Efficient Detection of COVID-19 Exposure Risk

Brian T. Nixon
Dept. of Computer Science
University of Pittsburgh
nixon.b@cs.pitt.edu

Rakan Alseghayer
Dept. of Computer Science
University of Pittsburgh
ralseghayer@cs.pitt.edu

Benjamin Graybill
Dept. of Computer Science
University of Pittsburgh
bcg36@pitt.edu

Xiaozhong Zhang
Dept. of Computer Science
University of Pittsburgh
xiz151@pitt.edu

Constantinos Costa
Dept. of Computer Science
University of Pittsburgh
costa.c@cs.pitt.edu

Panos K. Chrysanthis
Dept. of Computer Science
University of Pittsburgh
panos@cs.pitt.edu

***Abstract*—In this demo paper, we present the new module of our *HealthDist* system that performs contact tracing in a privacy-preserving manner and considers the COVID-19 exposure risk. This is achieved by answering a new spatio-temporal query, dubbed *ST-Aggregate Join*, which calculates the COVID-19 exposure risk of an individual on their devices. It utilizes a special-purpose access structure to record the trajectories of users on their devices and optimize the *ST-Aggregate Join* processing. We demonstrate interactively using a smartphone application how our system can provide effective contact tracing within a university campus. We also illustrate how our new module is working through an intuitive web interface that shows the exposure risk of a person by coloring the trajectory of the infected person and the person(s) in high risk in a preloaded real dataset.**

***Index Terms*—indoor, outdoor, congestion forecasting, contact tracing, spatio-temporal aggregate join**

I. Introduction

COVID-19 is an airborne disease, which is highly contagious with the larger percentage of infected people not exhibiting symptoms. With the rise of new COVID-19 variants, such as Delta and Omicron, governments around the world have reiterated the need for face masks, social distancing, *contact tracing* (CT) and isolation [1]. The US Centers for Disease Control and Prevention (CDC)[1] has broadly defined the following three criteria for two people to be declared as came to *close contact* indoors and outdoors: (i) they both need to be within six feet away ($\approx$1.8 meters) without wearing masks; (ii) one of the two persons is infected and this is starting two days before the infected person has developed symptoms; and (iii) the cumulative total time is 15 minutes or more [2]. The first two requirements establish a contact whereas the third one captures the exposure to the virus.

Existing CT applications (e.g., [3], [4]) focus on a single contact with duration constraints while ignoring the cumulative duration of multiple contacts. As a result, these applications are unable to assess the exposure risk, measured as viral load or quanta accumulation over a period of time (e.g., a day). In this paper, we present a solution for efficient contact tracing, that considers both contacts and their cumulative duration, implemented as a new module of our innovative *HealthDist* system [5]. *HealthDist* was designed as part of the *CovidReduce*[2] project to implement a holistic approach of *proactive* (contact avoidance) and *retroactive* (contact tracing) functionalities without violating privacy. In the first version of *HealthDist* we implemented safe path recommendations and a scenario analysis tool [3], which can calculate the infection exposure based on the user's input. The new module, called *Exposure Risk Detection*, also uses the scenario analysis tool in calculating the exposure risk based on the contact cumulative durations and operates along the lines of SmartTrace [6] to preserve privacy and achieve scalability.

There are two forms of spatial intersection that must be considered when attempting to perform contact tracing: *(i) direct contact*, and *(ii) indirect contact*. Direct contact occurs when an infected and susceptible person share the same location, while indirect contact occurs when the two individuals share the same location at different times within a given interval. The new *Exposure Risk Detection* module needs to calculate an

