Research Group Social Computing
Department of Informatics
Technical University of Munich

Simulation and mining of social networks to predict individual infection risks

Miriam Anschütz

April 19, 2021

Abstract — The ongoing covid-19 pandemic is a huge challenge for everybody thus scientists and healthcare experts are searching for new methods to get control over it. A common approach is the usage of contact tracing apps that store the user's contacts and notify him in case he had contact with an infected person. In this paper, we propose an extension to these app, a novel approach to predict a continuous risk instead of a binary prediction of normal or increased risk. This approach includes a daily update based on the contact's risk thus it does not need a positive test to identify risky contacts. Based on this model, we investigate the important features of the contacts as well as the required amount of users in the population to yield reliable predictions. This paper is part of a project in which the covid risk spread is simulated among an artificial population and uses this simulation data for its models.

1 Introduction

In recent years, the world has faced many pandemic situations. Not only the latest covid-19 pandemic but also the spread of the HI virus or tuberculosis are examples of such pandemics. In these situations, the government and scientists want to do everything possible to get control of the situation. However, especially for covid, it took some time to find effective countermeasures and the research about this is still going on. Especially for sensitive data, such as health data, ethical and legal issues arise that prohibit using the full data in studies. In this project, we, therefore, simulate an artificial population. We know everything about this population and can use this information without privacy concerns to take a god-like perspective about the disease spread.

The project itself is separated into three parts. The first part is the generation of a social graph, in which the relationship between individuals as well as their home- and workplace is stored. The generation approach for this graph will be explained in this paper. The second part of the project is the simulation of the daily movement of the people in the population as explained in Lopez, 2021. This information about the social structure and the meetings between people is used in the third part to model the disease spread among the population (Lee, 2021).

A common countermeasure for controlling the risk spread is the usage of contact tracing apps that recognize contacts to other people and warn the user in case they met an infected person. In this paper, we propose an extension to the Corona Warn app, the German contact tracing app. In the current version, the app only has a retrospective view that checks if the user's contacts were tested positive. Our version of the risk prediction uses the encountered person's risk to live-update the user's risk and is not dependent on a positive test result. This risk calculation can show a continuous probability of being infected instead of the binary prediction the app currently uses. With this advanced risk calculation, we identify relevant features of the encounters to see which additional user information can help to improve the prediction in real life. Moreover, we discuss the importance of a high app coverage in the population.

This contribution consists of three parts, namely the generation of the social network for the simulation, the advanced risk prediction, and an ethical evaluation of the approach. In the following section, we compare our graph generation and risk generation approach with already existing approaches. After that, we describe our graph approach and provide statistics about the network and the resulting population. Then we explain and evaluate our individual risk prediction approach. The last section discusses the ethical and legal considerations that are relevant to this work. The sections 2.1 and 3 were written in collaboration with Aldi Topalli.

2 Related work

In this section, we compare approaches for random graph generation as well as how these graphs can be used for machine learning predictions, especially in the context of disease spread.

2.1 Random graph generation[1]

The simplest model for a random graph is the model suggested by Erdős and Rényi, 1960. It randomly connects all the nodes in the graph with a probability p. Unfortunately, these graphs lack some properties that are needed to model a real-world graph (Mark E. J. Newman, 2002). The graph model presented by Duncan J Watts and Steven H Strogatz, 1998 is an extension to the ER model as these networks show a small world phenomenon. The model generates n nodes and aligns them into a circle where every node is connected to its k nearest neighbors. Then, each edge is reconnected to another node with a probability p. When p is set to 0, the graph preserves its ordered structure while $p = 1$ creates a completely disordered graph. It has been shown that this model builds graphs with real-world properties M. E. J. Newman, D. J. Watts, and S. H. Strogatz, 2002 while scaling for large networks due to the model's simplicity.

Talaga and Nowak, 2020 report these properties of real-world social networks such as high levels of clustering, positive degree assortativity, short average path lengths (small-world property), and right-skewed but rarely power-law degree distributions. While they do generate a social network, the size of this network is smaller than what would be sufficient to have conclusive observations about various theories, and at the same time, the goal of this work is not the network simulation itself, but rather to prove the importance of a set of specific properties of social networks. Moreover, the work presented in this paper produces a network that contains only one kind of connection between nodes, which is not fine-grained enough when considering the spread of a virus based on the intensity of human contact.

Bojchevski et al., 2018 work on a similar network generation problem that can be transferred to a social network. They use Generative Adversarial Networks (Goodfellow et al., 2014) to train a neural model that produces a network similar but not identical to the input graph. The main drawback of this approach is that it already requires a social network as input data, which is not available to us.

