SYNTHETIC DATA TO SUPPORT US-UK PRIZE CHALLENGE FOR DEVELOPING PRIVACY ENHANCING METHODS: PREDICTING INDIVIDUAL INFECTION RISK DURING A PANDEMIC

GALEN HARRISON, JIANGZHUO CHEN, HENNING MORTVEIT, STEFAN HOOPS, PRZEMYSLAW POREBSKI, DAWEN XIE, MANDY WILSON, PARANTAPA BHATTACHARYA, ANIL VULLIKANTI, LI XIONG, AND MADHAV MARATHE

Department of Mathematics and Computer Science, Emory University. Email: lxiong@emory.edu

Biocomplexity Institute and Initiative, Department of Engineering, Systems and Environment and Department of Computer Science, University of Virginia. Email: gh7vp,chenj,hsm2v,shoops,alw4ey,pjp2p,dawenx,pb5gj,asv9v,marathe@virginia.edu. Contact author: Madhav Marathe (mvm7hz@virginia.edu. U. Virginia Biocomplexity Institute, Technical Report: TR BI-2022-1701)

SYNTHETIC DATA TO SUPPORT US-UK PRIZE CHALLENGE FOR DEVELOPING PRIVACY ENHANCING METHODS: PREDICTING INDIVIDUAL INFECTION RISK DURING A PANDEMIC

GALEN HARRISON, JIANGZHUO CHEN, HENNING MORTVEIT, STEFAN HOOPS, PRZEMYSLAW POREBSKI, DAWEN XIE, MANDY WILSON, PARANTAPA BHATTACHARYA, ANIL VULLIKANTI, LI XIONG, AND MADHAV MARATHE

Abstract

Abstract. This document describes synthetically produced epidemic data to support the 2022 US-UK Prize Challenge focused on advancing privacy-enhancing technologies (PETs). Announced by the White House in December 2021, this challenge is part of a series of International Grand Challenges on Democracy-Affirming Technologies; see `https://www.whitehouse.gov/ostp/news-updates/2021/12/08/us-and-uk-to-partner-on-a-prize-challenges-to-advance-privacy-enhancing-technologies/` for more details.

In this challenge, participants will attempt to predict the likelihood of an individual getting infected by a disease in a privacy-preserving manner. This task, while of high public health relevance, has been hampered by data availability. The challenge participants will be able to develop and demonstrate their proposed solutions using a synthetic dataset that we have specifically created for this challenge. The synthetic data is developed by integrating realistic data to produce a synthetic social contact network, along with a synthetic outbreak that is similar to the COVID-19 pandemic. This dataset, which is very detailed and realistic, serves as ground truth. In addition to the description of the synthetic data, this document also describes three centralized baselines that can be used to evaluate the performance of the proposed methods.

1. Introduction and motivation

The COVID-19 pandemic has brought to the forefront the need for developing privacy-enhancing solutions to address important questions related to pandemic response. This document describes a synthetic, yet realistic, dataset designed in support of the recently announced US-UK Prize Challenge [1] to encourage development of privacy-enhancing technologies. This paper proposes the Individual-Risk-Prediction (henceforth referred to as IRP) learning task using this synthetic data. Informally speaking (see Section 2 for formal definitions), given a (noisy) observational dataset that captures the individual disease states from the start of a pandemic until a given time T, the IRP task aims to predict the risk of individuals being infected within a relatively short period after T. The IRP task, while of concrete practical use, is a challenging machine learning task, and has been severely

Department of Mathematics and Computer Science, Emory University. Email: `lxiong@emory.edu`

Biocomplexity Institute and Initiative, Department of Engineering, Systems and Environment and Department of Computer Science, University of Virginia. Email: `gh7vp,chenj,hsm2v,shoops,alw4ey,pjp2p,dawenx,pb5gj,asv9v,marathe@virginia.edu`. Contact author: Madhav Marathe (`mvm7hz@virginia.edu`. U. Virginia Biocomplexity Institute, Technical Report: TR BI-2022-1701)

constrained by data availability issues. Further, the IRP task requires data that is highly sensitive. Therefore, the use of privacy-enhancing techniques is necessary to make solutions to this problem practical in the real world.

The challenge is part of a series of International Grand Challenges on Democracy-Affirming Technologies. Two synthetic datasets are produced for this effort, one for the Commonwealth of Virginia in the United States (US), and the other for the United Kingdom (UK). These datasets represent a synthetic, yet realistic, epidemic outbreak in a region of the US and all of UK. We finally demonstrate the feasibility of the task with three example machine learning methods. These methods are not intended to be the best in class, but, rather, to demonstrate the feasibility of data-driven, as well as causal machine learning methods for this problem.

1.1. **Need for privacy-enhancing techniques (PETs) for pandemic response.** During the COVID-19 pandemic, a large number of public data infrastructures had to be developed and deployed very quickly in order to respond to immediate public health concerns [39]. Because of the many pressing needs, many of these systems were by necessity ad hoc, with the primary goal of providing the necessary data to analysts. In many ways, significant progress was made, and a lot of traditional constraints were overcome. However, in spite of best efforts, the amount of data that was shared was relatively small. There were many reasons for this, including commercial interests, a lack of well-accepted methods to share data while preserving privacy, and a shortage of large-scale data infrastructures to support real-time public health decision making. As the emergency subsides and many of these systems wind down, we have an opportunity to revisit these infrastructure decisions in a more systematic manner.

More specifically, a central problem epidemiologists need to confront is that the data necessary for many types of projections and models is distributed across multiple sources and is often not openly available. State health departments implement similar, but not identical, data collection programs for artifacts such as case counts, vaccines administered, and hospital capacity. Hospitals may maintain medical records that could provide estimates of medically vulnerable people. Global Positioning System (GPS) tracking may collect records on individual people, including the locations they visit and who they come in contact with. Social media companies, such as Google or Facebook, may have information about mobility, attitudes, and (mis)information viewership.

Understanding large-scale dynamic processes requires integration of all these data sources. Accurate modeling of disease dynamics requires understanding not only how many people are infected, but also who is infected, and where they became infected. Integrating the sorts of data necessary to make this type of analysis has in the past been done in an ad hoc manner, stitching together disparate sources. On top of the difficulties inherent in aggregating data across multiple data sources with overlapping and inconsistent schema, integration of (and access to) these data sources is stymied by valid privacy concerns.

Creating such a uniform data infrastructure poses serious privacy challenges. To get a sense of the challenges, it is useful to consider the contact tracing apps which were deployed in many localities at the beginning of the pandemic (see, e.g., [3]). While privacy was a concern at the outset, and a great deal of effort was expended in order to reassure users about privacy guarantees, there were multiple instances of these apps being used for purposes

other than pandemic control or otherwise exercising inadequate data controls [26, 30]. These sorts of issues could be addressed through socio-technical design processes, but, at present, the sorts of systems necessary to ensure reliable and useful epidemiological surveillance as well as proper privacy preservation require much more study.

INDIVIDUAL-RISK-PREDICTION (IRP) provides a natural example of a (distributed) machine learning task that can benefit from PETs. It requires highly sensitive data, but, if solved, would be of incredible public health benefit.

1.2. **Current data availability.** Much of the data listed above is not available, or available only on a limited basis. Medical records, for example, can only be accessed after navigating through several levels of bureaucracy. Location data is generally not available in a format easily reconciled to other epidemiologically relevant characteristics. Currently, integrating these sources (even when they are available) has been done in an ad hoc manner. Because of privacy constraints, data is generally made available only in an aggregated form.

However, aggregation also introduces certain complications. To cite a real example, vaccine data provided by the Centers for Disease Control and Prevention (CDC) exhibits large spikes in the vaccinated populations for several Virginia counties, and, in general, underestimates the vaccinated population when compared to the numbers tabulated by the Virginia Department of Health (VDH). Similar issues have led to the construction and utilization of alternative vaccination datasets [45]. Data quality issues with clinical data have spurred skepticism about the application of Artificial Intelligence (AI)/Machine Learning (ML) methods to the COVID-19 pandemic, both among researchers and in the popular press [38, 53]. Many of these issues are associated with the difficulties inherent in cleaning and aggregating data from a variety of (possibly inconsistent) sources.

1.3. **A synthetic pandemic dataset to develop and evaluate PETs.** We have provided synthetic pandemic datasets and an associated competition task for evaluating (*i*) the efficacy and (*ii*) the privacy of federated learning methods on dynamic social processes for broader scientific use. A *synthetic pandemic* (also known as *digital twins* and described in Section 3) creates a realistic disease outbreak over a realistic physical proximity network (aka social contact network) simulate populations whose statistical and dynamical properties are similar to those of real populations. The synthetic pandemic will allow competitors to explore privacy-preserving federated learning approaches in a manner where no real person's data is exposed. Through the competition, one can develop and evaluate methods for producing policy-relevant modeling on real data. For subsequent public health emergencies, we will have validated privacy architectures and learning techniques that can be deployed quickly.

An open data set for broader scientific use. The data was originally generated as a part of the US-UK Prize Challenge on Privacy-Enhancing Technologies [1]. It was part of the set of prize challenges for Democracy-Affirming technologies announced at the Summit for Democracy (`https://www.state.gov/summit-for-democracy/`. The challenges have been led by the U.K.'s Centre for Data Ethics and Innovation (CDEI) and Innovate UK, the U.S. National Institute of Standards and Technology (NIST), and the U.S. National Science Foundation (NSF), in cooperation with the White House Office of Science and Technology Policy. The US competition was coordinated by Driven data. The UK competition was conducted by Innovate UK and the Centre for Data Ethics and Innovation. More information on the

competition and the winners can be found at `https://www.drivendata.org/competitions/group/nist-federated-learning/`.

We have now made the data open to the broader scientific community. The data can be used to design and test novel algorithms and methods in varied settings, including network science, data privacy, contagion science, social networking and high performance computing — we look forward to its varied use and further conversations with the scientific community. The data can be accessed from `https://doi.org/10.18130/V3/ZOG1FF`. More details on our work on synthetic information can also be found in the tutorials we have given as well as our papers; see [4, 14, 32, 41, 40]. Over the last few years, the term *digital twin* has been used to describe such integrated set of models and data. In this sense, the data and the associated methods can be thought of as a digital twin of an epidemic outbreak. We have chosen to retain the term *synthetic data* as opposed to *digital twin* for two reasons: (*i*) the data is produced by synthesizing multiple data sources; (*ii*) the synthetic data is not **identical** to the real world (we argue that this is not even a well defined concept for social systems like the ones considered here).

2. Task Description and Learning Setup

In this section, we describe the IRP problem. This problem presents an important use case for synthetic data. While solutions could leverage existing datasets, collecting that data would pose grave privacy risks. Synthetic data allows us to make progress on this problem without putting anyone's information at risk.

2.1. **Preliminaries.** We introduce the notions needed here for the problem statement and discussion. Let $G(V, E)$ denote an edge-weighted undirected graph, with each node representing an individual $u \in V$ (also referred to as a node) and each edge $(u, v) \in E$ representing a contact between individuals u and v for a total time w_{uv} (the weight of edge (u, v)). We denote the neighbors of node u by $N(u)$. We note that G is inferred through a complex model of activities and co-location, as discussed in Appendix 4.

We consider a disease spreading over G, starting from some initial set of infected individuals. For simplicity, we only describe the SIR epidemic model here; the actual model we use in the data generation is more involved, and is described in Appendix 5. Let $s_v(t) \in \{S, I, R\}$ denote the disease state of node v at each time $t = 1, \ldots, T_{final}$, where S, I and R refer to **S**usceptible, **I**nfected, and **R**ecovered states, respectively, and T_{final} denotes the length of time for the synthetic outbreak. At each time t, an infected node v (i.e., with $s_v(t) = I$) may independently infect each susceptible neighbor u with probability p_{vu}, which depends on the weight w_{vu} of the contact. Due to the independence assumption, the probability that a node u becomes infected at the end of time t equals $1 - \prod_{v \in N(u), s_v(t)=I}(1 - p_{vu})$. At time $t = 0$, a small set of nodes is infected (i.e., in state I), and the remaining nodes are in state S. We use $\mathbf{X}(t)$ to denote the observed history of the disease evolution until time t; this specifies the disease states of all nodes at every time step $t' \leq t$.

Example. Figure 1 illustrates the SIR model on a small network. We will use this as a recurring example to illustrate all definitions. The sequence of configurations is a possible time evolution of the SIR process, starting with the initial configuration at time t_0, where only node v_0 is in state I. The probability of reaching the configuration at time t_1 is $(1/2)^3 = 1/8$. Figure 19 (in Appendix A) shows all configurations which the system could transition

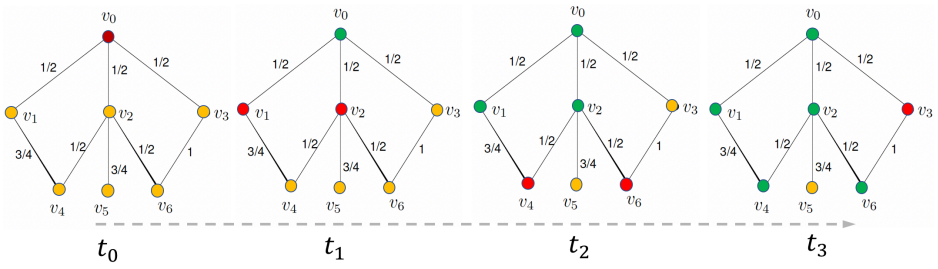


FIGURE 1. Example illustrating the SIR model on a network with seven nodes. The node colors yellow, red and green represent the states S, I and R, respectively. The edge weights represent transmission probabilities. The configuration at time $t = t_0$ with node v_0 in state I is the initial configuration. The figure represents a possible evolution of the epidemic process.

Time	t_0	t_0	t_0	t_0	t_0	t_0	t_0	t_1	t_1	t_1	t_1	t_1	t_1	t_1
Vertex	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_0	v_1	v_2	v_3	v_4	v_5	v_6
State	I	S	S	S	S	S	S	R	I	I	S	S	S	S

TABLE 1. An example of $\mathbf{X}(t_1)$ for the system shown in Figure 1

to in one step from the initial configuration. For $T = t_1$, $\mathbf{X}(t_1)$ will specify the states of all the nodes at times t_0 and t_1. Following the disease outcomes structure specified in Table 14 (in Appendix 4), $\mathbf{X}(t_1)$ is shown in Table 1.

The SIR model described here is used to illustrate the (slightly more complicated) model used to actually generate the data. The actual model we used differs in two respects: the model used to generate the data has an exposure phase and the includes unobserved asymptomatic transmission. Nodes that become infected enter a distinct state prior to being able to spread the disease in the infected state. Nodes may also be infected asymptomatically. Asymptomatic infections are infections that can infect others, but which do not appear as infections in the data we release. This is similar to the kind of information that would actually be available in the real world; generally, asymptomatic people are much less likely to be detected. The model we use to synthesize disease states is described in detail in Section 5

2.2. **The INDIVIDUAL-RISK-PREDICTION problem.** This problem attempts to predict whether an individual will become infected in the near future, given past and current infection statuses in a network.

