LUISS

Department
of BUSINESS AND MANAGEMENT

Bachelor's Degree in Management and Computer Science

Course of Artificial Intelligence and Machine Learning

Applying Reinforcement Learning to optimize Vaccine Distribution:

an example of how Artificial Intelligence can help address Social Issues

---

Supervisor:
Prof. Giuseppe
Francesco Italiano

Candidate:
Francesco Redaelli
ID No. 235301

Academic Year: 2020/2021

Contents

List of Figures

List of Tables

List of Abbreviations

AI	**A**rtificial **I**ntelligence
DP	**D**ynamic **P**rogramming
DQN	**D**eep **Q** **N**etwork
MDP	**M**arkov **D**ecision **P**rocess
ML	**M**achine **L**earning
RL	**R**einforcement **L**earning
SGD	**S**tochastic **G**radient **D**escent
SIR	**S**usceptible-**I**nfected-**R**emoved
SNA	**S**ocial **N**etwork **A**nalysis
SOCRATES	**SO**cial **C**ontact **RATES**

Introduction

"Can machines think?" was what the well-known computer scientist Alan M. Turing wondered in the middle of the last century, paving the way for the exploration of the field of study that goes today under the name of *Artificial Intelligence* (AI), which can be briefly defined as *intelligence exhibited by machines or software*. AI has undoubtedly been among the main drivers of theoretical and technological advancement during the last decades, with a wide range of applications across heterogeneous fields, and it promises to be of the utmost importance in the near future as well.

Nowadays, after more than 70 years of research in the field and innovation, it is safe to state that the focal question should not be whether machines *are actually able* to formulate thoughts or take decisions, but rather *how well* they can do it. Machines have already proved capable of achieving human-like performance in several tasks. However, the concept of mimicking in itself embeds, in fact, an intrinsic limit. From this perspective, the *"artificial"* adjective would assume a negative connotation, in opposition to the *"natural"* one, identifying human intelligence. Anyway, the idea that AI represents just a mere simulation of human thinking seems to be outdated, since AI has proved itself able to play a crucial role in addressing tasks that even humans fail to attain flawless results in. The Machine Learning (ML) technique known as Reinforcement Learning (RL) enabled the achievement of super-human performance in many application fields, with the AlphaGo computer program defeating a Go world champion being only one of several significant examples.

As a response to the pandemic of SARS-CoV-2, a vaccination campaign has been carried out in Italy starting in January 2021. During the last months, the level of efficiency of the adopted strategic plan, which had to take into account both the limited amount of available vaccine doses and the maximum achievable daily rate of administration, has been questioned extensively. Computational models of infection spread have been largely employed in the past in order to forecast outbreak severity and test different intervention strategies on a huge number of stochastic simulations. The plan of action finally selected was usually the one providing the best expected performance, averaged over all the scenarios. However, when dealing with a real outbreak, only one of the possible situations occurs in the end, with the set of final outcomes shrinking constantly according to the observed realization at each time step; there is no guarantee for the intervention strategy that performed best on average to be also the optimal one for the final realized scenario (Probert et al., 2019).

The present work suggests an approach to optimize vaccine distribution, under supply constraints. The project has been developed starting from the implementation of a model of infectious disease spreading over a graph, which reproduces interconnections among people, labelled by age.

At first, the social network graphs and the epidemic model have been built accordingly to publicly available Italian social contact data and COVID-19 statistics. In an effort to reproduce such a complex environment, simplifying assumptions had to be adopted.

In the second phase, an RL algorithm has been used. The RL agent interacted with the simulated environment through a process of trial-and-error learning, with the final goal of minimizing the expected cumulative number of deaths.

When the agent is allowed to vaccinate only people in a target age group,

the RL model, trained on graphs of different sizes - 100 and 500 nodes - exhibits superior performance with respect to a naive implementation of the same policy. Furthermore, if the constraint is relaxed and the agent is free to choose among all the nodes, the obtained outcomes suggest an additional improvement in terms of number of prevented deaths. The aforementioned results could indicate a real chance for humans to actually adopt an efficient AI-designed intervention strategy; from a general perspective, this seems to represent a new opportunity for human beings of tackling social issues.

A possible practical implementation of the discussed approach for the optimization of vaccine distribution, together with some open questions, suitable for future analysis, is proposed at the end.

Chapter 1

The Epidemic Model

1.1 The Social Network

1.1.1 Introduction to Graphs

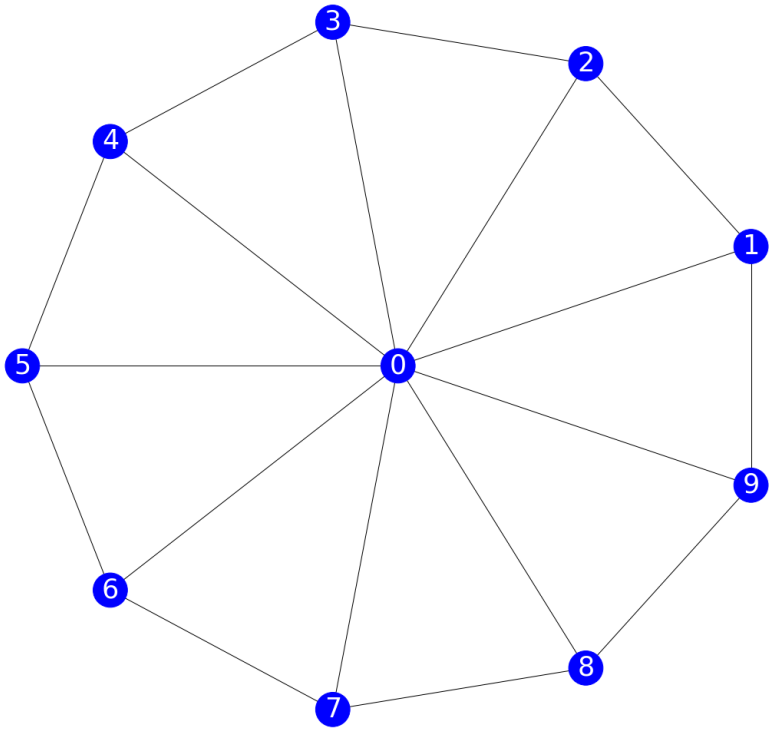
A social network refers to a defined set of social actors – which may include individuals, organizations, or other entities – and the social relationships that connect them to each other in a larger structure (Cornwell and Schafer, 2016). Social networks are naturally modelled as graphs.

A graph $(N; g)$ consists of a set of nodes $N = \{1, \ldots, n\}$ and a real-valued $n \times n$ matrix, g, where g_{ij} represents the relation between i and j. This matrix is often referred to as the *adjacency matrix*, as it lists which nodes are linked to each other, or in other words which nodes are adjacent to one another (Jackson, 2008).

A network is *directed* if it is possible that $g_{ij} \neq g_{ji}$, i.e. the graph can be defined in terms of ordered pairs of vertices in N, while it is *undirected* if it is required that $g_{ij} = g_{ji}$ for all nodes i and j, and the matrix g is therefore symmetric.

The graph is referred to as a *weighted* graph if the entries of g assume more than 2 values, in order to represent the relevance of the link; otherwise, it is said to be *unweighted*, and g_{ij} can be either 0 or 1, indicating disconnection or connection respectively.

Graphs without any self-link (connecting a node to itself), and without multiple links between two nodes, take the name of *simple* graphs. If self-links and multiple links are permitted, the resulting structure is called a *multigraph*. FIGURE 1.1 shows a *wheel graph* as an example of a *simple*, *undirected*, *unweighted* graph, and the corresponding *adjacency matrix*.



$$\begin{pmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}$$

FIGURE 1.1: *An example of a wheel graph (top) and its adjacency matrix (bottom)*

1.1.2 Social Contact Rates (SOCRATES) Data Tool

The statistical data required to build the social network model has been collected with the help of the Social Contact Rates (SOCRATES) Data Tool, an online interactive tool allowing to retrieve social contacts matrices from open-source data, with the aim of informing COVID-19 mitigation modelling (Willem et al., 2020).

Through the user interface, data can be filtered by country, according to:

- *Age Breaks* (Age groups)
- *Type of day*
- *Contact Duration*
- *Contact Intensity*
- *Gender*

1.1.3 Statistical characteristics of the Italian population

The present analysis is based on the POLYMOD dataset (Mossong et al., 2008). POLYMOD information on social contacts has been obtained using cross-sectional surveys conducted by different commercial companies or public health institutes. Survey participants have been recruited so as to be broadly representative of the whole population. Italian data was collected in May 2006.

In order to account for different habits, lifestyles and risk levels for severe illness when contracting COVID-19, while still preserving model simplicity, the population has been modelled into 3 age groups (TABLE 1.1).

Age	Age Group
[0,35)	Youth
[35,65)	Adult
65+	Senior

TABLE 1.1: *Categorization of the population by age*

POLYMOD data has been filtered, via the SOCRATES tool, according to the following parameters:

- *Age Breaks*: 0, 35, 65
- *Type of day*: All contacts
- *Contact Duration*: More than 15 minutes
- *Contact Intensity*: Physical contacts
- *Gender*: All

TABLE 1.2 and FIGURE 1.2 show the population distribution by age group.

Age Group	Population	Proportion
[0,35)	22571554	0.3872870
[35,65)	24251709	0.4161154
65+	11457946	0.1965976

TABLE 1.2: *Population by age group*

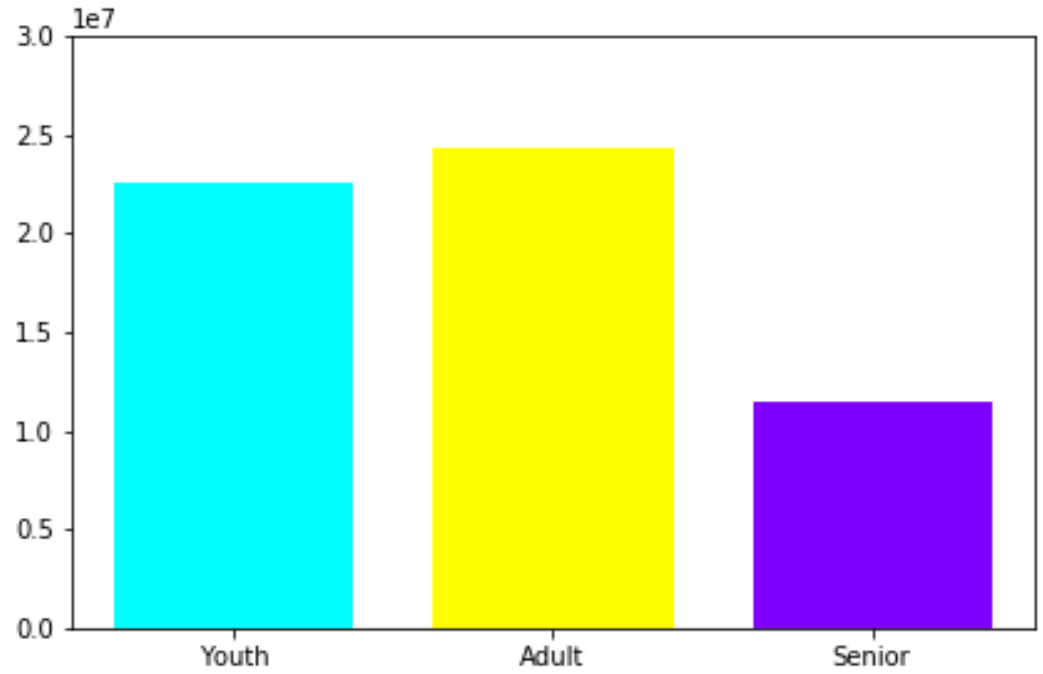


FIGURE 1.2: *Bar Chart - Population by age group*

In an effort to model the social behaviours strongly affecting the spread of COVID-19, social contacts have been considered. In FIGURE 1.3, each entry cm_{ij} of the *Contacts Matrix*, CM, represents the average number of daily contacts of every individual of the age group j with the members of the age group i.

