A Novel Graph Indexing Approach for Uncovering Potential COVID-19 Transmission Clusters

XULIANG ZHU, XIN HUANG, LONGXU SUN, and JIMING LIU, Hong Kong Baptist University

The COVID-19 pandemic has caused the society lockdowns and a large number of deaths in many countries. Potential transmission cluster discovery is to find all suspected users with infections, which is greatly needed to fast discover virus transmission chains so as to prevent an outbreak of COVID-19 as early as possible. In this article, we study the problem of potential transmission cluster discovery based on the spatio-temporal logs. Given a query of patient user q and a timestamp of confirmed infection t_q, the problem is to find all potential infected users who have close social contacts to user q before time t_q. We motivate and formulate the potential transmission cluster model, equipped with a detailed analysis of transmission cluster property and particular model usability. To identify potential clusters, one straightforward method is to compute all close contacts on-the-fly, which is simple but inefficient caused by scanning spatio-temporal logs many times. To accelerate the efficiency, we propose two indexing algorithms by constructing a multigraph index and an advanced BCG-index. Leveraging two well-designed techniques of spatio-temporal compression and graph partition on bipartite contact graphs, our BCG-index approach achieves a good balance of index construction and online query processing to fast discover potential transmission cluster. We theoretically analyze and compare the algorithm complexity of three proposed approaches. Extensive experiments on real-world check-in datasets and COVID-19 confirmed cases in the United States validate the effectiveness and efficiency of our potential transmission cluster model and algorithms.

CCS Concepts: • **Information systems** → **Graph-based database models**; **Temporal data**; *Database query processing;*

Additional Key Words and Phrases: Graph index, transmission cluster, COVID-19

ACM Reference format:
Xuliang Zhu, Xin Huang, Longxu Sun, and Jiming Liu. 2023. A Novel Graph Indexing Approach for Uncovering Potential COVID-19 Transmission Clusters. *ACM Trans. Knowl. Discov. Data.* 17, 2, Article 24 (February 2023), 24 pages.
https://doi.org/10.1145/3538492

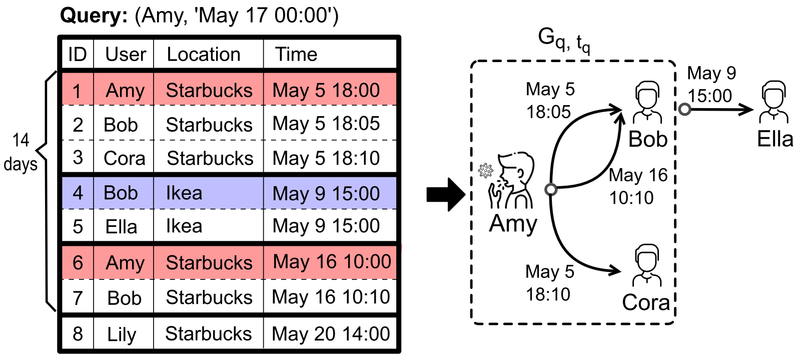
1 INTRODUCTION

Since late December 2019, an outbreak of a novel coronavirus disease COVID-19 has subsequently led to millions of COVID-19 cases globallyover the world [40, 48]. Severe disease onset has resulted in thousands of death in USA, Brazil, India, Italy, Spanish, China, and other countries/regions, due to massive alveolar damage and progressive respiratory failure caused by COVID-19. Viruses

This work was supported by the Hong Kong RGC Grant Nos. HKBU 12200021, 22200320, and 12201318.

Authors' address: X. Zhu, X. Huang, L. Sun, and J. Liu, Department of Computer Science, Hong Kong Baptist University, 55 Renfrew Road, Kowloon Tong, Hong Kong, China; emails: {csxlzhu, xinhuang, cslxsun, jiming}@comp.hkbu.edu.hk.

1556-4681/2023/02-ART24 $15.00
https://doi.org/10.1145/3538492



Query: (Amy, 'May 17 00:00')

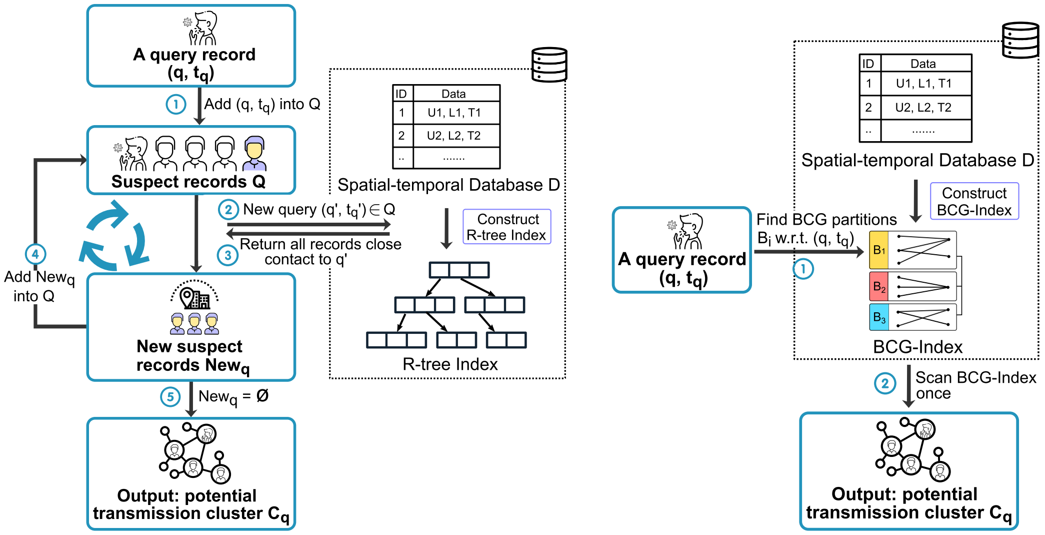
ID	User	Location	Time
1	Amy	Starbucks	May 5 18:00
2	Bob	Starbucks	May 5 18:05
3	Cora	Starbucks	May 5 18:10
4	Bob	Ikea	May 9 15:00
5	Ella	Ikea	May 9 15:00
6	Amy	Starbucks	May 16 10:00
7	Bob	Starbucks	May 16 10:10
8	Lily	Starbucks	May 20 14:00

Fig. 1. An example of potential COVID-19 transmission cluster discovery on spatial-temporal logs. The query consists of a patient user q = "Amy" and a timestamp of confirmed COVID-19 infection t_q = "May 17 00:00". The personal contact graph of user "Amy" is G_{q,t_q} as depicted in dashed rectangle. The answer of potential transmission cluster is C_q = {"Amy", "Bob", "Cora", "Ella"}.

are transmitted by close contact, droplets, and fomites [42]. Recently, many studies have been conducted on COVID-19 research [1, 5, 9, 16, 18, 24, 37], including quantifying the underlying transmission patterns of COVID-19 outbreak [25], security-aware mobile tracking system [34], forecasting [1, 37], and data privacy [30]. In terms of public health measures, it is importantly necessary to call for quick and effective tracking of virus transmission chains and early detection of outbreak, which can timely prevent the broad infections outbreaking to avoid the economic and society lockdown [22, 30].

To achieve COVID-19 containment, one effective approach is to discover potential transmission clusters effectively and cut off their further transmissions. It is well known that transmission lies on the close contact of individual citizens within a small spatial-temporal proximity. Therefore, it is critically important to obtain spatial-temporal logs of an individual citizen, i.e., a trajectory record of users, locations, and timestamps, describing the movement when and where people stay. Although privacy concerns are long-termed raised up, fortunately, these data are still available to be obtained and shared in a reasonably controlled management, e.g., telecommunication records in mobile services of AT&T and Hong Kong Telecom [31], the significant locations service in Apple smartphones [29], and so on. In worst cases, each user has a personal software environment (either on the smartphone or in the cloud) to store his/her raw data of spatial-temporal logs, which can help to be offered to health authorities in case they are tested positive to COVID-19 [30]. Moreover, in those countries with the highest COVID-19 virus cases, there are hundreds of thousands of confirmed cases in a single day, which demands for a highly efficient search to identify potential transmission clusters.

In this article, we motivate and investigate the problem of discovery potential transmission clusters, that is, given a query of patient q and time t_q, finding all potential infected users who have directly/indirectly social contacts to q before t_q, in terms of close spatial-temporal distance. For example, consider a spatial-temporal database D shown in Figure 1. Assume that the user "Amy" is diagnosed as the COVID-19 infectious patient at time t_q = "May 17 00:00". We take "Amy" as the query patient q and the diagnosis time as the query time t_q. It intends to find all close contact users of q in the incubation period of past 14 days, who are suspected to have high probabilities of getting infectious COVID-19. During the period [May 3, May 17], the virus transmission may happen from "Amy" to "Bob" and "Cora", due to their appearing in the same location of "Starbucks"



(a) Classical online search framework

(b) Our graph indexing framework

Fig. 2. The frameworks of potential transmission cluster discovery in spatial-temporal databases.

and at the close time of 18:00, 18:05, and 18:10 on May 5. Moreover, as "Bob" became a potential infected person on May 5 and had a close contact with "Ella" in the location of "Ikea" at "15:00 May 9". Thus, "Ella" is also a potential infecting people. Thus, the potential cluster involves four users {"Amy", "Bob", "Cora", "Ella"}. To play safe, the potential cluster should identify all persons that are close contact reachable from a query user.

However, an efficient extraction of query-dependent potential transmission clusters is challenging. The reason has two-fold. First, given millions of people in a city and multiple spatial-temporal records visited by one person, it may incur combinatorial blow-ups for enumerating all possible transmission clusters. Second, the infectious diseases usually have a time window of incubation period, reflecting that the virus transmission already happen before the patient is identified. Thus, the discovery of direct close contact is not enough for potential transmission clusters, which needs the search of all users involved in the high-risk clusters by underlying transmission.

To tackle the problem efficiently, we consider two different approaches of *classical online search* and *our graph indexing based search*. First, we consider the classical online search framework as shown in Figure 2(a), which finds all potential infecting users for every suspected record in a spatial-temporal database. Specifically, it first adds a query record (q, t_q) into the suspected records Q (at the step ① in Figure 2(a)). Then, it iteratively finds users and their records that have close contact to any record in Q in database D (at the step ② and ③ in Figure 2(a)) and adds the new close contact records into Q (at the step ④ in Figure 2(a)), until no new close contact user is identified. The final results of potential transmission cluster are returned (at the step ⑤ in Figure 2(a)). This classical online search framework needs to scan the suspected spatial-temporal logs multiple times, which is inefficient without any index. A useful spatial-temporal index of R-tree [13, 28, 33] can be used to accelerate the querying of close contact w.r.t. one query record. However, it cannot reduce the total number of query times and ask massive such close contact queries in spatial-temporal database, leading to inefficiency in the discovery of large-scale transmission clusters. In this article, we propose a novel *graph indexing based search framework* as shown in Figure 2(b), which can significantly reduce the query time to only once. Intuitively, one straightforward graph indexing approach is to build an offline multigraph index, which keeps all close contact relationships between any pair of spatial-temporal records. Multigraph indexing approach can extract potential clusters directly using one breadth-first search from a given query, but incurs an expensive cost of index construction in running time and consuming space. To improve the efficiency, we propose a **bipartite contact graph** (**BCG**) to describe close contact relationships between users and spatio-temporal records. Through compressing and partitioning a bipartite graph, we propose a compact BCG-Index to optimize the index construction and potential cluster queries in a fast way. Different from online search method, it only needs to search in the partitioned graph index once instead of scanning the suspected spatial-temporal logs multiple times (at the step ① and ② in Figure 2(b)). Overall, our proposed techniques have wide applications on disease expansion control and early outbreak prevention. More importantly, our potential cluster discovery algorithms not only work for COVID-19 but also benefit for other kinds of close-contact based infectious diseases.

