Contact Tracing Query Processing on Spark

MOHAMMED ABDALLA, Faculty of Computers and Artificial Intelligence, Beni-Suef University, Egypt
ABDELTAWAB HENDAWI, Department of Computer Science and Statistics, University of Rhode Island, USA

Infectious disease outbreaks like COVID-19, Spanish Flu, and Ebola represents a threat to governments and medical sectors worldwide due to millions of deaths caused by these diseases. Besides, infectious diseases reduce economic growth due to fear of workers going to their workplaces. As a result, interventions to reduce the onward transmission of infectious diseases are a must. In this regard, the contact tracing process can be considered a mitigation tool that is used in fighting the spread of infectious diseases. Meanwhile, the users’ traces data-sets grow exponentially with time; processing contact tracing queries over these data sets is a big challenge. This paper proposes a novel technique to process contact-tracing queries using Spark. This technique analyzes users’ trajectories based on two factors named; exposure time and proximate social distance to reveal if they are suspected cases. Besides, the designed technique takes full advantage of the spark framework to handle scalability challenges for answering contact tracing queries over a large scale of trajectories efficiently.

Additional Key Words and Phrases: Big Data, COVID-19, Spark, Contact Tracing, Infectious Diseases, GPS Traces.

ACM Reference Format:
https://doi.org/10.1145/nmnnnn.nmnnnn

1 INTRODUCTION

A record-breaking outbreak of the illness Coronavirus sickness 2019 occurred in December 2019, mainly in Wuhan, China (COVID-19). According to the World Health Organization (WHO), COVID-19 is an infectious illness that can spread rapidly and constantly, impacting millions of individuals around the world. In response to this danger, the World Health Organization (WHO) issued a global emergency warning about this rare virus, urging everyone to take all necessary safeguards. Around 500 000 individuals are estimated to have perished as a result of COVID-19, and an additional 10 million people are currently battling for survival, according to a May 2020 report from WHO. It is quite easy for COVID-19 to spread from one person to another and the illness is very contagious. Therefore, for months, governments all over the world implemented lockdown measures to combat the coronavirus’s spread. However, these metrics have had a huge impact on social and economic activities around the world[33, 43].

The Spanish flu, which killed between 20 and 100 million people in 1918, caused a huge decline in the gross domestic product in the following years. COVID-19 is not the first sickness to strike people worldwide. Countries Australia (3%), Canada (15%), the United Kingdom (17%), and the United States (11%)[35, 46]. In conclusion, the pandemic epidemic has an impact on economic growth and causes both long-term and short-term shocks. The pandemic outbreak’s effects also

Authors’ addresses: Mohammed Abdalla, Faculty of Computers and Artificial Intelligence, Beni-Suef University, Egypt. Abdeltawab Hendawi, Department of Computer Science and Statistics, University of Rhode Island, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2022 Association for Computing Machinery.
Manuscript submitted to ACM
Manuscript submitted to ACM
change people’s behavior, such as making them afraid to go to work, which reduces productivity through decreased production.

Indeed, pandemic outbreak diseases need significant actions and interventions to decrease the transmission spread of the infection between individuals. Hence, the contact tracing process can be considered one of the intervention methods that help in fighting the transmission of infectious diseases. Moreover, medical sectors and governments utilize the contact tracing process as a mitigation and monitoring tool to capture the root of infectious diseases and therefore can control the outbreak in an adequate time. More precisely, contact tracing starts by examining the network of contacts that communicated with the infected patient during the disease incubation period, based on the examination results, decision-makers in medical sectors can identify high-risk and low-risk contacts. Besides, semantic models of contact tracing support the analysis of infection behavior and potential paths of disease spread which helps to make a better understanding of pandemic hot-spots, and reasoning, therefore these lead to accurate recommendations to tackle major pandemic issues and management.

This paper proposes a technique that traces the contacts who are exposed to the infected patients and based on the examination results the technique identifies the high-risk and low-risk contacts and asks them to perform needed diagnostic tests. The proposed technique classified the contacts that communicate with the infected patient into two types: (1) close, and (2) proximate contacts. This classification is occurred based on two identified factors: (1) exposure time and proximate social distance. Additionally, this technique considers the incubation period as a tracing period, during this period, the technique needs to trace and examine all contacts exposed to the infected patient. Besides, this technique enables two types of tracing: direct contact tracing and in-direct contact tracing. In the case of direct contact tracing which is also named first order tracing[18, 23], the technique tries to trace the first level of contacts that make a direct physical connection with the infected patient, and therefore these individuals require self-isolation and specific medical care handle. In the case of in-direct contact tracing which is also named iterative contact tracing[18, 23], the tracing is done in an iterative manner (multi-level tracing), which means that the contacts of contacts will be also examined.

Initially, the technique performs the contact tracing process by starting to collect all trajectories of contacts moved nearby the infected patient during the incubation period. Next, based on these trajectories, the technique starts to identify if these contacts satisfy the conditions of exposure time and social distance identified values to classify these contacts as close or proximate contacts exposed to an infected patient. The technique’s final output alerts the user to perform quick diagnostic testing by returning the suspected contacts and their trajectories.

As regards tackling large-scale of big data traces needed for processing contact tracing queries, traditional methods fail to handle big datasets of traces. For this purpose, the proposed technique employs GeoSpark[60] framework to handle dealing with a large scale of traces big datasets. For processing massive amounts of spatial data, GeoSpark is an extension of the Spark framework for in-memory cluster computing. Additionally, the GeoSpark makes use of spatial data indexing components to divide the input Spatial RDD (Resilient Distributed Data sets) into grids and allocate the grids to different computers for parallel execution.

**Contributions.** We make the following contributions:

1. We develop a technique for addressing contact tracing inquiries over large data sets.
2. We develop a method that differentiates between close contacts and proximate contacts who are exposed to the infected patient.
(3) We introduce two tracing modes; first-order tracing and iterative tracing to catch either low-risk contacts or high-risk contacts or both of them.

(4) We use GeoSpark as an in-memory cluster computing system to process enormous spatial data traces that are necessary for effectively responding to contact tracing inquiries.