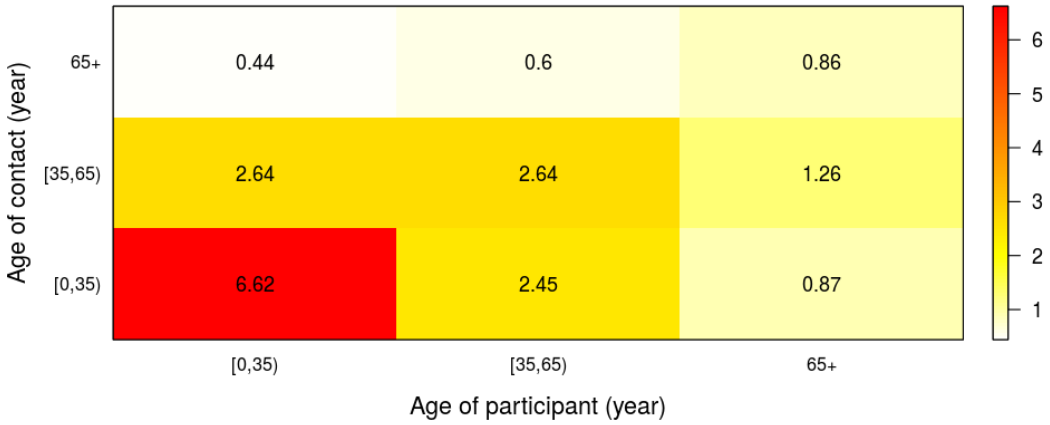


FIGURE 1.3: *POLYMOD Contacts Matrix for the Italian population*

1.1.4 A representative random graph

POLYMOD data has been used to develop a system for the generation of social random graphs: each node is assigned to an age group, and each pair of nodes develops a link, independently of the others, with probabilities evaluated from data, so as to reproduce Italian characteristics in terms of population distribution by age group and contacts among them.

The social network model has been implemented in the Python programming language using the NetworkX library (Hagberg, Schult, and Swart, 2008). For visualization purposes, the Matplotlib library has been used (Hunter, 2007).

Taking as input the total number of nodes N, the system generates a *simple*, *undirected*, *unweighted* graph, which represents one of the possible probabilistic realizations.

FIGURE 1.4 shows an example of an output with $N = 100$.

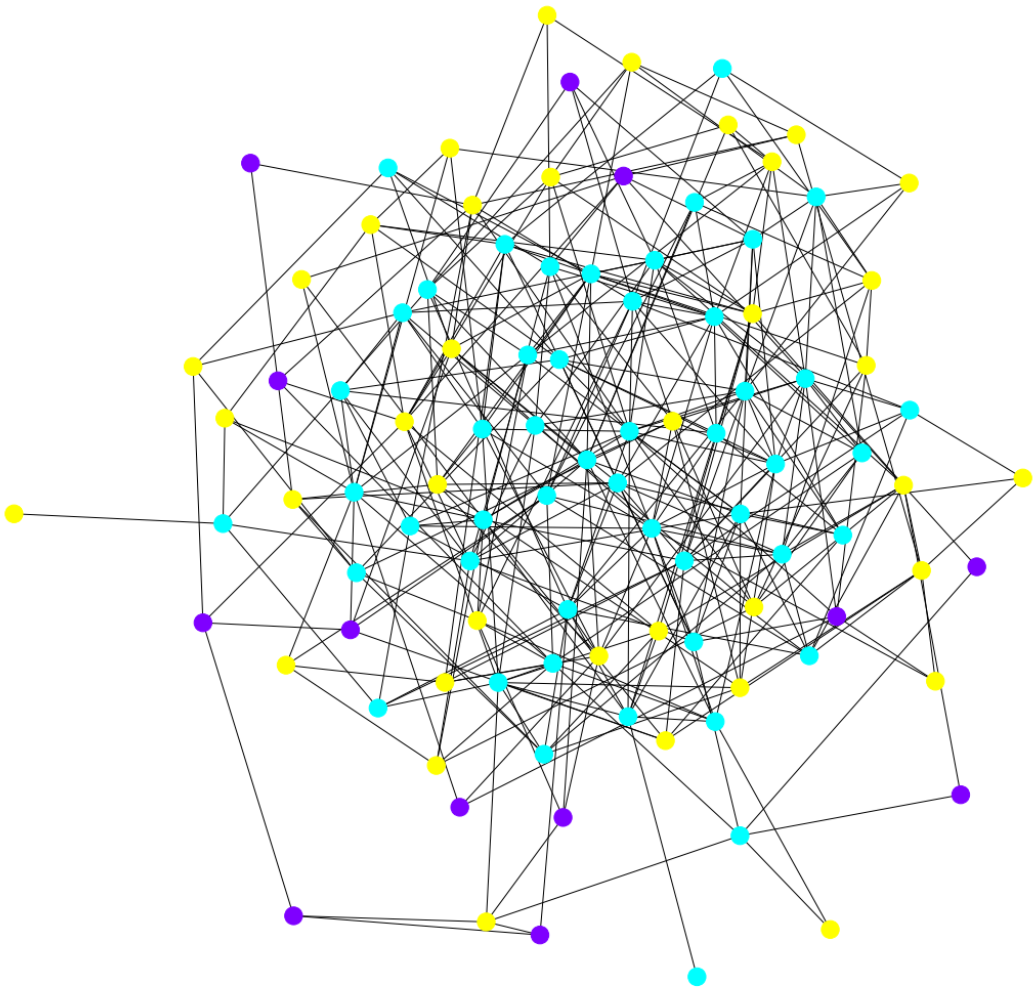


FIGURE 1.4: *A representative random graph with N = 100. Light Blue, Yellow and Violet colors represent respectively the Youth, Adult and Senior age group*

1.2 The Model for Epidemic Spreading

A probabilistic model of infectious disease spreading on networks has been developed and implemented. It is a low complexity model that inherits the classical Susceptible-Infected-Removed (SIR) framework. In an effort to reproduce such a complex environment, simplifying assumptions had to be adopted. The main characteristics of the model are listed below.

- The model considers only 4 statuses of infection: *Susceptible, Infected, Recovered/Vaccinated, Dead*. No distinction is made between diagnosed and non-diagnosed *Infected* people (i.e. no control strategy is adopted in case of infection) and between *Recovered* and *Vaccinated* individuals, both assumed not to be infectable.

- The temporal resolution of the model is a day, i.e. the status of the nodes is updated daily, simultaneously.

- There is no incubation period for the disease.

- Each *Infected* node infects every *Susceptible* neighbor (i.e. node it is directly linked with), independently of the other infections, with probability p_{SI}.

  According to this assumption, at each time step every *Susceptible* node will become *Infected* with probability i_h, which is a function of the number h of its *Infected* neighbors. It is enough that one single node manages to successfully infect the *Susceptible* target for it to change its status.

  The number of nodes infecting the target individual, X, is a binomial random variable; its probability mass function with parameters n and p is given by (Ross, 2004):

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k} \qquad k = 0, 1, \ldots, n$$

Hence:

$$i_h = P(X > 0) = 1 - P(X = 0) = 1 - \binom{h}{0} p_{SI}^0 (1 - p_{SI})^{h-0} = 1 - (1 - p_{SI})^h$$

where p_{SI} has been set equal to 0.1, the value corresponding to the estimated probability of contagion for a physical interaction at a distance of about 1m, according to (Agrawal and Bhardwaj, 2021).

- Each *Infected* node dies with probability d_a. For every age group a, according to the data from (Stewart, 2021) (FIGURE 1.5), a mean death probability has been calculated as a weighted average of the death rates, with weights given by the percentage of population in each age bracket.

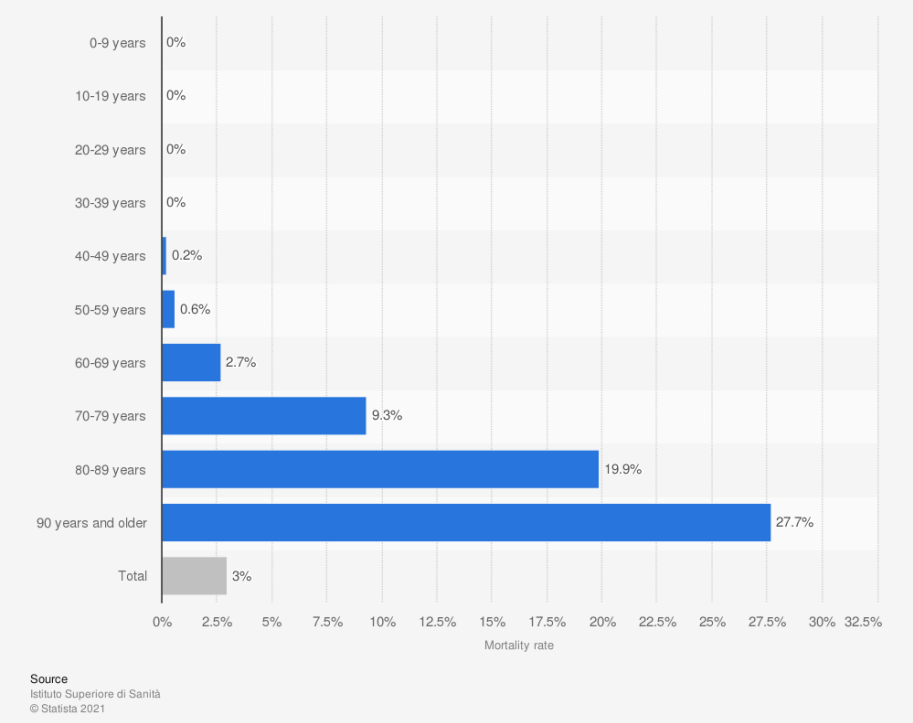


FIGURE 1.5: *COVID-19 death rate in Italy, by age bracket*

Furthermore, a mean recovery probability has been computed as (1 - {mean death probability}). To obtain d_a and r_a, these probabilities have been divided by the number of days for which COVID-19 symptoms last on average, supposed to be 15 days. In this way, at each time step, an *Infected* node is assumed to die or recover with the same probability, d_a and r_a respectively (TABLE 1.3).

	Youth	Adult	Senior
d_a	0%	0.004%	0.72%
r_a	6.7%	6.6%	6%

TABLE 1.3: *Death & Recovery probability distributions by age group*

- At each time step, the status of a node is updated according to the probability values shown in TABLE 1.4.

	Susceptible	Infected	Recovered	Dead
Susceptible	1-i_h	i_h	0	0
Infected	0	1-r_a-d_a	r_a	d_a
Recovered	0	0	1	0
Dead	0	0	0	1

TABLE 1.4: *The probability distributions for status update. Each entry p_{ij} represents the probability that a node having status i at time t, will have status j at time $t+1$.*

The settings of the epidemic model are thought to generate a disease spreading consistent with a *"worst-case scenario"*.

1.3 Infection Spreading Example

The model for infectious disease spreading has been implemented using the NDlib library (Rossetti et al., 2018). FIGURE 1.6 and FIGURE 1.7 show an example of the spreading caused by 1 initially *Infected* node during a week.

