

based contact tracing can be automatically switched off for better energy savings. Similarly, if the density of WiFi AP is less, then Bluetooth and acoustic based solutions could be activated. They can help each other in a crowded and occluded (e.g., by walls and other types of obstacles) environment for accurate and reliable contact tracing. Note that Bluetooth gives relative location information whereas WiFi gives absolute location information. There might be a mismatch in the locations as identified by WiFi data modality in conjunction with Bluetooth data. In order to resolve this issue, we can model the location of the user using probabilistic techniques, and then use filtering techniques in order to derive a more accurate location. Readers interested in more details about these techniques can refer to Kalman filtering and related topics in [17, 13].

Integrated Contact Tracing with Cellular Network. The location of a smartphone can also be identified from its communication with nearby cell towers. Since a phone has to connect with cell towers in order to send and receive data through cellular network, it constantly searches nearby cell towers and initiates connections during movements. In each established connection, a cell tower not only knows which phone is trying (each phone has a unique identifier) to connect at which time, but can also calculate the distance from itself to such a phone (e.g., using the time elapsed between a ping command, and the corresponding reply). As such, having access to the locations of cell towers, as well as the distance of a phone from each of the involved towers, we can use a “triangulation” technique to pinpoint the location of a phone. However, in practice, such techniques often can only locate a smartphone in an area instead of an exact position. Moreover, using this technique for location tracking could raise privacy concerns, in that, it requires access to the identifier of each phone that may disclose user identify as well. Therefore, when only having the corresponding permissions, cellular network can be used for contact tracing, and meanwhile the user identity must also need to be protected.

How to collect the encounter records. Even though there are distributed models for contact tracing which allow each user to individually control whether or not to disclose its own encounter records, we advocate a centralized model in which each individual user’s contact is collected by a central agency, and then stored at a central backend. This is bound to raise privacy concerns, and hence we need to introduce privacy-preserving mechanisms. To this end, we will generate pseudonyms for each user periodically and the linkage between a pseudonym and the real user is only resolved at the trust authority. The authority is only allowed to link pseudonyms to real users when the pseudonyms belong to (i) infected individuals that are confirmed by healthcare authorities or (ii) individuals who have close contact with infected ones. As such, the privacy of individuals who have no risk of infection will be preserved.

A similar approach has been proposed and adopted in ROBERT [3].

Moreover, privacy concerns might arise from using the cell tower information for locating users because the identifier of a user’s phone needs to be accessed. In order to mitigate these, we can also link such identifiers with pseudonyms. Similar approaches can also be applied in WiFi positioning. Therefore, the entry of data for upload involves self pseudonym, encounter pseudonym, timestamp, Bluetooth proximity, ultrasound proximity.

After processing each upload data entry, the output of this improved data collection procedure is data entries that involve pseudonyms of two encounters, the timestamp, the adjusted proximity, and the infection risk. In particular, the adjusted proximity is the weighted average from combining proximity measured from different sources (i.e., Bluetooth, ultrasound, WiFi, and cell tower), and the infection risk can be obtained by using environment detecting heuristics. For example, when there is no proximity measurement from ultrasound and the WiFi proximity indicates encounters are in different rooms, the infection risk can be adjusted to a low level. Besides the above privacy-preserving data collection methods, we can also apply tools from the field of differential privacy [8]. These utilize different kinds of data processing and noise injection methods, thereby making it difficult for any party to determine whether or not a particular individual is in the original data records and providing privacy protection to the users. Such a guarantee on the privacy would encourage more users to join the system.

3. DATA INTEGRATION AND SUSCEPTIBILITY GRAPH

Here, the goal will be to create a “susceptibility graph” that describes compactly the different ways in which disease is likely to spread. We begin by introducing this graph, and then also describe how to construct this graph by integrating the data from multiple sources. The graph would be time-variant.

3.1 Graph Structure

A basic version of the graph would contain the following components, and the designer is free to make reasonable modifications on it.