Li et al., 2018 simulate a social network from scratch by using Deep Learning methods. Their networks' resemblance to real-world counterparts (e.g. molecules) is very realistic. The major problem with using this approach for a social network is the model's complexity. The authors use this approach to simulate networks of small to medium size. The time needed to generate a network with 100,000 nodes is unreasonable thus this approach is not applicable for us.

2.2 Predicting the covid-19 pandemic

Machine learning is widely used to cope with the covid-19 pandemic, for example, to predict the patient's mortality based on his clinical features (Yadaw et al., 2020) or biomarkers (Yan et al., 2020) or to forecast the development of the pandemic in the next few days based on time series networks (Chimmula and L. Zhang, 2020). However, these approaches focus on an overview perspective and not on the individual risk prediction as in contact tracing apps.

Many countries have published apps to trace contacts and warn the users if they had contact with or shared location with an infected person. These apps use GPS location data[2], Bluetooth proximity information[3] or all sorts of combination of this data[4]. Besides, a group of researchers from the EPFL, ETH Zürich, and other universities published a framework for privacy-preserving contact tracing based on Bluetooth proximity (Troncoso et al., 2020). This framework was implemented in some of the country's apps already[5]. However, all of these apps use a rule-based approach to update the user's risk based on his encounters and do not use machine learning to train the risk prediction.

We base our work on the German Corona Warn app[6] that traces contacts with Bluetooth technology while ensuring data privacy. Every person that downloads the app gets a random ID that is unique for the person's phone. This ID cannot be used to identify the person using this phone. Whenever this person meets another user of the app, they exchange their random IDs, and the phones store the day, the duration, and distance of this contact. The distance is determined by the damping of the Bluetooth signal from the respective device as determined by the Exposure Notification Framework by

[1] This section was written together with Aldi Topalli

[2] https://govextra.gov.il/ministry-of-health/hamagen-app/download-en/

[3] https://www.health.gov.au/resources/apps-and-tools/covidsafe-app

[4] https://tracing.covid19.govt.nz

[5] https://foph-coronavirus.ch/swisscovid-app/

[6] https://www.coronawarn.app/en/

Google[7] and Apple[8]. Information about encounters that are older than 14 days is deleted [9].

Whenever a user reports a positive test, its ID is added to a publicly available list of positive IDs. Every phone checks if its encountered IDs are in this list and reports a risky contact to its user if it finds a match. The app can display a low or a high risk to its user. Low risk is displayed if no encounter with a positive person exists or if the encounters were very short or far away. Otherwise, the app shows a high risk[10]. This risk calculation only works retrospectively, i.e. the user's risk increases only if one of its encounters was tested positively. It does not include a live update that considers if the person met already has a high infection risk.

2.3 Exploiting graphs for predictions

Rorres et al. (2018) build an animal transfer graph for deer farms in the US to model the movement of animals and find possible infection paths of a deer disease. This disease has an incubation period of up to two years. They identify strongly connected components in this network and analyze the reachability of two farms. Due to the long incubation period of this disease, the reachability helps to find farms that could be infected by an ill animal in one of the farms. This model can easily be transferred to other diseases and contacts between two people instead of an animal transfer.

Another approach that uses the graph structure and properties to propagate prediction through it is the page rank algorithm (Page et al., 1999). It was originally designed to determine the importance of a web page. In this algorithm, a node's rank, i.e. importance, is determined by the ranks of the pages that link to it as well as the number of outgoing links these pages have. Therefore, nodes with extremely many incoming edges or one incoming edge from a very important node are deemed important and get a high rank. This algorithm can also be used in other domains, e.g. ranking authors by citations as done by Ding et al., 2010 or to model epidemic spread as done by Liu et al., 2013.

3 Generation of social network[11]

In this section, we explain the approach to generate an artificial social network.

3.1 Properties of real-world network

Real-world (social) networks have some distinctive properties that distinguish them from random graphs. These properties are a high clustering coefficient, a power-law-like degree distribution, and a small-world phenomenon present.

The clustering coefficient is the proportion of triangles of connected nodes compared to the number of connected triples. It can be interpreted as a kind of community structure thus real-world networks have clear communities, sometimes even only one large one. In contrast, random graphs are mostly loosely connected hence, they have a rather small clustering coefficient (Travers and Milgram, 1977).

Another very descriptive property of social networks is the power-law-like distribution of degrees for some attributes (M. E. J. Newman, 2010). The general formula of a power-law function looks like this:

$$f(x) = a \cdot x^{-k} \tag{1}$$

In terms of degree distribution, this means that having the value of the property, e.g. size of a facility, results in the doubled amount of nodes with this property, thus the distribution is heavy-tailed.