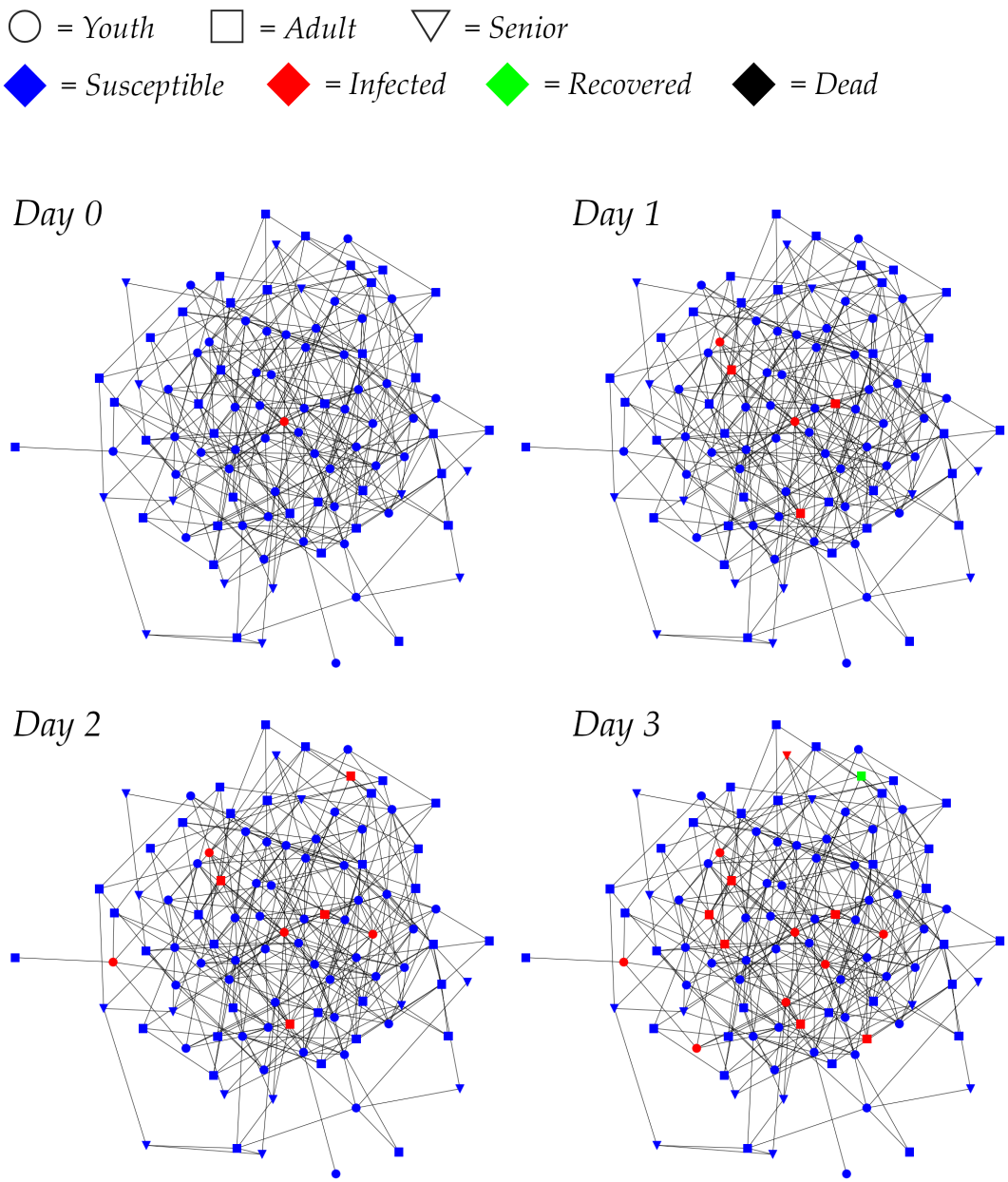


FIGURE 1.6: *Simulation of infection spreading (Day 0 - 3)*

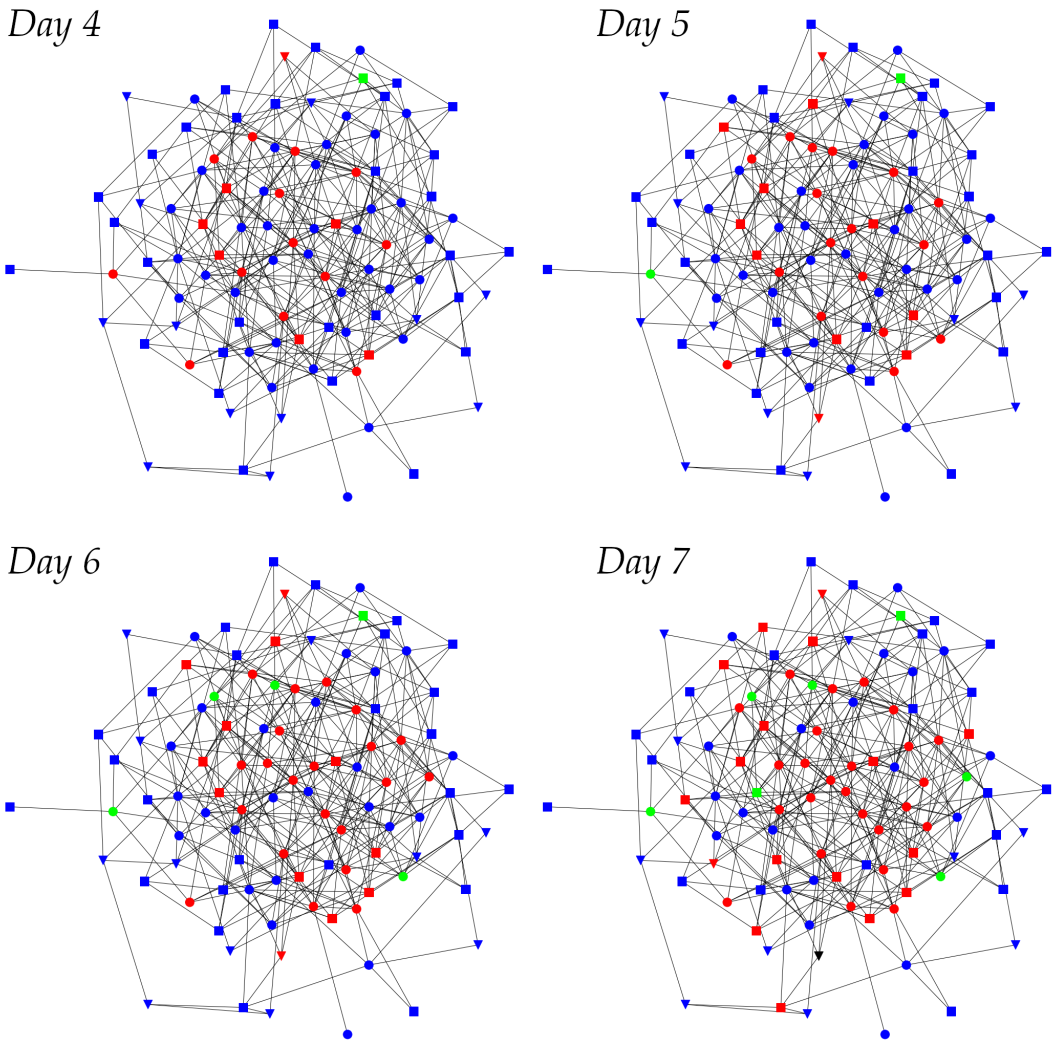


FIGURE 1.7: *Simulation of infection spreading (Day 4 - 7)*

Chapter 2

The Reinforcement Learning Model

2.1 Introduction to Reinforcement Learning

Reinforcement Learning is defined as the science of learning to make decisions from interaction. In a nutshell, RL refers to a trial-and-error learning process, during which the *agent* should discover a good *policy* that maximizes a *reward* by interacting with the *environment* (Madani, 2020). It is therefore an active ML type of learning, that allows the agent to improve his performance without examples of optimal behaviour. Some of the fundamental concepts of RL are described below.

- *Agent*: the agent is a goal-directed actor who interacts with the environment for a fixed number of time steps, taking actions picked from a given action space. Interactions are usually sequential; previous choices can therefore affect future ones.

- *Environment*: the environment encompasses everything the agent can interact with. At each time step t, the environment characteristics are defined by its internal *state* S_t. Depending on the specific problem framework, the agent may or may not be aware of the whole S_t. When

it cannot determine the state of the system at every time, the environment is said to be *partially observable*. Instead, if the agent is able to see the full state, the environment is referred to as *fully observable*.

- *Reward*: the reward R_t is a scalar feedback signal informing the agent about the quality of his actions; a value greater/smaller than 0 indicates a positive/negative feedback respectively. The goal of the agent is to maximize the (cumulative) reward $\sum R_t$, which takes the name of *return*. Reward signals could be either *dense*, meaning that they are provided at almost every time step, or *sparse*, if only certain events cause the signal to be sent to the agent.

When the agent is able to observe to whole environment state (i.e. in a *fully observable* environment), it is in a *Markov decision process* (MDP), defined as follows:

the process $Y = \{Y_t; t \geq 0\}$ with finite state space E is a *Markov process* if the following holds for all $j \in E$ and $t, s \geq 0$

$$Pr\{Y_{t+s} = j | Y_u; u \leq t\} = Pr\{Y_{t+s} = j | Y_t\}$$

In other words, the probability of the next state, conditioned on the current state, is equal to the probability of the next state, conditioned on all the previous states. The key Markov property can therefore be described with the sentence *"The future is independent of the past given the present"* (Feldman and Valdez-Flores, 2010).

A Markov state contains all useful information from the history. In RL settings, this means that *once the current environment state is know*, *the history* (made up of all the previous states) *can be ignored*. When the environment is *fully observable*, the present state completely characterizes the process.

During the interaction between the agent and the environment, at each time step t:

- the agent receives an *observation* O_t (and possibly a *reward* R_t), and generates an *action* A_t
- the environment receives the *action* A_t and produces an *observation* O_{t+1} (and possibly a *reward* R_{t+1})

Combining an *exploitation* approach (taking advantage of the best known option) and an *exploration* one (running the risk of collecting information about unknown options), the agent should discover an efficient mapping from states to actions, called *policy*, allowing it to maximize the final return.

2.2 A model-free approach

2.2.1 Q-Learning & Deep Q Network algorithms

The *Q-Learning* algorithm is a form of *model-free* RL: it addresses the RL task by directly mapping environment states to actions, without attempting to construct an exact model of the environment. It can be viewed as a method of asynchronous dynamic programming (DP). Q-Learning is shown to converge with probability one under reasonable conditions on the learning rates and the Markovian environment (Christopher, 1992).

For the present analysis, the *Deep Q Network* (DQN) RL algorithm (Mnih et al., 2013) has been used. DQN is a variant of the *Q-Learning* algorithm which makes use of deep neural networks, with Stochastic Gradient Descent (SGD) approach for weights update, in order to efficiently process training data.

2.2.2 The Agent-Environment interaction

The assumptions described in SECTION 1.2 enable the status of the nodes to be a Markovian environment state. No information regarding the social network graph or the epidemic model is provided to the RL model. The RL agent is trained over a sequence of iterations, or *episodes*; every episode is initialized with the same starting settings, in terms of social network graph and nodes originally infected.

At each time step t, the agent receives the ordered list of statuses (*observation* O_t), and proceeds to select k different nodes (*action* A_t). The environment is then updated according to its choice: the *Susceptible* nodes among the selected ones become *Vaccinated*. Since the agent is not aware of the environment functioning, it might end up picking nodes that are either *Infected*, already *Recovered* or *Dead*. In this case, the action does not affect the status of those nodes. Every episode lasts seven days; at the end of each one, the agent receives a scalar value R (the *Reward*), providing information about the final environment state.

The *Reward* is calculated at the end of each episode as the sum of death probability of every *Infected* or *Dead* node, multiplied by -1.

In formulas, for the i-th episode:

$$R_i = \sum_{Infected, Dead} - d_a$$

where d_a is the death probability for a node in age group a.

The RL model has been implemented by means of the Stable Baselines library (Hill et al., 2018). A Multilayer Perceptron (MLP) neural network has been used as the DQN policy. A Grid Search approach has been followed for the optimization of the *learning rate* hyperparameter.

Chapter 3

Model application & Results

3.1 Simulation settings

The initial number of *Infected* nodes h has been chosen according to the incidence rate of 250 COVID-19 cases per 100 000 people, set by the Italian government as the *"red zone"* threshold in March 2021. Therefore:

$$h = \left\lceil N \cdot \frac{250}{100\,000} \right\rceil = \left\lceil \frac{N}{400} \right\rceil$$

where N is the total number of nodes in the graph.

The h nodes are selected at random at the beginning of the model training phase and remain the same for all the episodes.