In summary, this article makes the following contributions:

- We motivate and formulate the problem of potential cluster discovery to prevent virus transmission chains. We formally define the close contact and contact reachability. Based on them, we propose a potential transmission cluster model (Section 3).
- We analyze the properties of our potential transmission cluster model, satisfying the good desiderata of *close social distance*, *arbitrary structural shape*, and *incubation-aware transmission* in real applications. Moreover, we discuss the particular usability in details to show our model flexibility (Section 4).
- We first develop an online search approach to find potential transmission clusters in an on-the-fly manner. To achieve the efficient search, we propose an offline indexing approach to construct multigraph for keeping all the records of close contact in history (Section 5).
- We further optimize the multigraph index by constructing a compact BCG-Index in an efficient space cost. We develop the techniques of spatio-temporal entity compressions and graph partitions, which shrink the index into multiple small bipartite graphs, which can support the fast potential cluster discovery. Moreover, we theoretically compare and analyze the algorithm complexity of online search methods (based on binary search and R-tree) and graph index methods (MG-Indexing and BCG-Indexing) (Section 6).
- We conduct extensive experiments on four real-world datasets of check-in records in geo-social networks. We generate the ground-truth of virus transmissions and infected users via the propagation simulation using a classical independent cascade model. We also conduct a case study of COVID-19 transmission following real-world dataset. The results validate the effectiveness and efficiency of our potential transmission cluster model and proposed algorithms (Section 7).

We discuss related work in Section 2 and conclude this article in Section 8.

2 RELATED WORK

Our work is related to *COVID-19 transmission analytics* and *spatio-temporal mining*.

COVID-19 transmission analytics. Under the COVID-19 pandemic, numerous studies have tried to discover suspected clusters in order to track the COVID-19 transmissions [1, 9, 11, 14, 16, 18, 26, 35–37, 41]. Recently, various COVID-19 relevant models are proposed for COVID-19 forecasting [1, 37] and spread prediction [16]. Luo et al. [26] developed a explorer system to monitor spatio-temporal data of COVID-19. A graph embedding approach is proposed to help identify COVID-19 cases [41]. A warning system is designed to predict the hazard area, by collecting data from websites and using machine learning approaches to analyze the relevant features [9]. Kim et al. proposed a deep learning approach and a hierarchical model for estimating the number of imported COVID-19 cases from abroad [18]. On the other hand, some researchers aim at exploring the impact of certain containment measures on the transmission of COVID-19. The system dynamic simulation model is used to discover how physical distance measures influence the infectious [36]. Reference [14] compares the impact of different control policies on the spread of COVID-19 and investigates the influence of heterogeneity of urban mobility during the propagation. Reference [11] focuses to lift mitigation measures using deep learning approaches. Previous tracking systems can be also found in [3, 46, 52]. This work of transmission cluster discovery is also related to clustering algorithms in graphs [20, 21]. Different from the above studies, our work leverages the spatio-temporal activities to efficiently find query-dependent suspected clusters using a graph indexing approach.

Spatio-temporal mining. There exist several studies on the spatio-temporal data mining [2, 4, 7, 8, 13, 23, 32, 33, 43]. A comprehensive survey of spatio-temporal data mining can be found in [4]. Wu et al. study the problem to mine the reachable area from a given location and temporal period and propose a data-driven method to tackle the problem based on historical trajectory dataset [43]. Francalanci et al. [8] propose to analyze the evolving information for spatio-temporal queries. Ahmed et al. [2] develop a tracking device to find the most frequent terms in spatio-temporal region for each query. The above spatio-temporal data mining studies work on the reachable region, evolving information analysis, and frequent terms discovery tasks. There exist also many works on developing the learning based models for spatio-temporal prediction [7, 23, 32]. In addition, there exists a line of work [45, 53] studying the reachability problem in temporal graphs. Wu et al. [45] proposed an indexing based technique to answer reachability and time-based path queries in a temporal graph. Zhang et al. [53] developed a labeling scheme of temporal vertex labeling over distributed temporal graphs. Both these two works use index based method to answer the reachability queries in the temporal graphs. R-tree [13, 28, 33] is a classical tree index to store spatial objects in database, which accelerates the nearest neighbor search or the objects within a given spatial interval. However, for detecting a transmission cluster in this article, the R-tree-based online search approaches cannot avoid a large number of query times interacting with spatial-temporal database, leading to the inefficiency. A detailed comparison to R-tree-based approaches can be found in Sections 1, 6.3, and 7, in terms of motivations, algorithm complexities, and also experimental evaluations, respectively. Different from these studies, we focus on the close contact modeling and develop fast querying algorithms for potential COVID-19 transmission cluster discovery, which uses spatio-temporal logs to build graph indexes.

3 PRELIMINARIES

We are given a database D of *spatial-temporal logs* in the form of a relation $\text{STlog}(\text{User}, \text{Location}, \text{Time})$, where each tuple $(u, l_u, t_u) \in D$ represents that a user u visited location l_u at time t_u. It contains such a tuple for each location visited by every user of the system. Assume that the projection of STlog on the first column forms a set of users V, and the incubation period of our interested disease is $\Delta_t \in \mathcal{R}^+$. In the following, we first define close contact.

Definition 1 (Close Contact). Given two parameters of location threshold δ_l and time threshold δ_t, we say that two users v and u have a close contact at time t, denoted by $u \leftrightsquigarrow_t v$, if and only if: there exists two tuples (u, l_u, t_u), $(v, l_v, t_v) \in D$ having a small spatial-temporal distance such that (i) $|t_u - t_v| \leq \delta_t$; (ii) $|l_u - l_v| \leq \delta_l$; and $t = \max\{t_u, t_v\}$

Let $\delta_t = 0$ and $\delta_l = 0$. Under this strict parameter constraint, two users v, u have close contact, which needs to satisfy $t_u = t_v$ and $l_u = l_v$. Due to the uncertainty in data collection process from sensors and humans, Definition 1 gives a flexible mechanism to quantify the close contact. Note that, in epidemiology, the close contact usually refers to the person who has been near enough to a person with COVID-19. However, we generalize the concept of close contact to any two person for potential COVID-19 transmission cluster identification. For example, consider the spatial-temporal database D in Figure 1. For $\delta_l = 0$ and $\delta_t = 20$ minutes, two users Amy and Bob have a close contact at time t = "May 5 18:05" as they appear at times "May 5 18:00" and "May 5 18:05", respectively, in the same location of "Starbucks". Note that we require an unique value to be the latest timestamp of close contact, i.e., $t = \max\{t_u, t_v\}$. The close contact in Definition 1 can be extended to allow multiple feasible values of t given two users' records. Moreover, Definition 1 can be extended to a given length of exposure, where two users v and u have a close contact for a continuous time window. To model the virus transmission chain within an incubation period Δ_t, we give a definition of contact reachability based on the close contact.

Definition 2 (Contact Reachability). Given two users $v, u \in V$ and a time t^*, we say that u reaches v via a series of close contact within the incubation period of $[t^* - \Delta_t, t^*]$, denoted as $u \rightarrow_{t^*} v$, if and only if there exist a contact path $\langle v_0, \ldots, v_l \rangle$ such that

(1) $v_0 = u$ and $v_l = v$;
(2) the close contact $v_{i-1} \leftrightsquigarrow_{t_i} v_i$ holds for $1 \leq i \leq l$ and $t^* - \Delta_t \leq t_1 \leq \ldots \leq t_l \leq t^*$.

Consider the example in Figure 1, for $\delta_t = 20$ minutes and $\Delta_t = 14$ days, Amy and Bob have a close contact at t_1 = "May 5 18:05", i.e., Amy $\leftrightsquigarrow_{t_1}$ Bob. Moreover, Bob and Ella have a close contact at t_2 = "May 9 15:00", i.e., Bob $\leftrightsquigarrow_{t_2}$ Ella. There exists a contact path $\langle Amy, Bob, Ella \rangle$ such that Amy reaches Ella, i.e., Amy $\rightarrow_{t^*}$ Ella, during the incubation period $t^* \in$ [May 9 15:00, May 19 18:05]. Therefore, the potential cluster is all the users that lie along the contact paths starting from the first query patient u. Based on the contact reachability, the potential transmission cluster is defined as follows.

Definition 3 (Potential Transmission Cluster). Given a query of (potential) patient user q (of patient) and time t_q, the potential cluster is defined as a group of users $C_q \subseteq V$ such that each user $u \in C_q$ is contact reachable from q, i.e., $C_q = \{u \in V : q \rightarrow_{t_q} u\}$.

In the following, we formulate the problem of potential cluster discovery on spatial-temporal database D studied in this article.

Problem statement: Given a spatial-temporal database D of $|V|$ users, a virus incubation time Δ_t, and a query of (potential) patient $q \in V$ and time t_q, the problem of potential cluster discovery is to find the potential cluster C_q, in which all users are potential to get virus from the direct and indirect close contacts to q within time $[t_q - \Delta_t, t_q]$.

Example 1. Consider a spatial-temporal database D shown in Figure 1, $\Delta_t = 14$ days, and a query of patient user q = Amy and time t_q = "May 17 00:00", our problem is to find all potential users contact-reachable from Amy within [May 3 00:00, May 17 00:00]. It is clear that Bob and Cora are potential to get virus directly from Amy at "May 5 18:05" and "May 5 18:10", respectively. Moreover, there is a contact path $\langle Amy, Bob, Ella \rangle$, which means Amy and Ella are contact reachable. As a result, the potential cluster is C_q = {Amy, Bob, Cora, Ella}.

Remark. In our problem definition, three parameters δ_t, δ_l, and Δ_t need to be set. It suggests to set three parameters based on the infectivity and incubation time of different diseases. The close contact parameters δ_t and δ_l are based on infectivity. Δ_t is based on the incubation time of specific

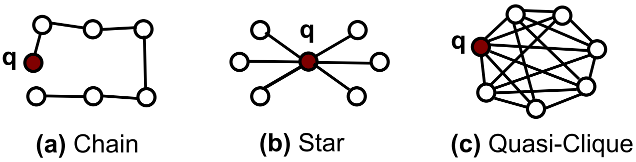


Fig. 3. Arbitrary structural shapes of transmission clusters

diseases. For example, to simulate COVID-19 transmission, we set $\delta_t = 15$ minutes, $\delta_l = 5$ meters, and $\Delta_t = 14$ days in the default setting of our experiments in Section 7.

4 PROBLEM ANALYSIS

In this section, we first analyze the property and particular usability of our transmission cluster model. Then, we introduce an overview of solutions proposed in this article.

4.1 Properties of Potential Transmission Clusters

We begin with the commonly accepted desiderata of good potential groups in the infectious disease transmissions in the following.