The small-world phenomenon describes the observation that the average distance between any two nodes in the network grows sublinearly, i.e. in $O(\log n)$, to the number of nodes. Usually, this is measured with the average shortest path length between all nodes that should be smaller or equal to 6 (Travers and Milgram, 1977). However, the calculation is in $\mathcal{O}(n^2)$ thus it is infeasible to calculate it for our big network.

3.2 Demographic assumptions

To design a population that is similar to a real population, we need to make demographic assumptions for our model. For this, we have a look at the OECD family database [12] that reports information about family structures and household size. As the mobility data is generated from travel plans in France, we also try to mimic a French population. The average

[7] https://www.google.com/covid19/exposurenotifications/

[8] https://developer.apple.com/documentation/exposurenotification

[9] https://github.com/corona-warn-app/cwa-documentation

[10] https://github.com/corona-warn-app/cwa-documentation/blob/master/cwa-risk-assessment.md

[11] This section was written together with Aldi Topalli

[12] oecd.org/els/family/database.htm

household size in France is 2.38 [13]. A household is defined as people that live in the same house or part of the house and share their costs of living. People in the same household do not necessarily have a relationship other than housemates.

In our model, we also want to mimic family structures. For this, we need to know about the average family size. The OECD reports that a French woman has 1.88 children on average [14]. This means that a family consists of four members on average, two parents and two children. Moreover, usually, there are also grandparents. To compensate for the slightly overestimated number of children (2 instead of 1.88), for families that may not have children yet or for families where some of the grandparents have already died, we only add three instead of four grandparents to the average family. This results in an average family size of 7.

3.3 Suggested approach

We use the Watts-Strogatz model as described in section 2.1 to generat ethe underlying network. In this network, every node represents a person and nodes are connected if the two persons have a long-term relationship. These relationships can be:

- friends
- colleagues
- family
- partner
- housemates

An edge in the network can have multiple labels, i.e. two nodes can be for example friends and colleagues at the same time. We do not allow the connection types family and partner or family and friend at the same time. The family relationship only accounts for family members in a direct line, e.g. parents, grandparents, or siblings but not aunts or cousins. Every person in the network has on average 20 long-term relationships.

In addition, every node has a home facility, i.e. the place the person lives, and a work facility, which could is also used to represent schools and other kinds of facilities where a regular activity takes place. The colleague label is only possible if the connected people work at the same work facility and the housemate label is only possible if the connected people share the same home facility. However, if two people share the same facility, it does not necessarily mean that they have a relation/an edge connecting them. With this design, we account for large living blocks or big companies.

We generate the network with nodes and edges from scratch and assign attributes and labels afterward. Therefore, we never manually add or delete an edge to preserve the network properties.

3.3.1 Label propagation to distribute node properties

The nodes in our network have attributes that describe their place of home and work. These attributes should be consistent with the social structure, e.g. it is more likely that a node shares a home facility with a family member than with a random person. To be persistent with this interpretation, we make use of the label propagation algorithm (Zhu and Ghahramani, 2002) that is based on the assumption that similar nodes should have similar labels. It was originally introduced for semi-supervised approaches and propagates node attributes through the network based on only a few labeled nodes. We receive a list of work and home facilities from the population the mobility group uses but it is also possible to generate completely random places. Each of these predefined facilities is assigned to a random node in the network. With this, we generate our small labeled sample. The algorithm generates a label matrix in the shape of (n_{nodes}, n_{labels}) that represents the label of each node in a one-hot encoding. This matrix is multiplied by the adjacency matrix of the network and normalized until convergence. Finally, we retrieve the optimal label for every node. As the place of work is independent of the home facility, we propagate the two attributes independently.

3.3.2 Resulting algorithm

Putting the network design, the underlying graph model and the propagation of node attributes together, we propose the algorithm as can be seen in figure 1. All parameters can be set in a configuration file and can be updated when designing a different population.

[13] http://www.oecd.org/els/family/SF_1_1_Family_size_and_composition.pdf

[14] http://www.oecd.org/els/family/SF_2_1_Fertility_rates.pdf

```
1. Use a Watts-Strogatz model with a
   reconnection probability $p=0.5$
   and an average number of
   connections $k=20$ to generate a
   random graph.
2. Assign the node attribute home
   facility
  2.1 Take the given facilities from
      the population file and assign
      them to random nodes in the
      network
  2.2 Propagate the node attributes
      through the network with the
      label propagation algorithm
3. Assign the attribute work facility
     with the algorithm from 2.
4. Label the edges, i.e. iterate over
     the existing edges:
  4.1 Edges connecting people from
      the same home facility are
      housemates with a probability of
       0.5.
  4.2 Edges connecting people from
      the same work facilities are
      colleagues.
  4.3 Randomly label edges as family
      with a probability of 0.35.
  4.4 Randomly label non-family edges
       as partners with a probability
      of 0.05.
  4.5 Randomly label non-family edges
       as friends with a probability
      of 0.15.
  4.6 All edges without a label are
      friends.
```

Figure 1 Algorithm to generate long-term social network

3.4 Evaluation

The resulting graph has 118,820 nodes with 42,459 home facilities and 27,639 education/work facilities.

