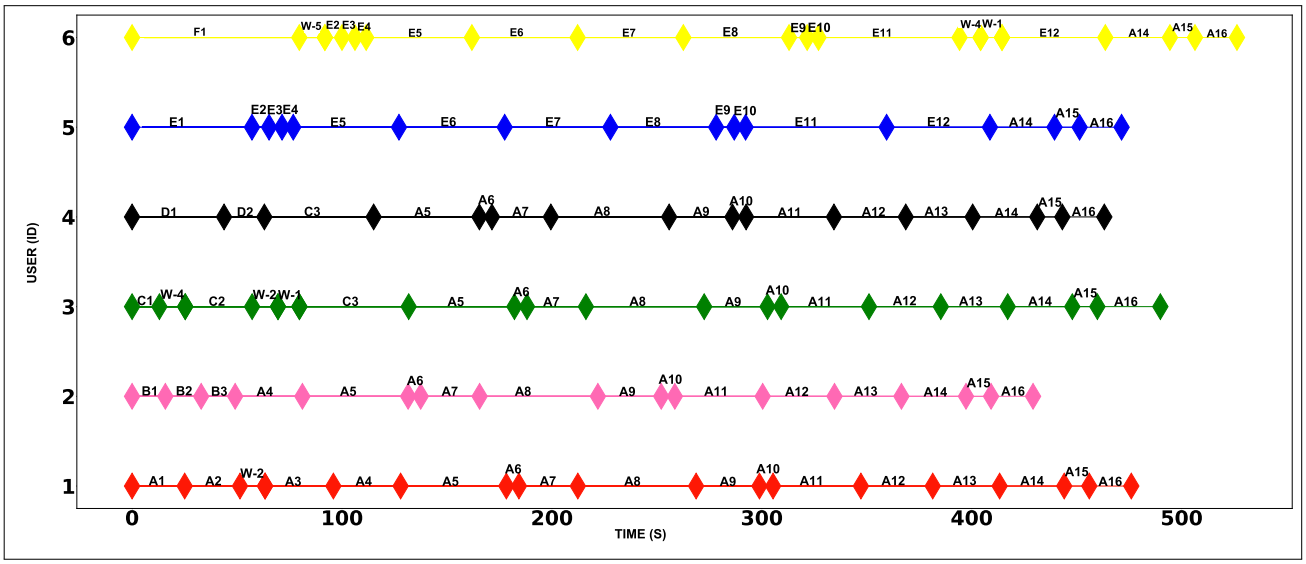


FIGURE 14. Time schedule for six users using the SINA algorithm. The length of each segment (connecting two rhombuses) indicates the time needed to navigate or wait. The labels on each line (e.g., E1) indicate the road segments crossed by a user. W-X indicates the waiting action to yield to the user X.

that PINA outperforms SINA in terms of navigation time due to the absence of the waiting process (e.g., users A, C, and F reach their destinations faster with PINA than with SINA, while the other users witness the same navigation time). However, SINA outperforms PINA in terms of risk score fpr the same safety coefficient α. Indeed, the waiting process in SINA is helpful if the users aim to navigate 100% safely. However, PINA is a suitable alternative if the users prefer to go quicker and avoid the waiting process. Finally, the user's decision is based on the trade-off between speed and safety.

VII. CONCLUSION

Ubiquitous IoT networks can provide real-time solutions to combat pandemic spread, such as COVID-19. In this paper, we have proposed an end-to-end smart navigation framework that helps enforce the practice of social distancing to combat the spread of viral viruses. The framework suggests a trajectory for pedestrians to reach their destination while avoiding areas with a high risk of exposure to the virus. Starting from an offline map and an IoT database, the smart navigation framework utilizes IoT devices' features, graph theory, and artificial intelligence techniques to determine risky areas, i.e., high-density areas, zones where social distancing is not practiced, or high-level of contamination to help people protect themselves in public areas or working zones. The framework operates in a real-time manner based on the user's needs while considering the network dynamics, i.e., mobility of devices and pedestrians. Accordingly, with the multi-agent navigation option, the proposed framework is accommodated for various practical use cases, including synchronizing pedestrians' navigation in private areas such as campuses and industrial workplaces. We have validated our results on real maps using a real-world IoT dataset. We have successfully generated safe trajectories given the IoT collected data while reaching a trade-off between safety and Speed.