The number of daily vaccination k has been set in line with the average administration rate, recorded in March 2021, of 250 000 vaccines every day. In order to achieve a similar vaccines-to-population ratio, k has been calculated as:

$$k = \left\lceil N \cdot \frac{500\,000}{60\,000\,000} \right\rceil = \left\lceil \frac{N}{120} \right\rceil$$

All the graphs have been plotted following the Kamada-Kawai algorithm, that allows for symmetric drawings, while keeping the number of edge crossing relatively small (Kamada, Kawai, et al., 1989).

3.2 Simulations with 100 Nodes

Four simulations have been performed on a social network graph with $N = 100$ ($k = 1, h = 1$). In a first phase, only *Senior* nodes were entitled to receive vaccines; this constraint was then relaxed in a second phase, and vaccination was extended to all the nodes. The two cases have been studied both before and after the application of the RL algorithm. TABLE 3.1 summarizes the four scenarios.

	Without RL Model	With RL Model
Vaccination by age group	*"Senior - No RL"*	*"Senior - RL"*
No Vaccination strategy	*"All - No RL"*	*"All - RL"*

TABLE 3.1: *Summary of the 4 simulations with 100 nodes*

Every simulation has been run for 1000 seven-day episodes; in each case, the RL model has been trained for a variable amount of time until the *Reward* values stabilized on a low level (e.g. FIGURE 3.1). When the RL model has not been applied (*"No RL"* cases), the action performed at every time step has been picked randomly from the action space.

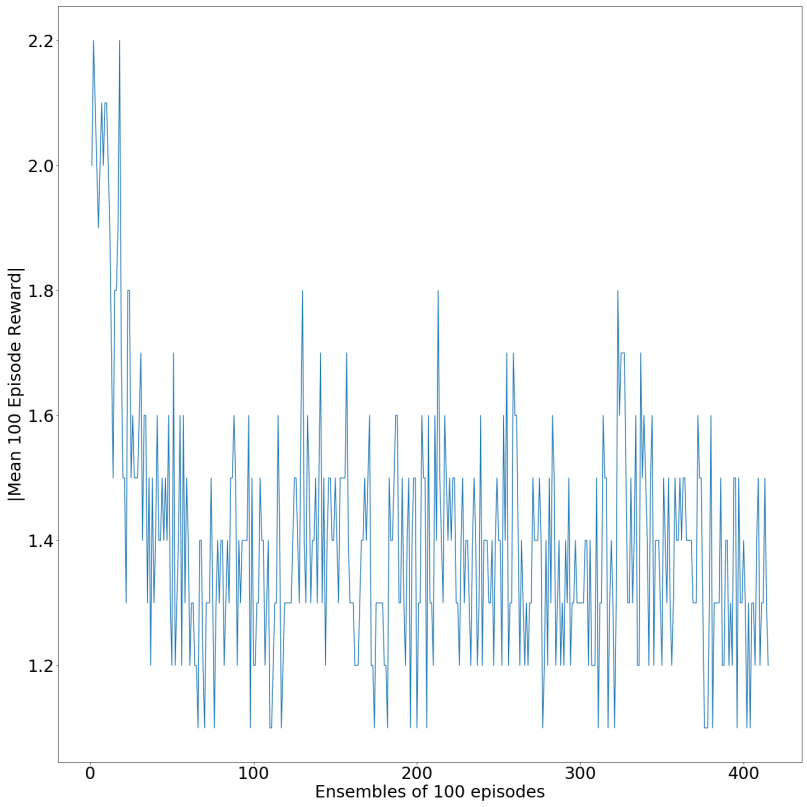


FIGURE 3.1: *Time evolution of the absolute value of the Reward (*1e-2) during training. Each value is averaged over 100 episodes ("Senior - RL" simulation)*

- ***"Senior - No RL"** vs **"Senior - RL"***

  In accordance with the main intervention strategy adopted during the first phase of the vaccination campaign in Italy, in these simulations only *Senior* nodes have been vaccinated.

  FIGURE 3.2 and FIGURE 3.3 show the results in terms of *Reward* distribution. The comparison evidences a shift of the distribution towards higher values, with an increase in the *mean* and a reduction in the *standard deviation*, as underlined by the Gaussian fitting the data.

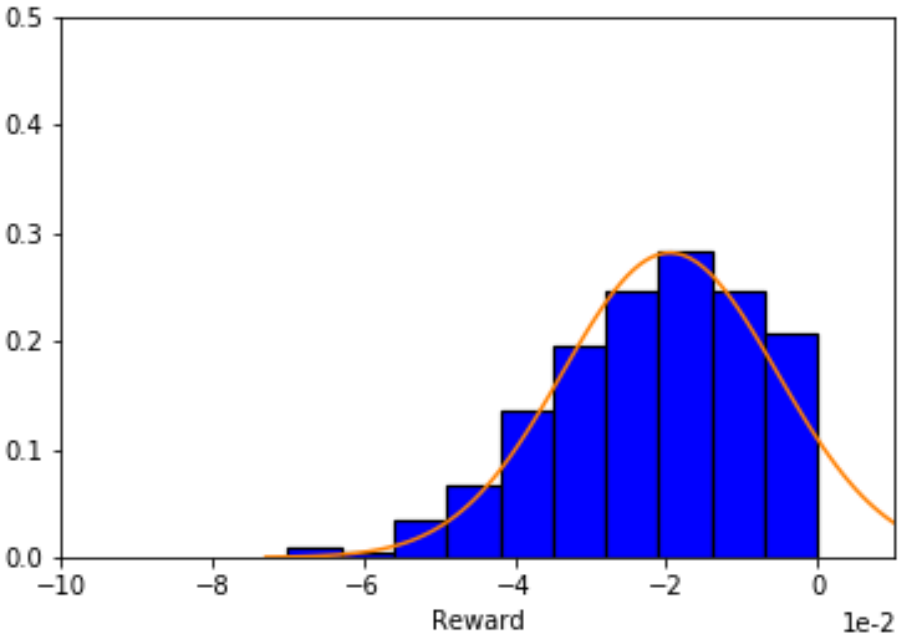


FIGURE 3.2: *Reward probability density (1000 episodes) in "Senior - No RL" simulation*

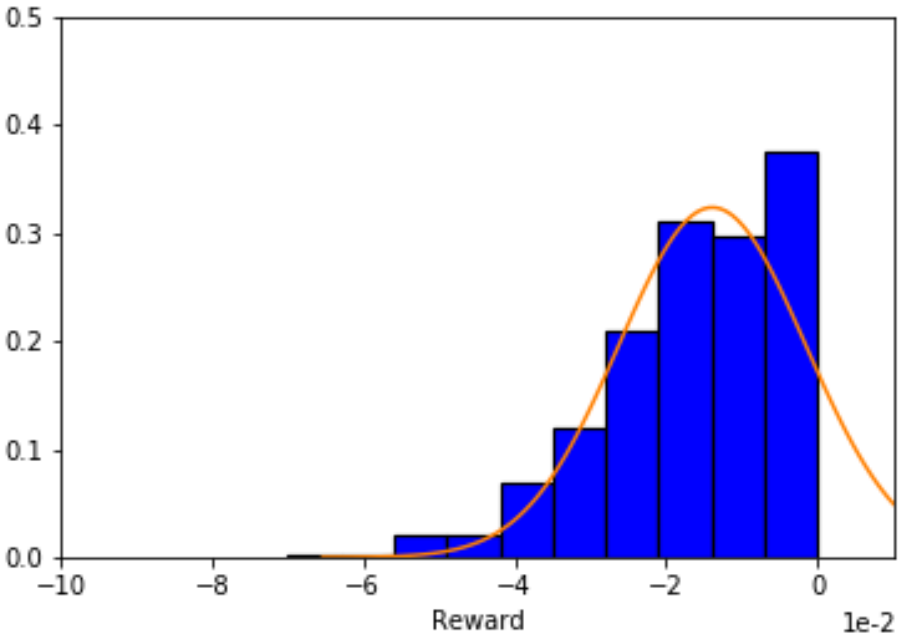


FIGURE 3.3: *Same as* FIGURE *3.2, but for "Senior - RL" simulation*

The comparison of FIGURE 3.4 and FIGURE 3.5 highlights strong differences in vaccine distribution. The RL agent strategy seems to outperform the simple homogeneous vaccination by age group: the aforementioned improvement in the average *Reward* appears to be achieved by treating some nodes preferentially and ignoring some others. These results might suggest the existence of some graph nodes playing a major role on the epidemic impact, evaluated in terms of total death probability.

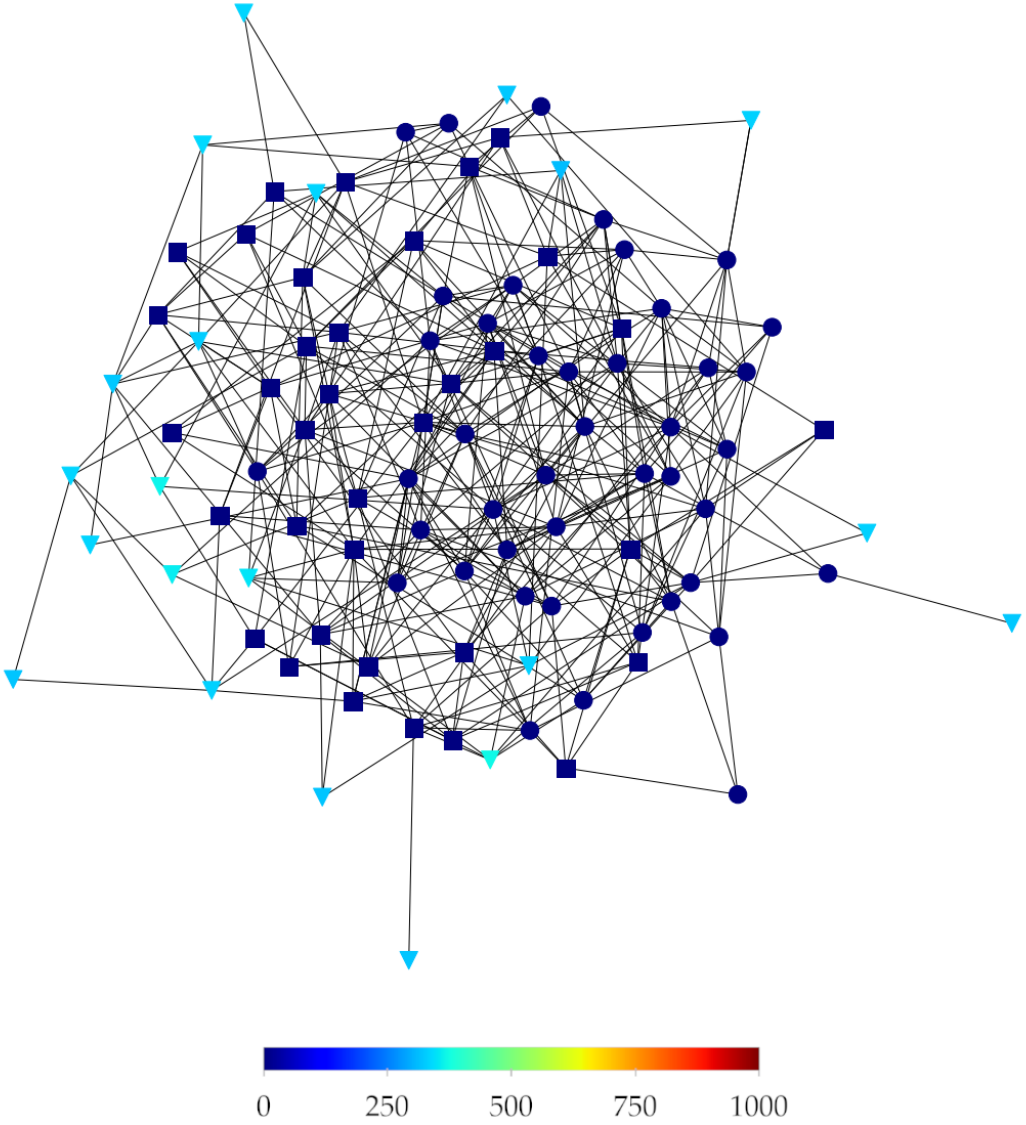


FIGURE 3.4: *"Senior - No RL" Simulation. The colors indicate the total number of vaccine doses received by each node over 1000 episodes*