3.4.1 Network properties

Figure 2 shows the degree distribution of the nodes in the network. As designed, the majority of nodes have 20 connections. The clustering coefficient for this network is 0.09 what is rather high for networks of this size. The network is fully connected and has one giant component. As the Watts-Strogatz-model has a small-world phenomenon present by design M. E. J. Newman, D. J. Watts, and S. H. Strogatz (2002), we assume this property for our graph as well. As can be seen in Figures 3 and 4, some of the edge types and node properties show a power-law degree distribution. We can conclude that the generation of a network with real-world properties was successful.

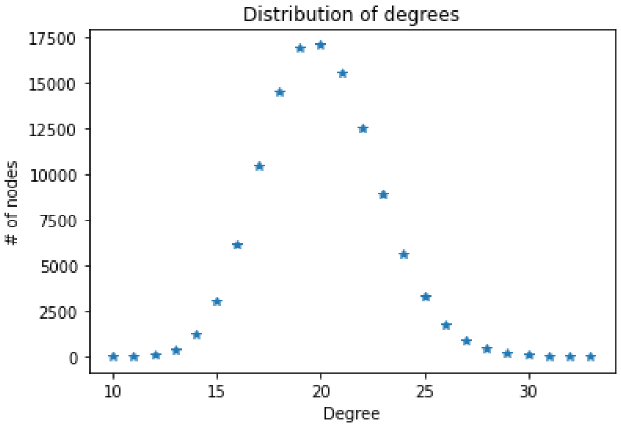

Figure 2 Degree distribution of generated network

3.4.2 Population properties

Figure 3 shows the distribution of the number of nodes that are connected with a specific edge label. We can see that most of the nodes have no or only one partner and only 10% of the population have more. This seems to be reasonable as also in real life some people have more than one regular sexual partner. Also, the friend and family distribution seem reasonable with friends being the most common relationship to other nodes. The colleague relationship peaks at 1-2 colleagues. It is important to mention that in our design, colleagues are employees at the same facility that regularly work together. Therefore, this small number of colleagues is acceptable. When discussing the colleague relationship, it is also interesting to look at the size of

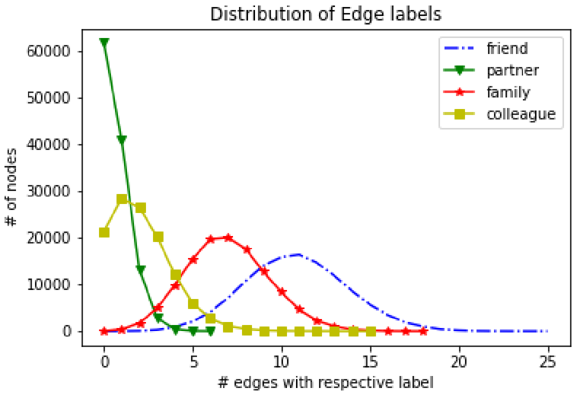

Figure 3 Degree distribution of different edge labels

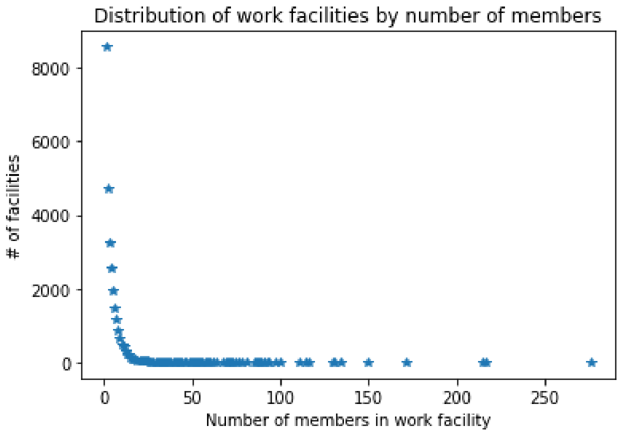

Figure 4 Distribution of work facilities by number of members

the work facilities. Figure 4 shows the distribution of work facilities by their number of employees. We can see that there are many facilities with less than 10 employees and a few facilities with more than 200 workers. This seems to mimic work in real life where most of the people work in small- or medium-sized companies and there are some very big global players.