In future work, we propose to investigate the scalability of the framework to incorporate larger datasets and integrate other IoT types of data and social relations to introduce a better trust management system and ensure an additional level of privacy, which remains one of the main challenging paradigms in these frameworks. We will also focus on developing a web-based dashboard and a mobile application that allows the effective use of real-world data and the direct application of the proposed framework in practice. The web-based dashboard can give city planners and authorities high-level insights about the precaution measures to adopt in the investigated areas given the real-time IoT data. At the same time, the mobile application can operate as a personal navigation assistant allowing its users to practice social distancing.

REFERENCES

[1] M. Lucic, H. Ghazzai, C. Lipizzi, and Y. Massoud, "Integrating county-level socioeconomic data for COVID-19 forecasting in the United States," *IEEE Open J. Eng. Med. Biol.*, vol. 2, pp. 235–248, 2021.

[2] Q.-V. Pham, D. C. Nguyen, T. Huynh-The, W.-J. Hwang, and P. N. Pathirana, "Artificial intelligence (AI) and big data for coronavirus (COVID-19) pandemic: A survey on the state-of-the-arts," *IEEE Access*, vol. 8, pp. 130820–130839, 2020.

[3] H. Friji, R. Hamadi, H. Ghazzai, H. Besbes, and Y. Massoud, "A generalized mechanistic model for assessing and forecasting the spread of the COVID-19 pandemic," *IEEE Access*, vol. 9, pp. 13266–13285, 2021.

[4] G. S. Leite, A. B. Albuquerque, and P. R. Pinheiro, "Applications of technological solutions in primary ways of preventing transmission of respiratory infectious diseases—A systematic literature review," *Int. J. Environ. Res. Public Health*, vol. 18, no. 20, 2021, Art. no. 10765.

[5] A. Mathew, F. A. Sa, H. Pooja, and A. Verma, "Smart disease surveillance based on Internet of Things (IoT)," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 5, pp. 180–183, 2015.

[6] R. P. Singh, M. Javaid, A. Haleem, and R. Suman, "Internet of Things (IoT) applications to fight against COVID-19 pandemic," *Diabetes Metabolic Syndrome, Clin. Res. Rev.*, vol. 14, no. 4, pp. 521–524, Aug. 2020.