(5) We confirm the scalability and interactive performance of the proposed solution by implementing a spatial data index (R-Tree) using Geo-Spark that divides the input Spatial RDD into grid segments using a grid structure and allots grids to machines for parallel processing.

(6) On actual datasets, we run thorough experiments. Experimental findings demonstrate that our approach accomplishes run-time optimizations.

Roadmap. The remainder of this paper is structured as follows. The previous research in the field of contact tracing is described in Section 2. The problem is properly defined in Section 3 together with all the foundational ideas. Section 4 details the suggested resolution. Our suggested solution is empirically assessed in Section 5. The paper is concluded in Section 6.

2 RELATED WORK

This section thoroughly examines earlier research on the contact tracing procedure. The three main topics covered in this paper are the value of contact tracing, contact tracing applications and unresolved issues in contact tracing procedures.

<table>
<thead>
<tr>
<th>Paper</th>
<th>Rational</th>
<th>Information Sources</th>
<th>Keywords</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>[19, 29]</td>
<td>Introduce pairwise-approximation and fully random simulation methods to catch suspicious cases with infection. These methods focus on analysing disease-transmission pathways for each person by examining the network of contacts belonging to each suspicious person.</td>
<td>Synthetic dataset</td>
<td>COVID19; contact tracing; query processing; decision support systems</td>
<td>1-2021 2.2003</td>
</tr>
<tr>
<td>[1, 50]</td>
<td>Highlight and define community surveillance roles and guidelines to control the spread of COVID-19 disease.</td>
<td>World Health Organization (WHO) about COVID-19 and SARS Epidemic.</td>
<td>COVID-19, Crisis management; Pandemics; Medical services; Big Data; Harmonic analysis</td>
<td>1-2020 2.2021</td>
</tr>
<tr>
<td>[10, 16, 54]</td>
<td>Developed an Ebola transmission model which adopted contact tracing processes. This model highlights the perfect timing for contact tracing actions to be done.</td>
<td>World Health Organization (WHO) reports of Ebola Virus Disease (EVD) in the forested rural region of south eastern Guinea.</td>
<td>Contact tracing; Ebola; compartmental model; activity-driven network; epidemic model</td>
<td>1-2015 2.2012 3.2018</td>
</tr>
</tbody>
</table>

2.1 Importance of Contact Tracing

There is a large volume of published studies [1, 4, 10, 16, 19, 29, 41, 42, 47, 50, 55] describing the role of contact tracing processes.

In [19, 29], to identify possible cases of infection, the authors suggest a number of contact tracing techniques, including fully random simulations and pairwise approximations. These techniques are more focused on studying...
disease-transmission pathways for each individual by looking at the contacts that each suspect person has. Finally, this study found a substantial correlation between the accuracy of modeling contact tracing and the disease reproduction ratio, which will make it easier for medical decision-makers to identify infected and suspicious cases and treat them right away or isolate them.

In [1, 50], to stop the COVID-19 disease from spreading, the authors identify and specify community surveillance roles and procedures. Additionally, the authors came to the conclusion that implementing COVID-19 contact tracing and quarantine will aid in preventing the disease from spreading from one person to another by nasal or oral spray, exhalation, or coughing of COVID-19 sick individuals.

The hazards associated with the Ebola virus, which can spread from one person to another through contact with an infected individual’s bodily fluids, were looked at by the authors. The authors next investigated how to use the contact tracing procedure to carry out surveillance for individuals who had been exposed to sick patients for 21 days. As a result, a contact tracing process-based Ebola transmission model was created. Additionally, this approach emphasizes the ideal moment to carry out touch-tracing actions. The findings showed that hospitalization and rapid contact identification can cause a significant decrease in the epidemic spread to 50% compared to delayed identification [10, 16, 55].

The authors present two models that use stochastic screening and contact tracking to combat the HIV epidemic. These models are the staged-progression model and the differential infection model (DI) (SP). The SP model examines the time differences for the same infected person, whereas the DI model analyses the differences between the persons throughout the disease transmission period. The contact tracing tactics are highly effective at tying up disease super-spreaders, according to results from the DI model, and this can enable a reduction in the epidemic’s overall magnitude [4, 41, 42, 47].

In [12], to determine the qualities of successful tracing programs, the authors analyze contact tracing effectiveness. They stated that the number of infected people, the speed at which new cases are tested and isolated, the number of their contacts who are traced and quarantined, the speed at which those contacts are quarantined, and the efficiency of quarantines in stopping further transmission all affect how effective contact tracing is in the community.

Generally, the contact tracing process plays the main role in controlling COVID-19 spread, but most protocols and applications only apply forward-trace to notify people who were recently exposed. In [44], the authors propose a stochastic branching-process model which depends on bidirectional tracing to identify infected individuals which leads to improving outbreak control.

In [45], the authors presented a novel framework for assessing manual and digital contact tracing. The framework assessment measures the effectiveness of mitigating the pandemic. Indeed, this assessment for either digital or manual contact tracing needs COVID-19 testing and return of results, engagement of communities, and high adherence with physical distancing and the use of masks.

Table 1 provides a summary of the key publications’ objectives and descriptions in this approach.

2.2 Contact Tracing Applications And Domain of Use

More recent attention has focused on the provision of contact tracing applications to control the disease outbreak. As a result, there has been an increasing amount of literature on proposing various contact tracing applications [9, 40, 48, 52, 59].

In [18, 24], The authors outline a number of methods for carrying out the contact tracing process. First-order tracing, single-step contact tracing, iterative contact tracing, and retrospective contact tracing are some of these techniques. The first order tracing process identifies the people who come into direct physical contact with the patient right away; these people need to be isolated and given particular medical treatment. Furthermore, after the first order has been
completed successfully, the second order tracing operation is started, which is unconcerned with tracking the contacts of the contacts. The single-step contact tracing approach detects all contacts of the proven sick person, classifies all of these contacts as infected, identifies their contacts, and then repeats the process for all of these contacts. The goal of the Iterative Contact Tracing process is to find the infection before it manifests itself by doing diagnostic tests on each individual in an iterative manner. These tests include symptom screening. The final type is the retrospective contact tracing process. It has the same phases as the single-step or iterative procedures, but it also looks into the infected person’s recent past and works backward to identify everyone with whom the patient comes into touch. The goal of the retrospective contact tracing process is to determine who first contracted the illness.