IRP problem: Given $\mathbf{X}(T)$, the history of the outbreak until time T, the problem is to determine

$$h_v(\mathbf{X}(T)) = \Pr[s_v(t) = I \text{ for some } t \in (T, T + \Delta] \,|\, \mathbf{X}(T)] \,,$$

for each node v. We use $h(\mathbf{X}(T))$ to denote the n-dimensional vector of all probabilities for all nodes.

Learning setup. The learner gets a dataset instance of $(\mathbf{X}(T), Y)$, where Y is an n-dimensional vector with $Y_v = 1$ if node v gets infected in the time $(T, T + \Delta]$, and $Y_v = 0$ otherwise. The learner will develop centralized and federated learning methods using this instance, which will be evaluated on a second dataset instance (not given to the learner).

Example (continued). Using the example shown in Figure 1, we can work out the probability of infection at t_3 precisely given knowledge of the node states at t_2, as well as the transmission probabilities. Specifically, v_3 will become infected with probability 1. This problem gets harder if we don't know the transmission probabilities, and as Δ becomes larger.

If we wanted to predict the likelihood that v_3 becomes infected at either t_2 or t_3, then the question is dependent upon the graph structure. The only path from an infected node to v_3 at t_1 is v_2, v_6, v_3. In this case, the likelihood of v_6 becoming infected is $\frac{1}{2}$ and of v_3 becoming infected from v_6 is 1. So at t_1, the precise solution to this problem would be $\frac{1}{2}$.

In the case of our example in Figure 1, for $T = t_1$ and $\Delta = 2$, we would have $y_3 = y_4 = y_6 = 1$ while $y_i = 0$ for $i \neq 3, 4, 6$. In this example, we can work out the true probabilities of infection precisely. $h_{true}(v_4) = \frac{7}{8}$, $h_{true}(v_5) = \frac{3}{4}$, and $h_{true}(v_6) = h_{true}(v_3) = \frac{1}{2}$.

In this example, these solutions can be worked out precisely because we (i) observe the disease state precisely, (ii) know the transmission probabilities precisely, (iii) have a sufficiently small graph, and (iv) are seeking to predict over a relatively short timeline.

Note. In this example, (assumptions about state visibility). Though the disease model has an asymptomatic state, this is not captured in the observed history (which is generally true in practice). Nodes which are in the asymptomatic state in the disease evolution will be labeled as being in the S state.

Epidemiological relevance. IRP is a very realistic abstraction of a public health problem during a pandemic. We note this form of the problem has also been suggested in recent work, e.g., [52, 9, 37], as being important for public health planning. If good surveillance is in place, the history $\mathbf{X}(t)$ until time t could be observed (at least partially). Most public health analyses rely on determining which individuals are at the highest risk, and how they would be impacted by changes in interventions. The function $h_v(\mathbf{X}(t))$ gives an estimate of this risk, and therefore a good solution can be useful in public health policy planning. If these estimates could be provided to individuals, they would also be useful for their own (decentralized) decision making, e.g., helping inform whether individuals should cut back on planned activities. From a practical perspective, we only have limited datasets, since only one epidemic would have been observed. In order to represent this constraint, we only provide one training instance, which represents the history of the outbreak up to that point.

Motivation from a Machine Learning perspective. IRP is closely related to the problem of learning an influence function, studied by Narasimhan et al. [22]. In this setting, the objective is to learn a function $F(X) = (F_1(X), \ldots, F_n(X))$, where X is a set of initial infections, and $F_v(X)$ is the probability that node v gets infected by the end of the disease process. They show that this problem is, in fact, efficiently PAC learnable. Their setting differs slightly, in that they assume that one has access to training examples of the form $(X^j, Y^j)_{j=1}^m$, where X^j is a set of initial seed nodes, and Y^j is the set of nodes infected at the end of the cascade process. These m examples are for cascades over the same underlying graph. In our setting, there is only one observation of a cascade process, but it happens large graph.

While Narasimhan et al. find learnability bounds, other work has sought to find workable implementations. Wilinski and Lokhov present a message passing algorithm with time

complexity $O(|E|T_{final}M)$, where M is the number of observed cascades [51]. The method they present works in the partial observation setting, but only for IC cascades, and finds, retrospectively, the likelihood of infection at each time step. Murphy et al. investigate neural network architectures for this problem, however only evaluate one time step forward, and use multiple cascade observations [33].

Other applied work on information diffusion has considered problems related to this one. In this setting, the goal, rather than predicting infection, is to predict the spread of some form of information, such as a retweet or a hashtag. Qiu et al. present DeepInf, a model architecture for predicting information cascades [36]. There are multiple other models in this vein. These models generally assume multiple cascade observations and vary in the precise aim [25, 17, 42, 27]. For some of this work, the goal is to predict the likely next "infection", and for others the goal is simply to predict the total infection size. They also vary somewhat in their assumptions about what the learner has access to. Since they generally are concerned with social media data, they sometimes do not assume that the underlying graph structure or edge weights are known.

While these graph neural network (GNN) models solve problems close to IRP, they differ in two regards. The first is that they assume multiple observations of the cascade process, whereas IRP has only one. The second is that they are concerned primarily with end-state behavior, that is, if a person was infected at all, and not the likelihood of becoming infected within a particular timeframe. To address these issues, we adapt our setting to use $\mathbf{X}(T)$ to generate examples. To this end, Qiu et al.'s DeepInf batching approach provides the most natural point of extension. However, we are obligated to significantly alter the procedure they describe. This is discussed further in Section 7.

We note that the IRP problem is in general very hard (e.g., [52, 9, 37]), and there are fundamental limits to the extent precise predictions can be made. In the synthetic data approach, we know the underlying mechanism precisely. In particular, the disease model is inherently stochastic. This leads to an upper bound on the ability of any learner to predict results, and motivates our choice of evaluation metric.

Relevance in the development of privacy-enhancing technologies (PETs). As mentioned earlier, individual-level attributes, such as infection status, demographics, and contacts, are sensitive attributes. IRP involves individual-level predictions, which are strongly dependent on the attributes of the individual and their neighbors. The IRP is a more general type of contact tracing – rather than looking only at immediate contacts of infected persons, IRP examines the contacts of the contacts, and the contacts of those contacts, and so forth. Therefore, we expect potential privacy risks with respect to these attributes, and developing methods to mitigate these risks will require innovations in PETs.

Evaluation. While the outcome of interest for IRP are binary, this problem *should not* be considered as simply a binary prediction problem. The setting in which we are working is inherently stochastic. Evaluating solutions as simple binary classifiers (e.g., in terms of precision, recall, F1-score, etc.) may cause solutions to appear better than they are due to random chance. In the example shown in Figure 1, if $T = t_1$, and $\Delta = 2$, the true probability of infection for v_3 is $\frac{1}{2}$. A model that guessed randomly, and evaluated only on whether it was correct or not, could appear to have relatively high accuracy if only evaluated against the single instantiation.

From a theoretical perspective, it would be better to try and predict probabilities explicitly. Unfortunately, in the real world, the probability of an individual becoming infected is never actually observed. Because of this, a ground truth based on probability of infection is not realistic.

Instead, the solution performance should be evaluated through AUPRC, or area under the precision-recall curve. If we consider a set of probability predictions for n instances, $\hat{p_1}, \hat{p_2}, ...\hat{p_n}$, sorted so that $\hat{p_i} \leq \hat{p}_{i+1}$, then we can define P_i as the precision if we were to predict 1 for all instances with predicted probability $\geq \hat{p_i}$. We can similarly define R_i as the recall if we were to predict 1 for all instances j where $\hat{p_j} \geq \hat{p_i}$. $R_0 = 0$ because if we set the cutoff probability to 0, then recall is 0. The AUPRC is then

$$AUPRC = \sum_{i=1}^{n} (R_i - R_{i-1}) P_i \tag{1}$$

In other words, it is a rectangular approximation of the area under the precision-recall curve. This measure captures a balance between the binary ground truth, as well as the need to avoid over-confident predictions. The AUPRC is preferable to other measures as it gives a more accurate measure of performance when better than random performance is not guaranteed [28].

3. Synthetic Data

The *synthetic pandemic* data described below consists of the following major components: (*i*) a synthetic population of a region; (*ii*) a synthetic social contact network: a labeled, spatially explicit network wherein nodes are individuals and edges between them signify proximity relationships that promote disease transmission; and (*iii*) a synthetic outbreak: a realistic epidemic outbreak over the synthetic social contact network based on realistic models of within-host and between-host models of disease transmission. The UK synthetic outbreak is modeled around seasonal influenza, while the Virginia synthetic outbreak is modeled around the early COVID-19 outbreak. All datasets (synthetic populations and synthetic outbreaks) are accessible here: `https://doi.org/10.18130/V3/ZOG1FF`.

3.1. Synthetic population and synthetic social contact network.

Our group has been developing methods for creating synthetic population datasets for over 25 years [41]. In our ongoing efforts to support the federal government in their response to the COVID-19 pandemic, we created a synthetic social contact networks of the state of Virginia, as well as other parts of the United States (US). We have also completed a synthetic network for the United Kingdom (UK), although this is somewhat less accurate than its US counterpart. In the network, nodes represent individuals, and edges capture physical proximity. Using these networks, we created epidemic simulations based on real-world outbreak information on vaccinations, social interventions, and other relevant data. The output of one simulation is a set of synthetic individuals who are infected over the course of the disease transmission. The synthetic data (also referred to as a *digital twin* in recent literature) provides a realistic account of how the disease spreads through the population in time and space.

A synthetic agent (e.g., person) is assigned states and interactions that make it statistically consistent with members of the (real) population without necessarily matching the

characteristics of any specific (real) person. A synthetic population represents a set of synthetic agents (e.g., people) that share common geographic, social or biological characteristics (e.g., people in a rural or urban region, individuals from a given tribe).

These populations and networks are formed by collecting a large and diverse set of publicly and/or commercially available datasets. These datasets include census, land use, mobility, activity, behavioral and transportation surveys and building maps. The datasets have been integrated in a first principles manner to construct these synthetic populations. They have been used for highly accurate, national level, agent-based modeling tasks [7].

The synthetic data is formed by taking the empirical distribution of particular attributes within an area (for example, the number of people per household, age, or income), then iteratively forming a population that matches those distributions while also preserving observed associations between those attributes. We apply a similar process to mobility data. This process means that there is no direct correspondence between any real person and any single synthetic agent within our synthetic population.

Our methodology ensures that privacy is maintained. We apply our synthesizing process to data that is already public, and, in the case of the census, has already had privacy-enhancing methods applied to it [16]. In the case of the 2020 census, the data that was released to the public had differential privacy applied to it [49]. Because differentially private data is known to be immune to post-processing, this means that we have a mathematical guarantee that this aspect of our synthetic data is private [13]. Furthermore, when we do use sources that have not been subjected to differential privacy processing, we rely on data sources that are already public. Any data an attacker might attempt to derive from the synthetic data could be much more easily obtained by looking at the sources we used to construct the data in the first place. Therefore, we are quite confident that our synthetic pandemic itself preserves privacy.

For the detailed description of the methodology, data description and format of our synthetic population, see Section 4. In brief, for the purpose of the challenge, we provided synthetic population datasets for Virginia and UK that consist of synthetic individuals, each of whom is assigned an age, a gender, a household and its home location, visited locations, and the activities performed at these locations. We use a synthetic person-location graph (also called an activity-location assignment graph) and examine which individuals were in the same place at the same time to develop a person-person contact network.

3.2. **Synthetic outbreaks.** The synthetic population can be used to simulate the spread of disease. We have highly granular disease models which can be applied to these populations. A disease model is a probabilistic model of disease progression. It specifies the likelihood of transitioning between disease states based on each synthetic agent's attributes. We specify the likelihood of infection given exposure based on a number of factors including (imputed) mask-wearing and vaccination status. We can further model whether a specific infection is asymptomatic, or the likelihood that the infection will result in a severe, reported case. Using this disease model, we generate synthetic outbreak data that reflects how an individual's disease state evolves over time. Additionally, for the purpose of method development, we generate a testing dataset that contains information about whether a given individual was infected during a forecast period. For the data description and a more detailed description of the generation of the synthetic outbreak, see Section 4 and Section 5

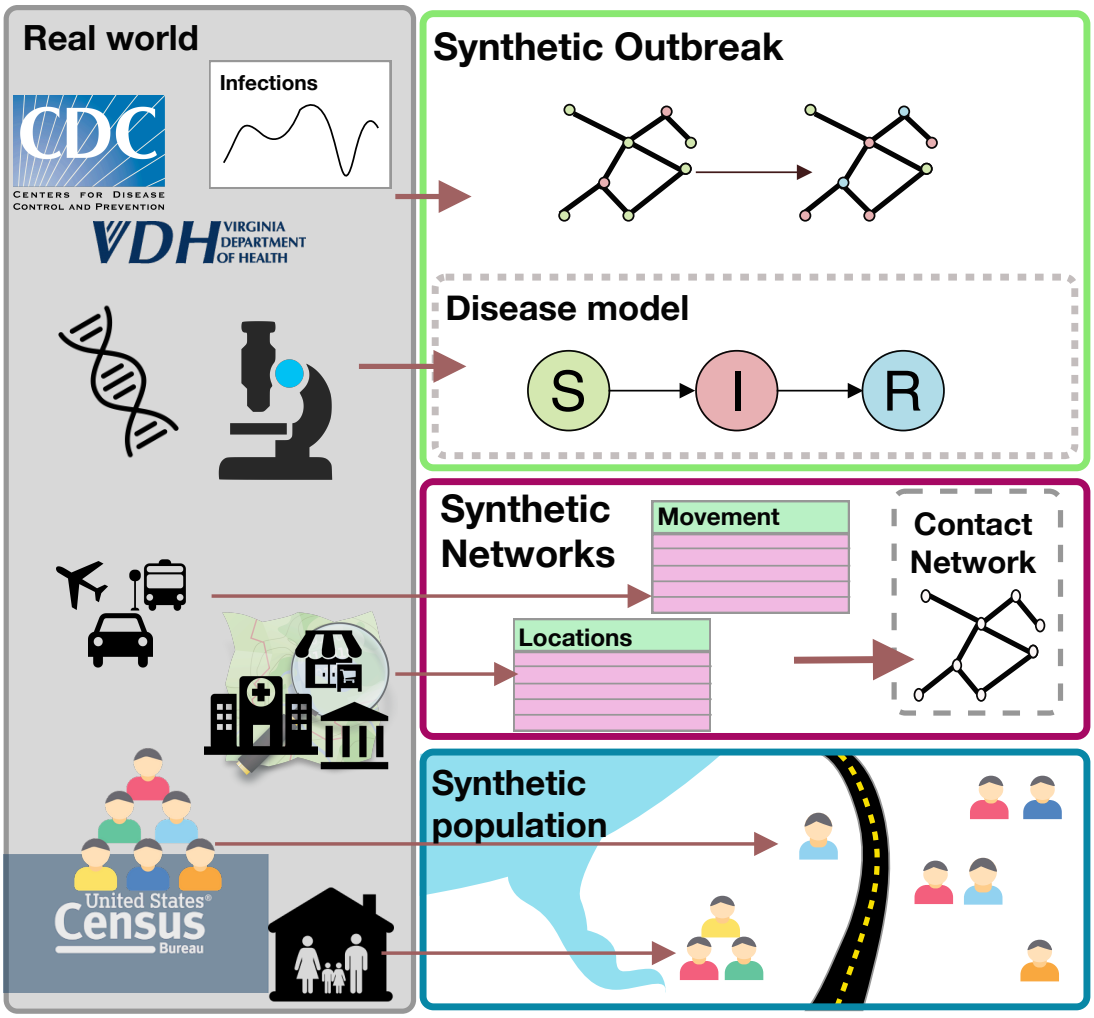


FIGURE 2. We produce our synthetic data at three different levels: (*i*) we use population data from sources such as the US Census to form synthetic populations, (*ii*) we build contact networks on top of aggregated mobility data to produce synthetic networks, and (*iii*) we use expert-informed models of diseases to model disease dynamics and create synthetic outbreaks.