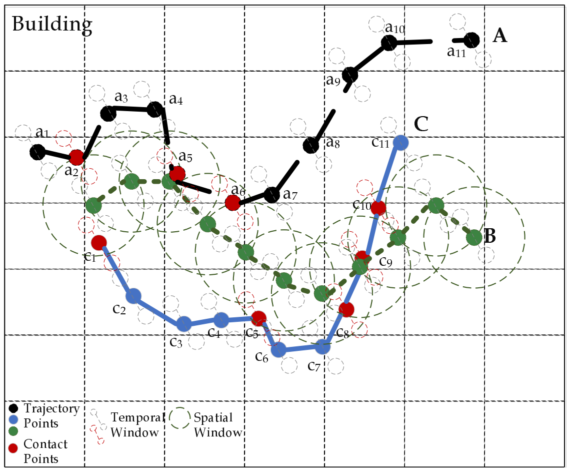


Fig. 1. Example of ST-Aggregate Join processing

[1]CDC: https://www.cdc.gov/

[2]CovidReduce: https://covidreduce.org

[3]Scenario Analysis: https://bit.ly/3HLhiJY

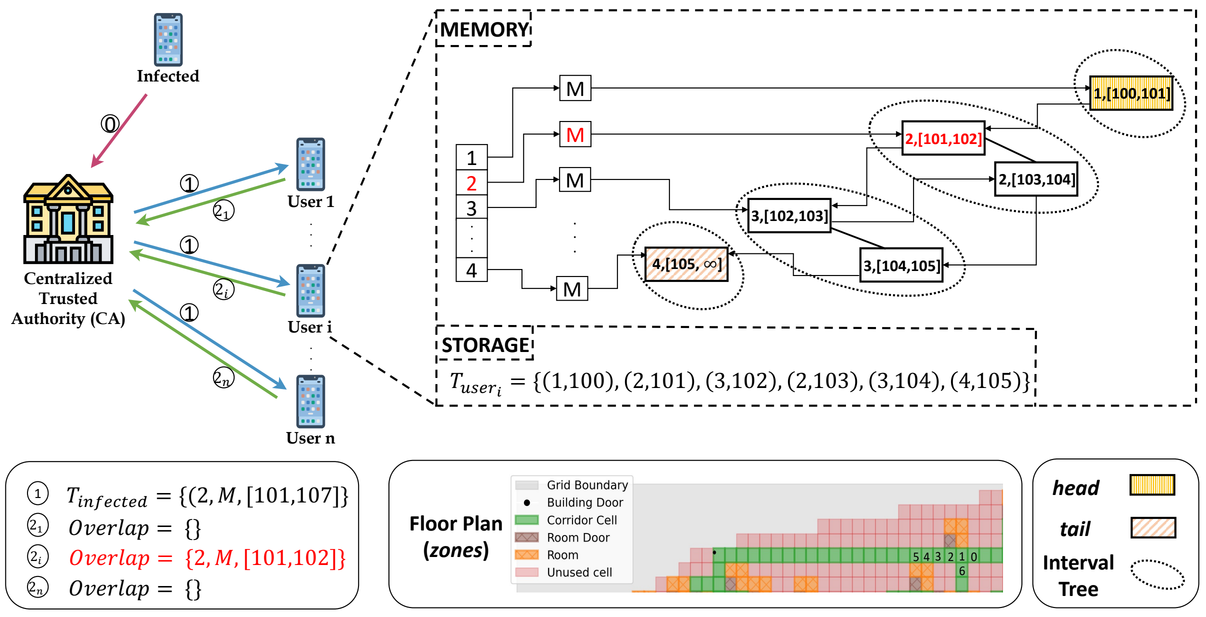


Fig. 2. e-Racoon access method memory and storage layout for a user (i.e., $user_i$) that has occupied the zones 1, 2, 3, and 4 on a Monday (M).

individual's total risk of COVID-19 exposure by considering the durations of all contacts both direct or indirect. This is achieved by answering a new spatio-temporal query, dubbed *ST-Aggregate Join*, which calculates the COVID-19 exposure risk of an individual on their devices.

