Spatial SIRD model with different risk groups – multiagent approach for modelling an epidemic

Mihailo Ilić and Mirjana Ivanović

Faculty of Sciences, University of Novi Sad, Novi Sad, Serbia

ABSTRACT
This paper highlights the use of software agents and simulating real-world medical phenomena. We start with a brief overview of different approaches and tools for developing software agents and running simulations. One of the more recent tools was utilized in this paper to develop a model of disease spread in a population of agents and for performing experiments. We proposed an extension of the SIRD model, which includes a new state for agents, spatial component, and also different risk groups based on age. Multiple factors affect the longevity of a pathogen in a given population, for instance, its infectivity and deadliness. This paper shows interesting results of extensive experiments performed with novel, modified SIRD model and impact of key factors on disease spread.

1. Introduction

Software agents, in the broadest sense, are programs that act on behalf of someone else. These agents are defined by their behaviour and their ability to act with a degree of autonomy. The first mention of an idea resembling today’s understanding of software agents dates back to the seventies, when Hewitt described his actor model (Hewitt et al., 1973). These actors communicate through messages and can modify only their own private state. If they intend to modify the states of other actors, they must do that by sending messages, to which the target actor reacts to.

The majority of authors agree software agents are based on the actor model (Bădică et al., 2011), and that they share common properties (Wooldridge & Jennings, 1994, 1995), which include: Autonomy – they act on their own; Social ability – agents usually interact with other agents; Reactivity – they are able to react when prompted by other agents or when changes in the environment occur; Proactiveness – agents are not only reactive, but they actively take action, which in most cases is goal-oriented, meaning they wish to achieve a specific goal.

Agents are capable of acting and reacting when confronted with other agents and the environment they’re in. During the history of developing and using software agents,
various programming languages and frameworks have been developed for different purposes. In the rest of the section, we will briefly present some characteristic languages and frameworks.

**Programming Languages for Agent Development** – Commonly used programming languages like Java, C, C++ and others have their own tools and frameworks for handling agents and multi-agent systems. On the other hand, there exist various prototype languages which provide useful abstractions when constructing agents and agent-based systems. According to Bădică et al. (2011), these languages can be classified into multiple groups based on their most relevant aspects regarding agent modelling: *Agent-oriented programming (AOP) languages*, *Belief-desire-intention languages (BDI)* and *Hybrid languages*.

**Agent Platforms** – Various tools and platforms for developing agent systems and agent modelling have been developed over the years. A large number of both open source and commercial tools are available today, since agent-based modelling is a powerful tool researchers can benefit from. The authors of Pal et al. (2020) split modern agent frameworks into two main categories:

1. **General purpose**: open source; commercial
2. **Special purpose**: Cognitive, social and affective agents; Artificial intelligence research; Modelling and simulation; Transport-related simulations

The advantage of *general purpose platforms* is their lack of domain focus, which makes them flexible and allows for a wide range of uses. Because of this, various widely used programming languages are employed to implement them, while some even have friendly *graphical user interfaces*. Open source variants, such as AgentScript (*Agentscript project page, n.d.*), JADE (*Bellifemine et al., 2007; Java agent development framework, n.d.*), Mesa (*Masad & Kazil, 2015*) and Repast (*Repast website, n.d.*) use general purpose programming languages such as *Java, C#, JavaScript and Python*, while other like JaCaMo (*Jacamo project website, n.d.*) rely on specialized languages for agent modelling like *AgentSpeak* (*Weerasooriya et al., 1994*) which is a BDI language. Commercial tools like AnyLogic (*Anylogic website, n.d.*) and Wolfram SystemModeler (*Wolfram systemmodeler website, n.d.*) provide a large number of features and supporting software, along with industrial strength simulations.

Special purpose platforms are designed to effectively handle specific problems. By sacrificing flexibility, they provide tools and mechanisms designed for solving specific problems. For example, modelling human behaviour is done using the ACT-R (*Ritter et al., 2019*) platform while for modelling relationships between societies and their environments, Cormas (*Cormas website, n.d.*) is used. In artificial intelligence research, tools like MAgent (*Zheng et al., 2018*) allow for multiagent reinforcement learning, while MADP (*Madp project website, n.d.*) is used in research of decision planning and learning in agent systems.