3.3. **Advantages.** In addition to the privacy and availability benefits, synthetic populations permit a larger degree of control over conditions. Since the populations are synthesized, one can create many variations of the population and network, and, in so doing, design a controlled experiment where the "ground truth" is completely understood. Indeed, synthetic population data is well-suited to this form of challenge. In 2015, similar techniques were used as part of a competition around forecasting Ebola; an agent-based disease model for Ebola running on a synthetic population was used to generate partial disease outcomes [2].

The goal of the challenge is to allow participants to develop and demonstrate privacy-preserving methods for risk prediction as it pertains to an ongoing pandemic. These methods can be developed to provide insights to public health officials, organizational decision makers,

or individual citizens. By modeling, in a realistic manner, the dynamics of complex socio-biological systems, synthetic populations of this type are a key tool for understanding and testing new technologies.

4. Methodology to Generate Synthetic Populations and Networks

A *synthetic population* of a region may be regarded as a digital twin of the real population of that region. In this section, we provide a compact summary of the models and methodologies behind constructing synthetic populations and contact networks. Section 4.1 focuses on Virginia, USA (see [32] for additional details). Our work builds on earlier techniques using a first principles approach for constructing synthetic populations [14, 15, 4]. We remark that the model and methodology for constructing the synthetic population for UK is slightly different from that for Virginia, the difference largely reflecting the availability and richness of data sources. We will describe the UK population construction in Section 4.2.

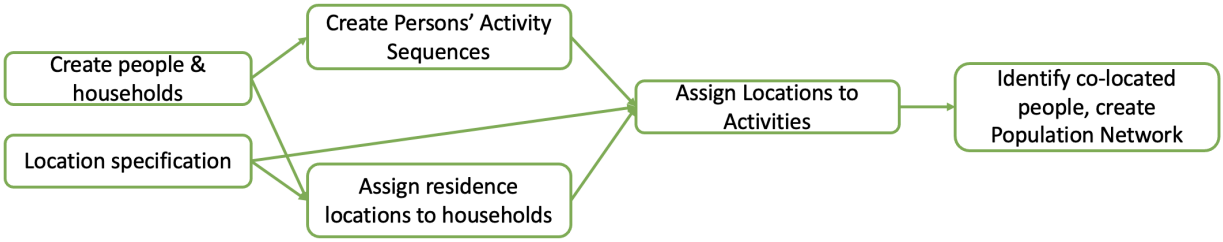


Figure 3. High-level sequence of models and steps used for constructing synthetic populations.

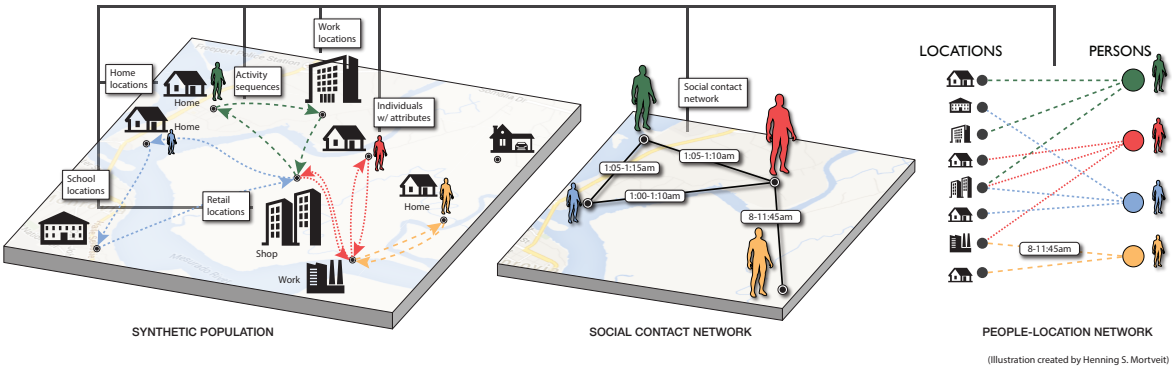


Figure 4. A high-level illustration showing the manner in which data is integrated in the modeling and construction of synthetic populations. The members of the populations will be equipped with a range of demographic attributes (details will depend on available data and application purpose), will have an associated contact network (denoted by G_P) as shown in the middle, and a person-location graph (denoted by G_{PL}) as shown on the right.

4.1. **Generating synthetic populations and networks for Virginia, USA.** To construct a population for a *geographic region* R (e.g., Virginia), we first choose a collection of *person attributes* from a set $\mathcal{D}$ (e.g., age and gender) and a set $\mathcal{T}_A$ of *activity types* (e.g., `Home`, `Work`, `Shopping`, `Other`, `School`, `College`, and `Religion`). The precise choices of $\mathcal{D}$ and $\mathcal{T}_A$ are guided by the particular scenarios or analyses the population will serve. Described at a high level (see Figure 4), we (*i*) construct virtual people and places, (*ii*) assign activity sequences to people, (*iii*) assign each activity a location and time of visit, and, from this, we derive (*iv*) a contact network using co-occupancy and a contact model to infer edges. A high-level workflow for this process is illustrated in Figure 3 where the contact network is illustrated in the middle. The construction factors into a detailed sequence of steps which can be outlined as follows:

Step 1: Person and Household construction. Using *iterative proportional fitting* (IPF) [6, 12], the **base population** model constructs a set of individual persons $\mathcal{P}$ where each person is assigned demographic attributes from $\mathcal{D}$. By design, this ensures that $\mathcal{P}$ matches the statistical distributions of the Public Use Microdata Sample (PUMS) data from the US Census [47], which is one of the input datasets for the model. Additionally, this model partitions $\mathcal{P}$ into a set $\mathcal{H}$ of *households*. Here, the term *household* encompasses the traditional notion of "family" as well as other subsets of individuals residing in the same *dwelling unit* (e.g., dormitories, apartments, army barracks, or prisons). In Figure 6 of Section 6.1 we show the population size and average age distribution of our Virginia synthetic population as aggregated by county (administrative level 2).

Step 2: Activity sequence assignment. In this step, each individual $p \in \mathcal{P}$ is assigned a week-long **activity sequence** $\alpha(p) = (a_{i,p})_i$ where each *activity* $a_{i,p}$ has a *start time*, a *duration*, and an *activity type* from $\mathcal{T}_A$. Data sources used as input for this step include the National Household Travel Survey (NHTS) [50], the American Time Use Survey (ATUS) [48] and the Multinational Time Use Study (MTUS) [44]. We write $\alpha\colon \mathcal{P} \longrightarrow \mathcal{A}$ for the mapping assigned to each person. For this construction, we use Fitted Values Matching (FVM) for adults [29], and Classification And Regression Tree (CART) for children (see, e.g.,[8]).

Step 3: Location construction. The **location construction model** generates a set of spatially embedded locations $\mathcal{L}$ partitioned into *residence locations* where households live, and activity locations where people conduct their non-`Home` activities. This construction is highly granular, and is based on several data sources, including the Microsoft US Building data [31], HERE/NAVTEQ [23] for points-of-interest (POIs) and land-use classifications, and the National Center for Education Statistics (NCES) [34] for public schools.

Step 4: Activity location assignment. For each person $p \in \mathcal{P}$, the **activity location assignment** model assigns a location $\ell_i = \ell(a_i)$ to each of their activities a_i. We denote the sequence of locations visited by p as $\lambda_p = (\ell_i)_{i,p}$. The location assignment model uses the American Community Survey (ACS) commute flow data [46] to assign a target county c for each `Work` activity, and a particular location randomly within c based on attractor weights assigned to each location in c. School activity locations are assigned using NCES data, while remaining activities are anchored near home and work locations. The activity location assignment induces the bipartite *people location graph* G_{PL} with vertex sets $V_1 = \mathcal{P}$ and $V_2 = \mathcal{L}$ (the set of locations) and labeled edges all (p, ℓ) for which p visits ℓ. The label includes the activity type, the start time for the visit, and the duration of the visit; for more details, see the right side of Figure 4.

Step 5: Contact network generation. In this step, the **contact network model** uses the people-location graph G_{PL} to first derive the *co-location graph* $G_{\max}$ with vertex set $\mathcal{P}$ and edges all $e = (p, p')$ for people p and p' that are simultaneously present at the same location. Applying sub-location contact modeling at each location, we determine which of the edges of $G_{\max}$ will be retained to form the *contact network* G, which is also referred to as the *person-person contact network* and denoted by G_P (rather than simply G) to make this explicit. In this work, we use a random graph model referred to as the *Min/Max/alpha model* at each location to obtain G_P. Let ℓ be a location and let $N = N_\ell$ denote the maximal number of simultaneous visits to ℓ. We define the function $p_\ell \colon \mathbb{N} \setminus \{0, 1\} \longrightarrow [0, 1]$ as

$$p_\ell(N) = \min\big\{1, \big[\mathrm{Min} + (\mathrm{Max} - \mathrm{Min})(1 - e^{-N/\alpha})\big]/[N - 1]\big\} \,, \tag{2}$$

where $\mathrm{Min} < \mathrm{Max}$ are non-negative numbers and $\alpha > 0$. Given $p = p_\ell(N)$, one samples from this random graph model in the same manner as for the standard model $G_{n,p}$ by independently applying to each edge e at random the probability p corresponding to the location ℓ where $e \in G_{\max}$ originates. Thus, the parameters Min and Max bound the degree of each vertex locally at ℓ (in expectation) at the beginning of each visit; note, however, that the degree of person p in the resulting graph G is the accumulation of degrees across their trajectory to locations visited while executing their activity sequence. Thus, the choices of Min, Max and α will induce the degree of each vertex in a bottom-up manner; see [32] for full details.

Remark. The Virginia networks G_P feature contacts and edges throughout an entire week. To support this challenge, we extract sub-graphs, e.g., G_1, from G_P to represent the contact network on the particular days. We have collected summary statistics, and structural network measures and visuals for the Virginia networks in Section 6.1.

4.2. **Generating synthetic populations and networks for the United Kingdom.** The quality of a synthetic population of a region R depends the data available for R. Whereas we have highly detailed data available for the United States, this is typically not the case elsewhere, and a subset of the methods described in Section 4.1 are replaced by other methods that are suited for such cases, as described below.

Step 1. Person and household construction. For the UK, we use data from Gridded Populations of the World (GPW) [21] where we linearly interpolate across age ranges to construct a base population for which the set of attributes $\mathcal{D}$ includes age and gender; the construction is done per cell using fractional rounding. The construction of the household partition $\mathcal{H}$ from $\mathcal{P}$ uses adapted household size distributions collected from the United Nations (UN); we ensure that all household members belong to the same GPW cell, and that households within the cells statistically match the given distribution at the aggregated level.

Step 2. Location construction. For general activity locations, we employ a two-step modeling approach integrating (*i*) Point-of-Interest (POI) data from [43], (*ii*) counts of location-by-type collected as part of other related work, and (*iii*) a regression model to infer the counts in (*ii*) when such data is unavailable. Starting from the POIs, additional locations of the appropriate categories are constructed to match the data from (*ii*) and (*iii*), and are placed proportionately to the GPW v.4.0 population density.

Step 3. Activity sequence assignment. In general, for regions other than the US, we use a combination of regression models to determine suitable activity sequence collections (e.g., from MTUS [44]). The activity sequences we construct for UK run from midnight to midnight on a typical weekday, and, aside from utilizing different data sources, are assigned using the same core methods (FVM, CART) as were used for Virginia.

Steps 4 and 5. Activity location assignment and network construction. These steps use the same models and computational tools as described in the corresponding parts of Section 4.1.

Remark. We have collected summary statistics, and structural network measures and visuals for the UK populations and networks in Section 6.1.

4.3. **Second Versions of the Datasets.** The second version of the datasets were generated to be similar to the initial instances using small perturbations of populations and networks. In the case of Virginia (US), the base dataset is a population with activities and contacts covering an entire week. The two instances of the Virginia network represent two different days from that week. By design, these networks will be similar to the existing, first version.

In the case of the UK, three additional networks were constructed using the same location assignment that was used for the first version. This ensured that people are assigned to the same locations for all activities, but, due to randomness and the use of different random seeds in subsequent constructions, the precise contacts that arise within each location will differ somewhat across the instances. This, too, will generate data instances that are similar to that in the first version.

5. Disease Models

We generate the synthetic outbreak data by running an agent-based simulation, as described in Section 5.1. The disease model used in the ground truth generation and the ranges of its parameters are shown in Figure 5. Note that, in the setting we have chosen, the learner does not observe the exposed or asymptomatic states. In the data released here, an individual who is marked as susceptible at some time, might be susceptible, according to this model or they may actually be exposed. An individual who is asymptomatic appears in the data as susceptible.

More broadly, NSSAC's disease models, as used in, for example, [20], are split into *disease transmission* and *disease progression*, the former capturing the infection aspect between individuals, and the latter capturing the evolution of a person's health state once infected. Disease progressions are captured as probabilistic timed transition systems (PTTS) [5] over the set of health states. These permit weighted, probabilistic transitions from a state to the next (e.g., transitions (E, I) and (E, A) in Figure 5, which has weights 0.4 and 0.6, respectively), and also include dwell-time distributions for time spent in states which are set by the public health literature. Infections may take place when one or more individuals in an infectious state are in contact with a person in a susceptible state, and are modeled using the notion of *propensities* in the sense of Gillespie [18, 19]. Each contact or interaction between a susceptible person P and an infectious person P' generates a propensity $\rho_{P,P'}$ (a non-negative number) which factors in aspects such as the duration of contact, vaccination histories, the use of NPIs, and other aspects modulating the infectivity and susceptibility of P and P'. At the end of each iteration, the collection of propensities for susceptible persons P are then used to determine (i) if P transitions to an exposed state, and, if the

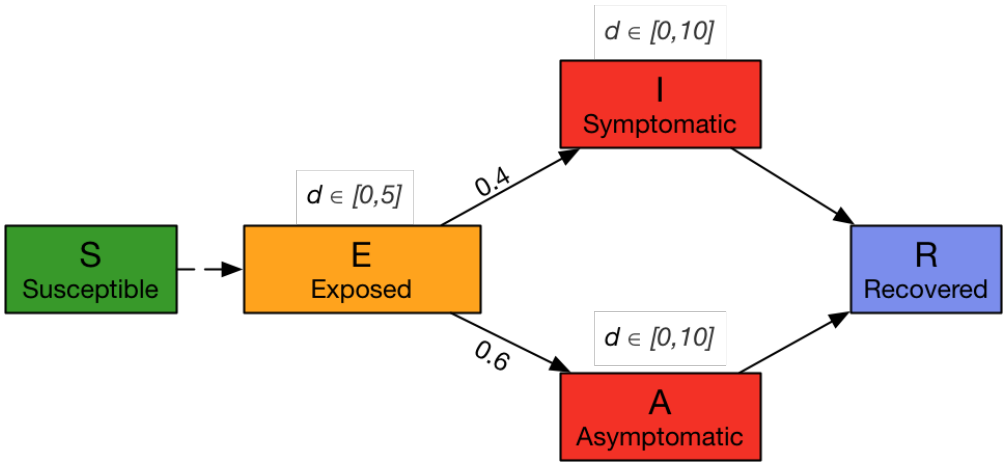


FIGURE 5. The disease model used in the ground truth generation consists of S (susceptible), E (exposed), I (infectious and symptomatic), A (infectious and asymptomatic), and R (recovered) states. Only infectious (I or A) nodes can infect susceptible (S) nodes. Ranges of the dwell time (d) of E, I, A states are specified. The asymptomatic state is not observable, so nodes in this state are taken as in the S state.

transition takes place, (ii) to whom in their list of contacts (P') we attribute the infection. Additional details can be found in [11].