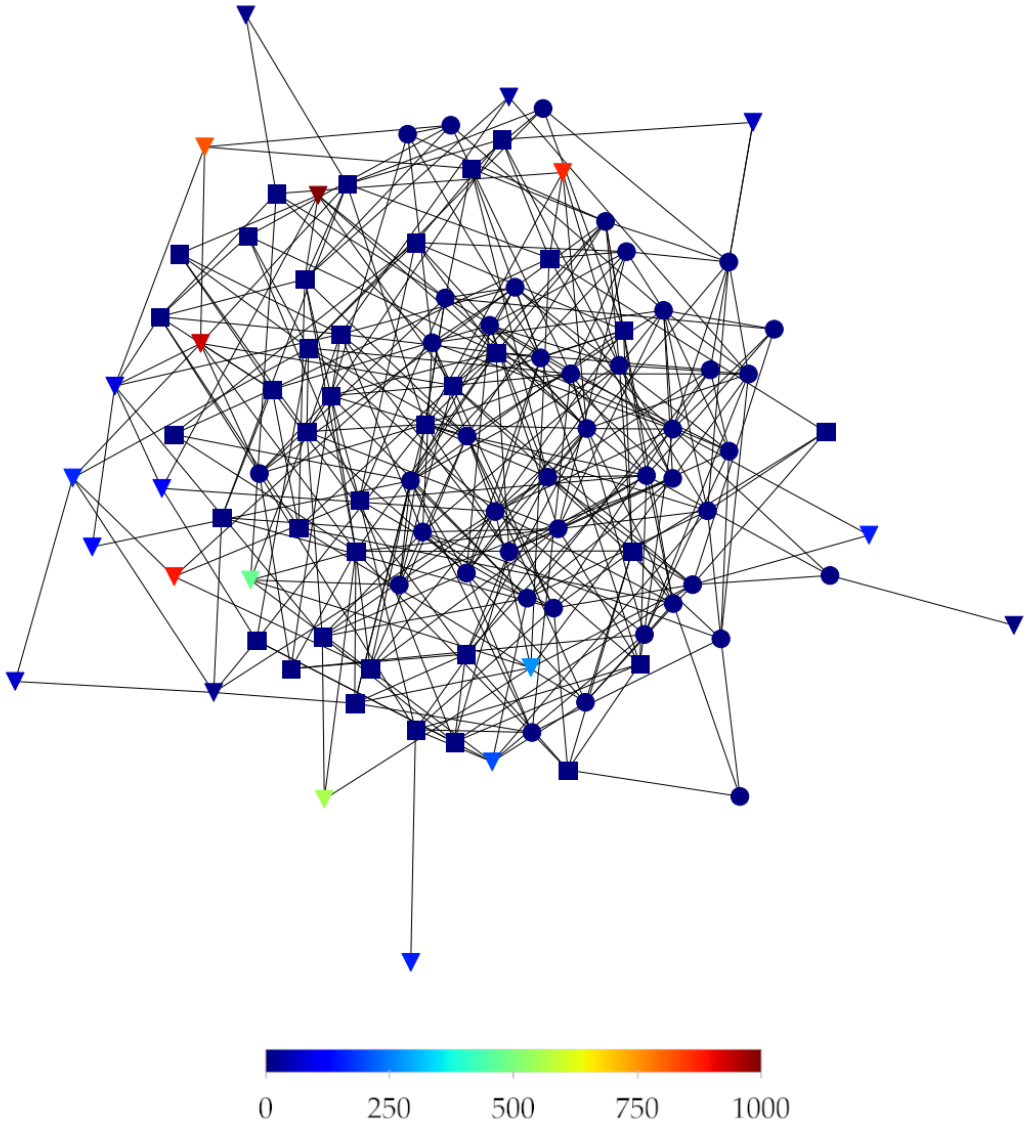


FIGURE 3.5: *Same as* FIGURE *3.4, but for "Senior - RL" simulation*

In an attempt to study the correlation between the total number of vaccine doses received by each node and its centrality value, based on the main centrality measures used for Social Network Analysis (SNA) (Jackson, 2008), their relationship has been investigated (FIGURE 3.6). The scatterplots show a moderate degree of correlation between the two variables, suggesting that the relative importance of a node, in terms of maximizing the efficacy of the vaccination campaign, can hardly be inferred *a-priori* from graph characteristics only. It is reasonable to

deem the final RL strategy to be resulting from a multifactorial environment, where the node status, together with the initial epidemic scenario, contributes to defining the actions of the agent.

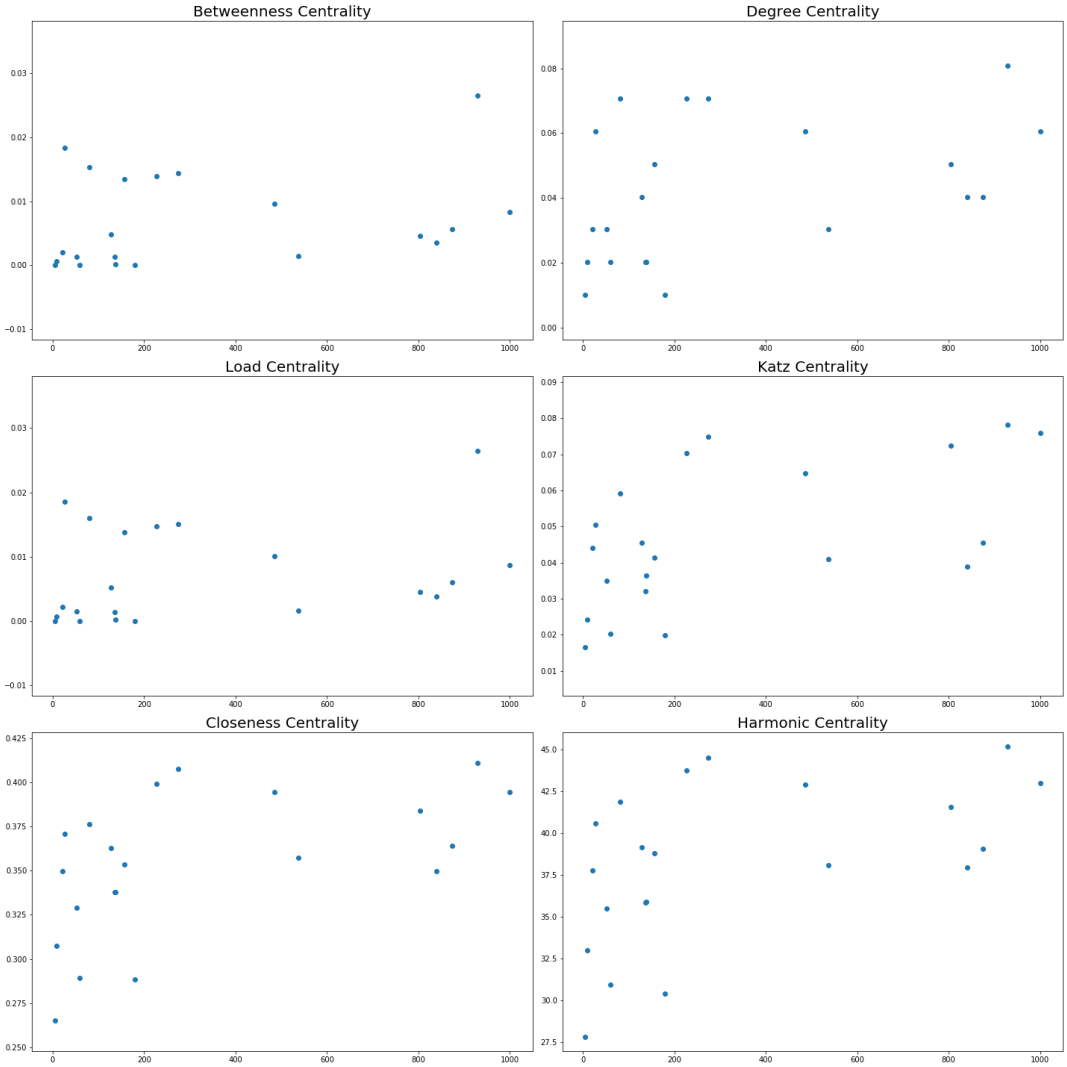


FIGURE 3.6: *Scatterplots showing the relationship between the total number of vaccine doses received (x-axis) and different centrality measures values (y-axis)*

- ***"All - No RL"** vs **"All - RL"***

  In the second set of simulations, the vaccination has been extended to every node indiscriminately, in an attempt to investigate the effectiveness of a similar action plan. Similarly to the *"Senior"* case, results shown in FIGURE 3.7 and FIGURE 3.8 highlight that the RL agent selects preferential nodes for the vaccination. The graphs seem to confirm the validity of a strategy aimed at vaccinating primarily *Senior* nodes. However, surprisingly enough, both *Youth* and *Adult* nodes can be found among the selected ones.