In [48, 59], the Australian government has introduced an app called COVIDSafe to stop the epidemic of COVID-19 from spreading. The predicted individuals who are exposed to the pathogen are informed via this application to take the necessary emergency safeguards.

In [40], the authors create a cutting-edge peer-to-peer smartphone app that conducts contact tracing while also protecting user privacy by not logging any personal information or location data. This application’s goal is to stop the COVID-19 epidemic from spreading further.

In [9], the Singaporean government launched TraceTogether on March 20, 2020, to track down contacts who interact with COVID-19 sufferers. When two users of the program are close to one another and a user alerts that he is a COVID-19 patient, the application starts to notify the Ministry of Health to identify the network of contacts logged to be nearby them. This application uses Bluetooth technology to trace contacts. The program also interacts with those contacts and reports them with the appropriate follow-up steps.

In [52], to manage Ebola-infected people in Sierra Leon, the authors created the Ebola Contact Tracing application (ECT app), a smartphone application. In order to notify the response centre about symptomatic people and their districts, this application is coupled with an alert system.
In [39], the authors create a cutting-edge peer-to-peer smartphone application that successfully completes the contact tracking process by leveraging the potential of smartphone application technology. The authors discuss the need for contact tracking, which is a crucial step in limiting the spread of the COVID-19 pandemic. The major goal of this study is to create a smartphone app that does contact tracing while also protecting user privacy by not obtaining any personal data or location information.

In [15], a small program was created by the authors and integrated into the WeChat application. It examines user data across the board and tracks all patients’ close contacts. In fact, doing so enables quick identification and containment of potential infection sources. Additionally, the information from the mini-program can be combined with data from other sources to forecast epidemic trends, estimate population and individual risks, and make suggestions for protective measures to be taken by individuals and the population.

Public health workers interview affected people during traditional contact tracing to determine contacts. The exposed or suspected infected contacts are then urged to monitor their own health for signs, self-quarantine, or seek medical attention. On the other hand, digital contact tracing makes use of electronic data to pinpoint those who have been exposed to an infection. Digital contact tracing does, in fact, address the drawbacks of conventional contact tracing, including scalability, notification delays, recall mistakes, and contact identification in public areas. In [39], digital contact tracing’s two main drawbacks are solved. The efficiency of a contact-tracing application depends on the level of adoption, which is the first restriction. For instance, when both the exposed and infected individuals have their phones close by and are running interoperable applications, only then can the program recognize contacts. Furthermore, the majority of contact-tracing programs have lost some of their usefulness in areas where smartphone ownership is restricted, where people share cell phones, or when users are unable or unwilling to utilize an application. The second drawback is measurement inaccuracies in the underlying technologies, which reduce the ability of the apps to discover interactions. For instance, it can be difficult for Bluetooth-based applications to estimate the distance between devices using signal strength. Because walls and floors impede Bluetooth signals, exposures are frequently misidentified. The accuracy of location-based measures is also constrained; during outdoor testing on a campus of an American university, location identification using an iPhone 6 with Wi-Fi enabled showed root-mean-square errors ranging from 3 to 16 meters at various testing locations and times.

In [34], the authors discussed three common tracing architectures for contact tracing applications: centralized, decentralized, and hybrid. Also, an overview of popular applications within these categories is discussed. Finally, the privacy and security concerns that could be possibly performed in each of the three architectures are discussed. In the centralized architecture, first, a user must pre-register with a central server. Then, the server generates a privacy-preserving Temporary ID (TempID) for each device. This TempID is then encrypted with a secret key (known only to the central server authority) and sent to the device. After that, the Devices exchange these TempIDs (in Bluetooth encounter messages) when they come in close contact with each other. Once a user tests positive, they can volunteer to upload all of their stored encounter messages to the central server. The server maps the TempIDs in these messages to individuals to identify at-risk contacts. On the other side, the decentralized architecture proposes to move core functionalities to the user devices, leaving the server with minimal involvement in the contact tracing process. The hybrid design also suggests dividing these functionalities between the server and the devices. To be more precise, in order to maintain privacy and anonymity, TempID generation and management must remain decentralized (i.e., managed by devices), whilst risk analysis and alerting should fall within the purview of the centralized server.

Table 2 provides a summary of the key publications’ objectives and descriptions in this approach.
2.3 Semantic Modeling and Patterns in Contact Tracing

In general, contact tracing is a process that public health officials are used to slowing down the spread of infectious diseases like COVID-19. As mentioned above, there are two types of contact tracing methods: digital and traditional contact tracing. However, traditional contact tracing has some limitations as significantly time and labor-intensive because this process mainly depends on interviewing and diagnosing cases with their related contacts and locations visited in a manual manner. On the other side, the digital contact tracing process still has some limitations related to privacy and security concerns. For this purpose, Ontology-based solutions and semantic modeling techniques can come into the picture to cover this gap by defining formal representation of infectious diseases like Covid-19. This representation includes symptoms, suspected infectious cases, drug modeling, and infections rate. Indeed, Ontology-based solutions work by integrating various pandemic data Sources which make a better understanding of pandemic hot-spots, reasoning, and knowledge-driven inference, therefore these lead to accurate recommendations to tackle pandemic major issues and management of smart lockdowns[25].

Overall, pandemic data modeling includes infection spread rates, infection tracing, and social distancing. This modeling helps in simulating and analyzing pandemic spread and impact in countries and communities. In [7], the authors collected data from public health and surveillance units for modeling pandemics’ long-term scenarios. While the authors in [37] introduced multi-disciplinary research models for infection modeling and predicting the pandemic’s spread.