The processing of *ST-Aggregate Join* is illustrated in Figure 1, where there are three trajectories $\{A, B, C\}$ representing three different people walking in a building. Each trajectory consists of multiple points (i.e., $A = \{a_1, \ldots, a_{11}\}$, $B = \{b_1, \ldots, b_{11}\}$, $C = \{c_1, \ldots, c_{11}\}$). Now assume that one person was infected with COVID-19 and the infected trajectory is B (i.e., the green dotted line). According to the CDC guidelines to find the contacts we need to consider a spatial window (i.e., the green dashed circles) and a temporal window denoted with the two non-filled points with a dashed outline and connected with a dashed line. By visually examining the example, we can easily identify the direct contacts, which are $\{a_5, c_9\}$ and the indirect contacts, which are $\{a_2, a_6, c_1, c_5, c_8, c_{10}\}$, but not the total duration of all the contacts. The accumulated duration of the contacts define the degree of exposure. In the example assuming the following, A has 3-second stops (i.e., a point with 3 seconds duration) and C has 1-second stops, A has the highest risk of exposure with accumulated duration of contact of 9 seconds, compared to C, which has 5 seconds.

ST-Aggregate Join processing is optimized by utilizing a special-purpose access structure, which is a modified version of the Racoon spatio-temporal index [7]. e-Racoon records the trajectories of users on their device using interval trees to encode the locations and the durations of the trajectories.

The rest of the paper is structured as follows. In Section II we introduce e-Racoon, which is a modified version of Racoon and describe how the *ST-Aggregate Join* works. Section III presents the new *Exposure Risk Detection* module and the mobile app, and Section IV describes the demonstration scenario.

II. System Architecture

The extended version of our system builds upon our previous architecture that consists of the *Data Layer*, *Processing Layer*, and *Application Layer* and incorporates the *Exposure Risk Detection* module to the Processing Layer [5].

The Data Layer is responsible for managing input data from various data sources such as local or distributed files, data streams, or external APIs.

The Processing Layer is responsible for the core functionalities and services of the system such as processing data for path recommendations, performing localization based on BLE devices, providing congestion information, and detecting potential exposure between intersecting trajectories.

The Application Layer represents the user interface and provides access to the Building Manager, BLE Manager, and Exposure Risk Detection Processing Layer modules for easy development with an open API. This layer utilizes a Leaflet JS map library and the LeafletPlayback plugin to abstract the complexity of the system from the user.

A. Infection Notification

If an individual tests positive for COVID-19, the user is expected to notify the system's Centralized Trusted Authority (CA) with their encrypted IDs and trajectories as shown in Figure 2. The trusted agent will periodically query users by sending the trajectories of infected individuals. Each user (local device) will determine the overall duration the individual was exposed to the infected trajectories and return this value to the CA along with their trajectories if found as infected.

B. e-Racoon Access Method

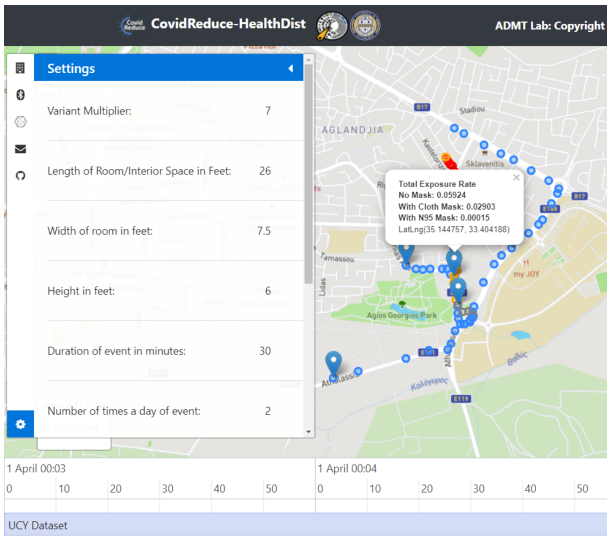
We extend the Racoon index presented in [7], to have the access method shown in Figure 2. e-Racoon facilitates efficient access to spatial and temporal information. We abstract spatial information in the system as *zones*, which are predefined areas in the Euclidean space. As a result, when a localization service samples a point at a user's device, the location coordinates are translated into a specific zone. Consequently, we represent time in our system by *intervals*, that captures the duration of occupancy in a certain zone. They are abstracted in the form: $[t_{arr}, t_{dep}]$, where t_{arr} and t_{dep} are the timestamps of arrival at and departure from a zone, respectively.