The goal of this paper is to illustrate how software agents can be applied in modelling real-world systems, such as the spread of disease. In the approach that we proposed in this paper, we are concentrated on an extension of SIRD model, which includes, one additional state for agents, a spatial component, and most significantly, the consideration of risk groups of patients/agents based on their age. A multitude of factors affect the
longevity of a pathogen in a population, some of which being its deadliness or infectivity. This paper brings interesting results of extensive experiments performed with our proposed model and shows the impact of key factors that influence the spread of disease.

The rest of the paper is organized as follows. The second section covers related work done in the field of epidemic modelling. In the third section, a brief overview of the Mesa agent modelling framework is given, along with the explanation of the new model that we proposed for virus spreading under specific circumstances for the purpose of this experiment. Comprehensive results of the simulations are shown in section 4. Concluding remarks are given in the last section.

2. Related work

Agent systems give the power to model behaviour of entities capable of making decisions when interacting with their environment. It is possible to simulate complex systems which include hundreds or thousands of agents. These models can be simple or complex, depending on the domains and problems that are planned to be solved by the agent system. This is the reason that they find use in both academic research and industry. Intensive use of agents can be found in numerous domains like video games, transportation and logistics, power grids, medicine and many others.

Extensive application of agent systems can be seen in epidemiology. The authors in Bădică et al. (2020) recognize the potential of applying agent systems in helping to understand the dynamics of an epidemiological outbreak. By using the GAMA simulation development environment, a standard SIR (Susceptible-Infected-Recovered) model where each agent can be either Susceptible, Infected or Recovered has been constructed. The agents are placed in a continuous space, where each agent is a point in 2D space capable of moving through the environment with traits such as speed and direction defined.

The authors in Shoukat and Moghadas (2020) give a general overview of agent-based systems, including an explanation of their computational construction. They emphasize that it is not imperative to include every single detail in modelling complex systems, as technical limitations don’t allow this. However, it is suggested that adding as much traits of the system as possible helps in recreating the real-world system, which allows researchers to create precise models. The agent model depicted in Shoukat and Moghadas (2020) is also focused on disease spread, relying on the standard SIR model which is also used in Bădică et al. (2020). The opposite to the approach shown in the paper (Bădică et al., 2020), in our approach agents are placed in a grid space, where each cell in the grid represents a ‘room’ which multiple agents can share. This way, the disease can only spread between agents located in the same cell. The simulation in Shoukat and Moghadas (2020) is conducted on a 20x20 grid, with each agent at a fixed position and the disease can only spread between agents in adjacent cells of the grid. Our approach presented in this paper builds upon the idea presented in Shoukat and Moghadas (2020). However, in our approach agents are allowed to be located in the same cell, and also it is allowed for them to migrate to adjacent cells at each step of the simulation.

The SIR model has been extensively used in literature. The authors of Adamu et al. (2019) and Fernández-Villaverde and Jones (2020) build upon the standard SIR model by accounting for deaths attributed to the disease being modelled, and thus creating a
so-called SIRD model. Valuable information can be gathered by analysing these mathematical models. These types of simulations offer a way of predicting the impact of certain diseases on various cities and countries, as well as give insight into the effects of government measures aimed at suppressing their spread. A comprehensive disease model can be found in Acemoglu et al. (2020) where different risk groups of patients are introduced. The main goal of the research was to find an optimal lockdown policy that has the least negative impact on the economy while also keeping the number of deaths to a minimum. According to the authors of Bisin and Moro (2020), spatial SIR models account for geographical factors and local interactions between individual agents, which have an impact on disease dynamics and are not captured by mathematical models.