In [14], the authors propose different ontology-based techniques to conceptualize infection symptoms or patients’ health and identify the main relationships between infection symptoms and patients’ health which finally leads to automated reasoning about data. In [14], the authors introduced the definition of Infectious Disease Ontology (IDO) which is an integration of interoperable ontology modules that cover all aspects of the infectious disease domain such as public health, clinical care, and biomedical research. Besides, IDO provides the basics of ontology-based reasoning for supporting COVID-19 use cases. In [58], the authors extend the concept of IDO for automated reasoning and data representation. This extension allows the monitoring of indoor people safely and analyzes patients’ data about the COVID-19 pandemic.

In [17], authors perform another extension for IDO named COVID-19 Ontology for cases and patient information (CODO) to represent infected patients’ data for creating a network that supports the infection behavior analysis of the disease, potential paths of disease spread and several factors of disease transmission.

Finally, the authors in [38] summarized that the ontology-based solutions have several benefits such as: (1) structural representation of infectious diseases data, (2) infectious diseases data semantics and (3) support predictive analytics for pandemic spread by providing behavioral analysis of pandemic data.

2.4 Open Directions and Challenges

Numerous studies illustrate the primary difficulties in using contact tracing in practical settings.

In [6, 31, 36, 56], the authors identify privacy concerns as one of the issues that prevent contact tracing apps from operating to their full potential. They examine the effects of these issues and offer strategies to address them. These are the privacy issues as summarised by the authors: Keeping contacts’ private protected from prying eyes, keeping contacts’ privacy protected from authorities, and protecting contacts’ privacy.

According to the authors in [5, 49, 53], the contact tracing procedure has the following difficulties: (1) Inability to fully identify contacts, (2) Lack of reporting systems adopted on papers, (3) Incomplete contact lists, (4) Inefficient
data collection, (5) Transcription errors during the early-stage of contact tracing, specifically during the contact’s identification, and (6) Delays in identification and isolation of the suspected individuals from these contacts.

In [3, 13], the authors point out various difficulties that arise during the contact tracing process, such as the requirement for a lot of work to efficiently and precisely trace the contacts. For instance, (1) the Canadian Health Sector opens the door for a volunteer workforce to carry out the tasks of contact tracing and other pertinent tasks, and (2) at the height of the COVID-19 disease outbreak in South Korea, the government employed a sizable number of medical staff and other workers to trace contacts and control this epidemic.

This study sets itself apart from the studies mentioned above by being one of the first attempts that perform the contact tracing queries using GeoSpark which helps in the execution of these queries on the large-scale data efficiently and in responsive time, this helps to make the system scale up with high performance. Additionally, this work defines three different types of infectious contact resulting from the query to identify the accurate risk level for these contacts, these types are (1) physical contact, (2) close contact, and (3) proximate contact. Finally, this work ensures the accurate results of the tracing process by investigating all checkpoints between the infected patient’s trajectory and other trajectories according to flexible adjustment of the exposure time and social distance parameters.

3 BACKGROUND

The concepts related to the contact tracing method that will be employed in the remaining sections of the paper are presented in this section.
Table 3. R0 Example (10 cases infect 2.1 contacts)

<table>
<thead>
<tr>
<th>Cases</th>
<th>#InfectedCases</th>
<th>Total # Infected</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>-</td>
<td>21</td>
</tr>
</tbody>
</table>

3.1 Overview

The contact tracing process is a tactic for monitoring and confirming the control of infectious illnesses. Additionally, since documenting the spread of infectious disease is this method's primary goal, it begins by looking into previously identified cases and confirming their infection to anybody who recently had physical contact with the patients. Because of this, the spread of infectious diseases is reduced as a result of this procedure: like the COVID-19 epidemic.[2, 11, 18, 26, 32, 54].

The contact tracing process identifies two types of contacts that need to be tackled during the incubation period: high-risk contacts and low-risk contacts. The incubation period is the time from when someone is infected until symptoms develop. Indeed, the high-risk contacts are the closest contacts with the highest risk for infection, such as; household members and people who spend significant time together (meals, travel, work-spaces). While the low-risk contacts are asymptomatic people who are less infectious than people who develop symptoms.

Several parameters need to be measured along the contact tracing process, these parameters include: Risk of transmission, infection fatality ratio (IFR), Generation time, and Serial interval. Indeed, the Risk of transmission, is determined by; (1) Opportunities of infection to transmit (how many contacts do they have?), and (2) intensity of the contact (proximity, duration). While the IFR can be defined as the proportion of infected people who die. Moreover, the Generation time is the time from the infection in one generation to the time of infection in the next generation; for example, 5 days. This is a measure of how quickly the outbreak will grow.

3.2 Measuring the Impact of Contact Tracing

The influence of the contact tracing technique is examined in this section.

Indeed, the Reproductive Number is used as an indicator that measures the impact of the contact tracing. More specifically, two parameters are used to evaluate the impact of contact tracing; (1) the basic reproductive number (R0), and (2) the reproductive number (R).

The basic reproductive number (R0), is the number of people one infectious person will infect if everyone that person has contact with is susceptible. The higher the basic reproductive number, the more people will be infected, R0 is 1 means that the new cases are constant as shown in figure 1. It is critical to note that R0 can hide the transmission, if R0=2.5, it does not mean that every person infects 2.5 others. For example, in table 3, many people never infect anyone else as shown in the first row, also some infect only 1 other person as shown in the second row and some infect a couple of people as shown in the third row. According to table 3, the final result of R0 is $R_0 = 21/10 = 2.1$, this means 10 cases will infect an average of 2.1 contacts. To sum up, R0 is the average number infected if all contacts are susceptible.
Reproductive number ($R$), is an indicator of transmission after interventions. Interventions include contact tracing, investigating cases, and treating susceptible and infected persons. If not everyone is susceptible, it will bring down $R$.

The difference between $R_0$ and $R$ is a good way to measure the impact of contact tracing programs.

**Example:** Figure 2 illustrates how to measure the impact of the contact tracing processes. Let’s assume that $R_0 = 2$, this means that each person infects 2 more on average. Interventions can reduce 1 infection at each step, and therefore this will change the overall size of the outbreak.

### 3.3 Definitions

This section presents the preliminary concepts that will be used throughout the rest of the paper.

**Definition 1:** Exposure Duration $T_{\text{Exposure}}$ is the amount of time that a healthy person is exposed to an infected person. After this duration, the normal individual becomes a risk-contact.