(1) **(Participation).** The query user $q \in C_q$.
(2) **(Close social distance).** Each potential member has the close contact to at least one another member via a small social distance, with regard to two spatial and temporal parameters δ_l and δ_t.
(3) **(Arbitrary structural shape).** Any shaped graph structure in underlying potential cluster may occur in real applications, e.g., a long chain, a star with multiple nodes and sparse connections, and a quasi-clique in dense structure as shown in Figure 3.
(4) **(Incubation-aware transmission).** Potential infecting members involve those people are affected by users during the incubation period Δ_t.

In view of the above properties, our potential transmission cluster model is validated to admit these four useful properties. Furthermore, the potential cluster model and our proposed techniques can be substantially modified to detect more rigorous potential groups using variant constraints, such as adding the minimum time window constraint for a close contact. We can also support efficient spatial proximity calculation in terms of latitude and longitude representations using R-tree [13, 33] and other spatial-temporal indexing techniques [27, 38, 38, 47].

In the following, we analyze the complexity of close contact relationship in the worst case.

Lemma 1. *Given a set of n users with the same temporal-spatial records $D = \{(v_i, l, t) : 1 \le i \le n\}$, it forms a n-clique of potential transmission cluster G where each pair of vertices v_i and v_j has an edge of close contact in G.*

Proof. Consider each pair of vertices v_i and v_j, there exist two records (v_i, l, t) and (v_j, l, t), which satisfies $|t - t| \le \delta_t$ and $|l - l| \le \delta_l$. Thus, v_i and v_j has an edge of close contact. This forms a n-clique of potential transmission cluster that consists of n potential infected users and $\frac{n(n-1)}{2}$ close contact edges. □

4.2 Particular Model Usability

We discuss the particular usability of our transmission cluster model. Here, we clarified several critical concern issues of our model in real-world applications for contact tracing and infectious disease transmission.

- First, a large number of potential clusters and frequently asked queries. The contact tracers may not initially reach out to a huge number of indirect contacts—they would call the direct contacts, ask about exposures, symptoms, and COVID testing results, and then only reach out to the direct contacts of direct contacts who tested positive, showed symptoms, and/or were at high risk for infection. However, when a large number of infectious diseases happened, it is costly and very inefficiently to ask only direct contacts by a few hand-on rounds of question answering, which needs a development of automated COVID-19 contact tracing system with efficient query processing techniques. Even worse, given the natural limitation of human-being's memory, it is easy to forget some close contact cases happened a few days ago. When such missing cases have high infectious probability, it may lead to critically bad situations with wide disease spread out.
- Second, the contagious effect w.r.t. the incubation period Δ_t. The infected persons maybe not contagious to others instantaneously, but only become contagious a couple of days before symptoms emerge. In our model, the incubation period Δ_t can accordingly adjust based on different contagious effect of diseases, where Δ_t can be extended to set up as the earliest contagious timestamp. Therefore, our potential cluster model can be adjusted to depict not only the low/middle contagious infectious disease, e.g., COVID-19, but also other highly contagious infectious diseases, e.g., Measles.
- Third, the contact risk w.r.t. the length of exposure. The contact risk is based on the length of exposure, i.e., the amount of overlapping time intervals of two persons at the same locations. Our Definition 1 of close contact determines the boolean value of such an overlapping exposure, which can be easily extended to a quantified length measure of close contact. This heavily depended on a large amount of spatial-temporal records. Our model can discover even low-risk potential clusters.

Overall, our potential transmission cluster model is applicable to detect potential groups caused by a given query patient via close contacts, even discovering many low-risk individuals for critical infectious disease, leveraging on the partially/fully recorded spatial-temporal logs.

4.3 Solution Overview

We consider three different algorithms to find the potential transmission cluster for a given query (q, t_q). An overview of our solutions is presented as follows.

First, we consider a basic solution of online search. This method is straightforward to find the personal direct close contact graph to a patient q during the timestamps of $[t_q - \Delta_t, t_q]$, which find all records satisfying the location and time constraints. This method does not make use of any indexes, which may cost an expensive time for query processing as shown in Section 5.1.

Next, we consider an indexing approach to construct a multigraph [10, 12, 15], where two users may have multiple edges of close contacts in D. The multiple-edge keeps all close contact relationships. A potential cluster C_q is a connected subgraph of G reached from q during the time period of $[t_q - \Delta_t, t_q]$. This method pays an expensive cost on multigraph construction and enjoys fast extraction of potential cluster answers as shown in Section 5.2.

Finally, to make a balanced tradeoff between index construction and online potential cluster discovery, we propose BCG-Index in Section 6 to construct a compact index based on BCGs, which keeps only a partial of contact information using a small index space.

5 ONLINE SEARCH AND MULTIGRAPH INDEXING-BASED APPROACHES

In this section, we first present an online search algorithm for potential transmission cluster discovery. We then give an indexing approach to construct a multigraph index for keeping close contact records and searching potential transmission cluster.

ALGORITHM 1: Online Potential Transmission Cluster Discovering

```
Input: Spatial-temporal database $D$, query user $q$, query time $t_q$, incubation period $\Delta_t$, location threshold
    $\delta_l$, time threshold $\delta_t$.
Output: All potential clusters $C_q$.
 1: Initialization: a priority queue $Q \leftarrow \{(q, t_q - \Delta_t)\}$, where $Q$ maintains the records in ascending order of
    timestamps; $C_q \leftarrow \emptyset$;
 2: while $Q \neq \emptyset$ do
 3:     Pop out a record $(v, t_v)$ from $Q$;
 4:     Generate a contact graph $G_{v,t_v}$ using Contact-Graph $(v, [t_v, t_q])$;
 5:     for each user $u \in V(G_{v,t_v})$ do
 6:         Find the earliest contact between $u$ and $v$ in $E(G_{v,t_v})$, i.e., $t^* \leftarrow \arg\min_{e(v,u,t) \in E(G_{v,t_v})} t(e)$;
 7:         if $u \notin C_q$ then
 8:             $C_q \leftarrow C_q \cup \{u\}$;
 9:             $Q \leftarrow Q \cup \{(u, t^*)\}$;
10: return $C_q$;

    procedure Contact-Graph $(v, [t_v, t_q])$
11: Find all $v$'s spatial-temporal records in $D$ during the incubation time of $v$ as $[t_v, t_q]$, i.e., $D_v = \{(v, t, l) \in$
    $D : t_v \leq t \leq t_q\}$.
12: for each tuple $(v, t_x, l_x) \in D_v$ do
13:     for each user $u \in V$ do
14:         if there exists $(u, t_u, l_u) \in D$ such that $|t_u - t_x| \leq \delta_t$ and $|l_u - l_x| \leq \delta_l$ then
15:             Add a vertex $u$ in $G_{v,t_v}$ if $u \notin G_{v,t_v}$;
16:             Add an edge $(u, v, \bar{t})$ between $u$ and $v$ in $G_{v,t_v}$, i.e., $u \leftrightsquigarrow_{\bar{t}} v$, where $\bar{t} = \max\{t_u, t_x\}$;
17: return $G_{v,t_v}$;
```

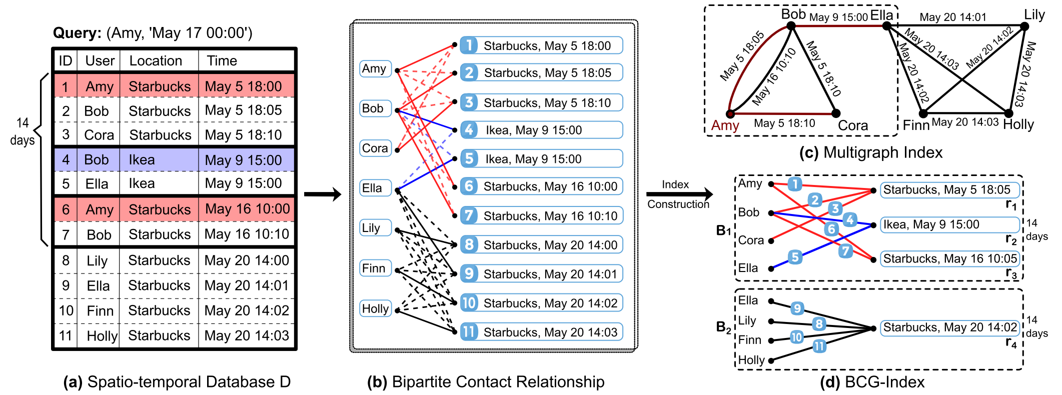
5.1 An Online Transmission Discovery

In this section, we propose an online search approach of discovering potential cluster for a given query (q, t_q). We begin with a definition of personal contact graph.

Definition 4 (Personal Contact Graph). Given a user u and a time t_u, the personal contact graph $G_{u,t_u}(V_u, E_u)$ is defined as follows:
(i) V_u is a subset of all users V, i.e., $V_u \subseteq V$. For each vertex $v \in V_u$, there is a time $t \in [t_u - \Delta_t, t_u]$ such that $u \leftrightsquigarrow_t v$ holds; and (ii) $E_u = \{e(u, v, t) : u \leftrightsquigarrow_t v, v \in V_u, t \in [t_u - \Delta_t, t_u]\}$.

The personal contact graph G_u is defined on the pair-wise close contact in Definition 1, which keeps all potential infecting users to u. Note that the personal contact graph is a temporal graph, in which two vertices may have multiple edge with different timestamps [44]. For example, consider a query user q = "Amy" and time t_q = "May 17, 00:00" in Figure 1. The personal contact graph G_q is shown as G_{q,t_q} in dashed rectangle in Figure 1. The graph has two multiple edges between two users "Amy" and "Bob", due to that they have two close contact in Starbucks in two timestamps of "May 5 18:10" and "May 16 10:10" w.r.t. δ_t = 20 minutes.

Algorithm. The online transmission cluster discovering algorithm is presented in Algorithm 1. Given a spatial-temporal database D, the algorithm iteratively extracts a potential user q from a priority queue Q, which is updated with new potential users (lines 1–9). For each user v, it first generates a personal contact graph G_{v,t_v} based on the close contact relationships between v and other users $u \in V$ (line 4), which invokes a procedure of finding personal graph for v and time t_v (lines 11–17). Specifically, it identifies the earliest infectious timestamp as t_v. Moreover, it collects all spatial-temporal records w.r.t. v during $[t_v, t_q]$ (line 11). It treats every person in contact to v within $[t_v, t_q]$ as a potential person and generates all close contact relationships to v in G_{v,t_v}



Query: (Amy, 'May 17 00:00')

ID	User	Location	Time
1	Amy	Starbucks	May 5 18:00
2	Bob	Starbucks	May 5 18:05
3	Cora	Starbucks	May 5 18:10
4	Bob	Ikea	May 9 15:00
5	Ella	Ikea	May 9 15:00
6	Amy	Starbucks	May 16 10:00
7	Bob	Starbucks	May 16 10:10
8	Lily	Starbucks	May 20 14:00
9	Ella	Starbucks	May 20 14:01
10	Finn	Starbucks	May 20 14:02
11	Holly	Starbucks	May 20 14:03

Fig. 4. An example of graph indexing framework.

(lines 12–16). For each record $(v, t_x, l_x) \in D_v$, it enumerates each user $u \in V$ and checks their contact distance whether are close or not by Definition 1, i.e., $|t_u - t_x| \leq \delta_t$ and $|l_u - l_x| \leq \delta_l$. Note that we can sort the records by time and use binary search to identify the time interval $[t_x - \delta_t, t_x + \delta_t]$. Then, we only enumerate the records in $[t_x - \delta_t, t_x + \delta_t]$. Nevertheless, we also can apply R-tree [13, 28, 33] to find all close contact records. After the above traversal, it constructs a complete personal graph G_{v,t_v} (line 17).