[7] S. Paiva and F. Mourão, "Mobility-as-a-service challenges and opportunities in the post-pandemic," in *Proc. IEEE Global Conf. Artif. Intell. Internet Things (GCAIoT)*, Dec. 2021, pp. 136–141.
[8] V. Chamola, V. Hassija, V. Gupta, and M. Guizani, "A comprehensive review of the COVID-19 pandemic and the role of IoT, drones, AI, blockchain, and 5G in managing its impact," *IEEE Access*, vol. 8, pp. 90225–90265, 2020.
[9] K. Panwar, S. Pandey, K. Rawat, and N. Bisht, "Predictor system for tracing COVID-19 spread," in *Use of AI, Robotics, and Modern Tools to Fight COVID-19*, 2021, pp. 79–87.
[10] N. C. Peeri, N. Shrestha, M. S. Rahman, R. Zaki, Z. Tan, S. Bibi, M. Baghbanzadeh, N. Aghamohammadi, W. Zhang, and U. Haque, "The SARS, MERS and novel coronavirus (COVID-19) epidemics, the newest and biggest global health threats: What lessons have we learned?" *Int. J. Epidemiol.*, vol. 49, no. 3, pp. 717–726, Feb. 2020.
[11] D. S. W. Ting, L. Carin, V. Dzau, and T. Y. Wong, "Digital technology and COVID-19," *Nature Med.*, vol. 26, no. 4, pp. 459–461, Mar. 2020.
[12] C. J. Wang, C. Y. Ng, and R. H. Brook, "Response to COVID-19 in Taiwan: Big data analytics, new technology, and proactive testing," *Jama*, vol. 323, no. 14, pp. 1341–1342, 2020.
[13] M. Z. Hasan, T. Y. Hamed, and F. Al-Turjman, "Particle swarm optimization for adaptive social-distance of neighborhood in the IoT and COVID-19 era," in *Proc. Int. Conf. Artif. Intell. Things (ICAIoT)*, Sep. 2021, pp. 7–14.
[14] A. Khanfor, H. Ghazzai, Y. Yang, and Y. Massoud, "Application of community detection algorithms on social Internet-of-Things networks," in *Proc. 31st Int. Conf. Microelectron. (ICM)*, Cairo, Egypt, Dec. 2019, pp. 94–97.
[15] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social Internet of Things (SIoT)–when social networks meet the Internet of Things: Concept, architecture and network characterization," *Comput. Netw.*, vol. 56, no. 16, pp. 3594–3608, Nov. 2012.
[16] A. Khanfor, H. Ghazzai, Y. Yang, and Y. Massoud, "Application of community detection algorithms on social Internet-of-Things networks," in *Proc. 31st Int. Conf. Microelectron. (ICM)*, Dec. 2019, pp. 94–97.
[17] J. Shuja, K. Bilal, W. Alasmary, H. Sinky, and E. Alanazi, "Applying machine learning techniques for caching in edge networks: A comprehensive survey," 2020, *arXiv:2006.16864*.
[18] A. Khanfor, H. Friji, H. Ghazzai, and Y. Massoud, "A social IoT-driven pedestrian routing approach during epidemic time," in *Proc. IEEE Global Conf. Artif. Intell. Internet Things (GCAIoT)*, Dec. 2020, pp. 1–6.
[19] A. Khanfor, A. Nammouchi, H. Ghazzai, Y. Yang, M. R. Haider, and Y. Massoud, "Graph neural networks-based clustering for social Internet of Things," in *Proc. IEEE 63rd Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2020, pp. 1056–1059.
[20] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.
[21] W.-Y. Loh, "Classification and regression trees," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 14–23, 2011.
[22] M. Hatzopoulou, S. Weichenthal, G. Barreau, M. Goldberg, W. Farrell, D. Crouse, and N. Ross, "A web-based route planning tool to reduce cyclists' exposures to traffic pollution: A case study in Montreal, Canada," *Environ. Res.*, vol. 123, pp. 58–61, May 2013.
[23] A. Pimpinella, A. E. C. Redondi, and M. Cesana, "Walk this way! An IoT-based urban routing system for smart cities," *Comput. Netw.*, vol. 162, Oct. 2019, Art. no. 106857.
[24] I. C. Konstantakopoulos, A. R. Barkan, S. He, T. Veeravalli, H. Liu, and C. Spanos, "A deep learning and gamification approach to improving human-building interaction and energy efficiency in smart infrastructure," *Appl. Energy*, vol. 237, pp. 810–821, Mar. 2019.
[25] M. S. Rahman, N. C. Peeri, N. Shrestha, R. Zaki, U. Haque, and S. H. A. Hamid, "Defending against the novel coronavirus (COVID-19) outbreak: How can the Internet of Things (IoT) help to save the world?" *Health Policy Technol.*, vol. 9, no. 2, pp. 136–138, Jun. 2020.