As stated in Hackl and Dubernet (2019), human mobility is one of the main factors which impact how fast a disease can spread through a population. The authors mention recent outbreaks such as SARS in 2003 and H1N1 in 2009. We are now experiencing a similar situation with the current COVID-19 pandemic. Given enough data, researchers can model the spread of disease on different levels (for example, city-wide or even world-wide). In Hackl and Dubernet (2019), a demonstration was conducted by modelling the metropolitan area of Zurich, Switzerland. Since the population size is roughly 1.5 million people, only 1% of the population (15 286 agents) was used to model the spread of seasonal flu. The authors propose linking transport and epidemiological modelling in order to simulate outbreaks of infectious diseases more precisely. They also mention the fact that the results of the simulation can vary depending on the number of parameters taken into account. Simpler models only account for healthy, infected and recovered individuals, while other parameters such as vaccination, immunity, gender, age, etc. could help in giving more accurate results.

In this paper a simulation of an epidemic is conducted based on a new model, which we proposed as an extended version of SIRD model. The main characteristics of our approach are as follows. Agents are placed in a grid and are able to move between cells. Cells represent some shared space (e.g. a room or building), and on this level infected agents have a chance of spreading the disease to healthy agents. This chance increases if the healthy agent is exposed to multiple infected agents occupying the same cell. Our simulation introduces risk groups to a spatial SIRD model in order to analyse the effects of disease suppression methods on mortality rates. These risk groups are based on age, where agents can be either young, middle aged or old. Similar groups are taken into account in other viral disease simulations, for example, they are considered for different COVID-19 measures and policy decisions in numerous countries.

3. Modelling the spread of disease

One of the main goals of this paper is to present how one could model a real-world problem from the medical domain using software agents. The topic of infectious diseases, more precisely viruses, and how they spread is catching more attention with the spread of COVID-19 (Karnyoto et al., 2021; Vrbačič et al., 2021). The ongoing pandemic was also an inspiration for our model and simulations. By understanding how viruses affect the human body and how they spread, we can reduce the speed of transmission and their
negative effect. The latter can be modelled using software agents, as shown in Hackl and Dubernet (2019).

3.1. The framework utilized in the experiments

As we briefly presented above there are still some flexible, widely used programming languages, which support adequate utilization of software agents in solving specific problems. In our experiments, we will use the Python-based Mesa framework.

Mesa Overview -- Mesa is a recent open-source framework written in Python, which gives users the ability to quickly construct agent models. These models can then be easily visualized and analysed using various other Python frameworks and libraries. One useful tool which this framework has built in Masad and Kazil (2015) is the data collector which will be showcased in this paper.

The main concepts this framework revolves around are the Model and the Agent. These core classes serve as the building blocks of any agent model. Mesa is highly modular (Masad & Kazil, 2015), which means it doesn’t make any assumptions in the way of how a model should operate and also provides multiple different schedulers which are in charge of handling the activation of agents.

The agent is the main actor in a Mesa model. A class which extends the mesa.Agent class needs to override its step method. A Mesa model defines the passage of time as a series of steps, and by overriding this method in the Agent class, we can define the behaviour of an agent at each step of the simulation.

The model class models the entire system, which includes the agents and their environment. The class which inherits the mesa.Model class serves as a container for all other elements and global parameters which are used to run the simulation.

The space component represents the space which is populated by agents. Multiple components are located in the mesa.space module, some of which are:

- Continuous space -- Agents are treated as points and they can have an arbitrary position.
- Grid -- This is the base class for a square grid. Multiple other classes extend the functionality of this one, like a grid realized as a torus.
- HexGrid -- Represents a hexagonal grid.

Each space component provides functionality for selecting agents at specific coordinates, along with iterating through neighbouring cells or finding neighbouring agents within a certain radius. A type of grid space will be used in our simulation (Figure 1).

Another core component is the scheduler, located in the mesa.time module. Schedulers are components that control when agents are to be activated and run their step methods. The activation order can be crucial for the outcome of the entire system, so it is important to choose the right one. Some predefined schedulers are:

- Base scheduler -- Calls each agent’s step function in the order they were added.
- Random activation -- Activates agents in a random order.
- Simultaneous activation -- Agents are activated at the same time.
Data collectors are a built-in tool which enables users to collect multiple types of data, such as model/agent level data, along with tables.