Algorithm 1 SynOutbreak Generation: Overall methodology to generate a synthetic outbreak.

procedure SynOutbreak Generation($G(V, E), D, I_0, \ldots, I_T, \Delta$)
  Input: Synthetic contact network, $G(V, E)$
  Input: SIR within host disease model, D
  Input: Empirically observed infections $I_0, \ldots, I_T$
  Output: Disease state of nodes over time, $\mathbf{X}(T + \Delta)$
  Select a set of nodes $V_I \subseteq V$ as initial infections
  ▷ τ is the probability in D that a node infects a neighboring node
  Choose τ^* to imitate $I_0, \ldots, I_T$
  Using τ^*, run simulations from time 0 to $T + \Delta$
end procedure

5.1. **Synthetic outbreak data (ground truth).** To obtain the training and testing data, we run SynOutbreak Generation to obtain $\mathbf{X}(T + \Delta)$. Data from 0 to T corresponds to $\mathbf{X}(T)$, the training data. Data for each day t consists of the disease state of each individual $v \in V(G)$. We assume that each individual is in exactly one of the disease states {S, I, R} on any day. Data from $T + 1$ to $T + \Delta$ is used to construct the vector Y which will be used in the evaluation.

6. Comparison of Networks

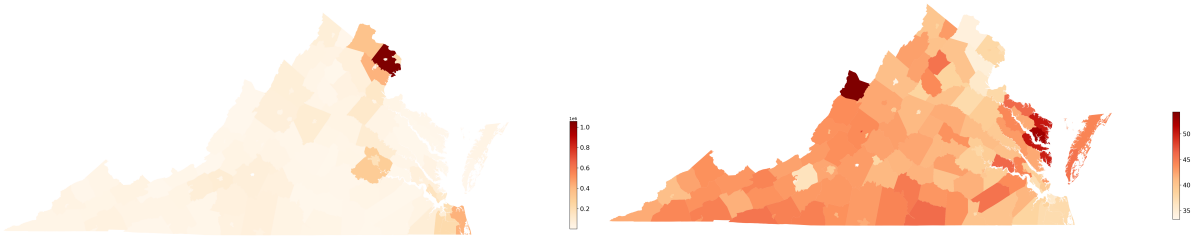
We use different networks for creating training data and evaluation data. This allows us to ensure clear separation between training evaluation and that global parameters were not inadvertently used during the model development process. For the Virginia network, we use

two instances of the network, representing activity on different days. These networks are based on synthetic daily activity. For the UK, because of different data availability, we use a different methodology: we regenerate several different networks using the same stochastic process.

In this, we show the ways in which the different networks used for Virginia and the UK differ. We find that while the networks are statistically similar, individuals have different connections on the different networks. This makes the different networks ideal for testing generalization because they ensure that the patterns being learned are learned based on the network dynamics rather than properties specific to individuals .

6.1. **Summary of population and network properties for Virginia (US) and the United Kingdom.**

6.1.1. *Population properties.* In Figures 6 and 7 we have provided spatial maps of population size distributions and average age distributions for the Virginia and UK populations. In the case of Virginia, the resolution is administrative area level 2, whereas for the UK it is level 1.



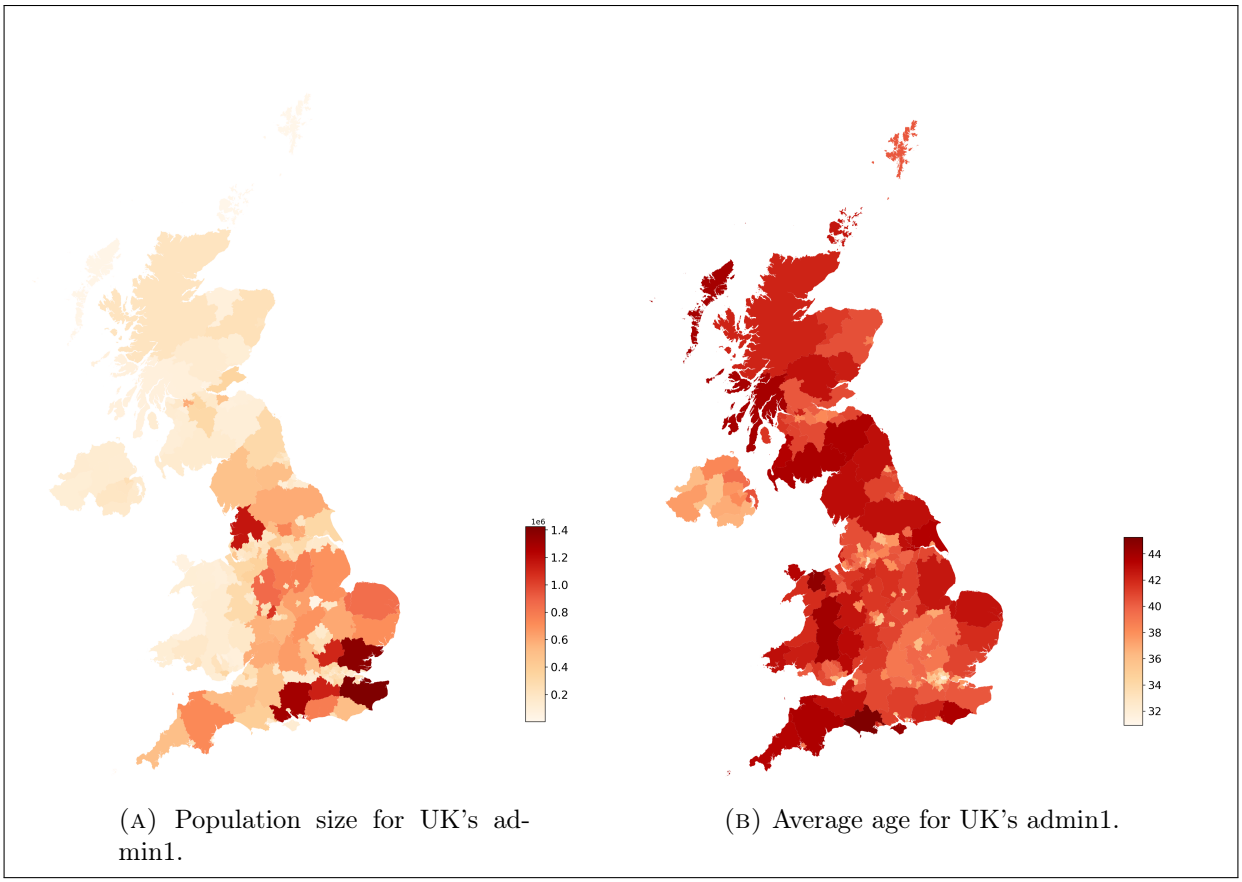
(A) Population size for VA's admin2.

(B) Average age for VA's admin2.

FIGURE 6. Regional comparison maps for Virginia.

Figure 8 provides spatial maps giving the population density for the synthetic populations of Virginia and the UK.

6.1.2. *Network properties.* Table 2 provides an overview of network properties across a range of measures. As can be seen, the differences are quite small across all measures.



(A) Population size for UK's admin1.

(B) Average age for UK's admin1.

FIGURE 7. Regional comparison maps for UK.

Network	$\|V\|$	$\|E\|$	$\bar{\delta}$	$\bar{T}$	$d_{\max}$	$T_{\max}$	$\bar{c}$	$k_{\max}$	r	D	λ_1	λ_2
VA 1*	7.6e+06	1.7e+08	43.53	235.64	565	3821.73	0.09	57	0.98	12	106.38	84.36
VA 2	7.6e+06	1.5e+08	40.53	225.87	642	3181.35	0.09	51	0.98	12	92.96	74.86
UK 1*	6.1e+07	5.3e+08	17.28	127.25	207	966.81	0.29	50	0.99	17	72.06	56.59
UK 2	6.1e+07	5.3e+08	17.30	127.40	194	944.75	0.29	51	0.99	18	72.20	56.75

TABLE 2. Comparison of structural properties of different replicates of synthetic population networks. Here $|V|$ is the number of nodes, $|E|$ the number of edges, $\bar{\delta}$ the average degree, $\bar{T}$ the average contact hours, $T_{\max}$ the maximal number of contact hours, $\bar{c}$ the average clustering coefficient, $k_{\max}$ the max core, r the normalized size of the giant component, D the diameter, and λ_1 and λ_2 are the largest eigenvalues. The networks marked by * are the reference instances.

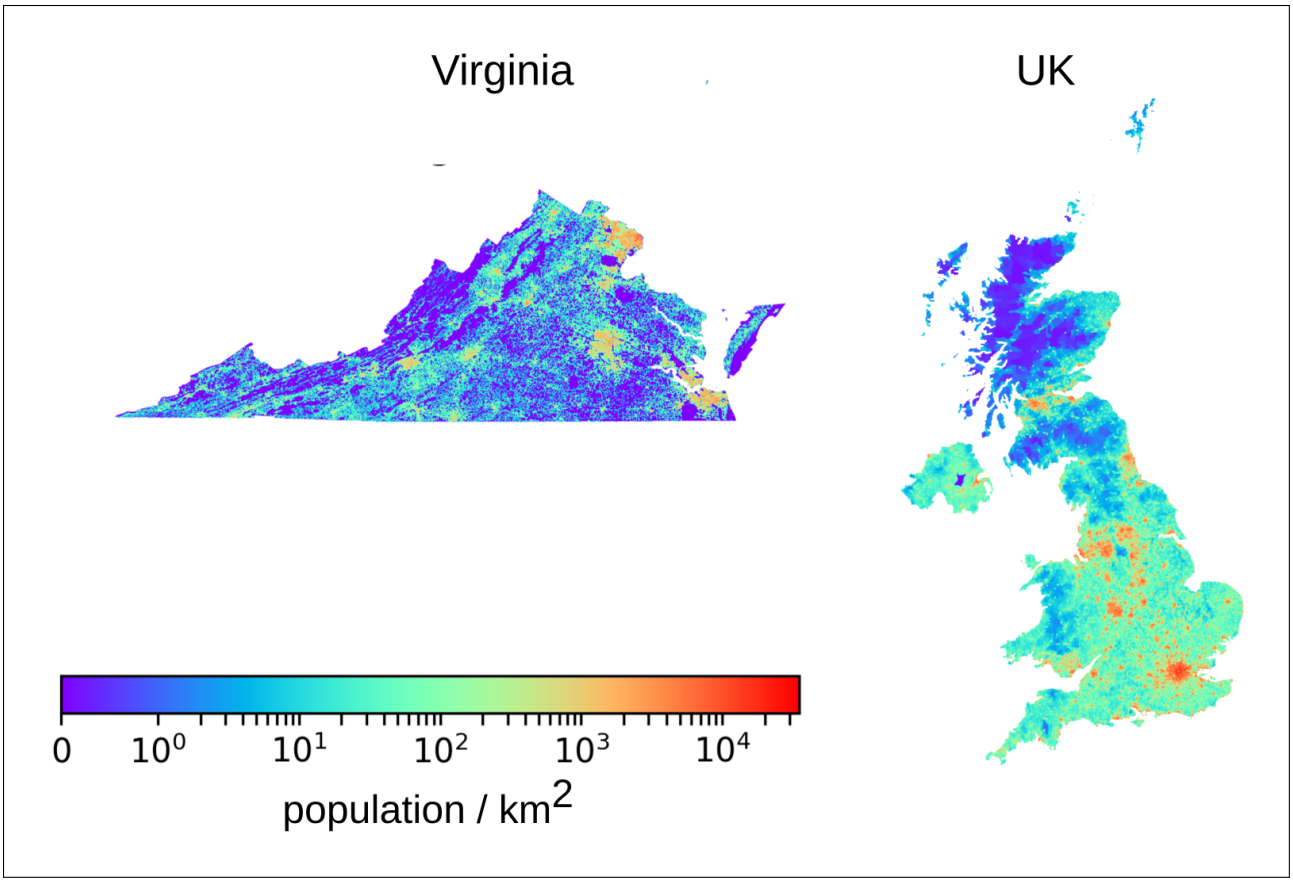


FIGURE 8. Population densities for Virginia and the UK.