- **Nodes.** Each node represents an individual that could be potentially infected. Individuals that are isolated will be removed from the graph. Also, we can remove individuals who have recovered from the virus from the graph. However, since recovered individuals lose their antibodies for most viruses (including COVID-19), re-infections are possible after a period of time, so they would have to be re-introduced into the graph after some time. We use \mathcal{N} to denote the set of nodes (individuals).

The first step is to infer the type or level of contacts between two individuals. Have they stayed closer than 6 feet or stayed in the same room for a while? The second step is to deduce the link probability. Accurate characterizations of the link probabilities could come from exposure data studies to the virus. However, a reasonable model would be to use a concave function of time to estimate the link probability.

- **Testing results.** For an individual, if this individual has taken a virus test and got the result, then we know whether this individual has the disease or not (with a certain confidence).

3.3 Data Integration

In real-world scenarios, there are multiple data sources. For example, different contact tracking data sources as described earlier (Bluetooth or ultrasound contact tracing data, WiFi logs, GPS, etc.) could be integrated to greatly improve the quality of contact tracing. The integration could be done by using filtering techniques, in which we compute the probability of an edge conditioned on the (multi-source) information available to us. We would typically rely upon generative models of the data in order to compute these conditional probabilities. With multiple data sources, we need to deal with inconsistent data. For instance, Bluetooth gives relative location information whereas WiFi gives absolute location information, and the information of two sources may be inconsistent. We can deal with this issue by assuming that the data collections of the sources are random and independent and assign probability distributions to them. Probabilistic description allows “soft recovery” of data after we use filtering algorithms. Bayesian updates can be used to merge or pool information from various source. We can use Kalman filtering or some other filtering algorithm. Such an integration can yield us the following kind of improvements:

- Reduced inaccuracies and better estimates of the link probabilities. Consider for example the case when people could have social contact by virtue of being located in a crowded facility such as students in the same classroom or people in the same flight. However, data sources such as building information, WIFI access might be noisy. In this case, one could combine GPS data (collected from probably smartphone usage) in order to yield an accurate estimate of social contacts.

4. TESTING UNDER RESOURCE CONSTRAINTS

The goal here is to leverage the information contained in the susceptibility graph in order to sequentially choose individuals for testing so as to minimize the spread of

the pandemic. *Note that this objective is quite different from focusing on testing individuals with the highest probability of infection, which is what current systems try to do. Rather our focus must be on testing individuals that have the highest expected impact on viral spread.* Consider the following example as an illustration.

Example: Assume that two individuals i and j are infected with probabilities 0.1 and 0.3, respectively. However, assume that the expected number of individuals that i encounters is 50 times larger than the expected number of individuals that j comes in contact with. In this case, it makes more sense to prioritize testing individual i over individual j . This is another reason why we should test healthcare workers more often, because of their frequent contact with a large number of individuals. Based on this key insight, our goal will be to:

- Develop learning based approaches that result in smart testing capabilities which balance the exploration and exploitation subject to testing constraints. Isolate individuals who have been tested positive and quarantine their contacts.
- Our model will also incorporate practical issues such as inaccurate estimates, testing errors, pool testing, etc.
- Develop efficient rules of thumb that can be easily implemented in practice. This could mean testing asymptomatic individuals who have not encountered a confirmed infected person, but have made a large number of contacts.

4.1 Suspicious Infection Inference

One significant task is to find the most likely infected individuals from the partial observations, i.e., the test results of some individuals. To do this, one way is to interpret the probability that a person is infected given the partial observations, such as (“noisy”) contact graph or test results of a few individuals from the graph etc. These algorithms could be based upon the susceptibility graph constructed by using the methods stated in Section 3.