3.2. Methodology used for defining the disease model

In order to run a simulation and collect data, all the components mentioned in the previous section need to be put in use. The goal of our experiment is to model disease spread among a population of agents. As stated in Hackl and Dubernet (2019), one of the main factors of disease spread is agent mobility. Hence, we need to define a space that the agents can traverse in this model. For this purpose, a MultiGrid from the mesa-space module will be used. In this paper, cells in the grid will be analogous to rooms in a building, with agents being able to move between adjacent rooms (cells). Agents, in this case, represent people moving around inside a building. The reason behind using a MultiGrid is the possibility of having multiple agents occupy a cell. Unlike the standard SIR model used in Bădică et al. (2020) and Shoukat and Moghadas (2020) where each agent can either be susceptible, infected or recovered, we considered an additional state — ‘deceased’, for agents which do not survive the infection. Accordingly, each agent can be in one of four states:

1. Healthy -- Healthy agents which are not carriers of said disease.
2. Infected -- Infected agents are carriers of the disease.
3. Deceased -- Such agents have succumbed to the disease which is being modelled and no longer have the ability to move between cells and cannot infect further agents.
4. Immune -- Agents which have survived the initial infection and are now immune to the pathogen.

If an agent is infected, it has a chance of recovery or death. Once an agent recovers, it becomes immune and can no longer be infected. These values can be defined globally, which means all agents are the same in terms of age, sex, immunity, etc. In order to

![Figure 1. Examples of different Grid space components. (a) A 10 x 10 grid space and (b) A hexagonal grid space.](image)
obtain more accurate models one could define agent-level properties, for instance, the agent’s age. Infected agents also have the possibility of spreading the disease to another healthy agent occupying the same cell. If a healthy agent is located in a cell that is also occupied by infected agents, the chance of the infection spreading to the new agent is calculated as in Shoukat and Moghadas (2020).

\[ P_{\text{infection}}[\text{state}_{a,s+1} = \text{Infected}|\text{state}_{a,s} = \text{Healthy}] = 1 - (1 - c)^k \] (1)

The chance of an agent \( a \) to transition from a healthy state to an infected state between steps \( s \) and \( s+1 \) of the simulation depends on the number of infected agent \( k \) occupying the same cell. In this equation, \( c \) represent the initial chance of infection, which is a global parameter of the model.

At the end of each step, an agent also has the ability to migrate to a different cell. The chance of this occurring is also configured globally. Each agent can perform the following actions: transmit the disease; migrate to another cell. They also have pre-defined global chances to: recover from the disease; pass away -- if this happens to an agent, it becomes inactive and performs no more actions during the simulation.

We used a random scheduler during modelling. To ensure the same results each time the simulation is run, it is imperative to set the seed of the random number generator to a fixed value.

A data collector was also used in order to collect data after each step in the simulation. The following values were computed after each step:

1. The number of healthy agents in the entire grid. Immune agents will also be included, since they have recovered from the disease.
2. The number of infected agents in the grid.
3. The number of deceased agents in the entire grid.
4. Multiple matrices containing the number of each type of agent according to their health status per cell.

Finally, all of these components are aggregated in the DiseaseModel class that extends the mesa.Model class. This class takes in multiple parameters for defining the simulation:

- \( N \) -- The number of agents in the system.
- \( Width \) -- The width of the grid.
- \( Height \) -- The height of the grid.
- \( Risk\ group\ ratio \) -- The ratio of agents belonging to each risk group. Risk groups include agents younger than 40 (young), older than 64 (old), and everyone else in between (middle aged).
- \( Chance\ of\ death \) -- The possibility of one succumbing to the disease based on the assigned risk group. It is evident that the chance of death from an infection depends on multiple factors. In our simulations, this risk is associated with age. Younger agents have a lower chance of death, while older agents have a higher possibility of death.
• **Chance of spread** -- The chance of the disease spreading from one infected agent to a healthy one. It is definitely interesting to consider different values of a chance of disease spread. Different types of viruses and their mutations spread more aggressively than others.