5.2 Multigraph Indexing Approach

In this section, we propose an offline indexing approach to construct a complete multigraph based on spatial-temporal database D and support any online tracking query (q, t_q).

Multigraph index $\mathcal{G}$. Based on the existing users' data logs D, we build up a multigraph $\mathcal{G}(V, E)$, where V denotes the set of users and E represents the set of multiple-edges. Specifically, the edge set $E = \{(u, v, t) : u, v \text{ has a close contact at time } t\}$ is determined by Definition 1 w.r.t. δ_l, δ_t, and Δ_t. Figure 4(c) shows an example of multigraph index $\mathcal{G}$ for the records D in Figure 4(a). Moreover, the subgraph of $\mathcal{G}$ in red is the answer for q = "Amy" and time t_q = "May 17 00:00".

Multigraph index construction. Algorithm 2 outlines the details of multigraph index construction. Specifically, it first sorts all data records of D in the decreasing order of timestamps (line 11). Then, for each record $(v, t_v, l_v) \in D$, it generates all close contact edges (u, v, t) and adds the multiple edges into multigraph $\mathcal{G}$ (lines 12–16).

Multigraph index based cluster discovery algorithms. Algorithm 2 presents the details of multigraph index-based query processing approach. Specifically, the algorithm starts from a query vertex q and checks the adjacent edges of close contact by involving all reachable vertices in a BFS manner (lines 1–9). Finally, it returns the potential cluster C_q (line 10).

For instance, we construct a multigraph index for the spatial-temporal database D in Figure 4(a). The multigraph index $\mathcal{G}$ is shown in Figure 4(c), which consists of 7 vertices and 11 contact edges. Based on the multigraph $\mathcal{G}$, Algorithm 2 finds the potential cluster for a query ("Amy", "May 17 00:00") as follows. The algorithm first adds a pair ("Amy", "May 3 00:00") into the priority queue Q. According to the BFS rule, Algorithm 2, in turn, finds the close contacts ("Bob", "May 5 18:05"), ("Cora", "May 5 18:05"), and ("Ella", "May 9 15:00"), and adds them into Q. After the traverse of queue Q, the potential transmission cluster is identified as C_q = {Amy, Bob, Cora, Ella}. The detailed complexity analysis of Algorithm 1 and Algorithm 2 can be founded in Section 6.3.

ALGORITHM 2: Multigraph Indexing Approach for Potential Cluster Identification

```
Input: A multigraph $\mathcal{G}(V, E)$, a query user $q$, a query time $t_q$.
Output: All potential clusters $C_q$.
A priority queue $Q \leftarrow \{(q, t_q - \Delta_t)\}$, where $Q$ maintains the records in ascending order of timestamps;
while $Q \neq \emptyset$ do
    Pop out a record $(v, t_v)$ from $Q$;
    if $v \in C_q$ then
        continue;
    $C_q \leftarrow C_q \cup \{v\}$;
    for each edge $(v, u, t^*) \in E$ with increasing $t^*$ do
        if $t^* \in [t_v, t_q]$ and $u \notin C_q$ then
            $Q \leftarrow Q \cup \{(u, t^*)\}$;
return $C_q$;

procedure Multigraph Index Construction
Sort all records $(u, t_u, l_u)$ of $D$ in decreasing order of timestamps $t_u$;
Let $\mathcal{G}(V, E)$, where $E = \emptyset$;
for each record $(v, t_v, l_v) \in D$ do
    Find all records $(u, t_u, l_u) \in D$ such that $t_u \in [t_v - \delta_t, t_v + \delta_t]$, and $u \leftrightsquigarrow_{\bar{t}} v$ where $\bar{t} = \max\{t_u, t_v\}$;
    $E \leftarrow E \cup \{(u, v, \bar{t})\}$;
return $\mathcal{G}(V, E)$;
```

6 FAST BCG-INDEXING APPROACH

In this section, we propose an efficient indexing approach for potential transmission cluster discovery. We first offline construct a BCG-Index based on spatio-temporal database D. We then develop the index compression and partition techniques to optimize potential cluster discovery for query (q, t_q).

6.1 BCG-Index Construction

We build a new data structure of BCG-index, which is more space-efficient than multi-graph index. Due to a large number of contact edges across between users and users existed in multi-graphs, it needs to make sparsification on the contact information by keeping contact edges between users and spatio-temporal records instead.

Bipartite Contact Graph. We present a new definition of BCG. Let be the bipartite graph $B(L, R, E)$ where $L \subseteq V$ denotes the left vertex set of users, $R = \{(t, l) : t \text{ is a time and } l \text{ is a location}\}$ denotes the right vertex set of spatio-temporal records, and $E \subseteq L \times R$ represents the set of bipartite contact edges between users and spatio-temporal records. Specifically, a bipartite contact edge $e \in E$ is defined as follows.

Definition 5 (Bipartite Contact Edge). Given a record $(u, t_u, l_u) \in D$ and a time-location pair $r^* = (t^*, l^*)$, if $|t_u - t^*| \leq \delta_t$ and $|l_u - l^*| \leq \delta_l$, there exists a bipartite contact edge $e = (u, r^*) \in E$ in graph B, associated with an extra information (t_u, l_u).

A contact edge $(u, (t^*, l^*)) \in E$ keeps that user u has close contact with other users at $(t^*, l^*)$by Definition 1 and also its exact visiting information of (u, t_u, l_u). Note that $(t_u, l_u) = (t^*, l^*)$ may hold. For example, in Figure 4, the contact edge (Amy, "Starbucks, May 5 18:05") associated with an extra information of ("Starbucks, May 5 18:00") represents that Amy visited "Starbucks, May 5 18:00" and she had close contact to other users around May 5 18:05 at Starbucks. However, there may exist a large number of contact edges in bipartite graph B. As shown in Figure 4(b), Amy has

ALGORITHM 3: BCG-Index Construction

```
Input: Spatio-temporal database $D$, an incubation time $\Delta_t$
Output: BCG-Index $\mathcal{B}$
 1: Sort all records $(u, t_u, l_u)$ of $D$ in decreasing order of timestamps $t_u$;
 2: Partition all records of $D$ by a time interval $\Delta_t$ into multiple record components $D_1, \ldots, D_m$ and time-location pair components $TL_1, \ldots, TL_m$;
 3: for $i = 1$ to $m$ do
 4:     $L_i \leftarrow V(D_i)$, $R_i \leftarrow \emptyset$, $E_i \leftarrow \emptyset$;
 5:     for each record $(t_v, l_v) \in TL_i$ do
 6:         $S \leftarrow \{(t_v, l_v)\}$;
 7:         for each record $(t_u, l_u) \in TL_i$ with $t_u \in [t_v - \delta_t/2, t_v + \delta_t/2]$ do
 8:             if $|l_v - l_u| \leq \delta_l/2$ then
 9:                 $S \leftarrow S \cup \{(t_u, l_u)\}$;
10:         $TL_i \leftarrow TL_i \setminus S$;
11:         Compute an average spatio-temporal record: $(\bar{t}_u, \bar{l}_u)$ where $\bar{t}_u = \frac{\sum_{(t_u, l_u) \in S} t_u}{|S|}$ and $\bar{l}_u = \frac{\sum_{(t_u, l_u) \in S} l_u}{|S|}$
12:         $R_i \leftarrow R_i \cup \{(\bar{t}_u, \bar{l}_u)\}$;
13:     for each record $(v, t_v, l_v) \in D_i$ do
14:         Find all records $(t_u, l_u) \in R_i$ such that $|t_v - t_u| \leq \delta_t$ and $|l_v - l_u| \leq \delta_l$;
15:         Let be the bipartite contact edge $e = (v, (t_u, l_u))$ associated with record $(v, t_v, l_v)$;
16:         $E_i \leftarrow E_i \cup \{e\}$;
17:     $B_i \leftarrow (L_i, R_i, E_i)$ for $1 \leq i \leq m$;
18: return $\mathcal{B} = \{B_1, \ldots, B_m\}$;
```

three contact edges with the right nodes with IDs "1", "2", and "3", which is actually generated from one record "Amy, Starbucks, May 5 18:00" in D.

BCG-Index construction using spatio-temporal entity compressions and graph partitions. To generate a compact index, we propose two useful techniques of *spatio-temporal entity compression* and *graph partition* to reduce the size of bipartite contact graph $B(L, R, E)$ and improve the query efficiency. First, similar spatio-temporal records can be merged into one vertex in R, which can compress R and E with low-redundant close contact edges. Second, the entire records of D spans a long period that far exceeds the incubation time Δ. It suggests to partition D into several disjoint components $D_i \subseteq D$ where all records falls in one timeslot of Δ_t length. Assume that we have a total of m components and $\bigcup_{i=1}^{m} D_i = D$. For each record component D_i, we construct a corresponding graph B_i. Overall, the whole bipartite graph index is composed of multiple bipartite graphs as $\mathcal{B} = \{B_1, \ldots, B_m\}$.

Algorithm 3 shows the details of our BCG-Index index construction. Specifically, it first partitions the timestamps of all records by interval Δ_t (lines 1–2). Next, for each spatio-temporal record (t_v, l_v), the algorithm will merge all the spatio-temporal records into a set S (lines 6–10). The average spatio-temporal of the elements in set S will be inserted into R_i (lines 11–13). Then, for each record $(v, t_v, l_v) \in D$, it finds the partition D_i and constructs the bipartite contact edge $e = (v, (t_u, l_u))$ associated with (t_v, l_v) and enlarge bipartite graph B_i (lines 14–18). Finally, it returns the set of all partitioned bipartite graphs $\mathcal{B} = \{B_1, \ldots, B_m\}$ (line 20).

Based on the BCG-Index construction, we can check whether two users have close contact in the bipartite graph B in the following property.

PROPERTY 1. *Given a bipartite graph $B(L, R, E)$ and two users $v, u \in L$, if the edges $(u, (t_x, l_x)) \in E$ associated with (t_u, l_u) and $(v, (t_x, l_x)) \in E$ associated with (t_v, l_v) satisfy $|t_u - t_v| \leq \delta_t$ and $|l_u - l_v| \leq \delta_l$, v has contacted with u at time t_x.*

ALGORITHM 4: BCG-Index Based Potential Transmission Cluster Tracking

```
Input: BCG-Index $\mathcal{B}$, a query user $q$, a query time $t_q$
Output: Potential transmission cluster $C_q$
 1: Find all the corresponding BCG-Index within the time interval $[t_q - \Delta_t, t_q]$ as $\{B_i, B_{i+1}, \ldots, B_j\}$ and
    combine them as $B(L, R, E)$;
 2: A priority queue $Q \leftarrow \{(q, t_q - \Delta_t)\}$, where $Q$ maintains the records in ascending order of timestamps;
 3: while $Q \neq \emptyset$ do
 4:    Pop $(x, t_x)$ from $Q$; //$x$ may be a user or a time-location pair.
 5:    if $visit(x)$ = true then
 6:       continue;
 7:    $visit(x) \leftarrow$ true;
 8:    if $x \in L$ then
 9:       $C_q \leftarrow C_q \cup \{x\}$;
10:    for edge $(x, y) \in E$ accessed in the order of increasing $t^*$ of associated edge information $(t^*, l^*)$ do
11:       if $t^* \notin [t_x, t_q]$ or $visit(y)$ = true then
12:          continue;
13:       if $(x \in R$ and $|t^* - t_x| \leq \delta_t$ and $|l^* - l_x| \leq \delta_l)$ or $(x \in L)$ then
14:          $Q \leftarrow Q \cup \{(y, t^*)\}$;
15: return $C_q$;
```

6.2 BCG-Index Based Tracking Algorithm

Based on the constructed BCG-Index, the potential transmission cluster tracking algorithm is described in Algorithm 4. Specifically, the algorithm starts from the query user q and searches the potential patients by involving all reachable users in a BFS manner. First, it finds the partition by binary search and merges them as the bipartite indexing graph (line 1). Then, it implements the BFS by ascending order of time (lines 2–14). Note that it should check whether two users are contacted following Lemma 1 (line 13). Finally, it returns the potential cluster C_q (line 15).