The access method is a multi-level index that stores zones and intervals. The first level is spatial, where we store zones that users occupy. The second level is a temporal level, which stores the days that users have occupied zones on. That is, accessing the first two levels leads to a pointer that points at a data structure that stores the time intervals during which a user occupied a particular zone on a particular day. We choose Interval trees to be the augmented data structure. It is implemented using a Red–black tree to achieve tree self balancing. At each node of the interval tree, we store the zone, the time interval, and three pointers. We store the interval in order to answer exposure measuring queries, two pointers are needed for the left and right children nodes of the Red–black tree, and the third pointer points at the node which contains the interval that temporally proceeds the stored one, hence we call it a *temporal* pointer. Note that the third pointer points at a node that exists in a different tree, since a user has to change their location to mark the time of departure of that zone.

We keep a global pointer *head* and a global pointer *tail*, where the head points at the very first node stored in the structure (first interval in the trajectory), and the tail points at the most recent node stored in the structure (latest interval in the trajectory). The most recent node will have an interval that has its t_{dep} marked as '∞' until a subsequent node is ready to be inserted.

The purpose of having all the nodes globally linked linearly is to achieve efficient spatio-temporal data streaming. That is, in case an infected user wants to send their trajectory to the CA, traversing the linearly linked nodes would make that task efficient. For example, in case $user_i$ in Figure 2 becomes infected, they need to send their trajectory data to the CA, then the CA would forward that trajectory to the rest of the non-infected users in the system. In that case, $user_i$ would traverse all the nodes through the global pointer 'head' and serialize the structure.

Figure 2 shows the memory and storage layout of $user_i$ where i is the ID of the user. We notice that the user occupied the zones 1, 2, 3, and 4 at different times. The raw trajectory points that are sampled by the device's localization service are stored in the secondary storage after translating the location information to zones. This is done for historical records purposes. When sampling consecutive points in the same zone (e.g., a user that is working in their office), we only store one node, and the end interval of it is marked '∞' until the users change their zone. Then, the timestamp of leaving that zone becomes the t_{dep} of the interval and the t_{arr} of the subsequent interval.



Settings

Variant Multiplier:	7
Length of Room/Interior Space in Feet:	26
Width of room in feet:	7.5
Height in feet:	6
Duration of event in minutes:	30
Number of times a day of event:	2

Total Exposure Rate
No Mask: 0.05924
With Cloth Mask: 0.02903
With N95 Mask: 0.00015
LatLng(35.144757, 33.404188)

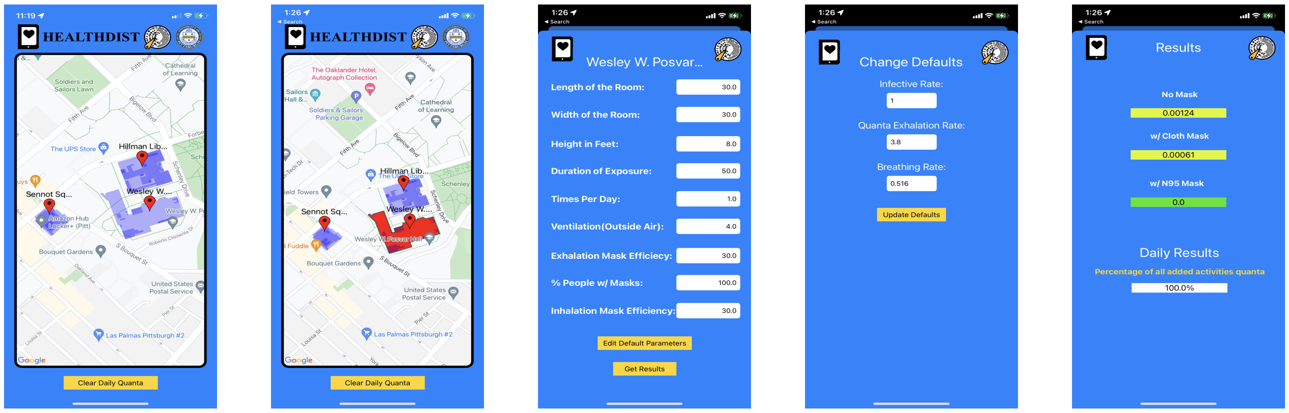
1 April 00:03 0 10 20 30 40 50
1 April 00:04 0 10 20 30 40 50

Fig. 3. The Exposure Risk Detection web interface shows the performance of our techniques and the COVID-19 exposure for three people (blue and orange trajectories) and one infected person (red trajectory) who are using public transportation in the University of Cyprus area.

The communication between the CA and the users is done as depicted in Figure 2. The CA sends the infected trajectory that was streamed to the CA in the format $\{(zone, day, [t_{arr}, t_{dep}]), ...\}$ as shown in the figure. However, in order to reduce energy consumption in communication and processing at the local devices, the communication of infected trajectories between the CA and users is carried out in two rounds. The first round is when the CA sends the zones and the days that belong to the infected user (i.e., $\{(zone, day), ...\}$). Then, each user in the system (excluding the infected ones) replies to that request by the zones and the days that overlaps with the infected one in the same format. Correspondingly, the second round starts by having the CA streaming the intervals in the format $\{(zone, day, [t_{arr}, t_{dep}]), ...\}$ for all the zones that the infected user occupied and overlapped with each particular user. Lastly, each user checks their in-memory structure for interval overlaps. In case an overlap occurs, the user sends back the zones and the days of the overlapping intervals in the same format. At the CA, the intervals are turned into durations of contact, which answers the exposure measuring query.

III. Implementation

The original prototype of *HealthDist*, consists of the *HealthDist Back-end*, *HealthDist Smart Clients*, and *HealthDist BLE Infrastructure*, which was developed using Play Framework 2.7 and MongoDB 4.4. We have extended the *HealthDist Smart Clients* to utilize the e-Racoon Index, which was developed in C++11. The *Exposure Risk Detection* web interface is implemented in HTML5/CSS3 along with extensive usage of Leaflet and the LeafletPlayback plugin.



Length of the Room: 30.0
Width of the Room: 30.0
Height in Feet: 8.0
Duration of Exposure: 50.0
Times Per Day: 1.0
Ventilation(Outside Air): 4.0
Exhalation Mask Efficiecy: 30.0
% People w/ Masks: 100.0
Inhalation Mask Efficiency: 30.0

Infective Rate: 1
Quanta Exhalation Rate: 3.8
Breathing Rate: 0.516

No Mask: 0.00124
w/ Cloth Mask: 0.00061
w/ N95 Mask: 0.0
Daily Results
Percentage of all added activities quanta: 100.0%

Fig. 4. (left) HealthDist Smart Client application for several University of Pittsburgh campus buildings, (left center) Smart Client application with an infection detected in a campus building, (center) user interface for entering information for a room in Posvar Hall, (right center) user interface for setting default values, (right) calculated risk of exposure using the parameters provided in the previous images.

1) Exposure Risk Detection Interface: The Exposure Risk Detection interface (shown in Figure 3) allows users to interact with a contact tracing scenario using trajectories from the SmartTrace dataset [6]. The red trajectory represents an infected individual while orange trajectories represent individuals with a high risk of exposure due to direct and indirect contacts with an infected person. Any individuals that were not in contact with an infected individual are displayed as blue trajectories and gray points indicating points that have not yet been visited. Users are able to select a trajectory to display its total exposure rate using the Bus Ride Half Density scenario from the Scenario Analysis tool of *HealthDist*. Users are able to visualize contacts spatially through the use of traces on the Leaflet JS map library and adjust the position of each trajectory as a function of time with the LeafletPlayback plugin.

2) Smart Clients: We have extended the *HealthDist Smart Clients* (shown in Figure 4) to utilize the results from the *ST-Aggregate Join* discussed in Section II to calculate a user's risk of exposure to COVID-19. In the two leftmost images of Figure 4, users are able to select a building for calculating their risk of exposure and any buildings with a reported case of infection are colored red. In the next central image, users are able to insert parameters used to calculate potential risk such as the number of times they visited the room and the percentage of people that were wearing masks. The duration of exposure is automatically inserted as a result of the *ST-Aggregate Join*, which returns a user's duration of exposure.

Finally in the rightmost image, the results of computing a user's potential risk based on the provided parameters and duration of exposure are calculated and displayed on the user interface. The results are categorized based on the type of mask that the user wears and a daily result is presented to indicate the total daily quanta for the user.

We further extended our Smart Clients by adding a new API endpoint to improve the scalability of the Building Management on local devices and support a spatial overlap query returning the geoJSON data of any buildings that overlap with the device's viewport.

IV. Demonstration Scenario

During the demonstration, attendees will be able to visualize the *E*xposure Risk Detection module through a user-friendly interface and the benefits of utilizing our novel index for calculating the COVID-19 exposure risk.

Equipment: The conference attendees will have the opportunity to interactively engage with the exposure risk detection web-interface using laptops, tablets, and smartphones.

Datasets: We will pre-load a variety of datasets collected from the University of Pittsburgh (PITT) and the University of Cyprus (UCY) to illustrate how the exposure risk detection works using our interactive map based interface. PITT dataset consists of six trajectories of 27 points on average, and UCY consists of four trajectories of 70 points on average.

Scenarios: A *HealthDist* server will be available to allow attendees to interact with the original prototype and its web interface, and the new mobile application for exposure risk detection.

Acknowledgments: This work was partially funded by NIH award U01HL137159 and NSF grant SES-2017614. The findings and expressed opinions are only those of the authors.

References

[1] N. Ahmed, R. A. Michelin, W. Xue, S. Ruj, R. Malaney, S. S. Kanhere, A. Seneviratne, W. Hu, H. Janicke, and S. K. Jha, "A survey of covid-19 contact tracing apps," *IEEE Access*, vol. 8, 2020.
[2] T. U. C. for Disease Control and Prevention. Covid-19 close contact. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/daily-life-coping/determine-close-contacts.html
[3] T. Altuwaiyan, M. Hadian, and X. Liang, "Epic: Efficient privacy-preserving contact tracing for infection detection," in *IEEE ICC*, 2018.
[4] M. O. Cruz, H. Macedo, and A. Guimarães, "Grouping similar trajectories for carpooling purposes," in *BRACIS*, 2015.
[5] C. Costa, B. T. Nixon, S. Bhattacharjee, B. Graybill, D. Zeinalipour-Yazti, W. Schneider, and P. K. Chrysanthis, "A context, location and preference-aware system for safe pedestrian mobility," in *IEEE MDM*, 2021.
[6] D. Zeinalipour-Yazti, C. Laoudias, C. Costa, M. Vlachos, M. I. Andreou, and D. Gunopulos, "Crowdsourced trace similarity with smartphones," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, 2013.
[7] R. Alseghayer, "Racoon: Rapid contact tracing of moving objects using smart indexes," in *IEEE MDM*, 2021.