• **Chance of the illness ending** -- The chance for an agent to reach the end of the illness, either by recovering and becoming immune or by succumbing to the disease. The duration of the illness and the difference between a positive and negative outcome also depend on various factors. A positive outcome, with the agent recovering, is more likely for younger agents than it is for older agents.

• **Chance of migration** -- The chance of an agent moving to an adjacent cell. Interactions between individuals in a population during an epidemic plays a key part in disease dynamics. An agent may decide to move during the simulation, placing them at risk of contracting the disease or spreading it to others.

• **Infected on start ratio** -- Defines the ratio of the entire population which will be disease carriers at the start of the simulation.

• **Curfew activation threshold** -- Percent of infected individuals in the population which will trigger government measures.

• **Compliance ratio** -- The ratio of agents from each risk group following all government epidemic measures.

A key factor that should be taken into account when modelling an epidemic is population demographics. Members of a certain population fall into different risk groups depending on their age, health condition, economic status, etc. The impact of a certain disease may vary drastically among each risk group. In order to more precisely simulate an epidemic, our model enables defining which risk group each member of the population belongs to.

In an effort to slow down or eliminate a disease, local authorities or governments may introduce certain measures like social distancing, mandatory masks indoors or even lockdowns. Even with these measures in place, a certain percentage of the population may not be able or willing to follow them, which is also a kind of factor that should be taken into consideration when simulating a disease.

Our first attempts in simulating virus spread are presented in Ilić and Ivanović (2021). In these simulations, we used a slightly modified SIR model. After achieving interesting and challenging results, we extended this simple model as explained above. The next section presents the results of both simulations, while focusing more on the results obtained with the newly proposed model.

### 4. Experimental results

Numerous details have an effect on the outcome of an epidemic, including attributes describing the disease, population demographics, countermeasures taken to slow down the spread of the disease along with geographical factors. Even slight differences in some of the parameters may drastically change the outcome of an epidemic. Varying the starting parameters of the simulated environment, it is possible to determine which of them have an effect on the longevity and impact of the disease.
4.1. Baseline simulation

Our previous work (Ilić & Ivanović, 2021) showcased simulations of 2 different diseases varying in only their deadliness. Other parameters were fixed, and no government measures have been introduced. All agents in the simulated population have also been placed into a single risk group. The aim of this simulation was to show how the deadliness of a pathogen impacts its longevity in the population.

The simulated environment consisted of a $10 \times 10$ grid containing 500 agents. The first disease had a chance of death equal to 2.1%, which is what the rate of death in confirmed COVID-19 cases was in April 2021, according to the World Health Organization. Having the chance of death of the second disease higher, at 7%, it is possible to easily see the effect this has on its longevity.

Figures 2 and 3 show the course of both diseases during the 100 simulation steps. It is evident that the first disease managed to last longer in the population, with the more deadly variant eliminating most of its hosts early on.

Table 1 shows clearly that the first variant of the disease persisted until the end of the simulation, while the more deadly variant was eliminated by step 50.

More details on performed experiments can be found in Ilić and Ivanović (2021).

4.2. Spatial SIRD simulation with different risk groups

This second simulation performed on the newly proposed model aims to determine how much of a negative effect does decreased compliance to COVID-19 safety measures has on the population. For this purpose, a population of 10 000 agents was divided into risk groups based on age. The ratio of agents in each risk group was modelled according to the demographics of Serbia\(^1\) and each group had a chance of death after contracting the disease as per the statistics gathered by the Centers for Disease Control and Prevention.\(^2\) The number of agents in each risk group and their associated risks of death can be seen in Table 2.

Multiple simulations were run with varying parameters of disease infectivity and population compliance. The chance of a host passing the disease to a healthy agent ranged from 2% to 15%, with a step of 0.1%, while the percentage of the population which

![Figure 2](image_url). The timeline of the disease during the 100 simulated steps with a mortality rate of 2.1%.
follows government measures was varied between 60% and 100%. For all simulations, government measures were put in place once infections reached 1% of the population. After 100 simulated steps, a count of deceased agents by age group was conducted.