4 Individual risk prediction

In this section, we describe how we use the simulation data to predict a continuous risk for every individual. This prediction is based on the encounters that person had.

4.1 Dataset

We retrieve data from a 28 day simulation with 105,245 people. In the following, we will discuss which of the output files are relevant for this work and how we use them.

In the person status file, all infection updates of every person are stored. We use the infection status of a person as a ground truth for the person's infection risk, i.e. a person showing COVID symptoms has a very high risk while a healthy person has a low risk. We translate the different infection states to an individual risk based on the mapping in table 1. As it is unlikely that a person notices an infection before showing symptoms, we slightly underestimate the risk of a contagious person by design.

In the events files, we can see all the actions that the people did for every day of the simulation, e.g. taking public transport or entering a building. Moreover, these files contain information about encounters between people. These encounters describe who is meeting whom and which relationship they have as well as the duration and the place they meet at. The relationship two people have comes from the social network described in section 3. The places include home, work or education facilities, public transport and others like leisure places or shops. In our data, we have 16,769,182 encounters between two people.

infection state	risk
none	0.0
infected but not contagious	0.2
contagious	0.5
showing symptoms, seriously sick	1.0
recovered	0.0

Table 1 Different infection states and their deduced risks

The last file contains the infection events, i.e. timestamps and places where an infection transmission happened. An infection has a distinct infector and infected person, thus infections only go in one direction. For every day, we observe ten infections that are not based on an encounter but random infections from the model itself to start or accelerate the risk spread.

4.2 Personal encounter network

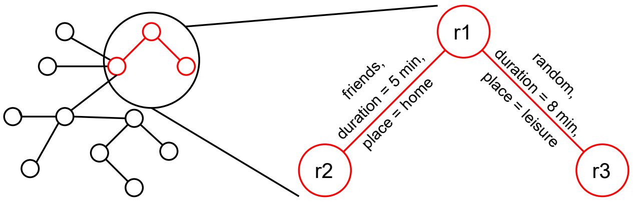
r1
r2
r3
friends, duration = 5 min, place = home
random, duration = 8 min, place = leisure

Figure 5 Encounter network of one day and a closeup of three nodes in this network

All encounters of one day are organized in an encounter network as shown in figure 5. Every node in this network is a single person, and an edge means that these two people met on this respective day. As shown in the closeup, the edges are attributed by the encounter features. Every node has a rank, i.e. its individual infection risk. Due to the decentralized approach, we do not build a large network but a single one for every person. In these person networks, the person's encounters are centered around the respective person. The closeup shows this individual network for node 1 with risk $r1$. Based on these networks, we develop a page rank inspired propagation method to calculate the individual infection risk of every person based on his daily encounters. We can interpret the current infection risk as the rank of a node and the features of the encounters weight the influence of the infection update.

The first step for this approach is to train a model that determines the weight of every encounter. Therefore, we try to match every encounter with the infection events and determine which encounters were infectious and which were not. Every infection event is uniquely identified by the two involved people, the id of the facility in which they meet, and the day of the simulation. The resulting dataset has 5,284 infectious encounters. As this is only a small subset of all encounters, we use a balanced version of the dataset to train our model.

As one of the goals of this work is to extend the risk prediction to a continuous scale, we use a regression model to calculate the weight of each encounter. Although the ground truth from the simulation is categorical, we use it to train the continuous model, i.e. we allow the model to determine if an encounter was risky even if it did not result in an infection. The regression itself uses these input features: relationship of the two people meeting, duration of the encounter, infection risk of the person met, and the facility type they meet in. The original warn app uses the distance between people. However, we do not get this information from our data thus we use the type of facility instead. We assume that it is possible to deduct information about the distance from the type. For example, people meeting at a home facility are closer together than people at a shop.

4.2.1 Evaluation

We evaluate the regression's performance against a test set. Figure 6 scatters the regression's predictions for infectious and non-infectious encounters. We can see that we only have a few false positives and no false negatives thus the prediction works very well.

To investigate the relevant features for this prediction, we leave out one of the features at a time and re-evaluate the performance against our test set. The differences in the absolute errors are shown in table 2. The most important feature is the encountered person's infection risk. If we leave out this feature, the interquartile range for noninfectious encounters ranges from 0.1 to 0.9, i.e. the model is not able to distinguish between the two encounter types at all. However, if we only use the person's infection risk for prediction, the results are slightly worse thus the other features have a least a small impact.