Example 2. Figure 4 shows an example of MG-Indexing and BCG-Indexing method. Figure 4(a) shows all the 11 records in databse D. Figure 4(b) depicts the user-record relationships without spatio-temporal compressions and graph partitions. In Figure 4(d), all the 11 records are compressed into 4 time-location and partitioned into two parts by the interval 14 days. It reduces the size of bipartite graph in Figure 4(b) and avoids a large size of clique graph.

6.3 Complexity Analysis and Comparison

We first analyze the time and space complexity of Algorithms 1 and 2. We denote the number of users as $n = |V|$ and the number of records as $m = |D|$ in spatial-temporal database D, respectively. Assume that we use d to represent the maximum number of records in $[t - \delta_t, t + \delta_t]$ for any t, i.e., $d = \max_t |\{(v, l, t) \in D : t \in [t - \delta_t, t + \delta_t]\}|$. Moreover, we denoted c by the number of infectious records involving users in the answer cluster C_q within $[t_q - \Delta_t, t_q]$, i.e., $c = |\{(v, l, t) \in D : v \in C_q, t \in [t_q - \Delta_t, t_q]\}|$.

Theorem 1. *The online cluster search in Algorithm 1 takes $O(|C_q|c(\log m + d))$ in $O(m)$ space.*

Proof. We analyze the query time of Algorithm 1. For each user $v \in C_q$, it generates a contact graph G_{v,t_v} by enumerating its infectious records within $[t_q - \Delta_t, t_q]$. The graph size of G_{v,t_v} is bounded by $O(c)$. For an infectious record, the step of identifying all its close contacts takes $O(\log m + d)$ time using the binary search (lines 13–16 of Algorithm 1). Overall, Algorithm 1 takes $O(\sum_{v \in C_q} c(\log m + d)) \subseteq O(|C_q|c(\log m + d))$ to find all possible contact records in $O(m)$ space. □

Table 1. A Comparison of Three Different Potential Cluster Discovery Algorithms Proposed in This Article, in Terms of Time Compelxity, Space Complexity, and Index Size

Method	Query Time	Query Space	Indexing Time	Index Size
Online using binary search (Algorithm 1)	$O(\lvert C_q\rvert c(\log m + d))$	$O(m)$	–	–
Online using R-tree (Algorithm 1)	$O(\lvert C_q\rvert c(\log m + d'))$	$O(m \log m)$	$O(m \log m)$	$O(m \log m)$
MG-Indexing (Algorithm 2)	$O(c \log c)$	$O(mn)$	$O(m(\log m + n))$	$O(mn)$
BCG-Indexing (Algorithms 3 & 4)	$O(b \log b)$	$O(md)$	$O(m(\log m + d))$	$O(md)$

For the online potential transmission cluster search in Algorithm 1, if we implement the close contact using the R-tree index instead of the binary search, the close contact of a new query takes $O(\log m + d')$ time in worst. Here, the parameter d' represents the maximum number of records satisfying $|t' - t| \leq \delta_t$ and $|l' - l| \leq \delta_l$ for any t and l, i.e., $d' = \max_{t,l} |\{(v, l', t') \in D : |t' - t| \leq \delta_t, |l' - l| \leq \delta_l\}|$. Therefore, Algorithm 1 using the R-tree index takes the query time in $O(|C_q|c(\log m + d'))$, the query space in $O(m \log m)$, the index construction time in $O(m \log m)$, and the index size in $O(m \log m)$.

Theorem 2. *The multigraph index construction in Algorithm 2 takes $O(m(\log m + n))$ in $O(mn)$ space. The disk size of multigraph index $\mathcal{G}$ takes $O(mn)$ space.*

Proof. Algorithm 2 first sorts all records in the increasing order of timestamps using $O(m \log m)$ time. Then, for each record in D, it finds its close contact with at most n (lines 13–15 of Algorithm 2) and takes $O(mn)$ time totally. Overall, Algorithm 2 takes $O(m(\log m + n))$ time in $O(mn)$ space. □

Theorem 3. *The multigraph index-based cluster discovery in Algorithm 2 takes $O(c \log c)$ in $O(mn)$ space.*

Proof. Algorithm 2 applies the BFS manner to identify the potential cluster. For each record in the answer cluster C_q, it will be inserted to the priority queue at most once in $O(\log c)$. So, Algorithm 2 takes $O(c \log c)$ in $O(mn)$ space. □

In this section, we analyze the complexity of BCG-Indexing index construction in Algorithm 3 and BCG-Indexing-based cluster discovery in Algorithm 4. We denote the maximum edge size of bipartite graph in $\mathcal{B} = \{B_1, \ldots, B_m\}$ as $b = \max_{1 \leq i \leq m} |E(B_i)|$.

Theorem 4. *The BCG-index construction in Algorithm 3 takes $O(m(\log m + d))$ in $O(md)$ space. The disk size of multigraph index $\mathcal{G}$ takes $O(md)$ space.*

Proof. Algorithm 3 first sorts all records in the increasing order of timestamps using $O(m \log m)$ time. Then, for each record in D, the algorithm connects its user to the spatio-temporal records in $O(\log m + d)$. So, Algorithm 3 takes $O(m(\log m + d))$ time in $O(md)$ space. □

Theorem 5. *The* BCG-Indexing *based cluster discovery in Algorithm 4 takes $O(b \log b)$ in $O(md)$ space.*

Proof. It contains at most two partitions in bipartite indexing graph, so the size of it is $O(b)$. Algorithm 4 also applies the BFS manner on bipartite indexing graph. Each insertion takes $\log(b)$ and the algorithm totally takes $O(b \log b)$ time in $O(md)$ space. □

Table 1 summarizes the algorithm complexity of three transmission cluster discovery algorithms, in terms of query time, index construction, and index size. Overall, our BCG-Indexing method achieves faster query time in $O(b \log b)$ than MG-Indexing method in $O(c \log c)$ time, as $b \leq c$ usually holds. This observation can be validated in our experimental results as shown in Table 3, due to our efficient spatio-temporal compression and graph partitions. Moreover, BCG-Indexing

Table 2. The Statistics of Check-In Datasets

Name	$\|V\|$	$\|D\|$	Period (days)
Brightkite	51,406	4,747,284	923
Brightkite-Syn	51,406	52,220,124	923
Gowalla	107,092	6,442,890	626
Gowalla-Syn	107,092	51,543,120	626
Foursquare	266,909	33,278,683	533
Foursquare-Syn	266,909	118,453,113	533
LBSN	2,733,324	90,048,627	666

method significantly takes much less index construction time and cheaper index space, in contrast to the MG-Indexing approach in Algorithm 2.

7 EXPERIMENTS

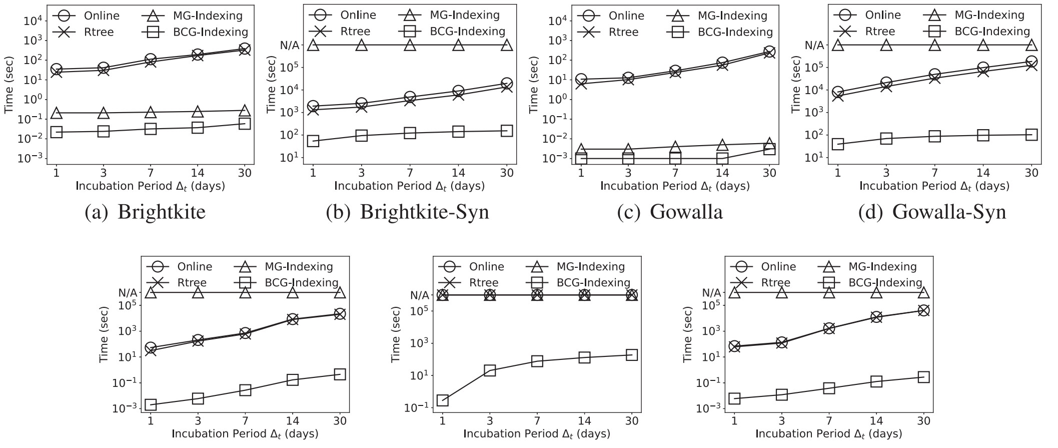
In this section, we conduct extensive experiments to evaluate the performance of our proposed solutions. All algorithms are implemented in C++. All the experiments are conducted on a Linux Server with Intel Xeon E5-2630 v4 (2.2 GHz) and 256 GB main memory.

Datasets. We use four real-world check-in datasets Brightkite (BK for short), Gowalla (GW for short) [19], Foursquare (FQ for short) [50, 51], and LBSN [49]. All these datasets are the location-based datasets where users shared their locations by check-ins. Each check-in record contains the user ID, time, and location. The location is presented by latitude and longitude. Brightkite [6] collects 4,747,284 check-ins of 51,406 users over the period of April 2008–October 2010. Gowalla [6] collects a total of 6,442,890 check-ins of 107,092 users over the period of February 2009–October 2010. Foursquare [51] collects 33,278,683 check-ins of 266,909 users over the period of April 2012–September 2013. Moreover, a large dataset LBSN [49] collects 90,048,627 check-ins of 2,733,324 users over the period of April 2012–January 2014, which are raw check-ins of Foursquare. To evaluate efficiency, we randomly add more records into Brightkite, Gowalla, and Foursquare, denoted as Brightkite-Syn, Gowalla-Syn, and Foursquare-Syn, respectively. All the newly generated records are composed of a random user ID and a random pair of time and location, both of which are selected from the original datasets. The details of seven datasets are reported in Table 2.

Benchmarks and queries generation. Besides the check-in datasets, we also generate a benchmark of infectious transmissions. We use a classical influential model of independent cascade [17] to generate influential individuals and simulate virus propagation. First, we randomly select a few users as the transmission source with a random infected time. We adopt the sampling rate of transmission sources by using real-world data of average infection rate as 10.1%, which comes from The New York Times, based on reports from state and local health agencies in US.[1] Thus, we select 5,136, 10,704, and 26,678 users as the transmission sources in Brightkite, Gowalla, and Foursquare, respectively. Then, we apply independent cascade model with an uniform probability 20% to simulate the influential individuals within 14 days after their infected time. Note that each individual only could be influenced once at the first infected time. For the queries, we randomly select 100 users from influential individuals at onset time (14 days after infected time) as queries.