The total number of deceased agents can be seen in Figure 4. The number of deceased agents rises along the infectivity axis, while accelerated increase can be seen when significant portions of the population do not follow government regulations set in place to combat the spread of infections.

Categorizing members of a population into different risk groups gives the added advantage of accurately portraying the impact the disease has on different layers of the population. The older population is most at risk, which can be seen in Figure 5. This figure shows the number of deceased agents belonging only to the highest risk group, agents older than 64, while Figure 4 represents the total number of deceased agents across all risk groups. Most of the casualties come from this specific group.

By decreasing the fraction of the population following proposed safety precautions, the number of casualties for the oldest population drastically increases. As this is the group that is most at risk of the disease, even a slight increase in the number of contacts with agents noncompliant with safety protocols may result in them contracting the disease, which in turn puts them at more risk of succumbing to the disease. In some of the worst-case scenarios simulated, having the lowest values for population compliance and highest infectivity, a staggering 4% of the entire risk group is lost during the epidemic.

**Figure 3.** The timeline of the disease during the 100 simulated steps with a mortality rate of 7%.

| Step # | Infected Count (2.1%) | Infected Count (7%) |
|--------|------------------------|---------------------|
| 0      | 5                      | 5                   |
| 5      | 12                     | 12                  |
| 10     | 5                      | 5                   |
| 15     | 11                     | 11                  |
| 20     | 12                     | 14                  |
| 30     | 25                     | 6                   |
| 40     | 42                     | 3                   |
| 50     | 39                     | 0                   |
| 99     | 4                      | 0                   |

**Table 1.** Number of infected agents at various steps for both simulations.
Figure 6 plots the number of deceased agents from each risk group against the fraction of the population following safety measures in simulations with the highest infectivity value. The number of deceased agents in the low and medium risk groups are relatively steady, showing a slight improvement with increased measures. Conversely, looking at the oldest risk group, a significant fall in the number of deceased agents can be seen. The maximum number of losses was 77 agents during the simulation conducted with 61% of agents complying with measures. With 100% of the population following safety precautions, this number drops down to 12 losses.

By looking at the COVID-19 pandemic, it is not uncommon to see individuals who have recovered from the disease become infected again later on. This may happen due to multiple reasons, such as them not developing enough antibodies to fend off a subsequent infection, or the virus may have mutated and infected the individual again. This effect can be achieved by modifying the simulation in such a way that agents no longer become immune but rather have a lower chance of infection once they recover from the disease. In this expanded simulation, recovered agents only have a third of the chance for a subsequent infection, which mimics what can be observed in COVID-19 cases in Serbia. The possibility for later infections drastically increased the number of deceased agents. Simulations with the lowest compliance rates show a steep rise in the number of casualties which can be seen in Figure 7.

This experiment aimed to explore the relationship between disease infectivity and compliance to safety measures, and the effects these two variables have on the spread of the disease and its impact on the population. It is shown that the negative effects of a disease are amplified when it is more easily transmitted. By spreading to more hosts, the number of possible casualties increases, which in turn increases the total death toll, especially among the highest risk groups. The transmission rate can be lessened by

| Risk Group     | Agent Count | Risk of Death (%) |
|----------------|-------------|-------------------|
| Under 40       | 4592        | 0.076             |
| 40 to 64       | 3663        | 1.009             |
| 65 and Over    | 1745        | 9.726             |

Table 2. Risk groups in the advanced simulation.

Figure 4. Surface plot of the number of deceased agents against different values of infectivity and compliance to government measures. The x-axis represents the fraction of the population complying to government measures, while the y-axis represents the infectivity.
Figure 5. Surface plot of the number of deceased agents aged 65 and over against different values of infectivity and compliance to government measures. The $x$-axis represents the fraction of the population complying to government measures, while the $y$-axis represents the infectivity.

Figure 6. The impact safety measures have on the number of casualties in each risk group in simulations with the highest infectivity.

Figure 7. Surface plot representing the total number of deceased agents in simulations where recovered agents are not immune, but rather have a decreased chance of subsequent infections. The $x$-axis represents the fraction of the population complying with government measures, while the $y$-axis represents the infectivity.
introducing and carrying out certain epidemic measures. The positive effects of following safety protocols can more easily be seen in instances where the disease is highly transmittable.