**Definition 2:** Proximity Distance $P_{\text{Dist}}$, is the minimum distance at which a healthy person can be found close to an infected person. Through this distance, the normal individual becomes a risk-contact.

**Definition 3:** Infectious contact can be, (1) Physical, (2) Close (within $P_{\text{Dist}}$ for time least equal to $T_{\text{Exposure}}$), and (3) Proximate (at more than $P_{\text{Dist}}$ but in an enclosed area for a time greater than $T_{\text{Exposure}}$).

**Definition 4:** Contact Point $P_{\text{Contact}}$, is the location where a healthy person and an infected person come into contact at the same time, $T_{\text{Diff}}(\text{infected, normal}) = 0$.

**Definition 5:** Direct Tracing, is the detection of contacts during a certain tracing time who have a direct physical relationship to an infected person, these discovered contacts are named as high-risk contacts.

**Definition 6:** In-Direct Tracing, is the detection of contacts during a certain tracing time who have a direct physical relationship to an infected person, these discovered contacts are named as low-risk contacts.

**Definition 7:** Contact Tracing Query (CTQ), is the search that locates the people who have been in contact with the infected patient (query user) either directly or indirectly within certain $T_{\text{Exposure}}$ and $P_{\text{Dist}}$. Contact tracing query example like select all contacts connected to infected individual either in (Physical, Close, Proximate) way where exposed time between them $\geq T_{\text{Exposure}}$ & connection distance within $P_{\text{Dist}}$ range & tracing mode equal (direct or indirect).

### 4 PROBLEM STATEMENT

This section presents the problem statement.

Given a tracing period $\text{Tracing}\_\text{period}$ which reflects the duration of the disease’s incubation, exposure time $T_{\text{Exposure}}$, proximity distance for exposure $P_{\text{Dist}}$, infected patient ($Q_{\text{user}}$) with trajectory $\tau_Q$ and set of several trajectories belonging to other objects may be in contact or not with infected patient $\tau_{\text{Others}}$. Each trajectory either $\tau \subset \tau_{\text{Others}}$ or $\tau_Q$ is represented by an ordered set of locations paired with a logged timestamp at this location; $\tau = ((l_1, t_1), (l_2, t_2)\ldots(l_n, t_n))$, longitude and latitude information for this place are used to represent every location. It is needed to generate a list of all suspected contacts $C$ either close or proximate contacts or both who are exposed to the $Q_{\text{user}}$ during $\text{Tracing}\_\text{period}$.

### 5 PROPOSED SOLUTION

This section presents the proposed solution for fetching a list of close and proximate contacts who are exposed to the infected patient and needs immediate diagnostic tests.

Manuscript submitted to ACM
5.1 Main Idea

The major goal of the paper is to go back and locate all suspected close or distant contacts who have been exposed to the infected person during an incubation period. First, the patient acts as the query user who marks himself as infected. Then, the system starts to backtrack all contacts who are exposed to him during the incubation period and retrieves all moved trajectories belonging to these contacts. After that, the solution starts to investigate each trajectory against the infected patient trajectory to identify the proximate distance and exposure time between each contact and the infected patient. When the proximate distance and exposure time are identified, the solution can report if the exposed contact is a suspected case that needs an immediate diagnoses test or not. It is critical to note that this solution is mainly designed to perform contact tracing queries that process massive big data of contacts and return efficient responsive answers.

5.2 Solution Architecture

This section presents a detailed description of the proposed solution.

The proposed solution mainly depends on the GeoSpark framework in answering contact-tracing queries and managing the storage and processing of the large-scale traces for infected patients. Indeed, GeoSpark is a cluster-computing framework for processing spatial big data. Employing GeoSpark as a part of our proposed solution makes the overall designed solution overcome the two big challenges during the answering of the contact-tracing queries and storing the large scale of the infected patients’ traces. The first challenge is scalability, the underlying database can store effectively Petabytes of infected patient traces, and allows the proposed solution to process it efficiently. The second challenge is an interactive performance, the response of answering the contact-tracing queries must occur in a sub-second response time.

Figure 4 depicts the GeoSpark architecture that is employed inside our proposed technique to answer contact tracing query, this architecture mainly consists of three major layers: (1) Apache Spark Layer, (2) Spatial Resilient Distributed Dataset (SRDD) Layer, and (3) Spatial Query Processing Layer.

Apache Spark Layer. The responsibility of the Apache Spark Layer is to provide the operations and functions that handle natively the tasks of loading/saving user’s traces from/to persistent storage. Unfortunately, Apache Spark layer as a standalone layer does not provide support for geometrical data and operations. Hence, GeoSpark comes with extra two layers to support the processing of large-scale spatial data and operations in Apache Spark. In this regard, GeoSpark
provides a novel data abstraction to support geometrical and distance operations called spatial resilient distributed datasets (SRDDs)[57]. Indeed, SRDDs are collections of objects partitioned across a cluster of machines.

**Spatial Resilient Distributed Dataset (SRDD) Layer.** In this work, the proposed technique utilizes this SRDD Layer in indexing the trace data of the infected patients (SRDD Indexing) for fast retrieval, SRDD Indexing is performed using the R-Tree index. The index was built by balancing the query selectivity, indexing overhead (memory and time), and the quantity of spatial objects. Earlier, SRDD it is necessary to index an SRDD Partitioning phase. Partitioning SRDD, SRDD By establishing a global grid file for data partitioning and assigning each element in an SRDD to the same 2-Dimensional spatial grid space, one may partition all saved/loaded SRDDs. Finally, the proposed technique explores the power of the Spatial Query Processing Layer in providing a set of geometrical operations in identifying the boundaries around the infected patient’s trajectory and also in identifying the larger area that other nearby trajectories of other objects move inside as shown in Fig 3. The boundaries identification is done through calling a function named \( \text{MinimumBoundingRectangle}() \) which is responsible for finding the minimum bounding rectangle for each infected object trajectory SRDD and finds a large minimum bounding rectangle that contains all of the other internal objects moving a nearby infected object. The large minimum bounding rectangle is commuted by adding \( P_{Dist} \) value to all sides of the minimum bounding rectangle of the infected object.