Methods compared. We evaluate and compare different models and algorithms, which address a novel problem of potential COVID-19 transmission cluster discovery. To evaluate the efficiency, we compare Rtree [13, 33] and our proposed algorithms as follows.

[1] https://github.com/nytimes/covid-19-data.



(a) Brightkite (b) Brightkite-Syn (c) Gowalla (d) Gowalla-Syn

(e) Foursquare (f) Foursquare-Syn (g) LBSN

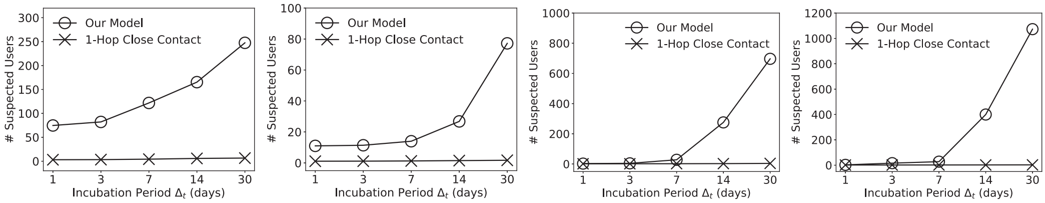
Fig. 5. The average query time of all methods on seven datasets varied by the incubation period Δ_t.

— Online: is an online search approach in Algorithm 1, which uses the binary search to find close contacts directly on spatio-temporal logs.
— Rtree: is an online search approach using a well-known Rtree indexing [13, 33] to find all close contacts.
— MG-Indexing: is a multigraph index-based method to construct a contact multigraph and search potential clusters in Algorithm 2.
— BCG-Indexing: is our efficient cluster discovery method, which constructs BCG-Index in Algorithm 3 and finds potential transmission cluster in Algorithm 4.

Moreover, to evaluate the quality of our potential transmission cluster model, we additionally implement and compare another 1-hop potential cluster model that finds all close contact users to a given query user.

Evaluation metrics and parameter settings. For efficiency evaluation, we report the average running time over 100 queries. For the indexing evaluation, we report the running time and index size on disk. To evaluate the quality and robustness of potential transmission cluster models, we report the number of potential infected users and also the percentage of missing users in comparison with ground-truth clusters. By default, we set two close contact parameters of time $\delta_t = 15$ minutes and distance $\delta_l = 5$ meters and the incubation period $\Delta_t = 14$ days. We treat the running time as infinite (INF for short) if the algorithm runs exceeding 30 hours.

Exp-1: Query time evaluation of all methods. We first evaluate the query time of all methods varied on incubation period Δ_t on all datasets. Figure 5 shows the average query time of four competitive methods Online, Rtree, MG-Indexing, and BCG-Indexing. Note that in the datasets Brightkite-Syn, Gowalla-Syn, Foursquare, Foursquare-Syn, and LBSN, the MG-Indexing approach fails to finish the multigraph index construction within a limited time of 30 hours. Furthermore, Online and Rtree are low-efficient because of massive close contact queries for online search. So we only run 10 queries of potential cluster tracking on Brightkite-Syn, Gowalla-Syn, Foursquare, Foursquare-Syn, and LBSN. All methods take more time with the increased incubation period Δ_t. Our efficient method BCG-Indexing clearly outperforms other competitors Online, Rtree, and MG-Indexing on all datasets. In the Foursquare-Syn, only our method BCG-Indexing identifies the transmission clusters successfully.



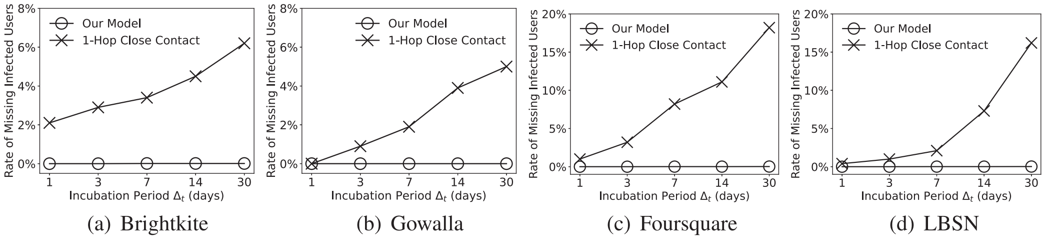
(a) Brightkite (b) Gowalla (c) Foursquare (d) LBSN

Fig. 6. The average number of potential infecting users in 1-hop close contacts and potential transmission clusters on four real datasets varied by the incubation period Δ_t.

Exp-2: Quality evaluation on the number of potential users. In this experiment, we compare our potential transmission cluster model and another 1-hop close contact model. Our model identifies all the transmission chains and potential infected users in potential clusters. Different from our model, the 1-hop close contact model only identifies the potential users who have contacted with the query user individually. Figure 6 shows the average number of potential users on all real-world datasets. Both methods identify more potential users with the increased incubation period Δ_t. Significantly, our model identifies much more potential users than the 1-hop close contact model. Although identifying more potential users may lead to more false positives in the result, the benefit is missing fewer cases of "important" suspected users in larger potential clusters for those critically dangerous disease pandemics. With 30 days incubation period, our model identifies at least 45 times of the potential users of 1-hop close contact model. Figure 7 shows the average rate of missing infected users on all real-world datasets. Our model miss no infected users in all datasets and 1-hop close contact model miss nearly 20% in the largest two datasets Foursquare and LBSN.

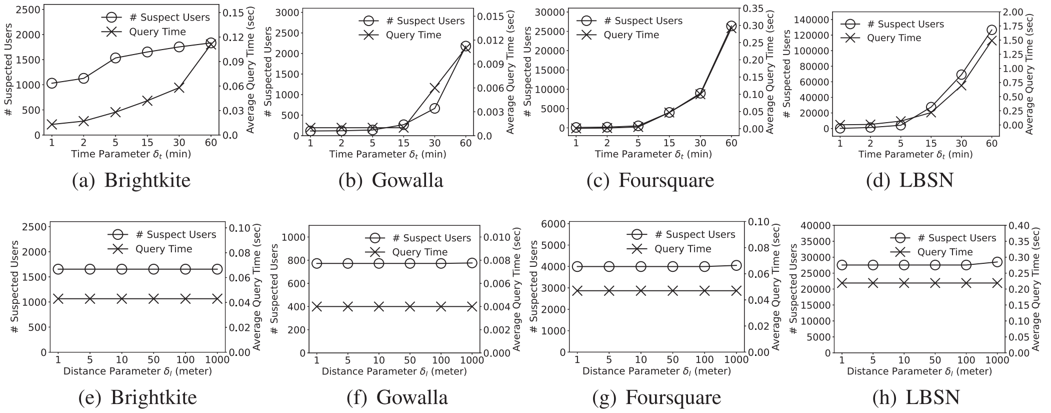
Exp-3: Parameter sensitivity evaluation on δ_t and δ_l. In this experiment, we conduct sensitivity evaluation of our potential cluster model by varying two parameters of time threshold δ_t and distance threshold δ_l. We test on all real-world datasets. Figures 8(a)–(d) show the number of suspected users and the average query time varied by the time threshold δ_t. The time parameter δ_t from 1 minute to 60 minutes. Both the number of suspected users and the average query time increase with the increased δ_t. In addition, we vary the distance parameter δ_l from 1 meter to 1,000 meters. Figures 8(e)–(h) show the results varied by the distance threshold δ_l on all real-world datasets. The number of suspected users and the average query time keep stable with the increased δ_l. The reason is that most check-ins are sparse in spatial locations in our used datasets. These check-in records cannot distinguish different tracking positions in the same building, e.g., one restaurant uses one consistent spatial location of check-in data. Figures 8(g)–(h) have a slightly increased number of suspected users when δ_l increases from 100 to 1,000 meters. If we set the distance threshold $\delta_l = 5000$ meters, the number of suspected users C_q increases to 94,725, which is 3.3 times of $|C_q| = 28{,}506$ when $\delta_l = 1000$. Here, we only test $\delta_l \in [1,1000]$, because the COVID-19 is difficult to transmit for a long distance in real world. As a result, the size of potential transmission cluster and the average query time have a slightly stable performance with the increased δ_l in Figures 8(e)–(h).

Exp-4: Indexing evaluation of different index construction methods. In this experiment, we evaluate the efficiency of two indexing methods MG-Indexing and BCG-Indexing. Figure 9 reports the index construction time on seven datasets. As we can see, our proposed method BCG-Indexing clearly wins MG-Indexing on all datasets. In the Brightkite-Syn, Gowalla-Syn, Foursquare, Foursquare-Syn, and LBSN, MG-Indexing fails to construct the index timely within 30



(a) Brightkite (b) Gowalla (c) Foursquare (d) LBSN

Fig. 7. The average rate of missing infected users in 1-hop close contacts and potential transmission clusters on four real datasets varied by the incubation period Δ_t.



(a) Brightkite (b) Gowalla (c) Foursquare (d) LBSN

(e) Brightkite (f) Gowalla (g) Foursquare (h) LBSN

Fig. 8. Parameter sensitivity evaluation varied by δ_t and δ_l on all datasets.

Table 3. The Index Size (in Megabytes) of Different Indexing Methods and Various Parameters on all Datasets

	Space Cost				Parameters					
Dataset	Online	R-tree	MG-Index	BCG-Index	m	n	$\lvert C_q \rvert$	c	b	d
BK	18**MB**	379**MB**	601**MB**	45**MB**	4**M**	51**K**	165	42**K**	3**K**	549
BK-Syn	208**MB**	4**GB**	/	495**MB**	52**M**	51**K**	3**K**	40**M**	638**K**	6**K**
GW	25**MB**	515**MB**	90**MB**	29**MB**	6**M**	107**K**	77	36**K**	2**K**	2**K**
GW-Syn	206**MB**	4**GB**	/	236**MB**	51**M**	107**K**	7**K**	63**M**	847**K**	12**K**
FQ	132**MB**	3**GB**	/	151**MB**	33**M**	266**K**	275	41**K**	4**K**	9**K**
FQ-Syn	473**MB**	9**GB**	/	806**MB**	118**M**	266**K**	14**K**	78**M**	1**M**	26**K**
LBSN	360**MB**	7**GB**	/	410**MB**	90**M**	3**M**	1**K**	122**K**	5**K**	15**K**

Here, $\mathbf{K} = 10^3$ and $\mathbf{M} = 10^6$. $n = |V|$ and $m = |D|$ are the number of users and records in database D, respectively. The parameters $|C_q|$ and c are the average number of users and close contact edges in ground-truth cluster C_q, respectively. b is the average size of bipartite graphs in BCG-Index. d is the maximum number of records within a close contact time window.

hours, while BCG-Indexing only takes nearly 1,000 seconds on Brightkite-Syn and Gowalla-Syn, and 5,000 seconds on Foursquare. In addition, Table 3 reports the index size on disk and various parameters in complexity analysis for R-tree, MG-Indexing, and BCG-Indexing. For the index size, BCG-Indexing takes about three times as the size of original dataset in the worst cases, which is

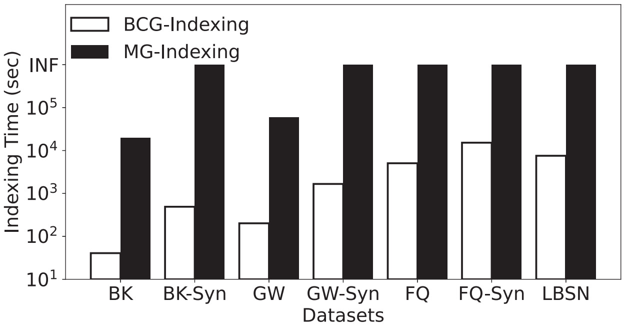
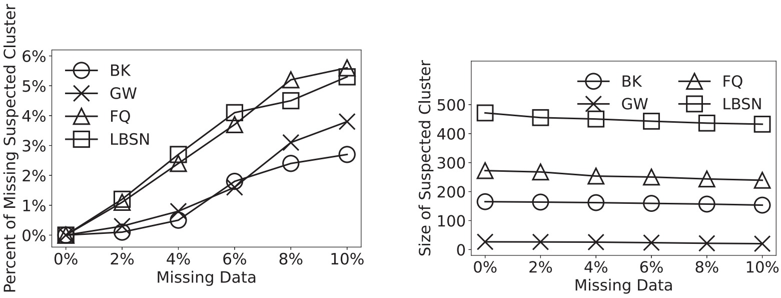


Fig. 9. The index construction time of different indexing methods on all datasets.