4.1.1 Partial Observed Markov Decision Process (POMDP)

We formulate the problem of sequential testing for COVID-19 as a Markov Decision Process (MDP). Population is composed of N individuals, and the state evolves at discrete times $t \in [1, T]$. Let $X_i(t) \in \{0, 1\}$ denote the hidden state of individual i at t , where $X_i(t) = 0$ means that i is free of disease at t and $X_i(t) = 1$ indicates that i is infected. We use the vector $X(t) := (X_1(t), X_2(t), \dots, X_N(t)) \in \{0, 1\}^N$ to represent the state of the entire system. Let $\mathcal{X} := \{0, 1\}^N$ denote the state-space of the network. Note that the state

individual i is not infected by the virus at time t , and $X_i(t) = 0$ if individual i is infected by the virus but cannot spread the virus, i.e., is in incubation, $X_i(t) = +1$ if individual i is infected by the virus and can spread the virus, and finally $X_i(t) = -2$ if they have recovered from the virus or have been isolated already. We call $X_i(t) = 0$ “inactive” and $X_i(t) = 1$ active.

The rest of the discussion in this section makes the following simplifying assumptions:

- States $X_i(t)$ and decisions $U_i(t)$ do not change within a slot.
- Upon becoming an active spreader, an individual can spread the disease only after the current time-slot ends. This might seem to be restrictive, but is justifiable since our modeling procedure already introduces “noise” due to erroneous tracing and testing.
- The spreading probability, denoted as p_s , is a constant that is independent of other parameters such as the values of the states, the number of days one has been infected, etc.
- After becoming an inactive infected person, at each time slot the individual becomes active with a probability equal to $p_{0,1}$.
- For an active infected person, for every time slot, this individual has a constant probability to get removed. We use $p_{1,-2}$ to denote this probability.
- For an individual at state $X_i(t) = a$, it has a constant probability $p_{a \rightarrow b}$ to be tested to be state $U_i(t) = b$.

Let \mathcal{F}_t be the filtration generated by $(X_i(s), Y_i(s), i \in \mathcal{N}, s \leq t)$. With the above assumptions in place, we can write the “dynamics” or transition probabilities governing $X(t) = \{X_i(t)\}_{i=1}^N$ as follows,

$$\begin{aligned} \mathbb{P}\{X_i(t+1) = 2 \mid \mathcal{F}_t\} \\ = \mathbb{1}_{X_i(t)=-2} + \mathbb{1}_{X_i(t)=1} \cdot p_{1,-2}, \end{aligned} \quad (11)$$

$$\begin{aligned} \mathbb{P}\{X_i(t+1) = 1 \mid \mathcal{F}_t\} \\ = \mathbb{1}_{X_i(t)=1}(1 - p_{1,-2}) + \mathbb{1}_{X_i(t)=0} \cdot p_{0,1}, \end{aligned} \quad (12)$$

$$\begin{aligned} \mathbb{P}\{X_i(t+1) = 0 \mid \mathcal{F}_t\} \\ = \mathbb{1}_{X_i(t)=0} \cdot (1 - p_{0,1}) \\ + \mathbb{1}_{X_i(t)=-1} \left(1 - \prod_{j:(i,j) \in \mathcal{E}, X_j(t)=1} (1 - p_s(t))\right), \end{aligned} \quad (13)$$

$$\begin{aligned} \mathbb{P}\{X_i(t+1) = -1 \mid \mathcal{F}_t\} \\ = 1 - \mathbb{P}\{X_i(t+1) = 1, 0, \text{ or } -2 \mid \mathcal{F}_t\}, \end{aligned} \quad (14)$$

and the dynamics or transition probabilities of $U(t) =$

$\{U_i(t)\}_{i=1}^N$ as follows,

$$\begin{aligned} \mathbb{P}\{U_i(t) = 1 \mid \mathcal{F}_t\} \\ = \mathbb{1}_{U_i(t) \neq 0} \left(\sum_{s=-1,0,1} \mathbb{1}_{X_i(t)=s} \cdot p_{s \rightarrow 1} \right), \end{aligned} \quad (15)$$

$$\begin{aligned} \mathbb{P}\{U_i(t) = 1 \mid \mathcal{F}_t\} \\ = \mathbb{1}_{U_i(t) \neq 0} \left(\sum_{s=-1,0,1} \mathbb{1}_{X_i(t)=s} \cdot p_{s \rightarrow -1} \right), \end{aligned} \quad (16)$$

where whether $U_i(t) = 0$ or not is determined by the tracing or testing algorithms.