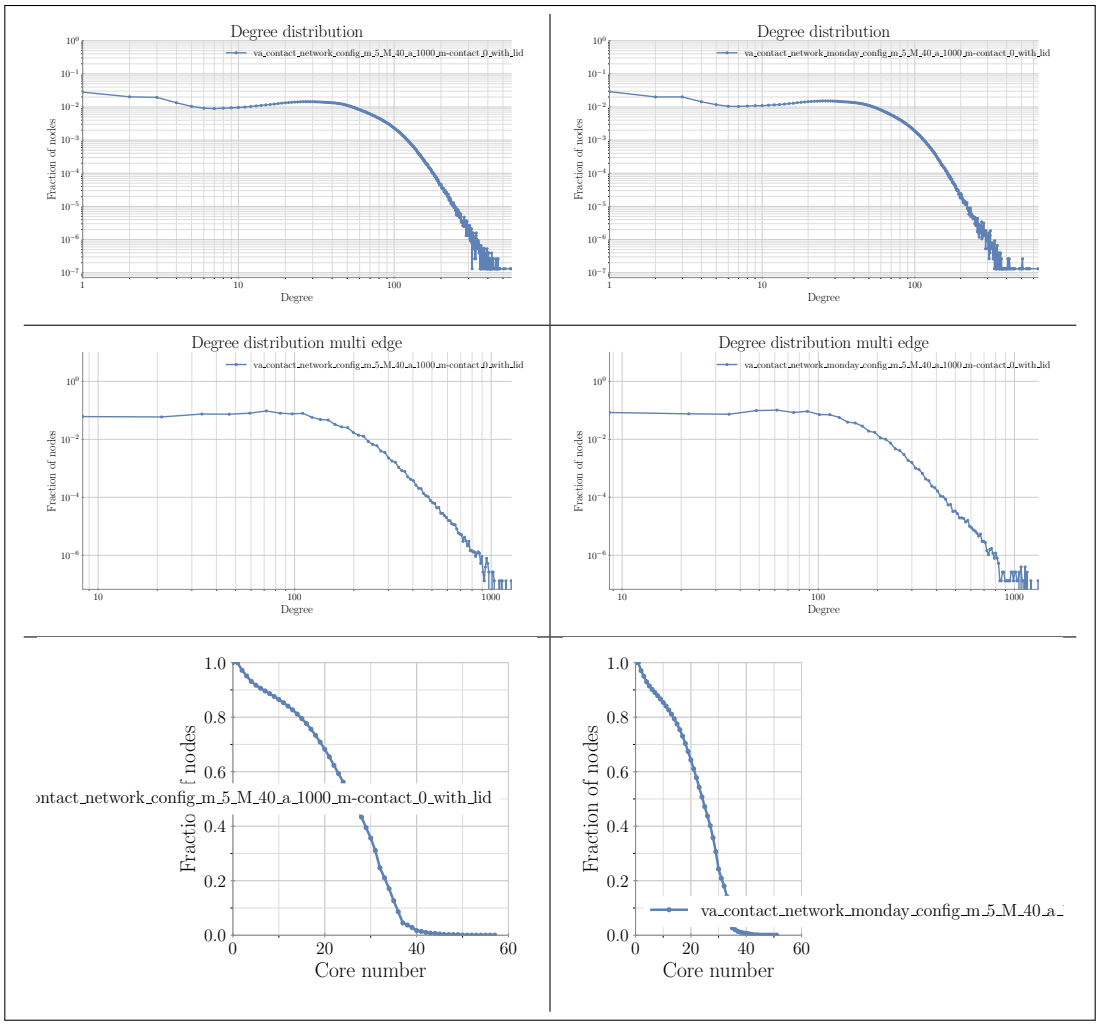


FIGURE 9. Network properties for contact networks of Virginia. Column one shows the base case, whereas the second column shows a replicate. The first row shows degree distributions for the simple contact networks (accumulated contact times), the second row shows degree distributions for the multi-edge versions, and the third row shows the distributions of the core number of the networks.

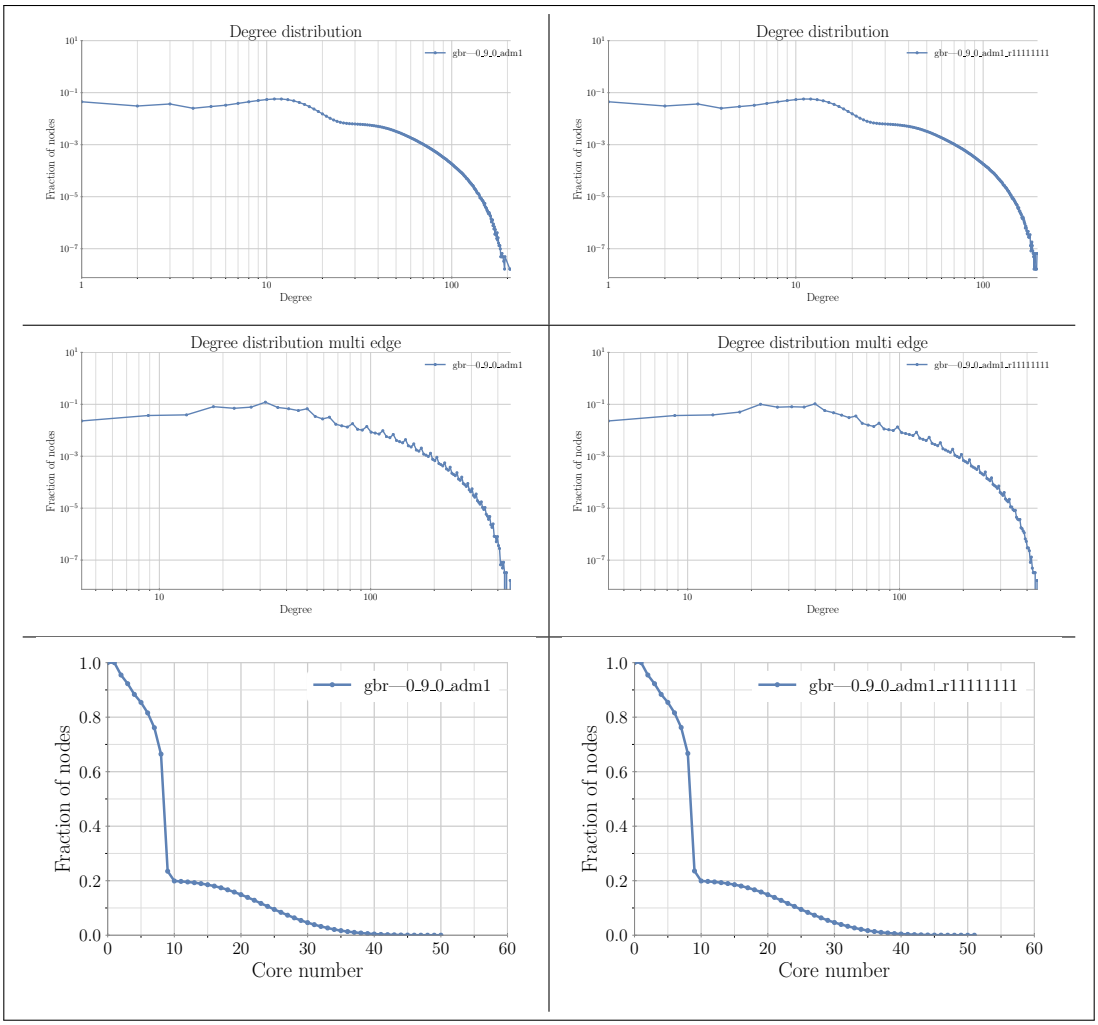


FIGURE 10. Network properties for contact networks of the UK. Column one shows the base case whereas the second column shows a replicate. The first row shows degree distributions for the simple contact networks (accumulated contact times), the second row shows degree distributions for the multi-edge versions, and the third row shows the distributions of the core number of the networks.

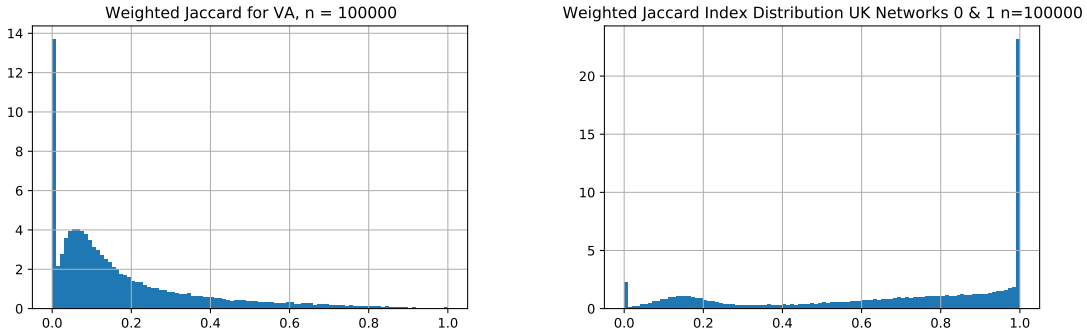


FIGURE 11. (a) The weighted Jaccard index for a sample of 100,000 individuals from the Virginia population. This compares two instances of the network. Note that individuals with no neighbors in both networks recieve a score of 0 (b) Weighted Jaccard index for a sample of 100,000 individuals from the UK population. This is a comparison of two replicate graphs for the UK.

6.2. **Comparing Network Neighbors.** We can compare networks using the weighted Jaccard index [35]. The weighted Jaccard index of neighbors measures how much overlap there is between the neighbors of a person in one network and the neighbors of that neighbor in a different network. For individual v in networks G_1 and G_2, this compares the weight of neighbors shared between G_1 and G_2 to the total weight of neighbors of v in G_1 and G_2. In other words, it measures the weight of the overlap in edges from v between G_1 and G_2. Given $G_1(V, E)$ and $G_2(V, E')$, two weighted graphs with the same vertices, but different edges, and weights $w_{1,u,v}$ and $w_{2,u,v}$, the weighted Jaccard index for $v \in V$ is

$$\frac{\sum_{(u,v)\in E\cap E'} \min(w_{1,u,v}, w_{2,u,v})}{\sum_{(u,v)\in E\cup E'} \max(w_{1,u,v}, w_{2,u,v})} \tag{3}$$

When $u, v \in E$ and $u, v \notin E'$, we consider u, v in E to have weight 0. The weighted Jaccard index ranges from 0 to 1.

Virginia Network Comparison We sampled 100,000 individuals from the Virginia network, and computed the weighted Jaccard index between the two instances of the network. The results of this are shown in Figure 11(a). This shows that, while there are certain similarities for individuals between the two instances of the network, there is not very much overlap between the activity profiles. When we performed an unweighted Jaccard Index using the same procedure, we found a distribution much more biased towards 0, which suggests that higher weighted contacts are more likely to occur in both instances.

UK Network Comparison The UK networks were synthesized using different data and using a different synthesis method, and therefore exhibit different similarity properties. This is a comparison using a similar measure. See Figure 11(b) The UK networks show some variation, but a large amount of overlap. A large proportion of individuals in these two UK networks have the exact same contacts in both. This is likely due to differences in data availability and granularity. While the Virginia networks each represent activity on different days, the UK networks do not, hence there is more overlap.

In Figure 12 the scatter plot shows different degrees of each node (from a sample of 1000 nodes) in different networks. We define the degree of a node to be the number of distinct neighbors of this node. If a node has the same degree (not necessarily the same neighbors) in two networks, it should lie on the $y = x$ line. We find that while degree distributions are similar between networks, node degrees do differ, with a correlation coefficient of only 0.230. We can also consider the weighted degree of a node, which is the total contact duration this node has with all its neighbors. In Figure 13, we find that the two Virginia networks are less different, with a correlation coefficient of 0.557, and data points more concentrated along the $y = x$ line in the scatter plot.

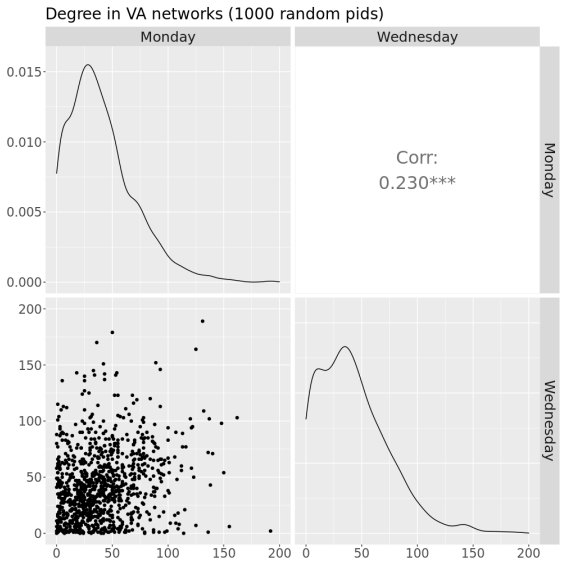


FIGURE 12. Compare node degrees in two Virginia networks.

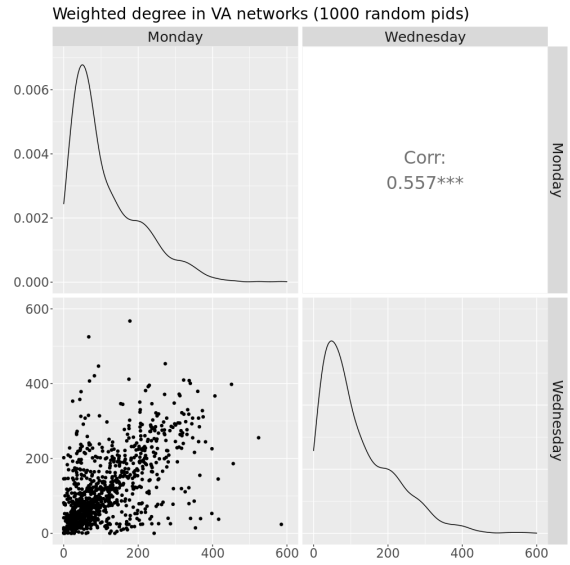


FIGURE 13. Compare node weighted degrees in two Virginia networks.

For the two UK networks, we expect them to be less different because they are generated with the same model and parameters, but with different random seeds. In Figures 14 and 15, we find that the node degree and weighted degree are indeed more consistent between two UK networks.

7. Examples of centralized implementations to show proof of concept

We describe the setup for the learning task and three example centralized implementations as proof of concept for the proposed task. These examples demonstrate that IRP, despite its limits, can be meaningfully learned. They further demonstrate the diversity of methods that may be applied to this problem. We employ a simple logistic regression (LR), a graph neural network based on prior work in this domain (GNN), and a mechanistic baseline based on a similar agent-based model (AC).

7.1. Example Centralized Implementation 1: Graph Neural Network (GNN).

Deep learning methods for classifying and predicting graph properties have been an area of research for several years. Because this area is so large, we adopt one of these deep learning methods as an example from prior literature. Specifically, we adapt Qiu et al.'s DeepInf framework [36]

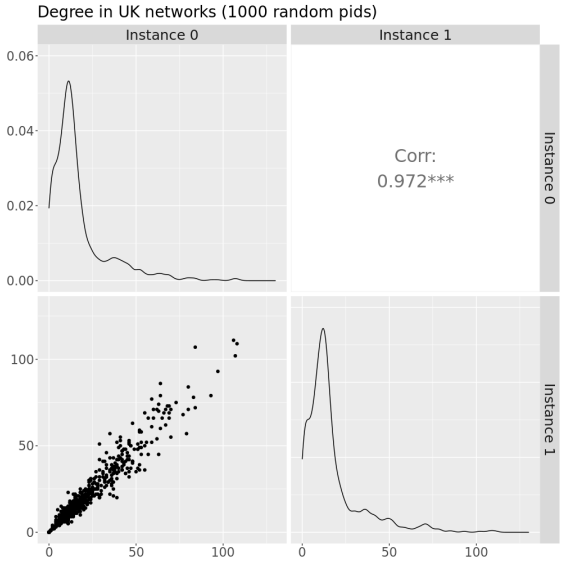


FIGURE 14. Compare node degrees in two UK networks.

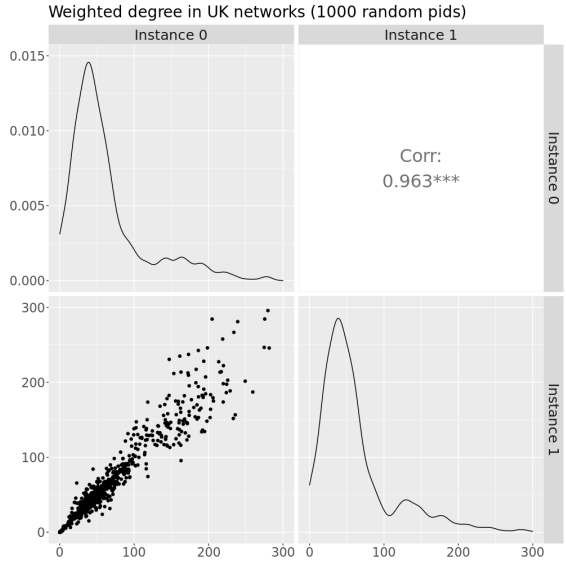


FIGURE 15. Compare node weighted degrees in two UK networks.

to learn the epidemic dynamics. We applied this framework to demonstrate the existence and efficacy of already-existing deep learning approaches for this challenge problem. Adapting DeepInf to IRP requires alterations to the way in which training and evaluation examples are constructed.