(a) The percentage of missing infecting users in potential clusters

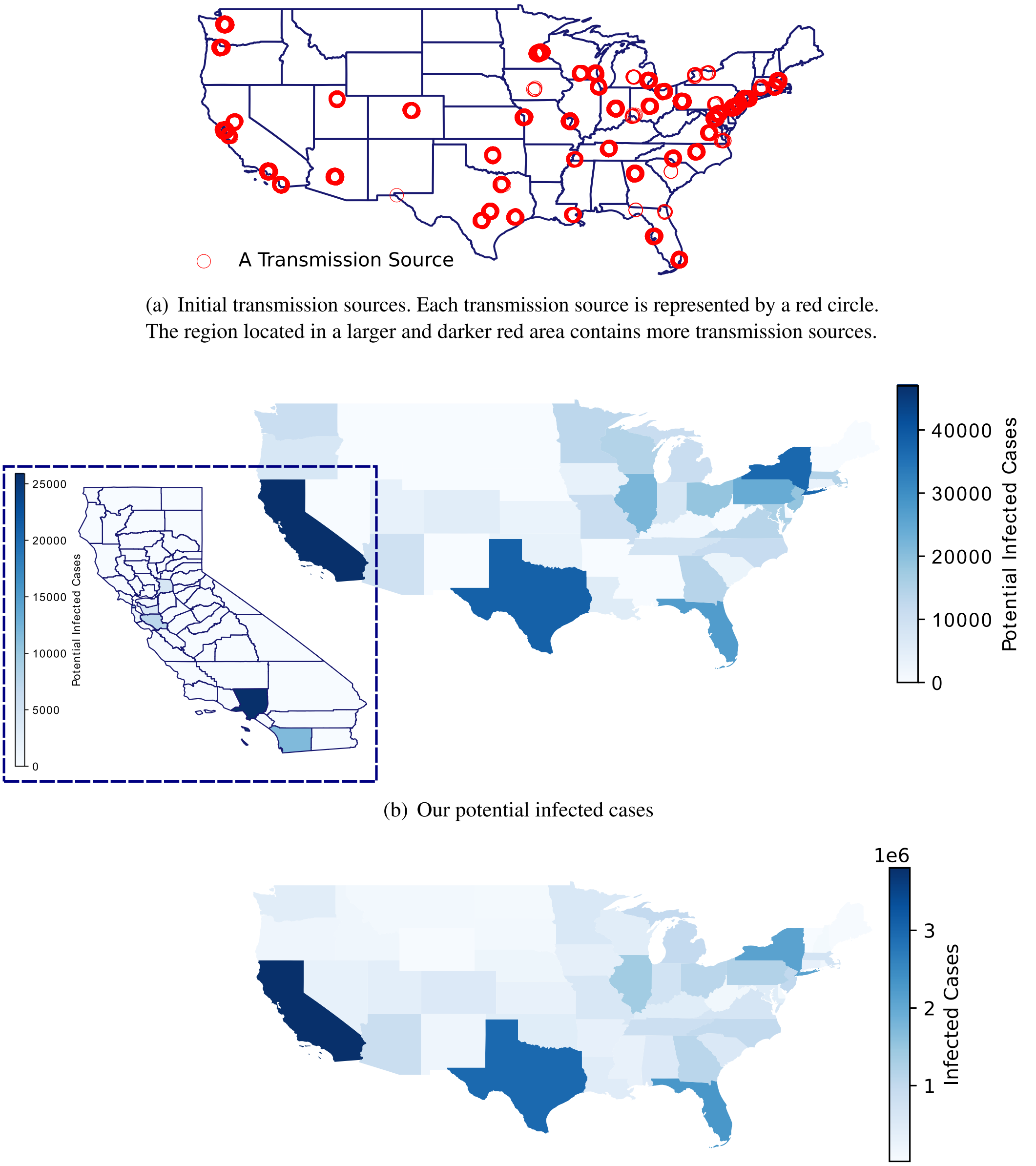
(b) The number of infecting users in potential clusters

Fig. 10. Quality evaluation of our potential transmission cluster model with missing records.

much smaller than R-tree and MG-Indexing. Moreover, the parameters $c \leq m$ and $b \leq c$ holds. This validates the complexity analysis and comparison in Section 6.3.

Exp-5: Quality evaluation of potential transmission cluster model on datasets with missing records. We also evaluate the quality of our potential cluster model when the missing data records happened in spatio-temporal D. We randomly remove 2% to 10% records from each dataset and compare the percent of missing infectious users in the ground-truth transmission clusters. Figure 10(a) shows the percentage of missing infectious users on Brightkite, Gowalla, and Foursquare. Our model successfully identifies the potential cluster members with missing at most 3% on Brightkite, 4% on Gowalla, and 6% on Foursquare with a proportionality of 10% data records missed from D, reflecting a robust modeling of our potential transmission cluster definition in real-life noisy environment. Figure 10(b) shows the size of potential transmission cluster with missing records of all datasets. Our model identifies most of infecting users even with 10% data records missed.

Exp-6: A case study of COVID-19 transmission using Foursquare dataset and US confirmed cases. We conduct a COVID-19 transmission case study to evaluate our potential transmission cluster model. We use two real-world datasets of users' check-ins in Foursquare [51] and COVID-19 confirmed cases in United States [39]. Specifically, we first randomly select 26,678 users in Foursquare dataset as the transmission sources, where the sampling rate of transmission sources in different states are different by following the infection rate of states in US [39]. Each transmission source is assigned with an infected time, which uses the timestamp of a random check-in



(a) Initial transmission sources. Each transmission source is represented by a red circle. The region located in a larger and darker red area contains more transmission sources.

(b) Our potential infected cases

(c) Real-world infected cases

Fig. 11. A case study of COVID-19 transmission using Foursquare dataset in United States.

record in US. Figure 11(a) depicts the distribution of transmission sources over all states in US, using a map visualization tool Geopandas.[2] We treat all these given transmission sources as our query. Next, we apply our potential transmission model to discover potential COVID-19 transmission clusters within 14 days, for all query infected users. The volume distribution results of our potential infected users over different US states are shown in Figure 11(b). To achieve a detailed

[2] https://geopandas.org/.

visualization, we plot a zoom-in view of potential transmission clusters in California state, where most users locate in three regions of Los Angeles, San Diego, and San Francisco Bay Area. Finally, to validate the effectiveness of our model, we also report the real-world COVID-19 confirmed cases in Figure 11(c). Compare the results in Figures 11(b) and 11(c), we observe that our model detect no potential transmission clusters in some states, i.e., Idaho and Montana. Because the Foursquare dataset has no check-in records located in these states. Nevertheless, our distribution results in Figure 11(b) are very similar with the real US confirmed cases in Figure 11(c). The four states California, Texas, New York, and Florida have the maximum potential infected users, which are the same as the real-world confirmed cases for all time in US [39]. This confirms the effectiveness of our proposed potential transmission cluster model in the case of COVID-19 transmissions.

8 CONCLUSION AND FUTURE WORK

In this article, we motivate and investigate the problem of discovering query-dependent COVID-19 transmission clusters on spatio-temporal logs, which finds all potential infected users to a given query of patient user and infection time. We propose three different methods including the online search approach and two indexing based solutions. Our novel BCG-indexing approach achieves a good balance of index construction and online query processing for fast suspected cluster discovery. Extensive experiments on real-world datasets validate the effectiveness of our suspected cluster model and query-dependent suspected cluster tracking algorithms.

Although the proposed BCG-indexing methods can efficiently uncover potential COVID-19 transmission clusters, the current algorithms leave three open issues for further improvements. First, the querying algorithms are designed only for one single query. In real applications, several patients may be confirmed simultaneously. Thus, it needs an efficient transmission cluster search for multiple queries. We could further explore the fast detection of potential transmission clusters for multiple queries in a batch. Second, the current algorithm of BCG-index construction is static, which lies on the predefined parameters δ_t, δ_l, and Δ_t. It is unclear how can efficiently update BCG-index over different dynamic settings for various disease transmissions. Third, the current result of transmission clusters does not involve any infected risk analytics. In reality, different persons may have quite different probabilities of infected risks even they are in the same transmission cluster. Last but not least, this work also opens up several other interesting problems, e.g., finding potential outbreaks hotspots, transmission source detection, BCG-index maintenance over evolving spatial-temporal data, and infected risk assessments.

REFERENCES

[1] Aniruddha Adiga, Lijing Wang, Benjamin Hurt, Akhil Sai Peddireddy, Przemyslaw Porebski, Srinivasan Venkatramanan, Bryan Lewis, and Madhav Marathe. 2021. All models are useful: Bayesian ensembling for robust high resolution covid-19 forecasting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2505–2513.

[2] Pritom Ahmed, Mahbub Hasan, Abhijith Kashyap, Vagelis Hristidis, and Vassilis J. Tsotras. 2017. Efficient computation of top-k frequent terms over spatio-temporal ranges. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 1227–1241.

[3] Md Musfique Anwar, Chengfei Liu, and Jianxin Li. 2019. Discovering and tracking query oriented active online social groups in dynamic information network. *World Wide Web* 22, 4 (2019), 1819–1854.

[4] Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. 2018. Spatio-temporal data mining: A survey of problems and methods. *Computing Surveys* 51, 4 (2018), 1–41.

[5] Chloë Brown, Jagmohan Chauhan, Andreas Grammenos, Jing Han, Apinan Hasthanasombat, Dimitris Spathis, Tong Xia, Pietro Cicuta, and Cecilia Mascolo. 2020. Exploring automatic diagnosis of COVID-19 from crowdsourced respiratory sound data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 3474–3484.

[6] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility: User movement in location-based social networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1082–1090.

[7] Rui Dai, Shenkun Xu, Qian Gu, Chenguang Ji, and Kaikui Liu. 2020. Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 3074–3082.

[8] Chiara Francalanci, Barbara Pernici, and Gabriele Scalia. 2017. Exploratory spatio-temporal queries in evolving information. In *Proceedings of the International Workshop on Mobility Analytics for Spatio-Temporal and Social Data.* 138–156.

[9] Zhenxin Fu, Yu Wu, Hailei Zhang, Yichuan Hu, Dongyan Zhao, and Rui Yan. 2020. Be aware of the hot zone: A warning system of hazard area prediction to intervene novel coronavirus COVID-19 outbreak. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 2241–2250.

[10] Fred Galvin. 1995. The list chromatic index of a bipartite multigraph. *Journal of Combinatorial Theory, Series B* 63, 1 (1995), 153–158.

[11] Salah Ghamizi, Renaud Rwemalika, Maxime Cordy, Lisa Veiber, Tegawendé F. Bissyandé, Mike Papadakis, Jacques Klein, and Yves Le Traon. 2020. Data-driven simulation and optimization for Covid-19 exit strategies. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 3434–3442.

[12] Jonathan L. Gross and Jay Yellen. 2005. *Graph Theory and Its Applications*. CRC Press.

[13] Antonin Guttman. 1984. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data.* 47–57.

[14] Qianyue Hao, Lin Chen, Fengli Xu, and Yong Li. 2020. Understanding the urban pandemic spreading of COVID-19 with real world mobility data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 3485–3492.

[15] Frank Harary. 1991. *Graph Theory.* Addison-Wesley.

[16] Xiaoyong Jin, Yu-Xiang Wang, and Xifeng Yan. 2021. Inter-series attention model for COVID-19 forecasting. In *Proceedings of the 2021 SIAM International Conference on Data Mining.* 495–503.

[17] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 137–146.

[18] Minseok Kim, Junhyeok Kang, Doyoung Kim, Hwanjun Song, Hyangsuk Min, Youngeun Nam, Dongmin Park, and Jae-Gil Lee. 2020. Hi-COVIDNet: Deep learning approach to predict inbound COVID-19 patients and case study in south korea. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 3466–3473.

[19] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. Retrieved June 2022 from http://snap.stanford.edu/data.

[20] Hui-Jia Li, Zhan Bu, Zhen Wang, and Jie Cao. 2019. Dynamical clustering in electronic commerce systems via optimization and leadership expansion. *IEEE Transactions on Industrial Informatics* 16, 8 (2019), 5327–5334.

[21] Hui-Jia Li, Lin Wang, Yan Zhang, and Matjaž Perc. 2020. Optimization of identifiability for efficient community detection. *New Journal of Physics* 22, 6 (2020), 063035.

[22] Hui-Jia Li, Wenzhe Xu, Shenpeng Song, Wen-Xuan Wang, and Matjaž Perc. 2021. The dynamics of epidemic spreading on signed networks. *Chaos, Solitons & Fractals* 151 (2021), 111294.

[23] Ting Li, Junbo Zhang, Kainan Bao, Yuxuan Liang, Yexin Li, and Yu Zheng. 2020. Autost: Efficient neural architecture search for spatio-temporal prediction. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 794–802.

[24] Yang Liu, Zhonglei Gu, and Jiming Liu. 2021. Uncovering transmission patterns of COVID-19 outbreaks: A region-wide comprehensive retrospective study in Hong Kong. *EClinicalMedicine* 36 (2021), 100929.

[25] Yang Liu, Zhonglei Gu, Shang Xia, Benyun Shi, X.-N. Zhou, Yong Shi, and Jiming Liu. 2020. What are the underlying transmission patterns of covid-19 outbreak?–an age-specific social contact characterization. *EClinicalMedicine* 22 (2020), 100354.

[26] Yuyu Luo, Wenbo Li, Tianyu Zhao, Xiang Yu, Lixi Zhang, Guoliang Li, and Nan Tang. 2020. Deeptrack: Monitoring and exploring spatio-temporal data: A case of tracking COVID-19. *Proceedings of the VLDB Endowment* 13, 12 (2020), 2841–2844.

[27] Nikos Mamoulis, Huiping Cao, George Kollios, Marios Hadjieleftheriou, Yufei Tao, and David W. Cheung. 2004. Mining, indexing, and querying historical spatiotemporal data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 236–245.

[28] Yannis Manolopoulos, Apostolos N. Papadopoulos, Apostolos N. Papadopoulos, and Yannis Theodoridis. 2006. *R-Trees: Theory and Applications: Theory and Applications*. Springer Science & Business Media.

[29] Lukas M. Marti, Michael P. Dal Santo, and Ronald Keryuan Huang. 2016. Modeling significant locations. US Patent 9,267,805.

[30] Mirco Nanni, Gennady Andrienko, Albert-László Barabási, Chiara Boldrini, Francesco Bonchi, Ciro Cattuto, Francesca Chiaromonte, Giovanni Comandé, Marco Conti, Mark Coté, Frank Dignum, Virginia Dignum, Josep Domingo-Ferrer, Paolo Ferragina, Fosca Giannotti, Riccardo Guidotti, Dirk Helbing, Kimmo Kaski, Janos Kertesz, Sune Lehmann, Bruno Lepri, Paul Lukowicz, Stan Matwin, David Megias Jimenez, Anna Monreale, Katharina Morik, Nuria Oliver, Andrea Passarella, Andrea Passerini, Dino Pedreschi, Alex Pentland, Fabio Pianesi, Francesca Pratesi, Salvatore Rinzivillo, Salvatore Ruggieri, Arno Siebes, Vicenc Torra, Roberto Trasarti, Jeroen van den Hoven, and Alessandro Vespignani. 2021. Give more data, awareness and control to individual citizens, and they will help COVID-19 containment. *Ethics and Information Technology* 23, 1 (2021), 1–6.

[31] Bing Ni, Qiaomu Shen, Jiayi Xu, and Huamin Qu. 2017. Spatio-temporal flow maps for visualizing movement and contact patterns. *Visual Informatics* 1, 1 (2017), 57–64.

[32] Maya Okawa, Tomoharu Iwata, Takeshi Kurashima, Yusuke Tanaka, Hiroyuki Toda, and Naonori Ueda. 2019. Deep mixture point processes: Spatio-temporal event prediction with rich contextual information. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 373–383.

[33] Dimitris Papadias, Yufei Tao, P. Kanis, and Jun Zhang. 2002. Indexing spatio-temporal data warehouses. In *Proceedings of the IEEE International Conference on Data Engineering*. 166–175.

[34] Zhe Peng, Jinbin Huang, Haixin Wang, Shihao Wang, Xiaowen Chu, Xinzhi Zhang, Li Chen, Xin Huang, Xiaoyi Fu, Yike Guo, , and Jianliang Xu. 2021. BU-Trace: A permissionless mobile system for privacy-preserving intelligent contact tracing. In *Proceedings of the DASFAA 2021 International Workshops: BDQM, GDMA, MLDLDSA, MobiSocial, and MUST*. 381–397.

[35] Zhe Peng, Cheng Xu, Haixin Wang, Jinbin Huang, Jianliang Xu, and Xiaowen Chu. 2021. P2B-trace: Privacy-preserving blockchain-based contact tracing to combat pandemics. In *Proceedings of the 2021 International Conference on Management of Data*. 2389–2393.

[36] Putsadee Pornphol and Suphamit Chittayasothorn. 2020. System dynamics model of COVID-19 pandemic situation: The case of phuket Thailand. In *Proceedings of the International Conference on Computer Modeling and Simulation*. 77–81.

[37] Amray Schwabe, Joel Persson, and Stefan Feuerriegel. 2021. Predicting COVID-19 spread from large-scale mobility data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3531–3539.

[38] Yufei Tao and Dimitris Papadias. 2001. The mv3r-tree: A spatio-temporal access method for timestamp and interval queries. In *Proceedings of the International Conference on Very Large Data Bases*. 431–440.

[39] The New York Times. 2021. Coronavirus (Covid-19) Data in the United States. Retrieved June 2022 from https://github.com/nytimes/covid-19-data.

[40] Vincent S. Tseng, Josh Jia-Ching Ying, Stephen T. C. Wong, Diane J. Cook, and Jiming Liu. 2020. Computational intelligence techniques for combating COVID-19: A survey. *IEEE Computational Intelligence Magazine* 15, 4 (2020), 10–22.

[41] Bowen Wang, Yanjing Sun, Trung Q. Duong, Long D. Nguyen, and Lajos Hanzo. 2020. Risk-aware identification of highly suspected COVID-19 cases in social IoT: A joint graph theory and reinforcement learning approach. *IEEE Access* 8 (2020), 115655–115661.

[42] WHO. 2020. Retrieved June 2022 from https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200306-sitrep-46-covid-19.pdf?sfvrsn=96b04adf_4.

[43] Guojun Wu, Yichen Ding, Yanhua Li, Jie Bao, Yu Zheng, and Jun Luo. 2017. Mining spatio-temporal reachable regions over massive trajectory data. In *Proceedings of the IEEE International Conference on Data Engineering*. 1283–1294.

[44] Huanhuan Wu, James Cheng, Yi Lu, Yiping Ke, Yuzhen Huang, Da Yan, and Hejun Wu. 2015. Core decomposition in large temporal graphs. In *Proceedings of the IEEE International Conference on Big Data*. 649–658.

[45] Huanhuan Wu, Yuzhen Huang, James Cheng, Jinfeng Li, and Yiping Ke. 2016. Reachability and time-based path queries in temporal graphs. In *Proceedings of the IEEE International Conference on Data Engineering*. 145–156.

[46] Marcin Wylot, Philippe Cudré-Mauroux, Manfred Hauswirth, and Paul Groth. 2017. Storing, tracking, and querying provenance in linked data. *IEEE Transactions on Knowledge and Data Engineering* 29, 8 (2017), 1751–1764.

[47] Xiaopeng Xiong, Mohamed F. Mokbel, and Walid G. Aref. 2005. Sea-cnn: Scalable processing of continuous k-nearest neighbor queries in spatio-temporal databases. In *Proceedings of the IEEE International Conference on Data Engineering*. 643–654.

[48] Zhe Xu, Lei Shi, Yijin Wang, Jiyuan Zhang, Lei Huang, Chao Zhang, Shuhong Liu, Peng Zhao, Hongxia Liu, Li Zhu, and Y. Tai. 2020. Pathological findings of COVID-19 associated with acute respiratory distress syndrome. *The Lancet Respiratory Medicine* 8, 4 (2020), 420–422.

[49] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudre-Mauroux. 2019. Revisiting user mobility and social relationships in LBSNs: A hypergraph embedding approach. In *Proceedings of the World Wide Web Conference*. 2147–2157.

[50] Dingqi Yang, Daqing Zhang, Longbiao Chen, and Bingqing Qu. 2015. Nationtelescope: Monitoring and visualizing large-scale collective behavior in LBSNs. *Journal of Network and Computer Applications* 55 (2015), 170–180.

[51] Dingqi Yang, Daqing Zhang, and Bingqing Qu. 2016. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology* 7, 3 (2016), 1–23.

[52] Zhao Yang and Nathalie Japkowicz. 2017. Meta-morisita index: Anomaly behaviour detection for large scale tracking data with spatio-temporal marks. In *Proceedings of the IEEE International Conference on Data Mining Workshops*. 675–682.

[53] Tianming Zhang, Yunjun Gao, Lu Chen, Wei Guo, Shiliang Pu, Baihua Zheng, and Christian S. Jensen. 2019. Efficient distributed reachability querying of massive temporal graphs. *The VLDB Journal* 28, 6 (2019), 871–896.

Received 23 June 2021; revised 24 December 2021; accepted 8 May 2022