Clearly, $X(t) = \{X_i(t)\}_{i \in \mathcal{N}}$ is a Markov process, and if we are provided with the values of $U_i(0), U_i(1), U_i(2), \dots$, then our goal is to find the most likely values of $X(t)$. In order to do this, we might use Markov chain Monte Carlo (MCMC) algorithms such as Gibbs Sampling. Readers may refer to [27] for a review of MCMC algorithms.

4.1.3 Graph Embedding

Computing the infection probabilities of individuals directly will be computationally cumbersome, and we can utilize graph embedding [5] techniques in order to find suspicious infected individuals. These techniques map the nodes of a graph to points in \mathbb{R}^d , where d is a natural number. If the graph embedding algorithm is properly chosen, then if two points are close in the space \mathbb{R}^d then they are also close in the susceptibility graph, so that the probability that the virus spreads from one individual to the other is high. Note that in the graph each node may have up to $|\mathcal{N}|$ edges, but in the embedded graph, each node only has d coordinates. Since the number of edges might be much more than d , performing computations with the embedded coordinates is much more efficient than directly working with the original graph.

5. SIMULATIONS

In this section, we use a simulation to indicate the necessities of contact tracing and building a contact graph. Contact tracing is an essential technique for finding potential infectious people. A commonly employed naive contact tracing technique is to trace and test only those who have had contact with a confirmed positive person. We call this simple and intuitively appealing contact tracing policy as Policy 1. However, this method may not be optimal, especially under circumstances when a sizeable proportion of the population is infected. To see why this might be the case, consider the scenario when two people are waiting to get tested. The first person had a close contact with a confirmed infected person, while the second person did not have any such close contact with a confirmed infected person; but did closely contact 500+ untested people (for example this person works in a supermarket). Policy 1 will suggest to us to test the first person; however, when a significant proportion of the population (e.g., 1%) are positive, in

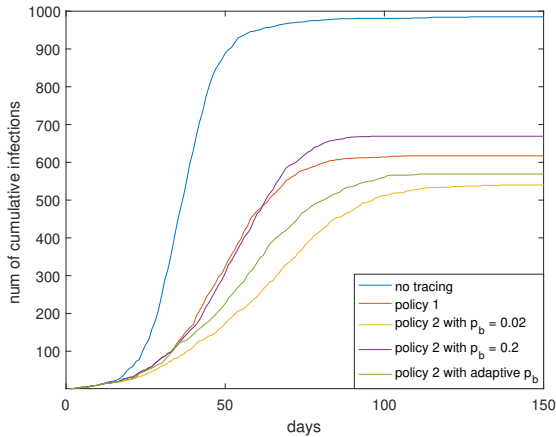


Fig. 2 – Simulation of the spreading of the virus under different tracing policies.

which is approximately the entire population. Even a simple contact tracing technique such as Policy 1, reduces the total infections to 617 (which is a 37% reduction). Thus, it is worth the effort to establish a system that can track the contacts of the people in order to better detect the infections.

- Compared to Policy 1 that only considers the contacts with confirmed infected people, Policy 2 also takes contacts with untested (but probable to be infected) people into account, and hence it has a better performance. This is clearly demonstrated in its superior performance in all of the three experiments, for example when $p_b = 0.02$, we get a gain of around 12%.
- When p_b is tuned properly (e.g., $p_b = 0.02$), Policy 2 performs better. However, the tuning effort is substantial, and might be deemed infeasible in practice. For instance, when $p_b = 0.2$, the performance of Policy 2 is worse than Policy 1. Thus, Policy 2 with adaptive values of p_b is a good option in practice.
- If we are to use only contact tracing and quarantine facilities, our performance is not very good. Even if tracing is possible for 2% of the population per day, the majority of the population will get infected after a few months. Hence, it is necessary to combine contact tracing and quarantines with other policies, e.g., avoiding contacts to reduce the number of contacts, and also wearing masks to reduce the virus spreading probability.

6. CONCLUSION

In this paper, we have provided a detailed blueprint on how to contain the spread of a pandemic by integrating the use of various wireless technologies with sequential learning based techniques. In particular, we show how different wireless technologies could be leveraged to improve contact tracing efforts and reduce the probabilities

of detection and false alarms. The idea is to use possibly disparate wireless data streams for data collection, then integrate this data to improve coverage, reduce probabilities of errors and false alarms and create a *susceptibility* graph that could be used for intelligent testing. Based on this susceptibility graph, we show how to develop efficient sequential learning based algorithms in order to minimize the spread of the virus infection. Another contribution is that we develop provably optimal algorithmic solutions that rely upon the theory of partially observable Markov decision processes. In particular, we show that current contact tracing plus testing strategies that are aimed at identifying (and testing) individuals with the highest probability of infection are inefficient. Instead, we find that it is better to test those individuals whose expected impact on virus spread is the highest. We formulate the testing problem as a Partially Observable Markov Decision Process whose goal is to minimize the expected spread of the virus subject to testing capacity constraints. We provide efficient algorithmic solutions to this problem and show via numerical results that our solution substantially reduces the spread of the virus.

ACKNOWLEDGEMENT

This work was supported in part by NSF grants CNS-1618520 and CNS-2028547.

REFERENCES

- [1] Covid-19 contact tracing: a briefing. <https://www.bmj.com/content/369/bmj.m1859>. (Accessed on 07/02/2020).
- [2] The global economy is expected to shrink by 3% this year. <https://www.economist.com/graphic-detail/2020/04/14/the-global-economy-is-expected-to-shrink-by-3-this-year>.
- [3] Robert – robust and privacy-preserving proximity tracing protocol. <https://github.com/ROBERT-proximity-tracing>. (Accessed on 09/17/2020).
- [4] Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.
- [5] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.
- [6] CDC. How to protect yourself & others. <https://www.cdc.gov/coronavirus/2019-ncov/prevention-getting-sick/prevention.html>.
- [7] Aaqib Bashir Dar, Auqib Hamid Lone, Saniya Zahoor, Afshan Amin Khan, and Roohie Naaz. Applicability of mobile contact tracing in fighting pan-

AUTHORS



Rahul Singh received his B. Tech. degree in Electrical Engineering from Indian Institute of Technology, Kanpur, India, in 2009, M.Sc. degree in Electrical Engineering from University of Notre Dame, South Bend, IN, in 2011, and his Ph.D. degree in Electrical and Computer Engineering from the Department of Electrical and Computer Engineering Texas A&M University, College Station, TX, in 2015. Currently he is an Assistant Professor at the Department of Electrical Communication Engineering, the Indian Institute of Science. His research interests include stochastic control, machine learning, applied probability, networks and large-scale complex cyber physical systems. He has earlier worked as a Postdoctoral Researcher at the Laboratory for Information Decision Systems (LIDS), Massachusetts Institute of Technology, at Intel, Santa Clara as a Machine Learning Engineer, and as a Data Scientist at Encored Technologies.

His research interests include stochastic control, machine learning, applied probability, networks and large-scale complex cyber physical systems. He has earlier worked as a Postdoctoral Researcher at the Laboratory for Information Decision Systems (LIDS), Massachusetts Institute of Technology, at Intel, Santa Clara as a Machine Learning Engineer, and as a Data Scientist at Encored Technologies.



Wenbo Ren received his B.S. degree in Electronic Information Engineering from University of Science and Technology of China in 2016. Since then, he has been a Ph.D. student in Computer Science and Engineering at The Ohio State University. His research interests include

machine learning theory and algorithms, with a focus on ranking, active learning, sequential decision making, recommender systems, etc. He has made publications or served as reviewers for top-tier machine learning conferences including ICML, NeurIPS, and AISTATS.



Fang Liu received his B.S. degree in Information Engineering from Shanghai Jiao Tong University, China, and his Ph.D. degree in Electrical and Computer Engineering from The Ohio State University. In 2020, he joined Facebook Inc. as a research scientist. His

research interests include machine learning theory and systems, with a focus on sequential decision making under uncertainty. Dr. Liu has served as a program committee member or reviewer for many top-ranked

machine learning conferences, such as ICML, NeurIPS, ICLR, and AAAI.



Dong Xuan received his B.S. and M.S. degrees in Electronic Engineering from Shanghai Jiao Tong University, China, and his Ph.D. degree in Computer Engineering from Texas A&M University. In 2001, he joined the Department of Computer Science and Engineering

at The Ohio State University, where he is currently a full professor. He was on the faculty of Electronic Engineering at Shanghai Jiao Tong University from 1993 to 1998 and a visiting scholar in Computer Science at City University of Hong Kong from 1997 to 1998. He was a research assistant/associate in the Real-Time Systems group in the Department of Computer Science at Texas A&M University from 1998 to 2001. His research interests include computer networking and mobile systems. Dr. Xuan has served as an editor for IEEE TPDS and ACM ToSN, and has been a TPC member for a number of IEEE and ACM flagship conferences such as IEEE INFOCOM, ICDCS, ICNP, and ACM MobiHoc. He is a recipient of the National Science Foundation CAREER Award and the College of Engineering Lumley Research Award at The Ohio State University.



Zhiqiang Lin received his Ph.D. degree in Computer Science from Purdue University in 2011. He was a faculty member in the Computer Science Department at University of Texas at Dallas between 2011 and 2017. Since 2018, he has

been a faculty member in the Department of Computer Science and Engineering at Ohio State University. His research interests are systems and software security, with an emphasis on binary analysis and vulnerability discovery. Dr. Lin is currently an associate editor of ACM TOPS, IEEE TDSC, and IEEE TMC. He has also served as a TPC member for numerous systems security conferences including IEEE S&P, ACM CCS, USENIX Security, and NDSS. Dr. Lin received the AFOSR Young Investigator award (2014) and the NSF CAREER award (2015). He is also a recipient of VMware Faculty Research Award (2012) and the Outstanding Junior Faculty Research Award at UT Dallas (2013).



Ness B. Shroff received his Ph.D. degree in electrical engineering from Columbia University in 1994. He joined Purdue University immediately thereafter as an Assistant Professor with the School of Electrical and Computer Engineering. At Purdue, he became a Full Professor of ECE and the director of a university-wide center on wireless systems and applications in 2004.

In 2007, he joined The Ohio State University, where he holds the Ohio Eminent Scholar Endowed Chair in networking and communications, in the departments of ECE and CSE. He holds or has held visiting (chaired) professor positions at Tsinghua University, Beijing, China, Shanghai Jiaotong University, Shanghai, China, and IIT Bombay, Mumbai, India. He has received numerous best paper awards for his research and was listed in Thomson Reuters' on The World's Most Influential Scientific Minds, and has been noted as a Highly Cited Researcher by Thomson Reuters in 2014 and 2015. He has served on numerous editorial boards and chaired various major conferences and workshops. He currently serves as the steering committee chair for ACM Mobihoc, and Editor in Chief of the IEEE/ACM Transactions on Networking. He received the IEEE INFOCOM Achievement Award for seminal contributions to scheduling and resource allocation in wireless networks.