7.1.1. *Sampling Strategy for* GNN. The graph neural network takes as input examples indexed by individual v, time t, and labeled by $y(t, t+\Delta, v)$. $y(t, t+\Delta, v)$ is 1 if there exists $t' \in (t, t+\Delta]$ such that $s_v(t) = I$, and 0 otherwise.

Because the graphs we consider are large (over 100s of millions of edges), it is neither feasible nor necessary to use every single node as a training instance. At any given t, only a small proportion of the nodes are infected, meaning only a subset of the network is subject to any kind of dynamic process. Therefore, using all timesteps and all nodes would not only take a large amount of computation, it is also likely unnecessary. We instead sample the data to prioritize the instances where dynamics are happening.

We do two types of sampling: (i) we sample the neighborhoods around each vertex we consider, and (ii) we select only instances where there is more than minimal likelihood of any kind of dynamic behavior being observed.

We use Data-Gen (see Algorithm 2) to generate this data. We sample the neighborhoods such that they all have the same maximum size of r nodes. These values, plus the state of v and of each $u \in N_r(v)$ are used to produce the prediction features for both LR as well as GNN. The precise features used for each example implementation differ (for example GNN makes use of graph embeddings, whereas LR does not).

Training Data: We use the neighborhoods and training instances generated by Data Gen to build features. Unlike Qiu et al., who apply certain data selection steps to address class imbalance issues, we use the output of Data Gen directly. This models the efficacy of an actually-deployed model.

Algorithm 2 Data Gen: The procedure to generate labeled data for data-driven methods

procedure Data Gen($G(V, E), \mathbf{X}(T)$)
  Input: Synthetic contact network, $G(V, E)$
  Input: Disease history, $\vec{X}_t$
  Output: Labeled instances $S = \{(v_1, t_1, y_1), \ldots, (v_m, t_m, y_m)\}$
  $S = \{\}$
  For each $v \in V$, do Random Walk with Restart to get $\tilde{N}_r(v)$
  for $v \in V$ **do**
    if $\exists v' \in \tilde{N}_r(v)$ infected at any time $t < T - \Delta$ **then**
      if v is infected at any point in $(t, t + \Delta]$ **then**
        $S.add((v, t, 1))$
      else
        $S.add((v, t, 0))$
      end if
    end if
  end for
end procedure

The features used are as follows: the embedding of v in $\tilde{N}_r(v)$, the state of each of the nodes in $\tilde{N}_r(v)$ at time t, a one-hot vector that indicates the index corresponding to v in the state vector, and the adjacency matrix formed by $\tilde{N}_r(v)$.

We additionally use information about v: the age of v, the coreness of v within $N_r(v)$, the authority of v, and the inverse of the degree of v. The embeddings are normalized before being concatenated to the rest of the features. These features are inputs to a graph attentional network with 8 attention heads and two hidden layers. The third layer is an output layer. The graph is trained with a learning rate of 0.1, dropout 0.2, and using the PyTorch Adagrad optimizer. We train the model for 500 epochs.

LR	# infected at time t in $N_2(v)$ # infected at time t in $N_1(v)$ # infected at time t in $N_2(v)$ Infection Weight$(v, t, 1)$ Infection Weight$(v, t, 2)$ Was v infected in the past? Age
GNN	64-dimensional embedding of v $I_t \in \{0, 1\}^r$ indicating infections at time t $E \in \{0, 1\}^r$ indicating v Coreness of v Authority of v Inverse of degree of v Age

TABLE 3. Features used in LR and GNN

7.2. **Example Centralized Implementation 2: Logistic Regression (LR).** To compare the GNN method with a simpler, but still data-driven, method, we tried predicting using a logistic regression with features based on the 2-neighborhoods of vertices. The 2-neighborhood of a vertex is the subgraph induced by all vertices within 2 edges of that vertex. We call these 2-neighborhoods $N_2(v)$. These features use more of the graph information than those used by GNN.

Instead of using the embeddings and representations of node states used in GNN, we use the infection weight. Essentially, this is a decaying weighted sum of the edge weights leading to v.

Let $p(u, v) = \{(v, w_1), \ldots, (w_k, u)\}$ be the shortest path from u to v, starting from v. We calculate the infection weight using Infection Weight (Algorithm 3). The infection weight is a measure of the duration of contact between v and an infected contact. In our logistic regression, we use $l = 1, 2, 3$.

As additional features, we use the number of infected nodes in $N_2(v)$ at time t', the number of infected neighbors of v at time t', and the number of infected people no more than 2 links from v infected at time t', as well as the "infection weight".

FeatureRoutine 3 Infection Weight: For computing the discounted infected contact duration.

procedure Infection Weight$(v, t, N_l(v), \vec{X}_t, l)$
  Input: vertex $v \in V$**, time** t**, maximum path length** l
  Input: neighborhood around v**,** $N_l(v)$
  Input: state of nodes at time t**,** $\vec{X}_t$
  Output: infection weight $w_{v,t}$
  $A(v, t) = u \in N_l(v)$ if u is infected at t and u within l steps of v
  $w_{v,t} = \sum_{u \in A(v,t)} \sum_{i=1}^{|p(u,v)|} \frac{1}{2^{i-1}} w_{e_i}$
end procedure

We resampled our training data using Synthetic Minority Over-sampling Technique (SMOTE) [10] to address class imbalance. We normalized our training features to remove the mean and to have unit variance, and applied a transformation to the evaluation data using the same parameters as those derived from the training features. We evaluated our regression on data that reflected the true distribution observed during sample extraction.

7.3. **Example Centralized Implementation 3: Aggregate Calibration (AC).** We use a generative approach which learns a distribution D of agent-based disease models based on our work in [7]. This involves following the steps described in Algorithm 4.

Note that despite the data available as an input to this algorithm being just SIR states, it fits a *more* complex disease model. Likewise, the disease model is *less* complex than the actual model used to generate the synthetic outbreak; it lacks an asymptomatic state. The disease model used in our mechanistic baseline is shown in Figure 16. Note that the disease transmissibility is not given to the learner, so our baseline estimates its value by calibration. This reflects how this type of modeling is actually done – one makes certain parameterized assumptions about how the disease progresses.

7.4. **Evaluation of the example centralized baselines.**

Algorithm 4 AC: Centralized algorithm based on causal models.

procedure AC($\mathbf{X}(T), \mathcal{D}, R$)
  Input: Past infection states $\mathbf{X}(T)$
  Input: Space of disease models $\mathcal{D}$
  Input: Number of runs R
  Let $\mathcal{D}_\tau$ be the set of disease models with all parameters other than transmissibility, τ, fixed
  Compute $I_0, \ldots, I_T$, total number of infections over time, from $\mathbf{X}(T)$
  Learn through calibration distribution D over $\mathcal{D}_\tau$ most consistent with the observed cases $I_0, \ldots I_T$
  Sample R times from D, and perform simulations to $T + \Delta$
  Retain $s_v(t; \theta)$ for each node $v \in U'$ from $T + 1$ to $T + \Delta$.
  Compute the probability of infection $p(v)$ for each node v from time $T + 1$ to $T + \Delta$ as a sample average.
end procedure

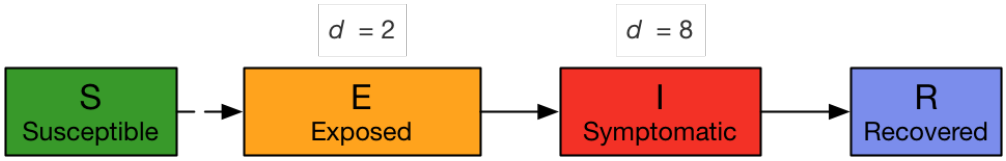


FIGURE 16. The disease model used in the mechanistic baseline consists of S (susceptible), E (exposed), I (infectious and symptomatic), and R (recovered) states. The dwell times of E and I states are chosen from the given range.

7.4.1. *Performance of Neural Networks (* GNN *) and Logistic Regression (*LR*).* We evaluated the logistic regression and graph neural network on the Virginia data. We extracted training examples up until day 56, and evaluated it on days 57-63, to assess projections for $\Delta = 7$. The sampling procedure for these methods does not necessarily produce a prediction for all nodes. Specifically, a node may be present which does not have any infected neighbors at time t. For these nodes, we choose a default value of 0.

We evaluated the logistic regression and graph neural network using the AUPRC (or area under the precision recall curve). To get probability estimates for the neural network method, we used the output layer weights. To compare performance, we used two baselines. We also did a logistic regression just on age to test whether the features we were using were valid.

In comparison to the baseline methods, we show significant improvement. Because of the class imbalance, as well as the inherent randomness of the data generation, getting performance that looks *good* is difficult. The precision-recall curves shown in Figures 17 and 18 are relatively low on their own. However, if we compare the areas under these curves to the AUPRC measured from the dummy methods shown in Table 4, we see significant improvements.

7.4.2. *Performance of the aggregate calibration (AC) model.* In Figures 17 and 18 we show the Precision-Recall curve of the predictions for Virginia and UK, respectively, by the AC model. We compare its performance against that of a random model, which assigns a value

	AC	LR	GNN	Age LR	Uniform	Random
US	0.05	0.045*	0.008	0.003	0.002	0.002
UK	0.136	0.098*	0.053*	0.025*	0.016*	0.016*

TABLE 4. The AUPRC for each dataset and method. Metrics marked with * were calculated on a random subset of individuals selected from the UK population data.

from uniform distribution $U(0,1)$ to each individual. Clearly, our AC model outperforms the random model on both Virginia and UK datasets.

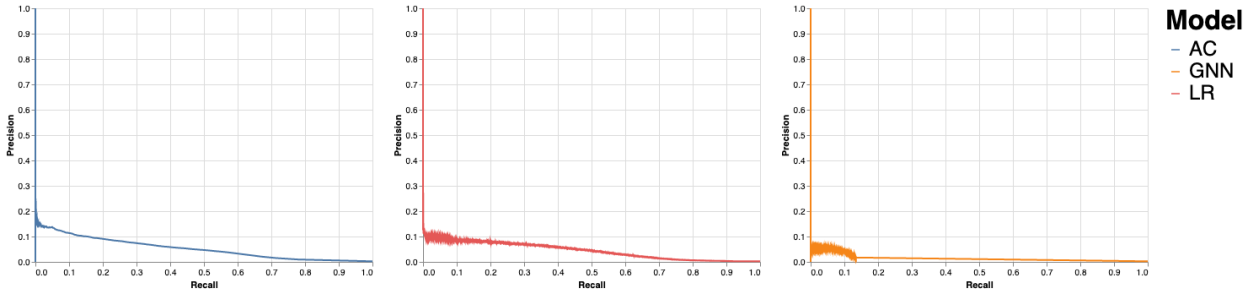


FIGURE 17. Precision-recall curves for the AC, LR and GNN models as computed over all Virginia individuals.

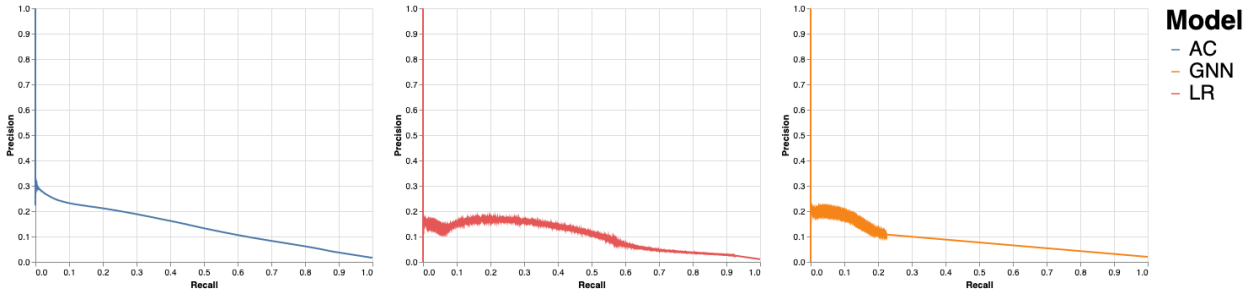


FIGURE 18. Precision-recall curves for the AC, LR and GNN models. Curve for AC is computed over all UK individuals, while curves for LR and GNN are computed over a sample of 2.35 million people.

8. FEDERATION MODEL

In the baselines described in section 7, there was full access to the graph and to node states. This type of data, if it were real, would be highly sensitive. Health data is generally considered to be one of the more private forms of data, and mobility data can be quite revealing about life aspects beyond one's commute [24]. Therefore, we consider a horizontal federation model where different regions are the data-holders for subsets of the graph.

More formally, given a social contact network $G(V,E)$, we partition the graph into k disjoint components, $G_1(V_1,E_1)\dots G_k(V_k,E_k)$. This is achieved by partitioning the set V of nodes of the graph into $V_1,\dots,V_k \subseteq V$, s.t. $V_i \cap V_j = \emptyset$ and $\cup_i V_i = V$. An edge $e=(u,v)\in E$

is in the set E_i iff $u, v \in V_i$. In other words, an edge is included in E_i iff both the end points belong to the partition. Edges that cross these partitions are excluded. Federation i has access to network $G_i(V_i, E_i)$ and $s_v(t)$ for each $v \in V_i$. As an example for the Virginia data, we have partitioned the dataset along county lines, creating 133 partitions with 113,176,615 edges total (out of 185,944,310, 39% loss). This models the situation where a county official or other data curator may have information about their local area of responsibility, but not about neighboring areas. Other horizontal partitions could be constructed, including partitioning by health district or by socio-demographic data associated with individuals (e.g. age).

8.1. **Creating an additional copy of the synthetic data.** As a way to overcome possible advantages that might be gained by methods that implicitly learn the entire network using certain hyper-parameters, we have generated another copy of the synthetic data. This data will not be revealed to the participants. The intent is that the developed methods would be evaluated using this second dataset to assess the method's generalizability. Our goal was to create a second synthetic dataset that was *similar* to the first copy. Our first network represented a normative weekday in a year. A natural second dataset then was a different normative weekday. Details of the differences are summarized in the Appendix.

Acknowledgements. This work is based on the dataset and methods developed under NSF RAPID: COVID-19 Response Support: Building Synthetic Multi-scale Networks, NSF Expeditions in Computing and the University of Virginia Strategic Investment Fund. The authors would like to thank members of the Biocomplexity COVID-19 Response Team, Network Systems Science and Advanced Computing (NSSAC) Division, University of Virginia. Special thanks to Erin Raymond, Lily Li, Golda Barrow, and Kristy Hall for their incredible and timely support. We also thank colleagues at NSF (James Joshi), NIST (Diane Ridgeway, Naomi Lefkovitz, Jim Horan, Joe Near), NSF/OSTP (Tess DeBlanc-Knowles), Driven Data (Jay Qi and Christine Chung) CDC (Matt Biggerstaff), and UK (Mark Durkee, David Buckley).