**Spatial Query Processing Layer.** This work utilizes the Spatial Query Processing Layer in performing spatial data queries that serve the answering finally of the contact-tracing query. First, the proposed technique runs the range query to get all trajectories of objects that fall inside the large rectangle in Fig 3. After that, the proposed technique performs a distance join query between query object’s trajectories and other trajectories that fall in the large rectangle, if they overlap each other within \( P_{Dist} \), they will be included in the final list. In the end, the proposed technique returns this list as these suspected contacts as the answer to the contact-tracing query.

Figure 5 depicts the contact tracing query processing steps. First, the input to the system is the contact tracing query, the query date/time, and the user id. Second, the system gets all the traces that belongs to the querying user who poses
the query, the traces are captured by the user id. Indeed, the traces are retrieved from the current query date/time to 14 days before the query date, in this work 14 days represents the incubation period of COVID-19 disease. After that, the system gets all traces of other individuals moved with the querying user within the period from the current query date/time to 14 days before the query date. All these traces are loaded to the GeoSpark memory for processing. Then, these traces are indexed using R-Tree\cite{27} for fast retrieval. Furthermore, the system performs a range query to retrieve traces moved with querying users within a close area. Hence, these traces are inspected according to Algorithm1 which will be explained in detail in section 5.3. In the final step, the result of the query is a list of the traces of all suspected individuals.

From the privacy perspective, in this work, there is no store of any sensitive data like (name, address or social security number) related to the users that are traced during the contact tracing process, only User_Id for each user is used to link the user stored in the system with each trajectory.

The proposed work offers some important insights into where and how long the data is stored in the system. For each user, his traces are stored as flat files consisting of location data paired with timestamp values in the file system, these traces are saved on the file system only for 14 days since the creation time of the trace file, the 14 days period represents the incubation period of the COVID-19 disease. When the system receives a contact tracing query CTQ from the infected patient (Querying user), the system starts to collect all related traces that will serve the query processing phase and load them into GeoSpark in-memory. After that, the system starts the contact tracing process against the loaded traces and finally generates the answer of the CTQ as an output. Indeed, the traces remain in the in-memory for the period equal to the period between start DateTime and end DateTim of the querying user’s trajectory. The process of contact tracing and how the system fetches the related traces with the querying user is described in a detailed manner in the next section 5.3

Manuscript submitted to ACM
Algorithm 1 Contact Tracer Solution

1: procedure Contact_Tracer
2: INPUT: Tracing Period $T_{\text{tracing}}$, Time Threshold $T_{\text{Exposure}}$, Proximate Distance $P_{\text{Dist}}$
3: /* Overlapped Traces list */
4: /* Region Bounding Query Trajectory */
5: Let $\text{RegionBounding}_{\text{TQ}} \leftarrow \phi$
6: /* Range Region */
7: Let $\text{RegionOverlapped} \leftarrow \phi$
8: Let $\text{OverlappedTracesList OL} \leftarrow \phi$
9: Let $\text{SuspectedList SL} \leftarrow \phi$
10: /* Query Trajectories List */
11: Let $\text{QL}_{\text{TQ}} \leftarrow \phi$
12: /* Thereabout Distance to get proximate contacts */
13: Let $\text{thereaboutDist} \leftarrow P_{\text{Dist}} + \text{distance threshold}$
14: $T_{\text{tracingStart}} = \text{CurrentDate}$
15: $T_{\text{tracingEnd}} = \text{CurrentDate} - T_{\text{tracingPeriod}}$
16: for $T_{\text{tracingStart}}$ to $T_{\text{tracingEnd}}$ do
17: /* Query object's trajectories */
18: Get $\text{QL}_{\text{TQ}}$ in iterated day
19: for each $\tau_{\text{TQ}} \in \text{QL}_{\text{TQ}}$ do
20: /* Others, Other object's trajectories */
21: Get $\tau_{\text{Others}}$ in iterated day
22: /* Region created from max/min points $\tau_{\text{TQ}}$ */
23: $\text{Region}_{\text{TQ}} = \text{Bound } \tau_{\text{TQ}}$
24: /* Add social distance to overlapped region */
25: $\text{Region}_{\text{Overlapped}} = P_{\text{Dist}} + \text{Region}_{\text{TQ}}$
26: /* Get traces in overlapped region */
27: $\text{OL} \leftarrow \text{Get } \tau_{\text{Others}} \text{ located in } \text{Region}_{\text{Overlapped}}$
28: for each $\alpha \in \text{OL}$ do
29: if $\text{isJoin}=\text{True}$ then
30: $t = \text{joined distance period}$
31: if $t \geq T_{\text{Exposure}}$ then
32: add $\alpha$ to SL
33: end if
34: end if
35: end if
36: end for
37: end for
38: end for
39: if $\text{TracingMode} = \text{Indirect}$ then
40: for $\theta$ to $T_{\text{tracingPeriod}}$ do
41: /* Increment SL by fetched contacts */
42: Call Contact_Tracer
43: end for
44: end if
45: end procedure
46: OUTPUT: Return objects belongs to SL

5.3 Algorithm
The primary algorithm that explains the whole functioning of the suggested solution is illustrated in this section. The three main steps of the suggested solution are briefly outlined as follows:

Step1: Set Boundaries for Query Object’s Trajectory. In this step, the solution identifies an envelope region around the query object’s trajectory to easily obtain other trajectories that are nearby this bounded region. This
approach establishes this envelope region by including the minimum and maximum points in the trajectory of the querying user and querying object, respectively. Based on these points, it confines the trajectory of the query object. The inner rectangle in Figure 3 presents this step.

**Step2: Recognize the Range Query Region.** In this step, the solution gets the envelope region created in step 1 around the query object’s trajectory and added the proximity distance on all sides of this region to obtain another larger region which easily helps in fetching all nearby objects. Moreover, all trajectories located in this region need to be investigated. The outer rectangle in Figure 3 describes this step.

**Step3: Extract Suspected Objects.** In this step, the solution gets all trajectories located inside the region created in step 2. Then, perform distance join between the query object trajectory and all trajectories located in the region created in step 2, then the overlapped distance between them is measured and the period occupied during this overlapped distance is evaluated. Particularly, the distance join is performed based on two identified distances to fetch either close or proximate contacts or both. For the close contacts, the distance join is occurred based on the proximity distance identified, and for proximate contacts, the distance join is occurred based on the proximity distance identified and added to it extra distance threshold named thereabout distance. Finally, if the fetched trajectories successfully joined to the query object either by thereabout or proximity distance the period occupied during this overlapped distance is greater than or equal to the identified exposure time, the system will consider the object that has this trajectory as suspected contact and notify him for making diagnostic tests.

**Pseudocode in Action.** Algorithm 1 shows the proposed pseudo-code for the contact tracing solution. There are three input parameters for the algorithm, (a) Tracing Period \( T_{\text{tracing period}} \); The disease’s incubation time is represented, (b) Time Exposure \( T_{\text{Exposure}} \); This value indicates the determined exposure time period, confirming the transmission of infection from an infected person to the healthy person, and (c) Proximity Distance \( P_{\text{Dist}} \); this parameter represents the identified distance between a normal individual and an infected individual. If the measured distance is below or equal to this distance, it is certain that contact was made with the infected person.

The algorithm returns a list of all suspected contacts either close or proximate contacts or both. First, the algorithm considers the query date as the date that will start tracing from it, so based on the query date and tracing period parameter, the algorithm identifies the start tracing date and end tracing date (lines 15-16). Then, the algorithm starts to iterate during the tracing period from start-tracing-date to end-tracing-date, in each daily date, the algorithm gets all trajectories that belong to the query object and other trajectories that belong to other objects located inside the range region on this day (line 17-28). After that, the algorithm iterates over the query object’s trajectories retrieved on each tracing day, and started to investigate it against other trajectories moved with the query object on this day. The algorithm defines two types of distance to get close and proximate contacts, first \( P_{\text{Dist}} \) to get close contacts, then added to \( P_{\text{Dist}} \) extra distance threshold and obtain another distance named thereabout and uses it to get proximate contacts.

Moreover, the algorithm utilizes \( P_{\text{Dist}} \) and thereabout values to perform distance join between query object’s trajectory and other overlapped trajectories, after that based on the distance common between them, the algorithm measures the overall period occupied by this distance, if it is larger than or equal to the identified exposure time, the algorithm added this trajectory and object belongs to this trajectory into the suspected list, (lines 29-39). The algorithm can identify if the contact is close or proximate based on the join that happened, if the distance join occurs based on \( P_{\text{Dist}} \), the algorithm marks the object as close contact else if it is joined by thereabout distance, the algorithm marks the object as proximate contact. The algorithm finally provides a list of every suspected contact. It is essential to remember that before starting, the algorithm configures the tracing modes. Lines 40–45 of the algorithm are used to make recursive
calls to retrieve the contacts of contacts when the tracing mode is “Indirect”. The algorithm then outputs the suspected list of objects together with their motions.

6 EXPERIMENTS

This section conducts an experimental analysis of our suggested solution and presents the findings.

6.1 Experimental Environment

6.1.1 Dataset. The experiments carried out in this study primarily make use of a dataset that is synthesis data gathered on real data set UJIIndoorLoc [8], which describes the indoor movements of human interactions inside the University of Jaume building. This data set consists of 1.1 billion records, each of which has the following characteristics: (1) Longitude, (2) Latitude, (3) Floor; which explains the altitude inside the building, (4) Building_Id; to identify the building, (5) Space_Id; to locate the area (office, corridor, classroom), (6) Relative position concerning space, it holds two values inside and outside (in front of the door), (7) User_Id; which denotes a special user identification, (8) Phone_Id; This is the identifier for the Android device and (9) Timestamp; which indicates the time the capture is made. These records show the user footprints inside the facility. These footprints come from 100K unique users who enter the building at various times throughout the day.

6.1.2 Environment Settings. The implementation is done through a cluster configuration of 8 worker nodes through GeoSpark, the CPU per each worker: is 8 Intel Xeon Processor, and the memory per each worker is 50 GB registered memory in Spark and Hadoop. Within the Eclipse Intellij IDE, all modules are in fact implemented in Java. Every evaluation is carried out on a computer running Windows 10 with an Intel(R) Core(TM) processor and 32GB of RAM.

6.2 Experimental Assessment

We then present our results.

Exp1: Impact of Scalability using GeoSpark. As shown in figure 6, this experiment compares the GeoSpark against SpatialHadoop with respect to the increasing number of nodes in the cluster, the experiment runs the same range query over a big data-set of size 60GB. SpatialHadoop [20] is a framework for Hadoop that supports spatial data; it enables users to run spatial SQL queries and conduct spatial operations, but it does not offer a thorough approach to query optimization. The cluster’s nodes are represented by the x-axis, and the time spent is shown on the y-axis. Overall, it is shown that expanding the cluster’s nodes will aid in cutting down on the total amount of time used. Also, it is observed that the GeoSpark consumes around 0.25 time consumed by SpatialHadoop in processing spatial queries. This is justified by the fact that GeoSpark loads the dataset containing the SRDDs from the storage system and cashes them in memory, enabling several memory-based spatial operations to be carried out over the SRDDs.

Exp2: Impact of Spatial Partitioning. In order to speed up a spatial join query, the fundamental goal of Spatial RDD (SRDD) partitioning is to group spatial items into the same partition based on their spatial proximity. This experiment measures the effect of various spatial partitioning types on range join queries using GeoSpark. Three distinct spatial partitioning techniques—KDB-Tree partitioning, Quad-Tree partitioning, and R-Tree partitioning—are compared in Figure 7. On join queries, it was found that the KDB-Tree partitioning method consumed the least amount of local join time (KDB-LocalJoin), but that the QUAD-Tree and R-Tree methods both consumed additional time at rates that were 2.5 and 1.7 times slower than the KDB-Tree method, respectively. The reasoning behind this is that, compared to R-Tree...
Fig. 6. Impact of Scalability using GeoSpark.

and Quad-Tree partitioning methods, KDB-Tree partitioning provides more load-balanced grid cells. Additionally, R-Tree does not always cover the entire space, and overflow data partition can happen\cite{22, 28, 51}.