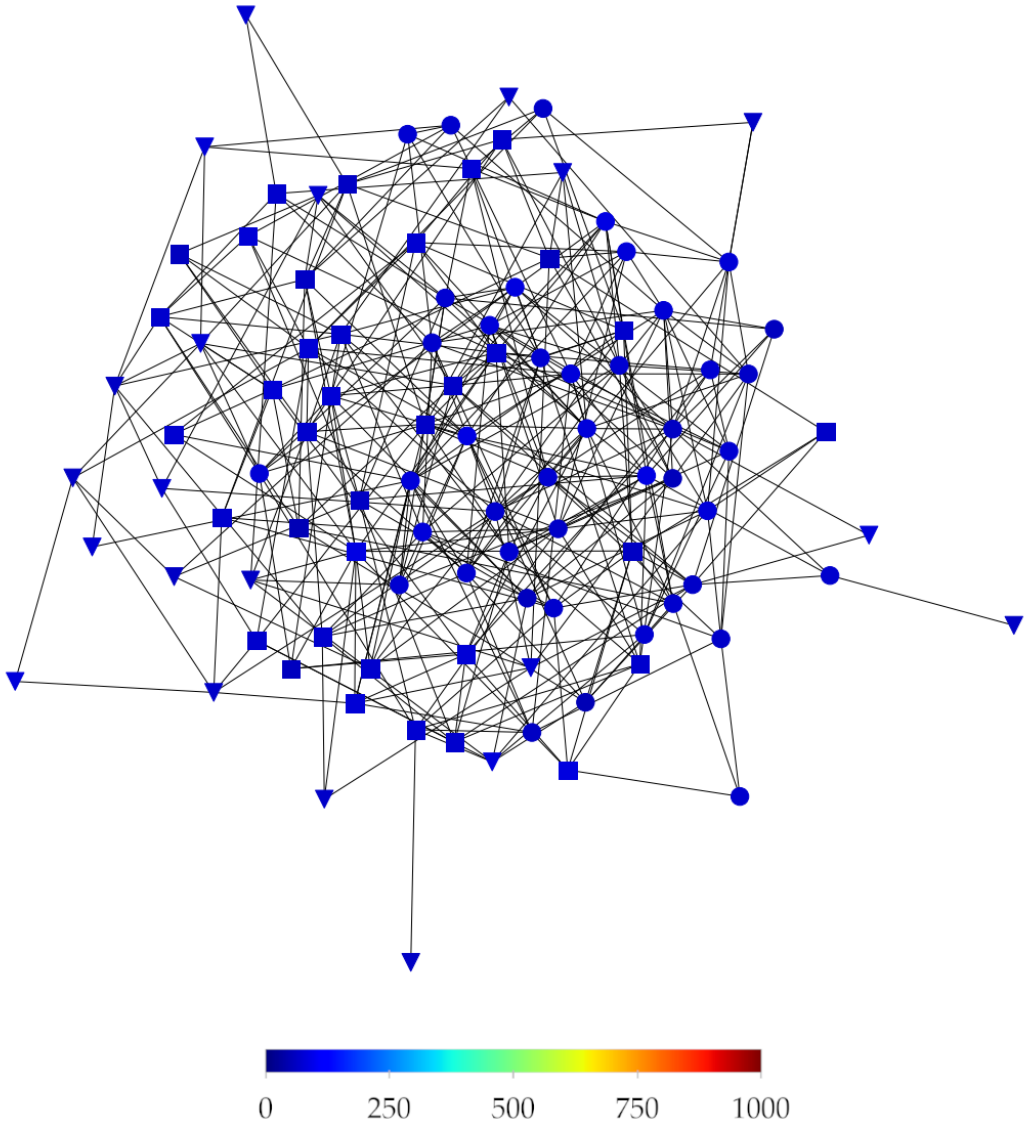


FIGURE 3.7: *Same as* FIGURE 3.4, *but for "All - No RL" simulation*

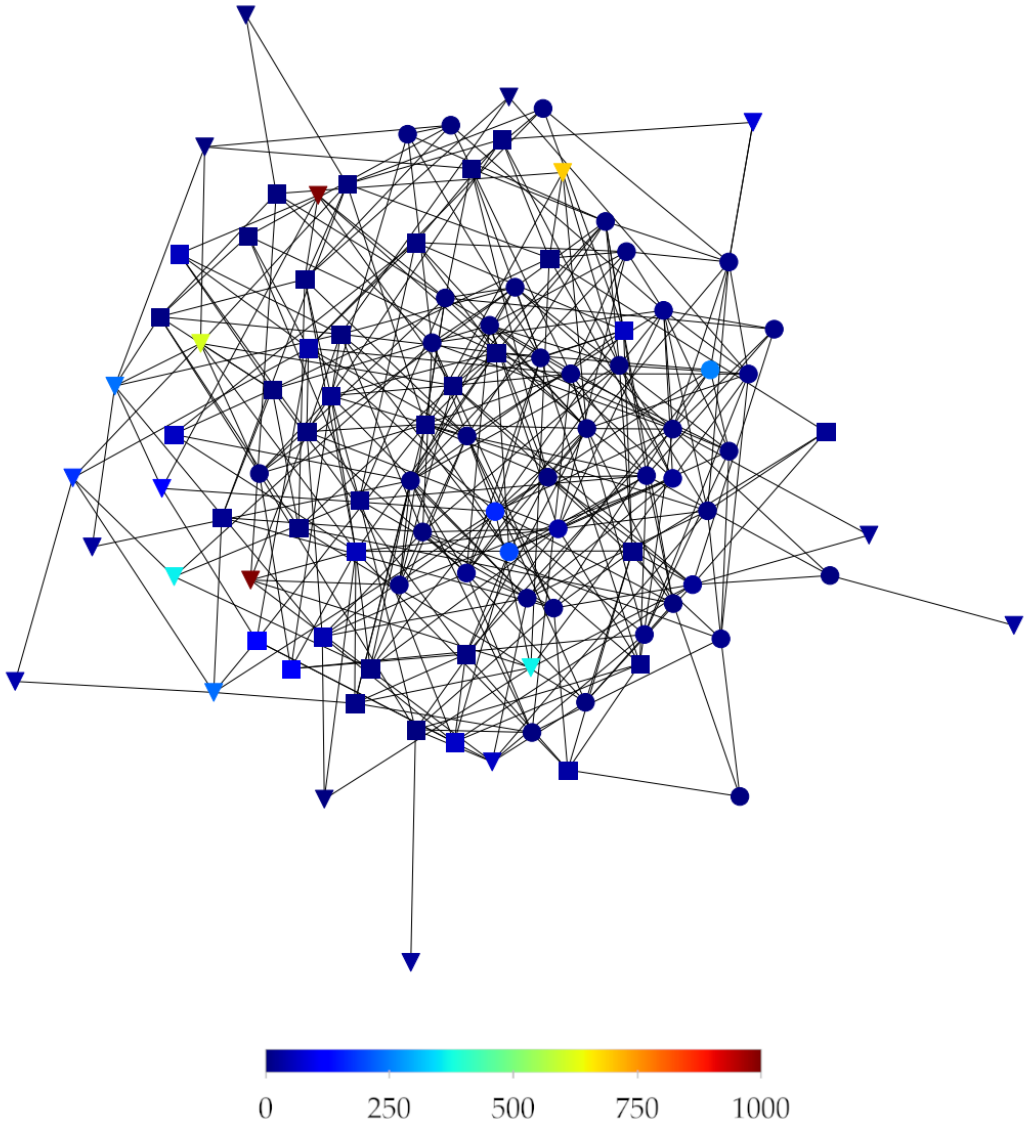


FIGURE 3.8: *Same as* FIGURE *3.4, but for "All - RL" simulation*

On average, on a seven-day episode, about 5 *Senior*, 1 *Youth* and 1 *Adult* nodes are vaccinated (FIGURE 3.9).

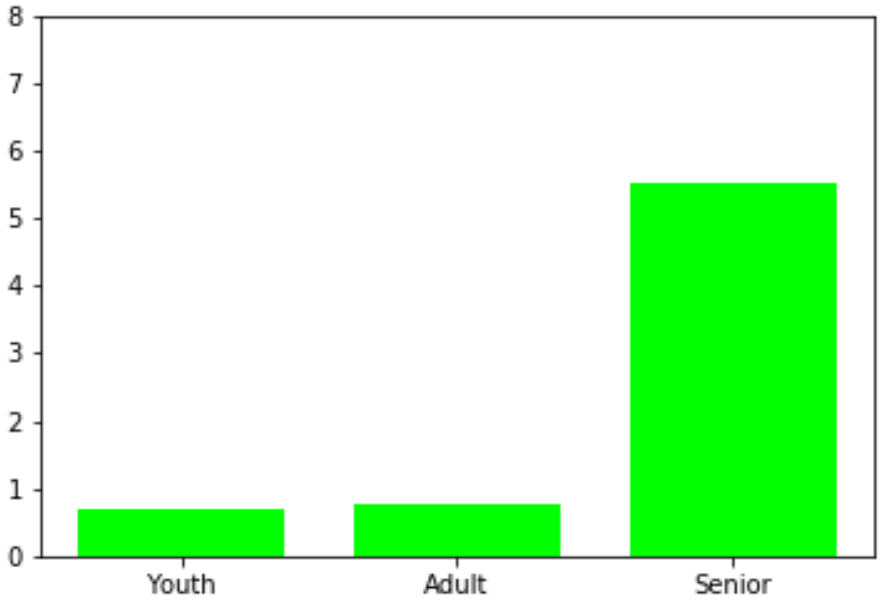


FIGURE 3.9: *Number of vaccinated nodes per age group, averaged over 1000 episodes*

FIGURE 3.10 and FIGURE 3.11 show again an improvement in the mean *Reward* per episode in the *"RL"* with respect to the *"No RL"* simulation. The stronger percentage increase, compared with the *"Senior"* case, leads in the *"All - RL"* simulation to an average *Reward* even lower than the one achieved in the *"Senior - RL"* ($-1.31 \cdot 10^{-2}$ *vs* $-1.39 \cdot 10^{-2}$). This result seems to indicate that, in a supply-constrained scenario, *Senior*-only vaccination might not represent the optimal strategy, since choosing not to vaccinate some *Senior* nodes in favour of *Youth* and *Adult* ones could constitute a better action plan.

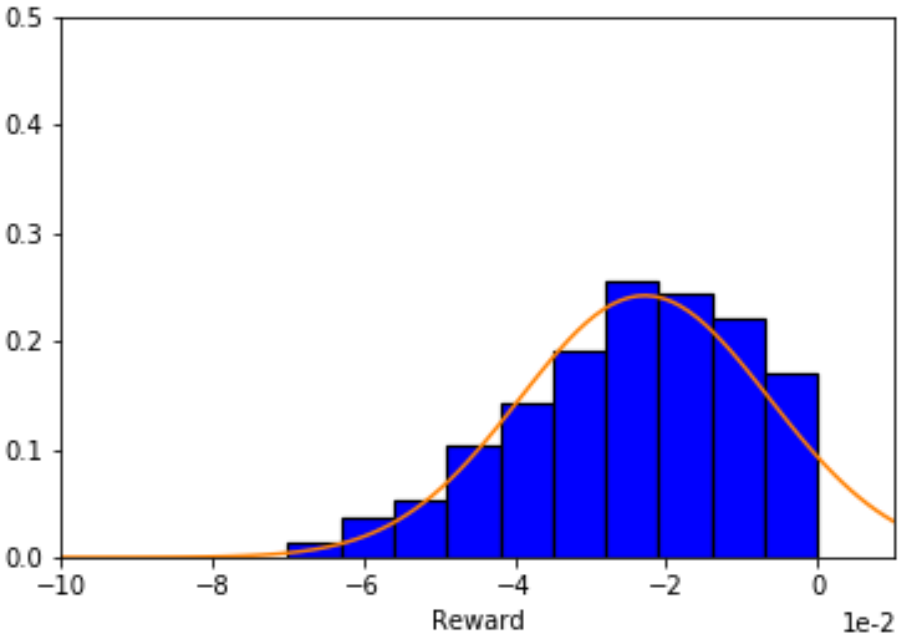


FIGURE 3.10: *Reward probability density (1000 episodes) in "All - No RL" simulation*

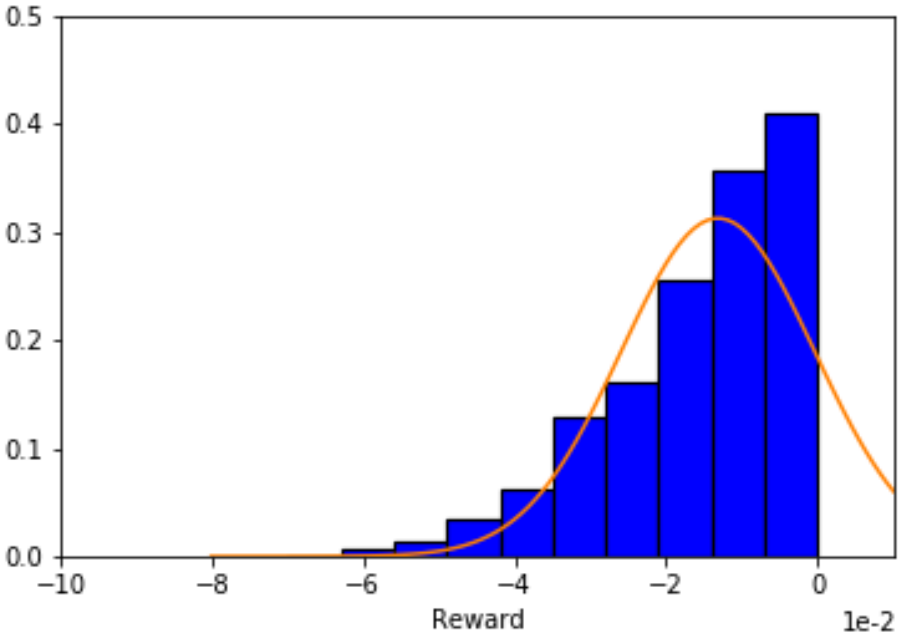


FIGURE 3.11: *Same as* FIGURE *3.10, but for "All - RL" simulation*

As in the *Senior* case, no strong relationship between a great number of vaccinations and a high values of centrality emerges from FIGURE 3.12.