5. Conclusion

Software agents are a powerful mechanism for simulating different processes in a wide range of domains. Given enough information of some real-world system, it is possible to simulate how it can evolve through time and how it will react to different starting conditions and different events which could unfold. Even subtle changes in starting parameters could give wildly different results. Systems such as proposed in Hackl and Dubernet (2019) which take into account fine details can help researchers get a better sense of the problems they face.

The value and versatility of software agents is further bolstered by the number of existing programming languages and tools which assist in working with them. These tools can cover a wide variety of use, with some being general purpose and others being more suited for specific fields of research.

This paper utilizes a newer general purpose agent framework (Masad & Kazil, 2015) in order to explore the impact an epidemic can have on a population based on the characteristics of the disease in question by making use of a spatial SIRD model. It also explores how government measures diminish the damage a disease can inflict on a population, along with negative effects of non compliance to these measures.

Various traits and parameters were included in the conducted experiments. With the inclusion of details such as subsequent infections, as seen in the real world, it is possible to run more intricate simulations and explore various outcomes. Fine tuning these parameters may give crucial insight of what can be expected during an epidemic.

Notes

1. Provided by the Statistical Office of the Republic of Serbia (https://www.stat.gov.rs/en-US/).
2. https://covid.cdc.gov/covid-data-tracker/#demographics (as of December 2021).

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The authors acknowledge financial support of the Ministry of Education, Science and Technological Development of the Republic of Serbia [grant number 620 451-03-68/2020-14/200125].

Notes on contributors

Mihailo Ilić currently works as a Teaching Assistant at the Faculty of Sciences, University of Novi Sad, Serbia, where he is also pursuing his PhD. His main fields of interests are agent technologies and machine learning, with a focus on federated learning techniques. As a member of the ASCAPE
H2020 project, his work is focused on the implementation of federated learning in the medical domain. In addition, Mihailo has notable experience from working in industry as both a software developer responsible for system design and a data scientist in the field of smart agriculture.

Mirjana Ivanovic holds the position of Full Professor at Faculty of Sciences, University of Novi Sad, Serbia. She is a member of National Scientific Committee for Electronics, Telecommunication and Informatics within Ministry of Education, Science and Technological Development, Republic of Serbia. She was a member of University Council for Informatics for more than 12 years. Prof. Ivanovic is author or co-author of 14 textbooks, several international monographs and more than 450 research papers, most of which are published in international journals and conferences. Her research interests include agent technologies, intelligent techniques, applications of data mining and machine learning techniques in medical domains and technology enhanced learning. She is member of Program Committees of more than 300 international conferences, Program/General Chair of several international conferences, and leader of numerous international research projects. Mirjana Ivanovic delivered several keynote speeches at international conferences and visited numerous academic institutions all over the world as visiting researcher (Germany, Slovenia, Australia, China, Korea). Currently she is Editor-in-Chief of the Computer Science and Information Systems journal. E-mail: mira@dpi.uns.ac.rs

ORCID

Mirjana Ivanović http://orcid.org/0000-0003-1946-0384

References

Acemoglu, D., Chernozhukov, V., Werning, I., & Whinston, M. D. (2020). A multi-risk sir model with optimally targeted lockdown (Vol. 2020). National Bureau of Economic Research.

Adamu, H. A., Muhammad, M., Jingi, A., & Usman, M. (2019). Mathematical modelling using improved SIr model with more realistic assumptions. International Journal of Engineering and Applied Sciences, 6(1), 64–69.

Agentscript project page (n.d.). Retrieved April 15, 2021, from https://github.com/backspaces/agentscript/.

Anylogic website (n.d.). Retrieved April 15, 2021, from https://www.anylogic.com/features/.

Bădică, A., Bădică, C., Ganzha, M., Ivanović, M., & Paprzycki, M. (2020). Multi-agent simulation of core spatial sir models for epidemics spread in a population. In 2020 5th IEEE international conference on recent advances and innovations in engineering (icraie) (pp. 1–7). IEEE.

Bădică, C., Budimac, Z., Burkhard, H. D., & Ivanovic, M. (2011). Software agents: Languages, tools, platforms. Computer Science and Information Systems, 8(2), 255–298. https://doi.org/10.2298/CSIS110214013B

Bellifemine, F. L., Caire, G., & Greenwood, D. (2007). Developing multi-agent systems with jade (Vol. 7). John Wiley & Sons.

Bisin, A., & Moro, A. (2020). Learning epidemiology by doing: The empirical implications of a spatial-sir model with behavioral responses (Tech. Rep.). National Bureau of Economic Research.

Cormas website (n.d.). Retrieved April 15, 2021, from http://+cormas.cirad.fr/indexeng.htm.

Fernández-Villaverde, J., & Jones, C. I. (2020). Estimating and simulating a SIRD model of covid-19 for many countries, states, and cities (Tech. Rep.). National Bureau of Economic Research.

Hackl, J., & Dubernet, T. (2019). Epidemic spreading in urban areas using agent-based transportation models. Future Internet, 11(4), 92. https://doi.org/10.3390/fi11040092

Hewitt, C., Bishop, P., & Steiger, R. (1973). A Universal Modular Actor Formalism for Artificial Intelligence. IJCAI. In Proceedings of the 3rd international joint conference on artificial intelligence (pp. 235–245). San Francisco, CA: Morgan Kaufmann Publishers Inc.
Ilić, M., & Ivanović, M. (2021). Modeling an epidemic-multiagent approach based on an extended SIR model. In International conference on computational collective intelligence (pp. 141–153). Cham: Springer.

Jacamo project website (n.d.). Retrieved April 15, 2021, from http://jacamo.sourceforge.net/.

Java agent development framework, X. (n.d.). Retrieved April 15, 2021, from https://jade.tilab.com.

Karnyoto, A. S., Sun, C., Liu, B., & Wang, X. (2021). Transfer learning and GRU-CRF augmentation for COVID-19 fake news detection. Computer Science and Information Systems, 53–53. https://doi.org/10.2298/CSIS210501053K

Madp project website (n.d.). Retrieved April 15, 2021, from https://github.com/MADPToolbox/MADP.

Masad, D., & Kazil, J. (2015). MESA: An agent-based modeling framework. In 14th python in science conference (pp. 53–60).

Pal, C. V., Leon, F., Paprzycki, M., & Ganzha, M. (2020). A review of platforms for the development of agent systems. arXiv preprint arXiv:2007.08961.

Repast website (n.d.). Retrieved April 15, 2021, from https://repast.github.io/.

Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition. Wiley Interdisciplinary Reviews: Cognitive Science, 10(3), e1488. https://doi.org/10.1002/wcs.1488

Shoukat, A., & Moghadas, S. M. (2020). Agent-based modelling: An overview with application to disease dynamics.

Vrbačič, G., Pečnik, Š., & Podgorelec, V. (2021). Hyper-parameter optimization of convolutional neural networks for classifying COVID-19 X-ray images. Computer Science and Information Systems, 19(1), 327–352. https://doi.org/10.2298/CSIS210209056V

Weerasooriya, D., Rao, A., & Ramamohanarao, K. (1994). Design of a concurrent agent-oriented language. In International workshop on agent theories, architectures, and languages (pp. 386–401). Berlin, Heidelberg: Springer.

Wolfram systemmodeler website (n.d.). Retrieved April 15, 2021, from https://www.wolfram.com/system-modeler/.

Wooldridge, M., & Jennings, N. R. (1994). Agent theories, architectures, and languages: A survey. In International workshop on agent theories, architectures, and languages (pp. 1–39). Berlin, Heidelberg: Springer.

Wooldridge, M. J., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. The Knowledge Engineering Review, 10(2), 115–152. https://doi.org/10.1017/S0269888900008122

Zheng, L., Yang, J., Cai, H., Zhou, M., Zhang, W., Wang, J., & Yu, Y. (2018). Magent: A many-agent reinforcement learning platform for artificial collective intelligence. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32).