Fig. 7. Impact of Spatial Partitioning.

**Exp3: Impact of Distance Join on Memory Usage.** In this experiment, figure 8 evaluates the performance of spatial join queries concerning the proximity distance $P_{Dist}$. The x-axis represents the distance covered by the count of overlapped points resulting from the join operation, and the y-axis represents the peak memory usage resulted from the join operation. Overall, when the distance increase the total memory consumed increase, this is because the scanning number of overlapped SRDDs, removing duplicates, and saving the final result in the disk will correspondingly increase.

6.2.1 **Baseline Comparisons.** We next report our findings.

**Exp4: Impact of varying $P_{Dist}$ on Memory Usage.** A baseline comparison is carried out in these experiments, more specifically in experiments 4 and 5. Since social distance and exposure time parameters are used to confirm the transmissions of infectious diseases, this work chooses the QR-tree index introduced in \cite{21} as a baseline method for comparisons. The rationale behind this decision is that QR-tree was proposed to provide answers to contact tracing
queries with regard to these parameters. Figure 9 compares the effects of increasing closeness distance on the proposed technique and the QR-tree index and gives information on how increasing proximity distance affects memory overhead. The x-axis describes the increased value of $P_{Dist}$ in meters, while the y-axis demonstrates the memory usage in gigabytes. It is observed that QR-Tree consumed more memory than the proposed solution because the proposed solution mainly depends on parallel executions of the queries, and indexes used like R-Tree or QUAD Tree are distributed on memory across worker nodes as local indexes, this saves memory. Additionally, mainly the proposed solution depends on spatial partitioning which ensures that the objects with the proximity characteristics will be grouped and this saves the search time and memory overhead.

**Exp5: Impact of varying $P_{Dist}$ on CPU Time.** In this experiment, figure 10 compares the QR-tree against the proposed solution in answering contact tracing queries. The x-axis describes the increased value of $P_{Dist}$ in meters, while the y-axis demonstrates the processing time in seconds. Overall, this experiment includes measuring the performance of distance join queries, it is noted that increasing $P_{Dist}$ will increase the processing time consumed. Furthermore, it was found that the proposed solution took less time than the QR-Tree solution. This is because the proposed solution
mainly used GeoSpark to manage query executions and because GeoSpark’s architecture caches intermediate results in memory rather than saving them to the disc, which speeds up processing.

Exp6: Effect of increasing query volume on CPU time. Figure 11 gives the impact of increasing the number of queries to test the efficiency of our proposed solution. According to the X-axis in Figure 11, the number of searches increased from 1000 to 10,000. The Y-axis displays the amount of CPU time (in milliseconds) used to process these queries, ranging from 0 to 100. In general, it is shown that as the number of requests rises, the CPU cost gradually rises. However, our proposed solution performs more efficiently when the number of queries increases than baseline methods. Indeed, the proposed solution outperforms the QR index in answering queries, the proposed solution approximately saved 70% of CPU cost compared to the QR index method in answering concurrent contact tracing queries. The justification for this great performance returned to the in-memory computing features and indexing methods used in GeoSpark which are empowered by our proposed solution; besides, the proposed technique enables clustering structure to answer the queries.
6.3 Experiments Summary

To conclude, GeoSpark is an efficient framework that manages big data. Experiments proved that GeoSpark outperforms SpatialHadoop in processing spatial queries over big data. Besides, KBD-Tree is the best choice as a spatial partitioning method than QUAD-Tree and R-Tree, it consumes less time because it generates more load-balanced grid cells. Also, it is confirmed that the proposed solution achieves good results in answering contact tracing queries compared to baseline methods like QR-Tree, the cause is proposed solutions cash intermediate results and use big data framework to process large-scale datasets.

7 SYSTEM REAL ON-SITE DEPLOYMENT

This section describes how the proposed system could be used in a real scenario.

The mobile devices embedded with GPS sensors and have internet connections of users represent the IOT data sources. These sensors are responsible for collecting users' traces linked by device id which determines each user registered on the system. Each trace consists of three fundamental parts (longitude, latitude and altitude). The system mainly depends on Bluetooth-enabled features to collect nearby traces of users. Indeed, the live system works as follows:
(1) the end-user downloads a mobile application which has Bluetooth and GPS location features enabled, (2) second, the user registers personal information in the mobile application, (3) after that, assume that the user moves inside a building and logged in the mobile application, (4) the application detects other users in the nearby region and sends traces to the back-end server and links these traces with the user device id, (5) the mobile application allows the user to mark himself as infected and enter the infection start date, (6) after that, the system back-end works to retrieve users' traces contacted with the infected person within incubation period, (7) users were divided into high-risk and low-risk contacts by the system, and (8) The system notifies users who may be infected by SMS notifications based on the type of contact risk. Actually, the system has three actors that can deal with the system: (a) the system administrator configures the disease incubation period and controls the proximate/close contacts thresholds through the system portal website; (b) the system user that sends their traces using a mobile application to the back-end server; (c) Database administrator monitors the system traces; makes backups needed; truncate traces that become outdated according to the incubation period.

8 CONCLUSION

This research addresses the problem of the outbreak of infectious diseases, and how the interventions like contact tracing strategies can help in fighting the spread of these diseases. This research suggests a novel method to accomplish this goal, one that assesses contact tracing queries (CTQ) to identify all conceivable contacts who may have been exposed to the infected patient throughout the incubation period. More specifically, this technique starts the investigation process by getting the trajectories of contacts that moved in the same time and region with the infected patient, then inspects them against two factors; (1) exposure time, and (2) proximate social distance. By employing Apache Spark, the proposed technique can process CTQ over a massive number of trajectories and return answers promptly. The proposed technique’s effectiveness and scalability are demonstrated through experimental evaluation using large real data sets.

REFERENCES


Manuscript submitted to ACM


Manuscript submitted to ACM
Contact Tracing Query Processing on Spark