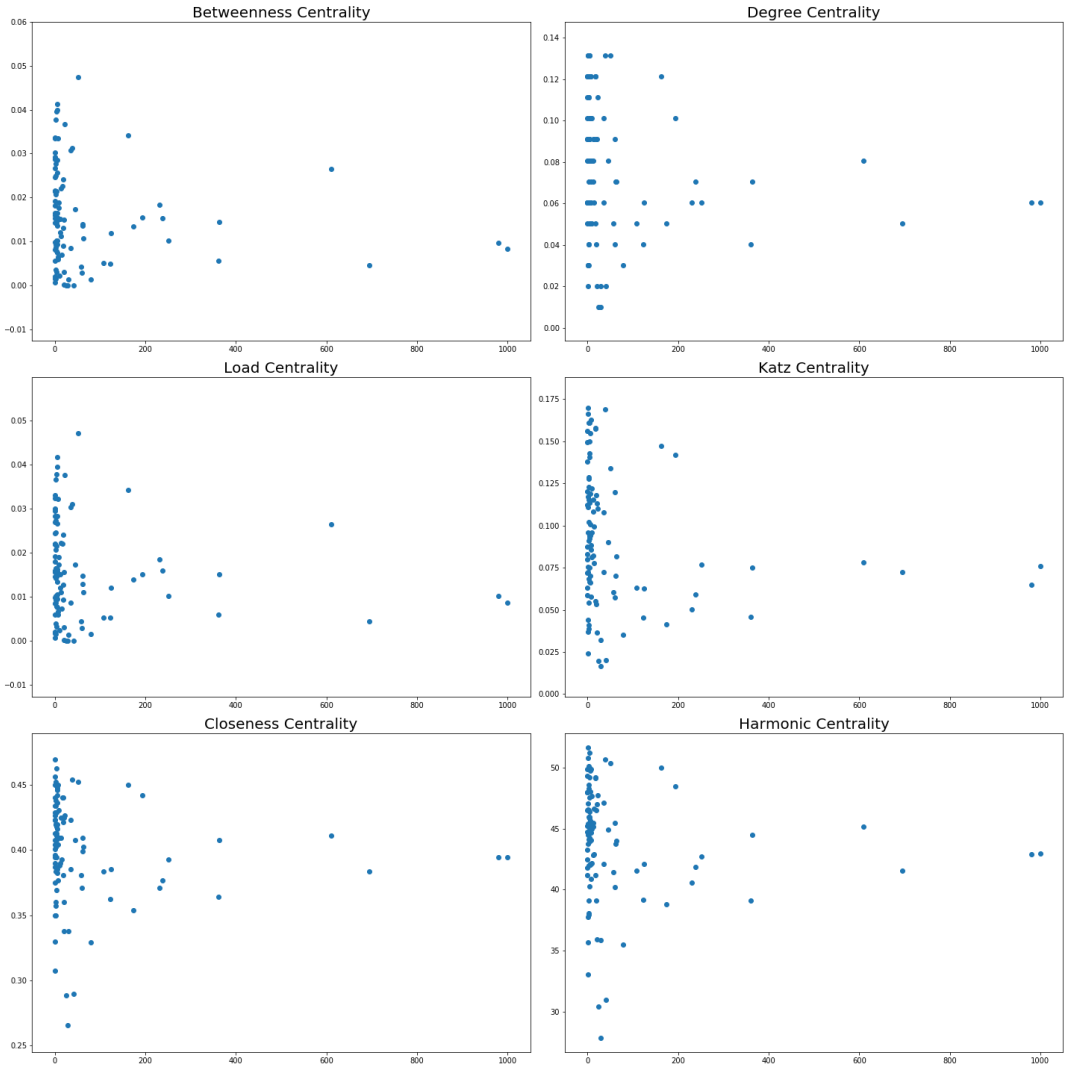


FIGURE 3.12: *Same as* FIGURE *3.6, but for "All - RL" simulation*

3.3 Simulations with 500 Nodes

Two simulations have been performed on a social network graph with $N = 500$ ($k = 2$, $h = 3$), in an effort to assess the RL model performance on a bigger network. TABLE 3.2 summarizes the two scenarios.

	Without RL Model	With RL Model
Vaccination by age group	*"Senior - No RL - 500"*	*"Senior - RL - 500"*

TABLE 3.2: *Summary of the 2 simulations with 500 nodes*

The outcomes seems to reproduce the ones obtained for the 100 node graph, both in terms of *Reward* distributions (FIGURE 3.13 and FIGURE 3.14) and nodes vaccination rate (FIGURE 3.15 and FIGURE 3.16), indicating the possibility of generalising the results to even wider social networks.

Unfortunately, computing power represented a technical barrier to further analyses.

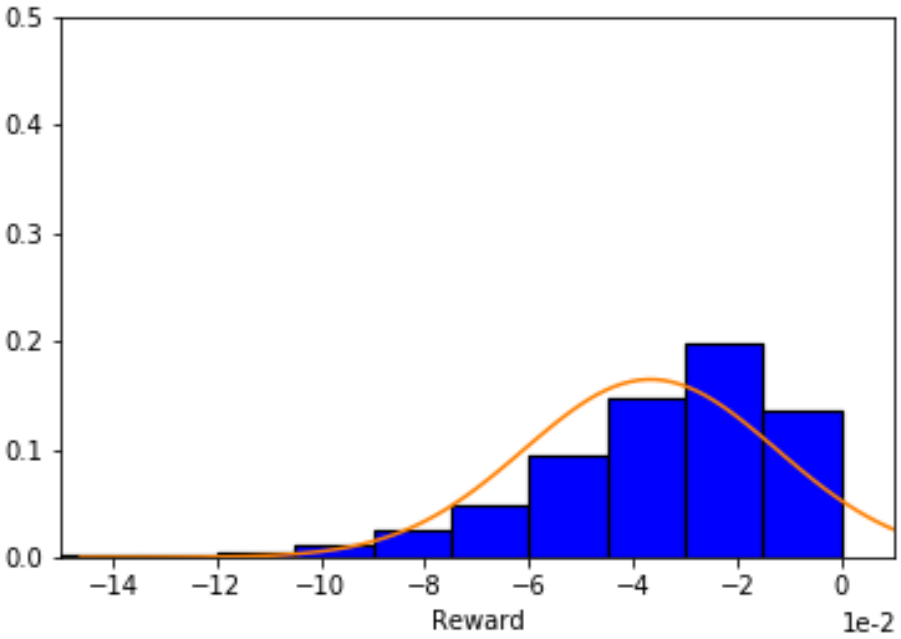


FIGURE 3.13: *Reward probability density (1000 episodes) in "Senior - No RL - 500" simulation*

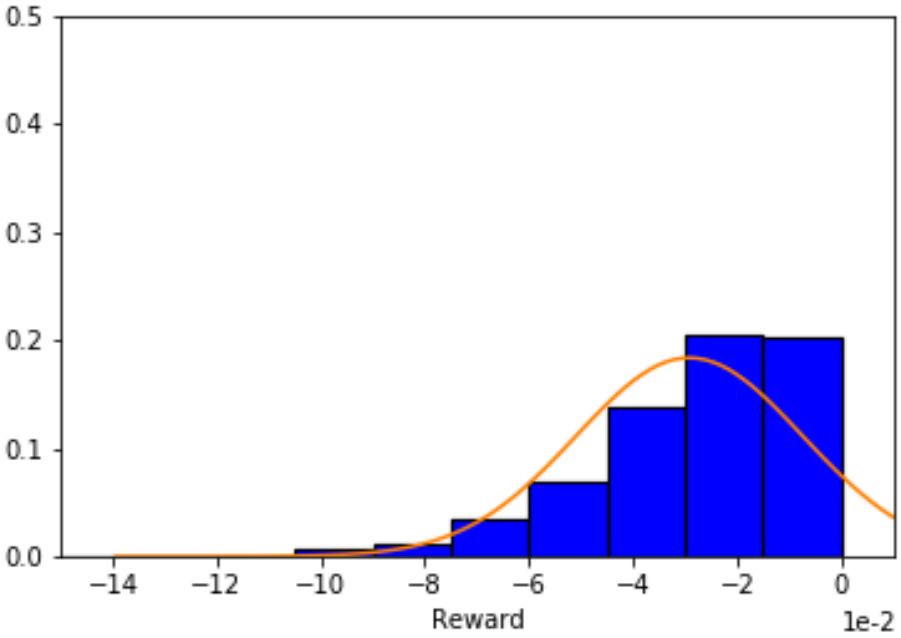


FIGURE 3.14: *Same as* FIGURE *3.13, but for "Senior - RL - 500" simulation*

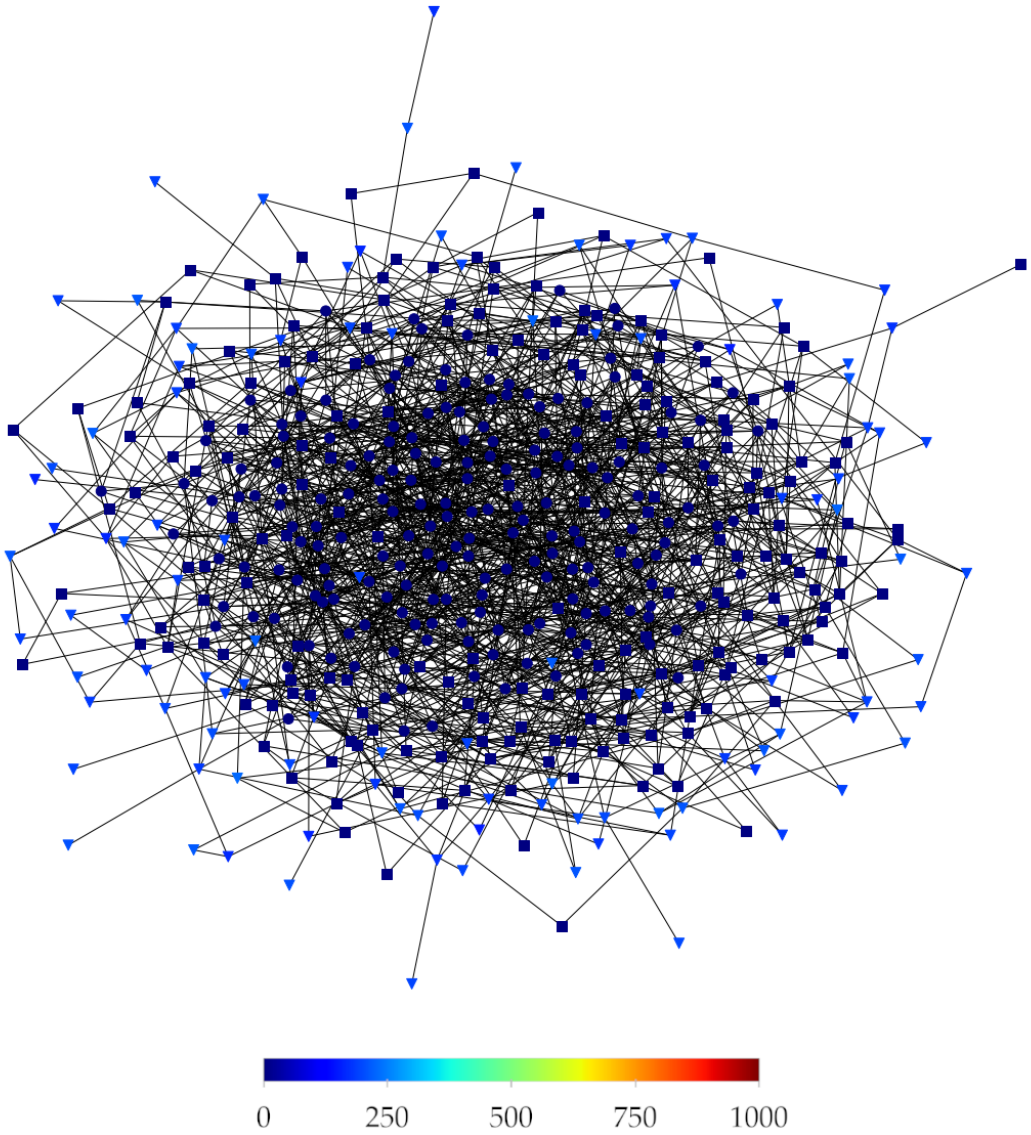


FIGURE 3.15: *"Senior - No RL - 500" Simulation. The colors indicate the total number of vaccine doses received by each node over 1000 episodes*