[26] N. Petrović and Đ. Kocić, "IoT-based system for COVID-19 indoor safety monitoring," in *Proc. IcETRAN*, 2020, pp. 1–6.
[27] M. Fazio, A. Buzachis, A. Galletta, A. Celesti, and M. Villari, "A proximity-based indoor navigation system tackling the COVID-19 social distancing measures," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2020, pp. 1–6.
[28] A. M. Abd El-Haleem, A. I. Salama, M. G. Anany, M. M. Abdelhakam, G. A. Khalaf, and M. M. Elmesalawy, "IoT enabled geofencing-based system for monitoring and enforcing COVID-19 control measures in workplaces, service areas and distributed home quarantine," in *Proc. 9th Int. Conf. Electr. Electron. Eng. (ICEEE)*, Mar. 2022, pp. 290–295.
[29] O. Akbarzadeh, M. Baradaran, and M. R. Khosravi, "IoT-based smart management of healthcare services in hospital buildings during COVID-19 and future pandemics," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–14, Jun. 2021.
[30] R. Cantarero, A. Rubio, M. J. S. Romero, J. Dorado, J. Fernandez, and J. C. Lopez, "COVID19-routes: A safe pedestrian navigation service," *IEEE Access*, vol. 9, pp. 93433–93449, 2021.
[31] M. Kamal, A. Aljohani, and E. Alanazi, "IoT meets COVID-19: Status, challenges, and opportunities," 2020, *arXiv:2007.12268*.
[32] O. Karaman, A. Alhudhaif, and K. Polat, "Development of smart camera systems based on artificial intelligence network for social distance detection to fight against COVID-19," *Appl. Soft Comput.*, vol. 110, Art. no. 107610, Oct. 2021.
[33] A. T. Sausen, M. de Campos, P. S. Sausen, M. O. Binelo, M. F. Binelo, J. M. da Silva, and M. Dos Santos, "Classification of the social distance during the COVID-19 pandemic from electricity consumption using artificial intelligence," *Int. J. Energy Res.*, vol. 45, no. 6, pp. 8837–8847, 2021.
[34] A. Ramchandani, C. Fan, and A. Mostafavi, "DeepCOVIDNet: An interpretable deep learning model for predictive surveillance of COVID-19 using heterogeneous features and their interactions," *IEEE Access*, vol. 8, pp. 159915–159930, 2020.
[35] K. Kumar, N. Kumar, and R. Shah, "Role of IoT to avoid spreading of COVID-19," *Int. J. Intell. Netw.*, vol. 1, pp. 32–35, Jan. 2020.
[36] F. Amin, A. Ahmad, and G. S. Choi, "Towards trust and friendliness approaches in the social Internet of Things," *Appl. Sci.*, vol. 9, no. 1, p. 166, 2019.
[37] C. Marche, L. Atzori, and M. Nitti, "A dataset for performance analysis of the social Internet of Things," in *Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Bologna, Italy, Sep. 2018, pp. 1–5.
[38] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Exp.*, vol. 2008, Oct. 2008, Art. no. P10008.
[39] A. Khanfor, H. Ghazzai, Y. Yang, M. R. Haider, and Y. Massoud, "Automated service discovery for social Internet-of-Things systems," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Seville, Spain, Oct. 2020, pp. 1–5.
[40] S. Wang and R. Koopman, "Clustering articles based on semantic similarity," *Scientometrics*, vol. 111, no. 2, pp. 1017–1031, May 2017.
[41] H. Almeida, D. Guedes, W. Meira, and M. J. Zaki, "Is there a best quality metric for graph clusters?" in *Machine Learning and Knowledge Discovery in Databases*, D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, Eds. Berlin, Germany: Springer, Sep. 2011, pp. 44–59.
[42] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[43] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios," 2021, *arXiv:2108.11539*.
[44] P. S. Madhukar and L. B. Prasad, "State estimation using extended Kalman filter and unscented Kalman filter," in *Proc. Int. Conf. Emerg. Trends Commun., Control Comput. (ICONC)*, Feb. 2020, pp. 1–4.
[45] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 748–756.
[46] M. Zeng, J. N. Kumar, Z. Zeng, R. Savitha, V. R. Chandrasekhar, and K. Hippalgaonkar, "Graph convolutional neural networks for polymers property prediction," 2018, *arXiv:1811.06231*.
[47] G. Boeing, "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Comput. Env. Urban Syst.*, vol. 65, pp. 126–139, Sep. 2017.
[48] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, p. 440, 1998.