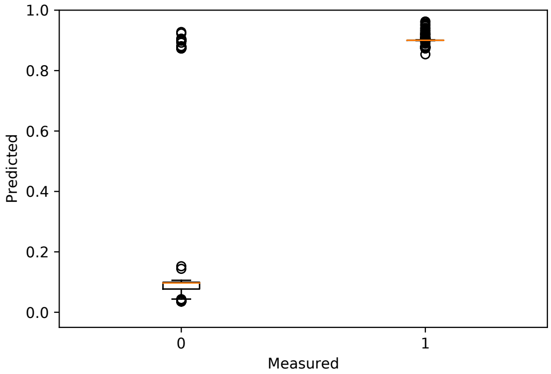

Figure 6 Predicted infection probability by encounter model

Feature left out	Mean absolute error
None	0.12
relationship	0.12
encountered person's infect risk	0.35
facility type	0.12
duration	0.12
all except person's infect risk	0.13

Table 2 Mean absolute error for different features used to train the encounter weights

4.3 14 day risk history

To determine the daily risk of a person p based on his n encounters of that day, we predict the weight w for every encounter based on the approach from the previous section. Then, we sum over all encounters and limit this sum to 1 to represent a proper probability. This results in a modified page rank update formula as shown in equation 2.

$$r_p = max\left(1, \sum_{i=1}^{n} w_i\right) \tag{2}$$

For our example from figure 5, we sum the predicted weight of the encounters with nodes 2 and 3 to get the encounter risk of node 1.

As the infection risk is dependent on multiple days, we store the encounters and daily risks of the last 14 days, similar to the original version of the app. With this history, we train a regression to predict the current infection risk based on encounter risks. We use the person's status of the next day as ground truth for this training.

4.3.1 Evaluation

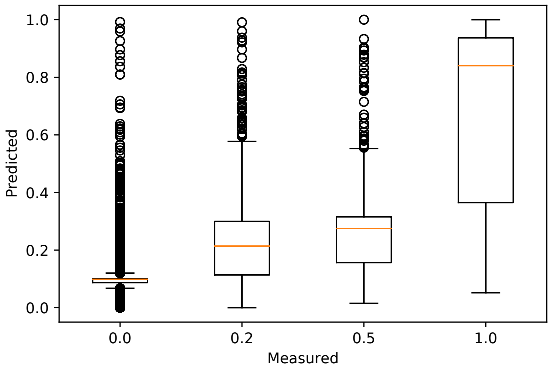

Figure 7 Predicting the infection risk based of a 14 day history of encounter risks

Figure 7 shows the predicted values for each of the ground truth labels. Again, we manually set values that are smaller than 0 to 0 and values that are higher than 1 to 1. The mean absolute error for this prediction is 0.14. Especially the predictions for the 0.5 risk is far too low, i.e. the model is not able to distinguish between risks for being infected and being contagious. To improve this prediction, we modify the ground truth to a binary case, where we merge the 0.2, 0.5, and 1.0 classes into one class with increased risk. The results of this prediction are shown in figure 8 and the mean absolute error is 0.21. As we wanted to build a continuous model, we allow the model to predict a higher or lower risk than the actual infection state from our simulation. Therefore, we consider this continuous risk prediction model as successful.

4.3.2 App usage in population

To measure the impact of people not using the app, we randomly sample a subset of our population as app users. We filter the encounter list by these users, i.e. we only work with encounters that are between two users of the app. The person status updates from the simulation stay the same as the people actually met, only the app did not recognize these encounters. We predict the 14-day risk with the model explained in the previous section. Figure 9 compares the predicted risks when only 50% or 75% of the population is using the app with full coverage. For infected people, the predictions get more accurate the more people are using the app. Especially for the 50% usage, the false-negative rate is extremely high. For healthy people, the 50% usage outperforms the 75% usage but this could also be due to randomly sampling the users. In any case, the 100% usage is the most accurate by far. Therefore, it is very important that many people use the app, e.g. by making it mandatory.