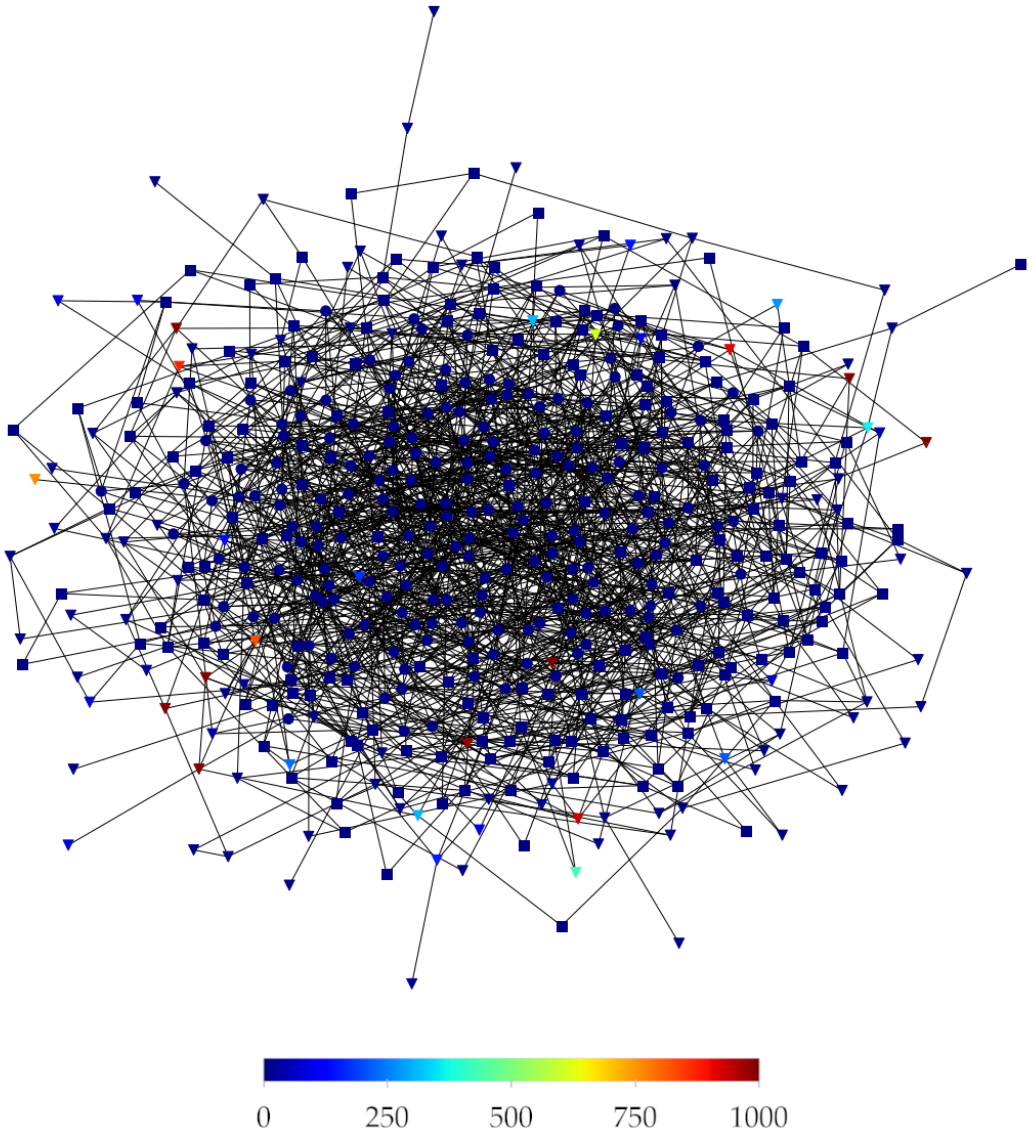


FIGURE 3.16: *Same as* FIGURE *3.15, but for "Senior - RL - 500" simulation*

Conclusion

The present work is aimed at addressing the performance of an AI-driven strategy for the optimization of vaccine distribution, under supply constraints. The results show that the RL agent, trained using the DQN model-free algorithm, is able to identify a policy that might outperform the widely adopted strategy of vaccination by age group.

The proposed approach represents just a preliminary idea, which could nonetheless open new scenarios in the management of future pandemic emergencies. A lot of open issues still need to be addressed, from both the technical and the theoretical perspective.

The whole analysis has been performed employing a *Reward* function which only aims at minimizing the total number of deaths at the end of every 7-day episode. More complex multifactorial functions, taking into account additional economic and/or social factors (e.g. the social cost linked with high hospitalization), could be implemented.

On the technical side, the chance of providing the *Reward* signal on a more frequent basis, as well as the opportunity of modifying the temporal window of each episode, has to be investigated.

The obtained outcome is anyway quite promising, suggesting that AI-based decision making systems may find themselves playing a crucial role in addressing future social issues.

Appendix A

About the possibility of a practical implementation

The vaccine distribution strategy has been a major social issue during the early stage of COVID-19 vaccination campaign. FIGURE A.1 shows how the U.S. Department of Health and Human Services identified three different phases within the planned COVID-19 vaccination program on the basis of the amount of doses available.

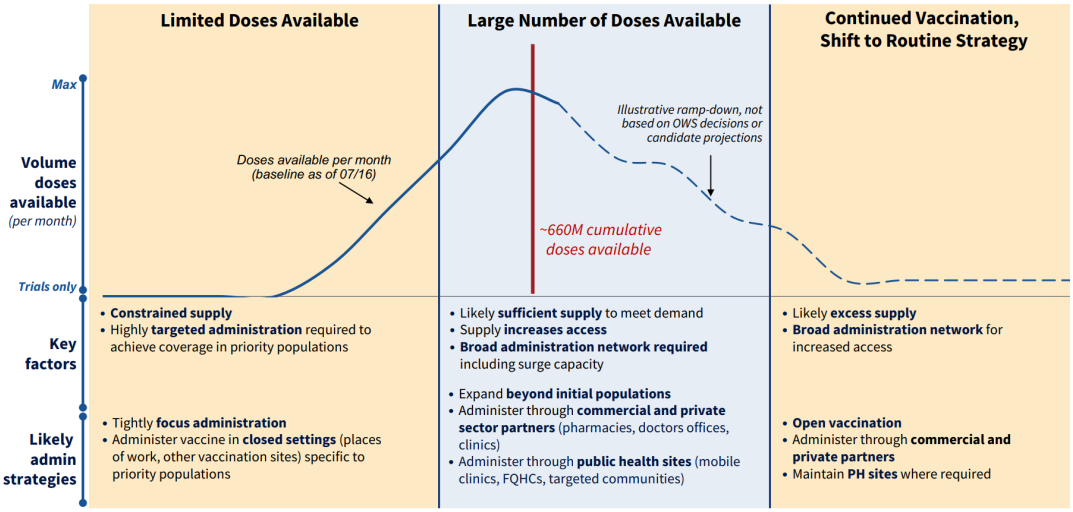


FIGURE A.1: *Illustrative scenario for COVID-19 vaccination planning*
(HHS.gov, 2020)

During the first phase, supply constraints bring the need for a highly targeted administration. The approach discussed in this work could position itself as a possible alternative to the vaccination by age strategy, applied extensively during the last months. The analysis shown in CHAPTER 3 might suggest that an even more tightly focused administration could produce better outcomes on a small network, with promising results on a larger scale. The proposed application of the RL model might represent a smart preventive approach.

However, while data reproducing the average characteristics of the Italian population has been used as input in this analysis, information about a specific area or community would be needed in case of a more focused employment. A practical implementation would certainly require an adequate contact tracing system and high computing resources. Geolocation data could help build an accurate social graph, reproducing a target community, modelled as an isolated network. Up-to-date backward tracing of the contacts of infected individuals would contribute to defining the initial epidemic situation.

Clearly, due to the adopted simplifying assumptions about both the social network and the epidemic spreading, the present work provides only a preliminary evidence of the effectiveness of an RL-driven control strategy, that would need for extensive further validation.

Bibliography

Agrawal, Amit and Rajneesh Bhardwaj (2021). "Probability of COVID-19 infection by cough of a normal person and a super-spreader". In: *Physics of Fluids* 33.3, p. 031704. DOI: 10.1063/5.0041596. URL: https://doi.org/10.1063/5.0041596.

Christopher, JCH (1992). "Watkins and peter dayan". In: *Q-Learning. Machine Learning* 8.3, pp. 279–292.

Cornwell, Benjamin and Markus H. Schafer (2016). "Chapter 9 - Social Networks in Later Life". In: *Handbook of Aging and the Social Sciences (Eighth Edition)*. Ed. by Linda K. George and Kenneth F. Ferraro. Eighth Edition. San Diego: Academic Press, pp. 181–201. ISBN: 978-0-12-417235-7. DOI: https://doi.org/10.1016/B978-0-12-417235-7.00009-3. URL: https://www.sciencedirect.com/science/article/pii/B9780124172357000093.

Feldman, Richard M. and Ciriaco Valdez-Flores (2010). "Markov Processes". In: *Applied Probability and Stochastic Processes*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 181–199. ISBN: 978-3-642-05158-6. DOI: 10.1007/978-3-642-05158-6_6. URL: https://doi.org/10.1007/978-3-642-05158-6_6.

Hagberg, Aric A., Daniel A. Schult, and Pieter J. Swart (2008). "Exploring Network Structure, Dynamics, and Function using NetworkX". In: *Proceedings of the 7th Python in Science Conference*. Ed. by Gaël Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, pp. 11 –15.

HHS.gov (2020). *From the Factory to the Frontlines. The Operation Warp Speed Strategy for Distributing a COVID-19 Vaccine*. https://www.hhs.gov/sites/default/files/strategy-for-distributing-covid-19-vaccine.pdf.

Hill, Ashley et al. (2018). *Stable Baselines*. https://github.com/hill-a/stable-baselines.

Hunter, J. D. (2007). "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3, pp. 90–95. DOI: 10.1109/MCSE.2007.55.

Jackson, Matthew O. (2008). *Social and Economic Networks*. Princeton University Press, pp. 39–41, 61–69. ISBN: 9780691134406. URL: http://www.jstor.org/stable/j.ctvcm4gh1.

Kamada, Tomihisa, Satoru Kawai, et al. (1989). "An algorithm for drawing general undirected graphs". In: *Information processing letters* 31.1.

Madani, Alif Ilham (2020). *Casual Intro to Reinforcement Learning*. URL: https://towardsdatascience.com/casual-intro-to-reinforcement-learning-4a78b57d4686.

Mnih, Volodymyr et al. (2013). "Playing Atari with Deep Reinforcement Learning". In: *CoRR* abs/1312.5602. arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

Mossong, Joël et al. (Mar. 2008). "Social Contacts and Mixing Patterns Relevant to the Spread of Infectious Diseases". In: *PLOS Medicine* 5.3. DOI: 10.1371/journal.pmed.0050074. URL: https://doi.org/10.1371/journal.pmed.0050074.

Probert, W. J. M. et al. (2019). "Context matters: using reinforcement learning to develop human-readable, state-dependent outbreak response policies". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 374. DOI: 10.1098/rstb.2018.0277. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rstb.2018.0277.

Ross, S.M. (2004). *Introduction to Probability and Statistics for Engineers and Scientists*. Introduction to Probability and Statistics for Engineers and Scientists. Elsevier Science, pp. 141–143. ISBN: 9780080470313. URL: https://books.google.it/books?id=m7oeCiMh-78C.

Rossetti, Giulio et al. (2018). "NDlib: a Python Library to Model and Analyze Diffusion Processes Over Complex Networks". In: *CoRR* abs/1801.05854. arXiv: 1801.05854. URL: http://arxiv.org/abs/1801.05854.

Stewart, Conor (2021). *Italy: coronavirus death rate by age*. URL: https://www.statista.com/statistics/1106372/coronavirus-death-rate-by-age-group-italy/.

Willem, Lander et al. (2020). "SOCRATES: an online tool leveraging a social contact data sharing initiative to assess mitigation strategies for COVID-19". In: *BMC Research Notes* 13.1, p. 293. ISSN: 1756-0500. DOI: 10.1186/s13104-020-05136-9.