This work was partially supported by the National Institutes of Health (NIH) Grant R01GM109718, VDH Grant PV-BII VDH COVID-19 Modeling Program VDH-21-501-0135, NSF Grant No.: OAC-1916805, NSF Expeditions in Computing Grant CCF-1918656, CCF-1917819, NSF RAPID CNS-2028004, NSF RAPID OAC-2027541, NSF PREPARE CNS-2041952, US Centers for Disease Control and Prevention 75D30119C05935, NSF XSEDE TG-BIO210084 and NSF Prepare grant. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies. This work used resources, services, and support from the COVID-19 HPC Consortium (`https://covid19-hpc-consortium.org/`), a private-public effort uniting government, industry, and academic leaders who are volunteering free compute time and resources in support of COVID-19 research.

References

[1] US and UK to partner on prize challenges to advance privacy-enhancing technologies. Last accessed: April 2023.

[2] M. Ajelli, Q. Zhang, K. Sun, S. Merler, L. Fumanelli, G. Chowell, L. Simonsen, C. Viboud, and A. Vespignani. The RAPIDD ebola forecasting challenge: Model description and synthetic data generation. *Epidemics*, 22:3–12, 2018.

[3] A. Akinbi, M. Forshaw, and V. Blinkhorn. Contact tracing apps for the covid-19 pandemic: a systematic literature review of challenges and future directions for neo-liberal societies. *Health Information Science and Systems*, 9:1–15, 2021.

[4] C. L. Barrett, R. J. Beckman, M. Khan, V. A. Kumar, M. V. Marathe, P. E. Stretz, T. Dutta, and B. Lewis. Generation and analysis of large synthetic social contact networks. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 1003–1014. IEEE, 2009.

[5] C. L. Barrett, K. R. Bisset, S. G. Eubank, X. Feng, and M. V. Marathe. Episimdemics: An efficient algorithm for simulating the spread of infectious disease over large realistic social networks. In *SC '08: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, pages 1–12, 2008.

[6] R. J. Beckman, K. A. Baggerly, and M. D. McKay. Creating synthetic baseline populations. *Transportation Research Part A: Policy and Practice*, 30(6):415–429, 1996.

[7] P. Bhattacharya, D. Machi, J. Chen, S. Hoops, B. Lewis, H. Mortveit, S. Venkatramanan, M. L. Wilson, A. Marathe, P. Porebski, B. Klahn, J. Outten, A. Vullikanti, D. Xie, A. Adiga, S. Brown, C. Barrett, and M. Marathe. AI-driven agent-based models to study the role of vaccine acceptance in controlling COVID-19 spread in the US. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1566–1574, 2021.

[8] L. Breiman. *Classification and regression trees*. Wadsworth statistics/probability series. Wadsworth International Group, 1984.

[9] M. Castro, S. Ares, J. A. Cuesta, and S. Manrubia. The turning point and end of an expanding epidemic cannot be precisely forecast. *PNAS*, pages 26190–26196, 2020.

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[11] J. Chen, S. Hoops, B. L. Lewis, S. V. Henning S. Mortveit and, and A. Wilson. EpiHiper: Modeling and implementation, 2019. NSSAC Technical Report Series: No. 2019–003.

[12] W. E. Deming and F. F. Stephan. On a least squares adjustment of a sampled frequency table when the expected marginal tables are known. *Annals Math. Stats*, 11(4):427–444, 1940.

[13] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3):211–407, 2013.

[14] S. Eubank, H. Guclu, V. S. Anil Kumar, M. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429:180–184, 2004.

[15] S. Eubank, V. S. Anil Kumar, M. V. Marathe, A. Srinivasan, and N. Wang. Structure of Social Contact Networks and Their Impact on Epidemics. In *Discrete Methods in Epidemiology*, volume 70, pages 179–200. American Math. Soc., Providence, RI, 2006.

[16] Federal Committee on Statistical Methodology. STATISTICAL POLICY WORKING PAPER 22 (second version, 2005). Technical Report 22, Office of Management and Budget, Office of Information and Regulatory Affairs, 2005.

[17] S. Feng, G. Cong, A. Khan, X. Li, Y. Liu, and Y. M. Chee. Inf2vec: Latent representation model for social influence embedding. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 941–952. ISSN: 2375-026X.

[18] D. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comp. Phys.*, 22:403–434, 1976.

[19] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361, 1977.

[20] Fighting covid-19 with hpc, 2021.

[21] Gridded population of the world (GPW), v4. Last accessed: March 2022.

[22] D. C. P. Harikrishna Narasimhan and Y. Singer. Learnability of influence in networks. *Advances in Neural Information Processing Systems*, 28:23, 2015.

[23] HERE, 2020. http://www.here.com, Accessed April 2020.
[24] J. Hsu. The strava heat map shows even militaries can't keep secrets from social data. *Wired*, 2018. Section: tags.
[25] M. R. Islam, S. Muthiah, B. Adhikari, B. A. Prakash, and N. Ramakrishnan. DeepDiffuse: Predicting the 'who' and 'when' in cascades. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1055–1060. ISSN: 2374-8486.
[26] Kirsten Han. Broken promises: How singapore lost trust on contact tracing privacy.
[27] C. Li, J. Ma, X. Guo, and Q. Mei. DeepCas: An end-to-end predictor of information cascades. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 577–586. International World Wide Web Conferences Steering Committee.
[28] M. Lopes and G. Bontempi. On the null distribution of the precision and recall curve. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14*, pages 322–337. Springer, 2014.
[29] K. Lum, Y. Chungbaek, S. Eubank, and M. Marathe. A two-stage, fitted values approach to activity matching. *International Journal of Transportation*, 4:41–56, 2016.
[30] S. Marsh and A. Hern. Government admits breaking privacy law with NHS test and trace.
[31] Microsoft. U.S. building footprints. https://github.com/Microsoft/USBuildingFootprints, 2020.
[32] H. S. Mortveit, A. Adiga, C. L. Barrett, J. Chen, Y. Chungbaek, S. Eubank, C. J. Kuhlman, B. Lewis, S. Swarup, and S. Venkatramanan. Synthetic populations and interaction networks for the U.S. Technical report, NSSAC, University of Virginia, 2020. NSSAC Technical Report: 2019-025.
[33] C. Murphy, E. Laurence, and A. Allard. Deep learning of contagion dynamics on complex networks. 12(1):4720. Number: 1 Publisher: Nature Publishing Group.
[34] T. National Center for Education Statistics (NCES). Last accessed: February 2020.
[35] E. C. Pielou. *The interpretation of ecological data: a primer on classification and ordination*. John Wiley & Sons, 1984.
[36] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang. DeepInf: Social influence prediction with deep learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2110–2119, 2018.
[37] D. J. Rosenkrantz, A. Vullikanti, S. S. Ravi, R. E. Stearns, S. Levin, H. V. Poor, and M. V. Marathe. Fundamental limitations on efficiently forecasting certain epidemic measures in network models. *Proceedings of the National Academy of Sciences*, 119(4):1–7, 2022.
[38] C. Sáez, N. Romero, J. A. Conejero, and J. M. García-Gómez. Potential limitations in COVID-19 machine learning due to data source variability: A case study in the nCov2019 dataset. *Journal of the American Medical Informatics Association*, 28(2):360–364, 2021.
[39] J. A. Salomon, A. Reinhart, A. Bilinski, E. J. Chua, W. La Motte-Kerr, M. M. Rönn, M. B. Reitsma, K. A. Morris, S. LaRocca, T. H. Farag, F. Kreuter, R. Rosenfeld, and R. J. Tibshirani. The US COVID-19 trends and impact survey: Continuous real-time measurement of COVID-19 symptoms, risks, protective behaviors, testing, and vaccination. *Proceedings of the National Academy of Sciences*, 118(51):e2111454118, 2021. Publisher: Proceedings of the National Academy of Sciences.
[40] Samarth Swarup and Madhav Marathe. Generating synthetic populations for social modeling. IJCAI, 2016.
[41] Samarth Swarup and Madhav Marathe. Generating synthetic populations for social modeling. AAMAS, 2017.
[42] A. Sankar, X. Zhang, A. Krishnan, and J. Han. Inf-VAE: A variational autoencoder framework to integrate homophily and influence in diffusion prediction. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, pages 510–518. Association for Computing Machinery.
[43] SLIPO. World-scale OpenStreetMap POIs in CSV. Last accessed: June 2021.
[44] The University of Oxford. The Multinational Time Use Study (MTUS). Last accessed: February 2020.
[45] A. Tiu, Z. Susswein, A. Merritt, and S. Bansal. Characterizing the spatiotemporal heterogeneity of the COVID-19 vaccination landscape. Technical report, medRxiv, 2021.
[46] United States Censuc Bureau. 2011-2015 5-year ACS commuting flows. Last accessed: April 2020.
[47] United States Census Bureau. American Community Survey 2013-2017 5-year estimates. Last accessed: February 2020.

[48] United States Department of Labor, Bureau of Labor Statistics. The American Time Use Survey (ATUS). Last accessed: February 2020.

[49] US Census Bureau. Disclosure avoidance for the 2020 census: An introduction. Technical report, US Government Publishing Office, 2021.

[50] U.S. Department of Transportation, Federal Highway Administration. The National Household Travel Survey (NHTS). Last accessed: February 2020.

[51] M. Wilinski and A. Lokhov. Prediction-centric learning of independent cascade dynamics from partial observations. In *Proceedings of the 38th International Conference on Machine Learning*, pages 11182–11192. PMLR. ISSN: 2640-3498.

[52] C. O. Wilke and C. T. Bergstrom. Predicting an epidemic trajectory is difficult. *PNAS*, pages 28549–28551, 2020.

[53] Will Douglas Heaven. Hundreds of AI tools have been built to catch covid. none of them helped.

Appendix A. Notations and additional terminology

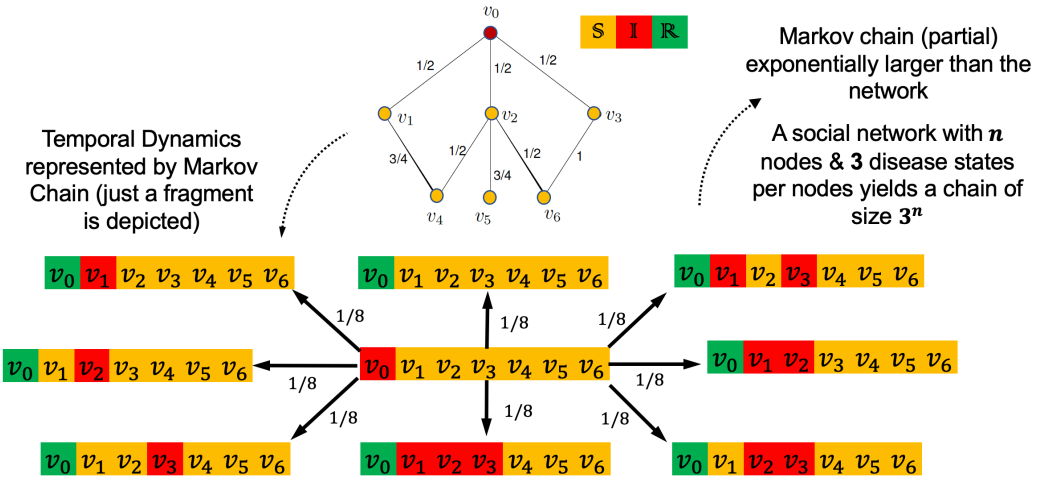


FIGURE 19. Part of the phase space of the SIR dynamics on the network in Figure 1. The node colors yellow, red and green represent the states S, I and R, respectively. The configuration at time $t = t_0$ with node v_0 in state I is the initial configuration, and is shown in the center. The transitions to possible configurations, along with their probabilities are shown.

Dendrogram: The disease state of each individual on each day. Assuming a disease model including S, I, R states, the dendrogram indicates for each individual u, whether u is ever infected, and, if yes, when u is infected and when u is recovered.

Transmission tree: Transmissions in the form of u infects v on day t. Assuming a single variant, no immune waning, and a single infector, transmissions can be represented as trees by connecting each infected individual with its infector.

Appendix B. Data Schema

Two synthetic population datasets are provided for this challenge: one for the UK (including Northern Ireland), and the other for the state of Virginia, USA. The UK dataset includes one weekday of activity assignments, while the USA dataset includes a normative day of activity assignments. Each synthetic population dataset is made up of data that will be delivered across several comma-separated-value (CSV) files as described in Table 7. Each file has data for a particular region and component (e.g., person, household, etc.) of the overall population schema. The fields contained in each file, along with their descriptions, are provided in tables 8– 14 and figure 20 shows the relationships between different files.

The schema of each file is described below:

TABLE 5. Notation and their values implemented in the synthetic dataset.

Notation	Meaning	Value in synthetic data
day 0	first day of simulation	2020-08-13
day T	last day of real cases data used in calibration for ground truth generation	2020-09-30
day $T + \Delta$	last day of evaluation data	2020-10-07
Δ	duration prediction period	7 days
V	set of all people in the study population	Virginia or UK population
U	subset of V whose disease states are given to a learning algorithm	$U = V$ for now
U'	subset of $V \setminus W$ where W represents the nodes in the learning data that are infected (and possibly recovered)	select nodes in $V \setminus W$ with probability π
π 7height$s_v(t)$	probability used to choose U' disease state of node v on day t	0.1, 0.2 one of {S,I,R}
$y(v)$	ground truth binary outcome of node v during test period $(T, T + \Delta]$	0 or 1
$\mathbf{X}(T)$	observed history of the disease evolution until time t (this specifies the disease states of all nodes at every time $t' \leq t$)	tuples $(v, t, s_v(t))$
$h_v(\mathbf{X}(T))$	Probability that $s_v(t) = I$ for some $t \in (T, T + \Delta]$, as predicted by the learned function $h(\cdot)$ using history $\mathbf{X}(T)$	value in $[0, 1]$

TABLE 6. Notations and their values implemented in the mechanistic baseline.

Notation	Meaning	Value in synthetic data
$p(v)$	probability that v will be infected during test period $(T, T + \Delta]$, as predicted by the mechanistic baseline	$[0, 1]$

Field	Description
hid	Household ID: An integer identifying a household as defined in the Household file
pid	Person ID: A unique integer identifying a person
person_number	The sequence identifier related to the indicated person's position within the household. A household with 3 people would have person_numbers 1, 2, and 3.
age	Age of person
sex	Gender of person

TABLE 8. Person File

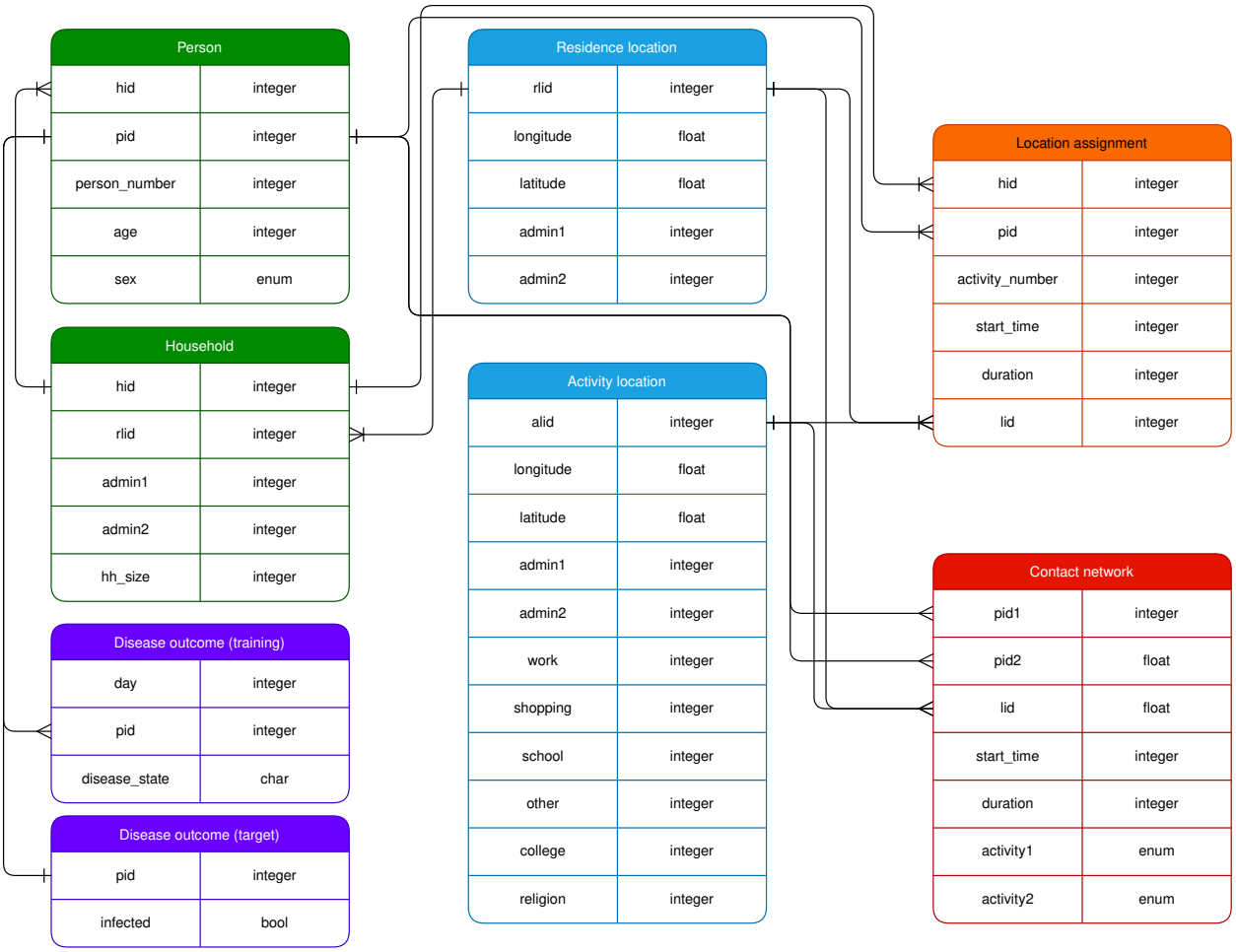


FIGURE 20. Data Structure Diagram.

Field	Description
hid	Household ID: A unique integer identifying a household
rlid	Residence location ID
admin1	For UK, this is the ADCW ID for the admin1 region; for Virginia, USA, this is the state FIPS code (51)
admin2	For UK, admin2 is the same as admin1 because ADCW does not provide admin2 level for the UK. For Virginia, USA, this is the county FIPS code
hh_size	Household Size: Number of persons in a household

TABLE 9. Household File

Data Component	Description
Person	Each row represents one synthetic individual in the population, including their age, gender, and the household to which they belong.
Household	Each row represents one synthetic household in the population, including its residence location, administrative regions, and the number of household members.
Residence Locations	Each row represents a residence location (e.g., where a household may reside.)
Activity Locations	Each row represents a non-residence location where people may go over the course of the day (e.g., work, school, shopping, etc.)
Activity Location Assignment	Each row maps an individual to an activity and the location where that activity took place. An individual will likely have multiple activity locations over the course of a day.
Population Network	Indicates when and where two people came in contact, and for how long.
Disease Outcomes (training)	Each row indicates an individual's disease status for a simulation day.
Disease Outcomes (target)	Each row indicates if individual was infected during the forecasting period.

TABLE 7. Various components of the synthetic population dataset. The naming convention is: {region}_{component}_ver_{major}_{minor}.csv, where {major} and {minor} indicate the version of the population and component indicates person, household, etc.

Field	Description
rlid	Residence Location ID: A unique integer identifying the residence location.
longitude	Longitude of the location
latitude	Latitude of the location
admin1	See household file description
admin2	See household file description

TABLE 10. Residence Locations File

Field	Description
alid	Activity Location ID: Unique integer identifying the location where non-HOME activities can take place
longitude	Longitude of the location
latitude	Latitude of the location
admin1	See household file description
admin2	See household file description
work	Does the location support work activities? (Value is 0 or 1)
shopping	Does the location support shopping activities? (Value is 0 or 1)
school	Does the location support school activities? (Value is 0 or 1)
other	Does the location support other activities? (Value is 0 or 1)
college	Does the location support college activities? (Value is 0 or 1)
religion	Does the location support religion activities? (Value is 0 or 1)

TABLE 11. Activity Locations File

Field	Description
hid	Household ID of the person
pid	Person ID of the person
activity_number	Activity Number: Number of the activity in the activity sequence to which it belongs
activity_type	Activity Type: Enumerations used for encoding activity types. 1: Home, 2: Work, 3: Shopping, 4: Other, 5: School, 6: College, 7: Religion
start_time	Start time of the activity in seconds since midnight Sunday/Monday
duration	Duration of the activity in seconds
lid	Location ID of the location where the activity takes place (rlid or alid)

TABLE 12. Activity Location Assignment File

Field	Description
pid1	Person ID 1 of this edge
pid2	Person ID 2 of this edge
lid	Location ID of the location where the contact takes place (rlid or alid)
start_time	Start time of the contact between Person ID 1 and Person ID 2 measured in seconds since midnight of Sunday/Monday
duration	Duration of the contact measured in seconds
activity1	Activity type of Person ID 1 at time of contact, see activity_type in activity location assignment file description
activity2	Activity type of Person ID 2 at time of contact, see activity_type in activity location assignment file description

TABLE 13. Population (Contact) Network File

Field	Description
day	Day: Simulation day
pid	Person ID of the person
disease_state	Disease State: State (S, I or R) of the person on the simulation day

TABLE 14. Disease Outcomes (training data)

Field	Description
pid	Person ID of the person
infected	Boolean field (0, 1) indicating whether an individual is infected during the forecast period

TABLE 15. Disease Outcomes (target data)

APPENDIX C. EVALUATION OF CENTRALIZED EXAMPLE METHODS AND HEURISTIC MODELS

We would like the IRP to have the following property: the performance of a learner is significantly better if it has access to the whole network data than if it can only access the network data of one partition. In this section we present results that compare performance of a centralized baseline, *Aggregate Calibration*, which uses network-based simulations, and a network-based heuristic, *Contact[t]*, under various settings.

- *Aggregate Calibration*: This model is described in 7.3.
- *Random classifier*: This heuristic assigns probability $p(v)$ for each node v by sampling from a uniform distribution $U(0, 1)$.
- *Contact[t]*: This heuristic assigns probability $p(v)$ for each susceptible node v proportional to the total contact duration v has with nodes that were infected during the last t days, right before the prediction period. For any nodes that is or was infected (possibly has recovered) we assign $p(v) = 0$. That is, $p(v) \propto \sum_{u \in I[t]} w_{uv}$, where $I[t]$ is the set of nodes that are infected (in I state) on any day in $[T - t + 1, T]$ and v is susceptible at T.
- *Informed*: This model generates probability predictions by running the same simulation that generates synthetic outbreak data forward for 1000 replicates. The simulation has the same model parameters as in Algorithm 1 and branches only after T. In fact the test data $\{\mathbf{X}(t)|T < t \leq T + \Delta\}$ is sampled from the same distribution as the additional 1000 replicates. This is the best-informed model and is an upper bound of the *Aggregate Calibration* method.

We apply the above four models on the first Virginia instance and evaluate their AUPRC scores. Figure 21 shows their precision-recall curves. Note that for *Contact[t]* we have chosen $t = 1$, i.e., we only consider neighbors of nodes infected on the last day. Not surprisingly, *Informed* outperforms all other models. Interestingly, *Contact[1]*, being a simple heuristic, performs almost as well as *Aggregate Calibration*. This suggests that network data and recent system state are the most important indicators for predicting near future individual risks. We apply three of the models on the second Virginia instance. From Figure 22, we observe similar results.

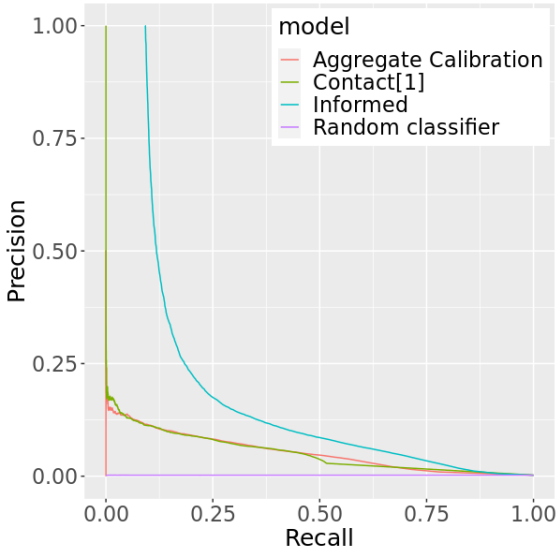


FIGURE 21. Compare performance of Contact[t] with that of Aggregate Calibration model and Random classifier, on the first instance of the Virginia network. AUPRC is 0.0023 for *Random classifier*, 0.0524 for *Aggregate Calibration*, 0.0515 for *Contact[1]*, and 0.1032 for *Informed* model.

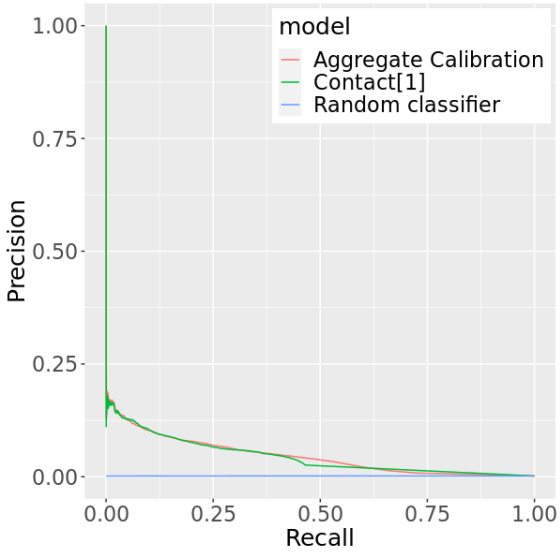


FIGURE 22. Compare performance of Contact[t] with that of Aggregate Calibration model and Random classifier, on the second instance of the Virginia network. AUPRC is 0.0017 for *Random classifier*, 0.0451 for *Aggregate Calibration*, and 0.0441 for *Contact[1]*.

We consider performance of models with a *mismatched* network. Suppose all datasets, including synthetic outbreak data for the second Virginia instance, are given to the learners, except the network. Instead of the first instance of the Virginia network, the learners are given the second instance of the network, which is similar to the first instance in terms of network structural properties, but has a lot of differences at the node level, as discussed in the last section. In Figure 23, we show the performance of the two network-based models when they are given the "wrong" network. As expected, *Contact[1]* has a much lower AUPRC score than in Figure 22. But it seems *Aggregate Calibration* has a similar performance as in Figure 22, suggesting that it is able to capture network dynamics beyond 1-hop contacts. We apply *Contact[1]* to each partition of the Virginia network to study its performance in a federated learning setting. Each partition corresponds to a county in Virginia, and consists of population data for the individuals residing in that county, network data for contacts between individuals from that county, and outbreak data for disease states of individuals from that county. We assume that *Contact[1]* does not have access to any data about any other county, and has to predict infection probability for each individual from the county. It does so by assigning $p(v)$, for each susceptible node v of the county, proportional to the total contact duration v has with nodes from the same county that were infected during the last t days. In Figure 24, we show distribution of the AUPRC scores it achieves on each partition of the first Virginia instance. It seems that *Contact[1]* performance is substantially lower in

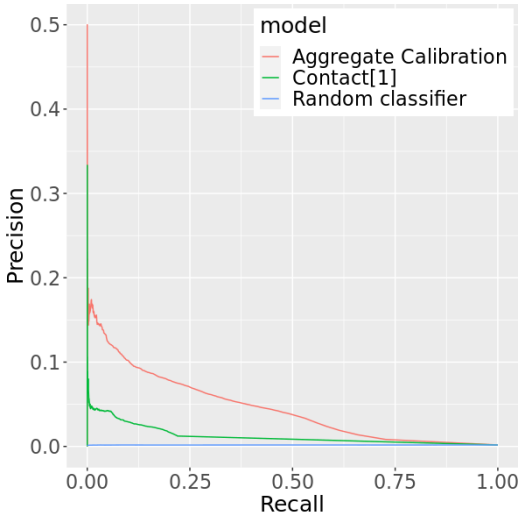


FIGURE 23. Performance of Aggregate Calibration and heuristic models when they use a mismatched Virginia network to make predictions. AUPRC is 0.0017 for *Random classifier*, 0.0449 for *Aggregate Calibration*, and 0.0126 for *Contact[1]*.

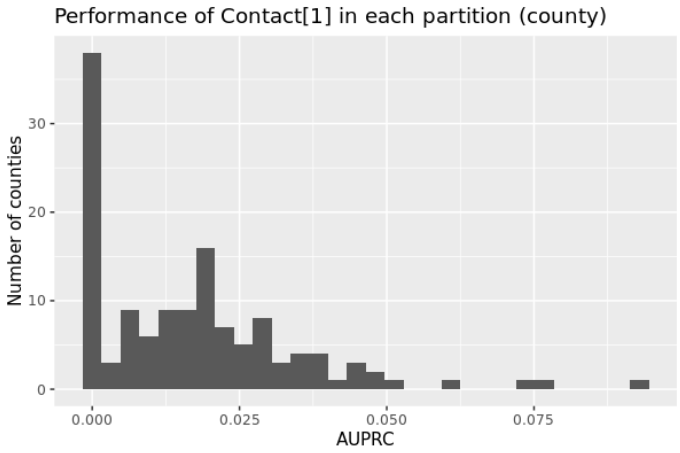


FIGURE 24. Performance of Contact[1] heuristic on partitioned Virginia network. In this setting, Contact[1] can access data only limited to a county, including network and outbreak data. Figure shows histogram of AUPRC score among all Virginia county networks.

most partitions than it is in Figure 21, where it achieves AUPRC=0.0515 with access to the whole Virginia network.