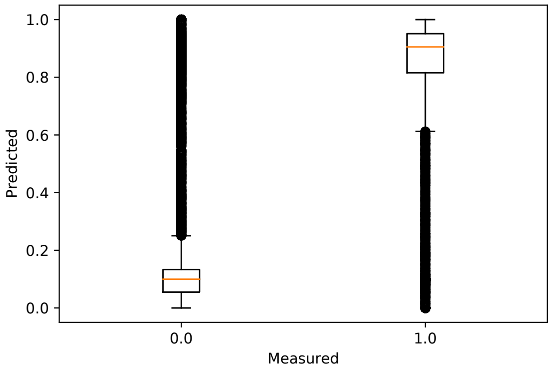

Figure 8 Infection risk prediction with simplified ground truth labels

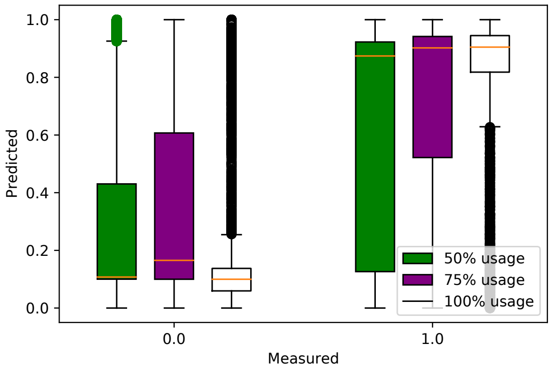

Figure 9 14 day risk prediction for different app coverage

5 Ethical considerations

Individual risk prediction is a very sensitive task that requires some ethical consideration and discussion to consider when designing contact tracing apps. In the following section, we will discuss four ethical and legal aspects: privacy, justice, transparency, and reliability. These aspects will be evaluated against a possible real-life application of the Corona Warn app's extension and not the artificial population we utilized to design it.

5.1 Privacy considerations

Data privacy is probably the most delicate topic when designing contact tracing apps, at least in Germany. Many people do not use these apps as they fear being tracked by the government. The German Corona Warn app has published its key aspects of data privacy that it ensures to the user [15]. As in the original app, the approach suggested in this paper randomizes the user and only stores and communicates a random ID of the installation. Moreover, the approach is decentralized which means that only the user knows about all the encounters and his personal risk. This encounter data will be deleted after 14 days. However, in this paper, we have to use the facility type instead of the distance of encounter due to the unavailable distance data. With the history of encounters and places, it would be possible to generate a movement profile from the user. However, we consider this to be a neglectable problem as this data is only stored on the user's device that probably already has access to concrete locations of the user and does not need the contact tracing app to build such a profile.

5.2 Justice considerations

One of the goals of a contact tracing app is that users change their behavior based on the risk that they have, e.g. reduce their contacts when they already have a high risk or proactively get a test. However, this may also lead to discrimination of people with a high risk in the app, e.g. when they are not allowed to enter specific buildings due to their high risk. Especially users with many encounters, e.g. at work, will have a higher risk by design, maybe although they are wearing face masks at work.

Even though the app does only store the day of the encounter and not the exact timestamp, for some users it may still be possible to infer whom the user was meeting at a specific encounter. This may be an issue when users start blaming other users for transferring their high risk or blaming them for harming other people.

5.3 Transparency considerations

Transparency is extremely important, especially when we want to deduce information from the risk in the app. The algorithm design is rather simple with two support vector regressions and a page rank inspired update formula. In the previous section, we have seen which features are important for the two regressions thus the model is transparent about the data it uses for its prediction. However, this decentralized approach lacks an overview of the situation. Every user only knows about his risk thus the government has no chance to identify high-risk areas. From a privacy point of view, this is desirable but it also makes the risk spread among the population less transparent.

5.4 Reliability considerations

A big issue with the original app is that users are not forced to enter their positive test result in the app, i.e. there may be undetected high-risk encounters. In our version of the app, the risk update does not require manual work thus the correctness of the risks should be higher. However, the app is still vulnerable to deliberate misinformation by the user, e.g. reinstalling the app in case of high risk. This reinstallation generates a new random ID thus the user does not have a high risk anymore. Moreover, users could fake a high risk or a positive test to frighten other users or to make them reduce their contacts. Even without malicious intentions, the app can only consider encounters that it recognizes, i.e. when a user does not take his phone with him or puts it too far away, the app can not properly update the user's risk.

With the decentralized approach, all calculations can be run on the user's phones thus no internet connection is necessary. Even with the original app, the phones only need to query the positive IDs once a day. The models are very simple thus they should run on any common device without using much power, especially as the updates are only calculated once a day. This makes the app very reliable and insusceptible to crashes.

The last reliability concern is that all models were designed and trained based on our artificial population. Although we tried to build this population as

[15] https://github.com/corona-warn-app/cwa-documentation/blob/master/pruefsteine.md

close to real-life as possible, there may still be mistakes in the simulation or the models, i.e. one would have to evaluate them against a real population.

Putting things together, the approach suggested in this paper puts the strongest focus on data privacy concerns and therefore has to suffer from some drawbacks in reliability and transparency. Still, this is a valuable extension to the original app, that can improve it to some degree. Also, the strong focus on data privacy makes it likelier that many people actually use the app. As shown in the previous section, a high number of users in the population is certainly the most important goal and it can compensate for some of the small issues mentioned.