HAMDI FRIJI (Student Member, IEEE) received the National Diploma of Engineering degree (Hons.) in information and communication technology from the Higher School of Communication of Tunis (SUP'COM), University of Carthage, Tunisia, in 2020. He has worked as a Research Assistant with the Stevens Institute of Technology, Hoboken, NJ, USA, in 2020. He has worked as a Research and Development Engineer at iMaxeam. Additionally, his study area is the intersection of artificial intelligence, mathematical modeling, optimization, the Internet of Things, and graph theory.

ABDULLAH KHANFOR (Member, IEEE) received the master's degree in computer science and the Ph.D. degree in systems engineering with a concentration in software engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2016 and 2020, respectively. He is currently working as an Assistant Professor at the College of Computer Science and Information Systems, Najran University, Saudi Arabia. His research interests include the Internet of Things, graph theory, and machine learning. He has received the Exceptional Achievement Award from the School of Systems and Enterprises, Stevens Institute of Technology, in 2020.

HAKIM GHAZZAI (Senior Member, IEEE) received the Diplome d'Ingenieur degree (Hons.) in telecommunication engineering and the master's degree in high-rate transmission systems from the École Supérieure des Communications de Tunis (SUP'COM), Tunis, Tunisia, in 2010 and 2011, respectively, and the Ph.D. degree in electrical engineering from the King Abdullah University of Science and Technology (KAUST), Saudi Arabia, in 2015. He was a Research Scholar with the Qatar Mobility Innovations Center (QMIC), Qatar; Karlstad University, Sweden; and the Stevens Institute of Technology, NJ, USA. He is currently a Research Scientist with KAUST. He is the author and the coauthor of more than 150 publications. His general research interests include artificial intelligence enabled applications, the Internet of Things, intelligent transportation systems (ITS), mobile and wireless networks, and unmanned aerial vehicles (UAVs). He was a recipient of appreciation for an Exemplary Reviewer for IEEE Wireless Communications Letters, in 2016, and IEEE Communications Letters, in 2017. Since 2019, he has been on the Editorial Board of the IEEE Communications Letters and the IEEE Open Journal of the Communications Society. Since 2020, he has been with the Board of *IoT and Sensor Networks* (Specialty Section of *Frontiers in Communications and Networks*) as an Associate Editor.

YEHIA MASSOUD (Fellow, IEEE) received the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, USA. He has held several experience at leading institutions of higher education and respected industry names, including Rice University, the Stevens Institute of Technology, USA, WPI, UAB, the SLAC National Accelerator Laboratory, and Synopsys Inc. In 2003, he joined Rice University as an Assistant Professor, where he became one of the fastest Rice faculties to be granted tenure at the Department of Electrical Engineering and the Department of Computer Science, in 2007. From January 2018 to July 2021, he was the Dean of the School of Systems and Enterprises (SSE), Stevens Institute of Technology. Prior to Stevens, he has worked as the Head of the Department of Electrical and Computer Engineering (ECE), Worcester Polytechnic Institute (WPI), from 2012 to 2017. He is currently a Professor and the Director of the Innovative Technologies Laboratories (ITL), King Abdullah University of Science and Technology (KAUST), Saudi Arabia. He has published more than 400 papers in leading peer-reviewed journals and conference publications. His research interests include design of state-of-the-art innovative technological solutions that span over the broad range of technical areas, including smart cities, autonomy, smart health, embedded systems, nanophotonics, and spintronics. He was selected as one of the ten MIT Alumni featured by the MIT's Electrical Engineering and Computer Science Department, in 2012. He was a recipient of the Rising Star of Texas Medal, the National Science Foundation CAREER Award, the DAC Fellowship, the Synopsys Special Recognition Engineering Award, and several best paper awards. He has served as an Editor for the Mixed-Signal Letters—*The Americas*, an Associate Editor for the IEEE Transactions on Very Large Scale Integration (VLSI) Systems and the IEEE Transactions on Circuits and Systems—I: Regular Papers, and a Guest Editor for a Special Issue of the IEEE Transactions on Circuits and Systems—I: Regular Papers. He was named as a Distinguished Lecturer by the IEEE Circuits and Systems Society, from 2014 to 2015.