6 Conclusion and future work

In this paper, we presented a simple algorithm to generate an offline social network. This network has real-world properties considering the network structure as well as the population it models. We use this network to build an extension to the German corona warn app that can predict a continuous risk based on daily encounters. As we were going from categorical ground truth to a continuous prediction, the accuracy of our models is not perfect, yet they perform very well. We have seen that the encounters do not have a specific feature that makes them infectious. Moreover, we have shown how important a high app coverage among the population is to yield the most accurate predictions.

The next steps would be to include more data in the social network, e.g. the age, and to extend it to more complex scenarios such as unemployed people or people having more than one work/education facility. Considering the machine learning approach, the original app and our approach should be merged to have a daily update together with the validation by positive test results. Moreover, users should be able to enter a negative test result into the app to reset their risk.

References

Bojchevski, Aleksandar et al. (2018). *NetGAN: Generating Graphs via Random Walks*. arXiv: 1803.00816 [stat.ML].

Chimmula, Vinay Kumar Reddy and Lei Zhang (2020). “Time series forecasting of COVID-19 transmission in Canada using LSTM networks”. In: *Chaos, Solitons & Fractals* 135, p. 109864. ISSN: 0960-0779.

Ding, Ying et al. (2010). *PageRank for ranking authors in co-citation networks*. arXiv: 1012.4872 [cs.DL].

Erdős, Paul and Alfréd Rényi (1960). “On the evolution of random graphs”. In: *Publ. Math. Inst. Hung. Acad. Sci* 5.1, pp. 17–60.

Goodfellow, Ian J. et al. (2014). *Generative Adversarial Networks*. arXiv: 1406.2661 [stat.ML].

Lee, Soh-Yee (2021). “Ethical AI for Pandemic Management”. MA thesis. Technical University of Munich.

Li, Yujia et al. (2018). *Learning Deep Generative Models of Graphs*. arXiv: 1803.03324 [cs.LG].

Liu, Z. et al. (2013). “A Parallel IRAM Algorithm to Compute PageRank for Modeling Epidemic Spread”. In: *2013 25th International Symposium on Computer Architecture and High Performance Computing*, pp. 120–127.

Lopez, Maria (2021). “Ethical Machine Learning for Pandemic Control”. MA thesis. Technical University of Munich.

Newman, M. E. J. (2010). *Networks: an introduction*. Oxford; New York: Oxford University Press. ISBN: 978-0198805090.

Newman, M. E. J., D. J. Watts, and S. H. Strogatz (2002). “Random graph models of social networks”. In: *Proceedings of the National Academy of Sciences* 99.suppl 1, pp. 2566–2572. ISSN: 0027-8424. DOI: 10.1073/pnas.012582999.

Newman, Mark E. J. (2002). “Random graphs as models of networks”. In: *Handbook of Graphs and Networks*. John Wiley & Sons, Ltd. Chap. 2, pp. 35–68. ISBN: 9783527602759.

Page, Lawrence et al. (1999). *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-66. Previous number = SIDL-WP-1999-0120. Stanford InfoLab.

Rorres, Chris et al. (2018). “Contact tracing for the control of infectious disease epidemics: Chronic Wasting Disease in deer farms”. In: *Epidemics* 23, pp. 71–75. ISSN: 1755-4365.

Talaga, Szymon and Andrzej Nowak (2020). “Homophily as a Process Generating Social Networks: Insights from Social Distance Attachment Model”. In: *Journal of Artificial Societies and So-*

cial Simulation 23.2. ISSN: 1460-7425. DOI: 10.18564/jasss.4252.

Travers, Jeffrey and Stanley Milgram (1977). “An experimental study of the small world problem”. In: *Social Networks*. Elsevier, pp. 179–197.

Troncoso, Carmela et al. (2020). *Decentralized Privacy-Preserving Proximity Tracing*. arXiv: 2005.12273 [cs.CR].

Watts, Duncan J and Steven H Strogatz (1998). “Collective dynamics of ‘small-world’networks”. In: *nature* 393.6684, pp. 440–442.

Yadaw, Arjun S et al. (2020). “Clinical features of COVID-19 mortality: development and validation of a clinical prediction model”. In: *The Lancet Digital Health* 2.10, e516–e525. ISSN: 2589-7500.

Yan, Li et al. (2020). “An interpretable mortality prediction model for COVID-19 patients”. In: *Nature machine intelligence* 2.5, pp. 283–288.

Zhu, Xiaojin and Zoubin Ghahramani (2002). “Learning from labeled and unlabeled data with label propagation”. In: