Deliverable Report

D6.1 Agent-based simulator for synthetic data generation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 882828. The sole responsibility for the content of this document lies with the author and in no way reflects the views of the European Union.

Document information and contributors

Deliverable No.	D6.1	Work Package No.	WP6	Task/s No.	Task 6.1
Work Package Title	DATA ANALYSIS AND INTERPRETATION ON PROFILES FROM POTENTIAL YOUNG VICTIMS AND OFFENDERS				
Linked Task/s Title	T6.1 Agent-based modelling and synthetic data generation				
Status	Final	(Draft/Draft Final/Final)			
Dissemination level	CO	(PU-Public, PP, RE-Restricted, CO-Confidential)			
Due date deliverable	30/09/2022	Submission date	30/09/2022		
Deliverable version	1.0				
Deliverable responsible	COMILLAS				

Contributors	Organization	Reviewers	Organization
Jaime Pérez	COMILLAS	M. Alvarez-Campana	UPM
Mario Castro	COMILLAS	Sonia Solera	UPM
Gregorio López	COMILLAS	Aivars Bērziņš	TILDE
		María Reneses	COMILLAS
		Abel Muñiz	ZABALA
		Violeta Vázquez	ZABALA

Document History

Version	Date	Comment
0.1	06/07/2022	1st draft – Definition of structure and content
0.2	05/09/2022	First complete draft
0.3	21/09/2022	Improved draft ready for review
1.0	30/09/2022	Final version including reviewers' comments

TABLE OF CONTENT

List of Abbreviations

Abbreviation	Description
ABM	Agent-Based Models
Ada	AdaBoost
AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
BN	Bayesian Network
DAC	Direct Acyclic Graph
DL	Deep Learning
DT	Decision Tree
HDI	Highest Density Interval
IRT	Item Response Theory
KNN	k-Nearest Neighbours
LSTM	Long Short-Term Memory
LR	Logistic Regression
ML	Machine Learning
MLP	Multi-Layer Perceptron
NLP	Natural Language Processing
RF	Random Forest
RTS	Risky to Safe
STR	Safe to Risky
WP	Work Package

Executive Summary

Task 6.1. "Agent-based modelling and synthetic data generation" is the first task of WP6 in the RAYUELA project. Therefore, this deliverable represents the first output of this essential WP, which focuses on:

- Process and interpret the data gathered using the serious game developed in WP3 via Bayesian data analysis algorithms and Machine Learning techniques.
- Build an ad hoc model to generate synthetic data to enable fine-tuning all the developed data analysis algorithms and techniques and increase the volume of viable available data.
- Obtain key insights to set the foundations for evidence-based recommendations, guidelines, policies and measures in WP7.

In this deliverable, an agent-based model is created based on the previous research and game development efforts carried out in WP1, 2 and 3. The primary objective of this simulator is to calibrate the algorithms and techniques that will be used in later stages of the project, as well as to generate synthetic data to increase the amount of available data in the project. Therefore, these synthetically generated data, combined with the incoming data from the pilot studies in WP5, will serve as baseline to perform the analysis described in Tasks 6.2 and 6.3.

The deliverable is organised as follows:

Section 1 introduces the context and motivation for this work within the RAYUELA project, together with the objective and summary of the content.

Section 2 discusses synthetic data's concept and its rise (and causes) in recent years. We also discuss the main types of synthetic data and introduce some key concepts to understand the work developed.

Section 3 represents the early stages of simulator development, outlining design considerations and the proposed architecture.

Section 4 details the implementation of each part of the simulator, explaining the proposed agent models and the approach used to take advantage of external information (e.g., expert knowledge and prevalence data).

Section 5 shows some examples of datasets generated with the simulator. Moreover, we perform identifiability tests that aim to ensure the usefulness of the data generated by training Bayesian and Machine Learning models.

Finally, Section 6 draws the main conclusions from the work carried out.

The code developed in this work is open and freely available in the GitHub repository of the RAYUELA project[1].

[1] GitHub repository of the RAYUELA project (Simulator): https://github.com/rayuelaproject/Simulator_Synthetic-Data

1. Introduction

In developing the serious game of the RAYUELA project, we face several challenges: **data scarcity** (even if the number of players is large, the prevalence of those players as potential victims or aggressors is expected to be small), matching between real-life attributes and the player attitudes during the game or the intrinsic behavioural and socio-economical variability of the players. This will lead to downstream problems such as **miscalibration** of parameters in the serious game itself or in the inference algorithms that will be used afterwards. However, this problem is not new to many fields of research, and there are several ways in which we can address it.

In an ideal scenario, we would have access to detailed information about the players so their choices during the game could be matched to the information gathered from the players. This is not possible (and that is precisely one of the reasons why we are developing the RAYUELA game) so, without knowing the "ground-truth" about the players, calibrating the statistical analysis is far from being trivial.

To address this problem, in this task we introduce an **agent-based simulator** that **mimics participants' behaviour**, also enabling to incorporate the **knowledge** gained from **previous Work Packages** (WP) research. The agents of this simulator follow (probabilistic) rules that attempt to capture the behaviour of players in the serious game, parameterising them according to different **sociological** or **psychological profiles**. In addition to addressing the problem of data scarcity, it will also be possible to generate large **synthetic datasets** that, among other uses, will drive and motivate research in this area.

The data generated in this task through the use of the simulator will serve as input for task T6.2, which will develop the feature selection methods and the training of Machine Learning (ML) algorithms thus providing a safe monitorable test bed for the algorithms before having data from actual players.

The remainder of the document is organised as follows. **Section 2** introduces core concepts of synthetic data generation and simulation. **Section 3** discusses design considerations for the simulator and explains the proposed general architecture. **Section 4** further details the technical implementation of the modules that compose the simulator. **Section 5** explains the generated datasets and performs some inference tests on them. Finally, **Section 6** draws the main conclusions from the conducted work.

2. Background

Player modelling aims to detect, predict and characterize the human player attributes that manifest while playing a game [1]. These models can be any mathematical representation, rule set or probability set that maps parameters to observable variables and are built on dynamic information obtained during game-player interaction. On the other hand, player profiling usually refers to categorizing players based on static data that does not alter during gameplay (e.g., personality, cultural background, gender, age). Despite their dissimilarities, these two concepts can complement each other, contributing to more reliable player models.

The main objective of studying game players is to understand their cognitive, affective, and behavioural patterns. Recent advances in AI have demonstrated an impressive ability for these same goals that player modelling sets out to achieve. However, at the moment, there is a significant lack of interpretability in complex models. Thus, the use of Artificial Intelligence (AI) for player modelling is a perfect fit solely when explainability is not a hard constraint.

Hooshyar et al. [2] conducted a systematic literature review that deeply analyses the computational and data-driven techniques used for player modelling between the period of 2008 to 2016. As this is such a broad and promising field, the variety of algorithms used is immense: descriptive statistics and correlations, path/network analysis, supervised learning (e.g., Neural Networks, Linear Regression, Hidden Markov Models, Decision Trees), unsupervised learning (e.g., k-means, Linear Discriminant Analysis, Self-Organizing Map), probabilistic algorithms (e.g., Bayesian / Markov Networks), evolutionary methods (e.g., Genetic algorithms), reinforcement learning methods (e.g., Multi-armed bandits), etc. Most of the computational methods used are model-free, meaning they do not impose strict assumptions on the model. However, there are also some model-based approaches (e.g., Bayesian hierarchical models) [3] [4] that yield more interpretable and explicit models (e.g., psychological or cognitive) than those which do not impose strict assumptions on the model. For instance, we can infer the player's hidden attributes or mental states with such models.

In academia, especially in Psychology, experiments have been conducted using games (serious and non-serious) for research, but primarily focusing on analysing how the player's personality manifests in the gameplay patterns [5] [6] [7] [8] [9]. However, studying psychological characteristics or phenomenology using serious games seems an up-and-coming field, especially if we introduce AI techniques into the equation.

Synthetic data consists of artificially generated information that is not produced by real-world events. Interest in synthetic data has grown enormously in recent years, driven by two simultaneous trends. The first is the **demand for large amounts of data** to build and train AI and ML models. In certain domains, this data may be challenging to find, expensive, or involve privacy issues for users (e.g., medical imaging, data from minors). Besides, providing more efficient access to data is essential for developing more reliable data-driven models and improve statistical analysis. The second trend involves recent work that has **demonstrated effective methods** for generating (at scale) high-quality, useful synthetic data. Companies like NVIDIA [10], IBM [11] [12], Google [13], and agencies such as the US Census Bureau [14] have adopted different data synthesis methodologies to support model building, application development, and data dissemination. The **most typical uses** of synthetic data are: testing/calibration of systems or software, training of AI/ML models, and data governance (e.g., helping to reduce biases present in real data, to be able to share data while preserving privacy).

We can identify **two main process** to generate synthetic data: **simulation** and **data augmentation**. A simulation consists of imitating the operation of a real-world process or system over time through the use of models [15]. The model represents the behaviour and features of the simulated system or some of its constituent parts. Data augmentation, on the other hand, consists of increasing the amount of data available by making (slightly) modified copies of existing data, maintaining similar statistical properties. That is, it implies the existence of real data and is limited by it, although in the ML field it is a widely used technique to regularise and reduce overfitting [16].

Within the field of simulation lies the paradigm of **Agent-Based Models** (ABM), which consists of simulating the actions and interactions of autonomous agents to understand a system, the interactions between such a system and the agents, or emergent rules. ABMs are used in a wide variety of scientific domains including Biology, Epidemiology, Ecology, Economics, and Social Science [17].

In this work, we implement an ABM simulator that allows us to start calibrating and testing the ML models that will be used later on with real data gathered through RAYUELA serious game. To this end, we have developed a realistic model of how participants will interact with the game. In the following sections, we go on to explain the overall proposed architecture and the details of the technical implementation.

3. Simulator Design

3.1 Design Considerations

This simulator's primary objective is to generate **synthetic data** to alleviate the problem of **data scarcity**. To this end, it seeks to generate realistic data to assist in the **calibration** of in-game parameters and inference algorithms. In addition, once actual player data is available, it can provide support for potential problems such as class imbalance in the data or serve as a data augmentation method.

In the serious game of RAYUELA, players will have to make **decisions** through which the story will progress. The main hypothesis of the RAYUELA project is that the decisions made by the minors during the game will be a projection of their **online behaviour (or risk propensity online)** in terms of certain sociological/psychological factors (e.g., tendency to isolation, family communication, time spent online, impulsivity, etc.).

Mimicking this human behaviour requires finding a sound parametrisation (namely, defining a vector of the agent attributes matching the player's internal state) that also accommodates individuality and variation among players. Defining such parametrisation is a challenge as, unlike in traditional (fun-oriented) games, our goal here is not "winning" [18]. In other words, our agents will not act by optimising a reward function, but rather following field-specific informed rules, in an approach more similar to [19] [20].

Specifically, the agents will have a parameterised **risk profile** and a **decision-maker** which, depending on the current state of the game, will decide which option to take at any given moment. Agents may also include a **state perception** module, which attempts to represent the player's imperfect collection of information in each game state.

While generating synthetic agents, we can also produce synthetic **demographic data** (e.g., age, gender, sexual orientation, etc.). However, for such demographic data to be consistent with reality, we must be able to include **expert knowledge** and **prevalence data**. It is therefore essential that the simulator has some mechanism to include this external information. A suitable example of this mechanism could be **causal models**, as they are represented through graphical models that depict the data generation process. Thus, it fits nicely into the generative or simulation mindset.

While playing, each player will be presented with a conversation/text description and a list of available actions. The serious game will be represented by an **environment** module in the simulator, in which the player finds himself in one **state** and depending on the chosen decisions, he/she will find himself in a different state. The simplest case is a series of sequential states that, regardless of the decision made, are always consecutive and in the same order.

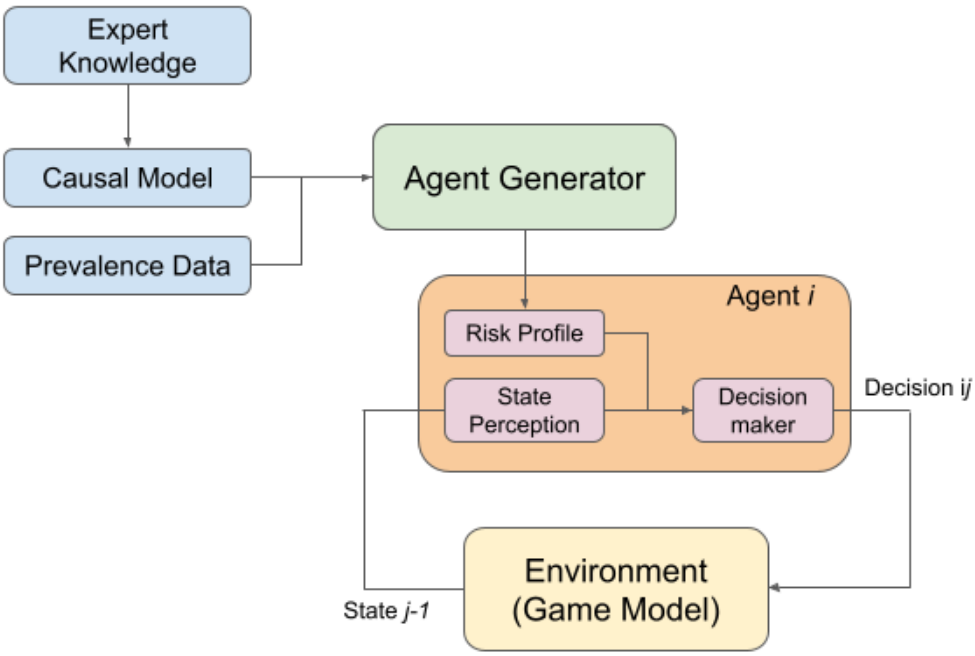


Figure 1. Simulator's general architecture and components. The agents (orange) respond to the environment (yellow) and act according to its risk profile, which is defined in a non-deterministic way by external information (blue).

3.2 General Architecture

Taking into account all the technical and design considerations outlined in the previous section, in Figure 1 we summarise schematically a generic agent design. The modularity of the proposed architecture allows us to change specific features of the simulator in the future without the need for a deep redesign (e.g., in case we want to create a new agent model, we would only have to modify that module).

Four main components are found in this design:

- **External Information**: [*blue*] The expert knowledge, in our case obtained from WP1 and WP2, converges in a causal model. This model, together with statistical prevalence data depending on the cybercrime to be considered, serves as input for the agent generation.
- **Agent Generator**: [*green*] This module is in charge of generating agents with distinct parameters representing varied sociological/psychological profiles. This generation process is driven by the external information of expert knowledge and prevalence data. Therefore, it can generate an intentional imbalance in the data or causal connections between the demographic data and the profiles, which faithfully represent reality.
- **Agent**: [*orange*] The output of the agent generation module is a series of synthetic agents that represent the players, consisting of three submodules:
    - Risk Profile: Value of the parameters defining the behaviour of each agent.
    - Decision-Maker: This submodule represents the sampling of a weighted random choice between the possible decisions. The agents will tend to take decisions aligned with their risk profile or traits.
    - State Perception: It would represent the players' imperfect retrieval of information from the text when interpreting game situations, so Natural Language Processing (NLP) models could be used. This is an experimental submodule.

- **Environment**: [*yellow*] This module represents the serious game storyline with which the agents interact. It may have a linear or a decision tree-based structure (Figure 2). It consists of a series of states and the associated possible decisions.

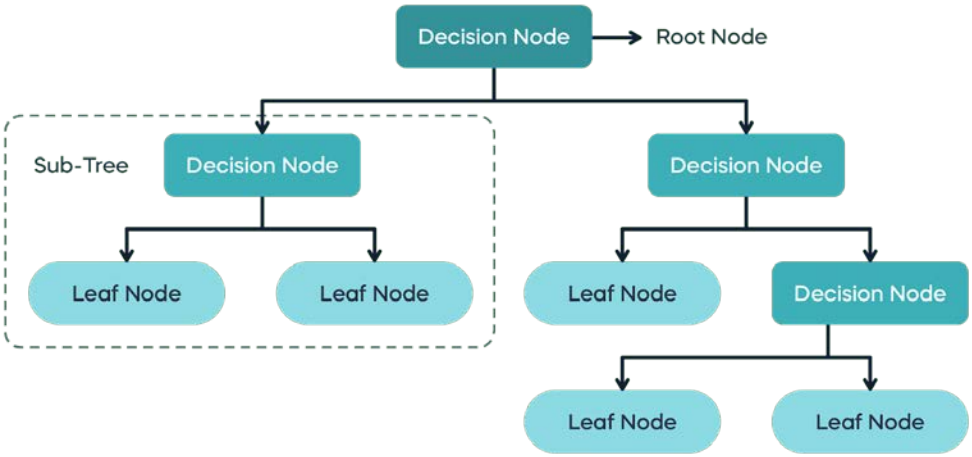


Figure 2. Decision tree structure

4. Simulator Implementation

In this section, we will explain in more detail some of the modules presented in the general architecture (Figure 1) to dive into the technical aspects of the implementation.

4.1 Probabilistic Agent Model

Our probabilistic agent model borrows ideas from the **Item Response Theory** (IRT) [21]. IRT is a paradigm for the analysis and evaluation of tests, questionnaires, or similar instruments measuring abilities, attitudes, or other variables. IRT is often superior to classical test theory for many reasons, but the main one is that, in addition to inferring the "ability" of the participant, it also takes into account the "difficulty" of each question when assessing (and other possible parameters in more complex models). In our particular case, we will use this paradigm to **generate data**, rather than inference players' abilities. In the case of our game, the questions are the analogues of the game decisions, so it is straightforward to extend this framework to our case.

The simplest "game" is modeled by a set of "adventures" in which the player takes dichotomous decisions (two possible answers), where one response will represent the choice with the highest propensity to online risk (i.e., *risky* option), and the other choice with the lowest (i.e., *safe* option). Each agent's responses will be obtained by a random sample, **weighted** by a probability parametrised through Alpha-α and Beta-β:

- Alpha-α ⇒ Represents the agent's "**risk propensity**". Positive α values would represent more risk-prone agents. Negative values would represent agents with lower risk propensity (i.e., safe players). Zero represents a random player (like *"a monkey with a typewriter" [Russell, S. J., Norvig, P (2021). Artificial intelligence a modern approach, 4th edition. Pearson Education, Inc.]*)
- Beta-β ⇒ Represents the "**amount of information**" of risk propensity given by each proposed dilemma/question (i.e. the amount of information that each social/psychological factor measured in the question gives us about online risk propensity). A value of 0 represents null information, and a value of 1 represents perfect information.

Equation 1 shows the formulation of the base (dichotomous) case. The proposed model composes a **probabilistic** (non-deterministic) solution for eliciting agents' responses to the dilemmas or questions presented, including a sociological/psychological profile parameterisation.

Figure 3 shows the variation of the probability values as a function of the α and β parameters. We can appreciate that for values of β approaching 1 (i.e., high amount of information), **positive α** values (i.e., **risky players**) imply a high probability for the agent to choose the risky response. Equivalently, **negative α** values (i.e., **safe players**) imply a low probability of choosing the risky response.

$$\text{Answer}_{ij} = \text{Bernoulli}(p_{ij}),$$
$$p_{ij} = \frac{1}{1+e^{\alpha_i \beta_j}},$$
$$\alpha_i(\text{player ability}) \in \mathbb{R}$$
$$\beta_j(\text{question information}) \in [0, 1]$$

Equations 1. Agent response acquisition model in the base (dichotomous) case. The agents' responses are a sample from a Bernoulli distribution with probability *p*, parameterised by α and β.

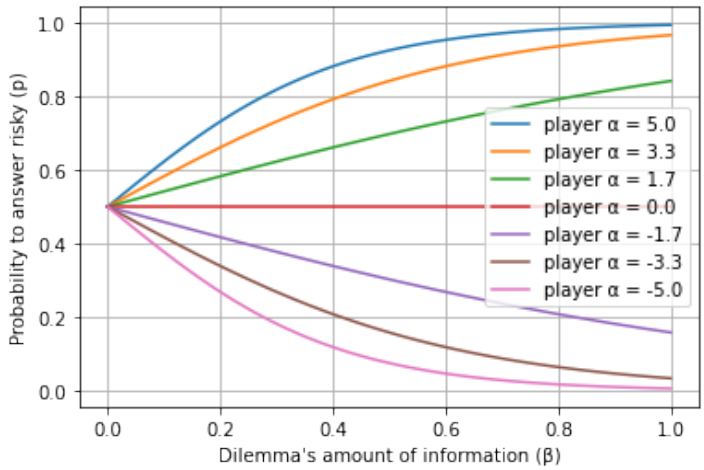


Figure 3. Visualisation of the probabilistic agent model as a function of the Alpha-α and Beta-β parameters, for the base (dichotomous) case. The Y-axis represents the agent's probability of choosing the risky option.

4.2 Memory Agent Model

Complementing the agent model explained in Section 4.1, an additional functionality has been implemented that is intended to represent in some way "the memory about past decisions". There are numerous options for representing a memory of past decisions and/or past risk appetite (Alpha-α). In a general way, it can be formalised in the form expressed in Equation 2, where *M* is a vector whose coefficients represent the relative weight of the past for altering the agent's initial profile (i.e., its risk propensity parameter Alpha-α); the past responses the agent has chosen is represented by *A*; I_n is an all-ones vector; and *n* is the length of the memory (i.e., number of past decisions that affect your next decision)

Depending on the values taken by the vector *M*, the agent with memory can either be reaffirmed in his profile or reach a point of 'profile change' (e.g., his Alpha-α value changes from a negative to a positive value, or vice versa). Therefore, it is possible to model players who start out by making risky decisions and, for some reason, start to play more safely, or vice versa. For instance, it might be the case that a player might get bored with playing "safe" so, after a few right choices, it shifts towards a more risky behaviour.

$$\alpha_k = (I_n - M) \cdot \vec{f}(A_{k-1}, A_{k-2}, \ldots, A_{k-n}) + M \cdot \vec{\alpha}$$

$$\text{where } \vec{\alpha} = \begin{bmatrix} \alpha_{k-1} \\ \alpha_{k-2} \\ \vdots \\ \alpha_{k-n} \end{bmatrix}$$

Equations 2. Mathematical implementation of the memory-based agent, which modifies its Alpha-α parameter based on past decisions and its previous Alpha-α values.

By adding this new functionality to the model presented in the previous subsection, we can obtain a total of five main profiles:

1. **Random**: Characterised by an Alpha-α value of 0, making their choices entirely random.
2. **Safe**: Characterised by negative Alpha-α values, making it a low probability of choosing risky options.
3. **Risky**: Characterised by positive or higher Alpha-α values than the *Safe* profile, making it more likely to choose risky options.
4. **Safe To Risky (STR)**: It uses the memory module, starting from a *Safe* profile, and, after a certain number of answered questions, turning into a *Risky* profile.
5. **Risky To Safe (RTS)**: It uses the memory module, starting from a *Risky* profile, and, after a certain number of answered questions, turning into a *Safe* profile.

4.3 Expert Knowledge and Prevalence Data

One of the most relevant aspects of the simulator presented in this deliverable is the ability to incorporate expert knowledge and prevalence data to make the synthetic data generation process more in line with reality. This feature serves mainly to generate also **demographic** or **personal data** of synthetic agents, based on actual data (e.g., from surveys or reports).

Causal models based on **Bayesian Networks** (BN) represent one of the most natural techniques to apply in the presented use case. BN are a probabilistic graphical representation to model the conditional dependencies between a set of variables via a Direct Acyclic Graph (DAG). BN fit perfectly for taking an occurred event and inferring the likelihood that each possible known causes was the contributing factor. Figure 4 shows a simple example of a BN relating a certain outcome (wet grass) to its possible causes (sprinkler and rain).

The nodes presented in a BN may be observable quantities, latent variables, unknown parameters, or hypotheses, giving extra flexibility for modelling a wide range of events. Moreover, with this technique, we can ask the model "**What would have happened if...?**" and obtain a quantifiable and coherent answer with the available evidence.

Considering the specific use case addressed in RAYUELA, the BN allow us to generate synthetic agents with specific demographic or personal characteristics, and from these, obtaining a probability of belonging to a certain risk profile (i.e., its alpha-α values) driven by actual prevalence data and expert knowledge.

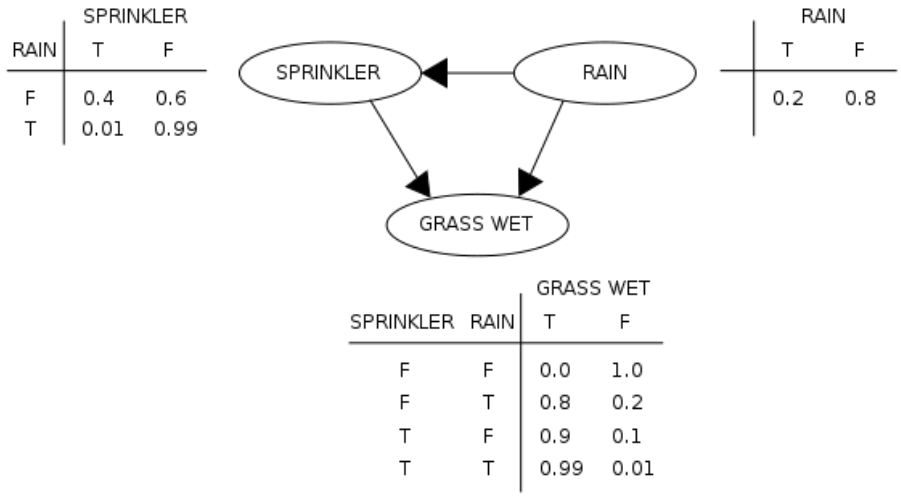


Figure 4. Simple example of Bayesian Network with conditional probability tables. Source: Wikipedia.

The proposed **method** to generate synthetic data informed by expert knowledge can be summarised in the following **steps**:

1. Construct a causal model (BN) hypothesis based on the expert knowledge obtained through the research carried out in previous stages of the project (e.g., in WP1 and WP2).
2. Train the parameters of the proposed BN using real data (e.g., surveys, literature, reports, etc.).
3. Generate synthetic demographic and personal data on actors (e.g., age, gender, sexual orientation, hours of internet use, etc.).
4. From the demographic or personal data generated, derive probabilities of belonging to a specific risk profile using the BN.
5. Use these probabilities to obtain, in a non-deterministic way, the risk profile to which each synthetic agent belongs.

We will now address a practical **example** of causal model design, with cyberbullying as the cybercrime to be analysed, and using as evidence the data collected in the survey of minors carried out in WP1 and WP2. Some of the obtained results in this survey have been reflected in the open deliverables D1.7 and D2.5 of the RAYUELA project. The survey gathered demographic data and risk factors, and subsequently asked whether they had experienced situations related to cyberbullying. Once the data is collected, together with the expert knowledge obtained in WP1 and WP2, we built hypothetical causal models explaining the influence of certain variables on the likelihood of having suffered situations related to cyberbullying. Figure 5 shows an example of these causal model hypothesis.

Once we have a causal model hypothesis, we can use specialised software (e.g., GeNIe[2]) to learn the BN parameters from the survey data. After this BN is constructed and trained, we can employ it to derive the probability of having had close experiences of cyberbullying based on fixed demographics. For example, in Figure 6 we infer that, from the data used, a 15-year-old male, non-heterosexual, that uses the internet for leisure more than 4 hours per day, has a 47% chance of having experienced a situation related to cyberbullying.

[2] Bayes Fusion, GeNIE Modeler. https://www.bayesfusion.com/genie/

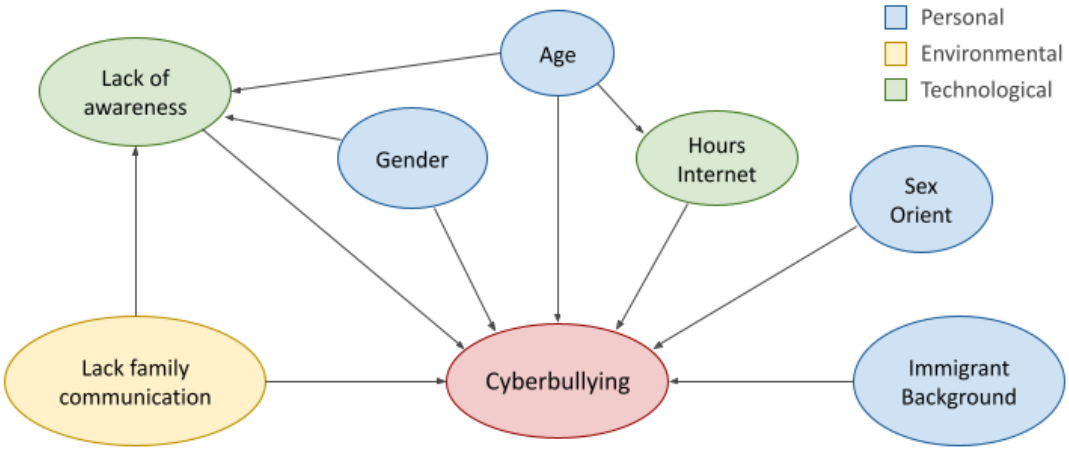


Figure 5. Example of causal model hypothesis (Cyberbullying).

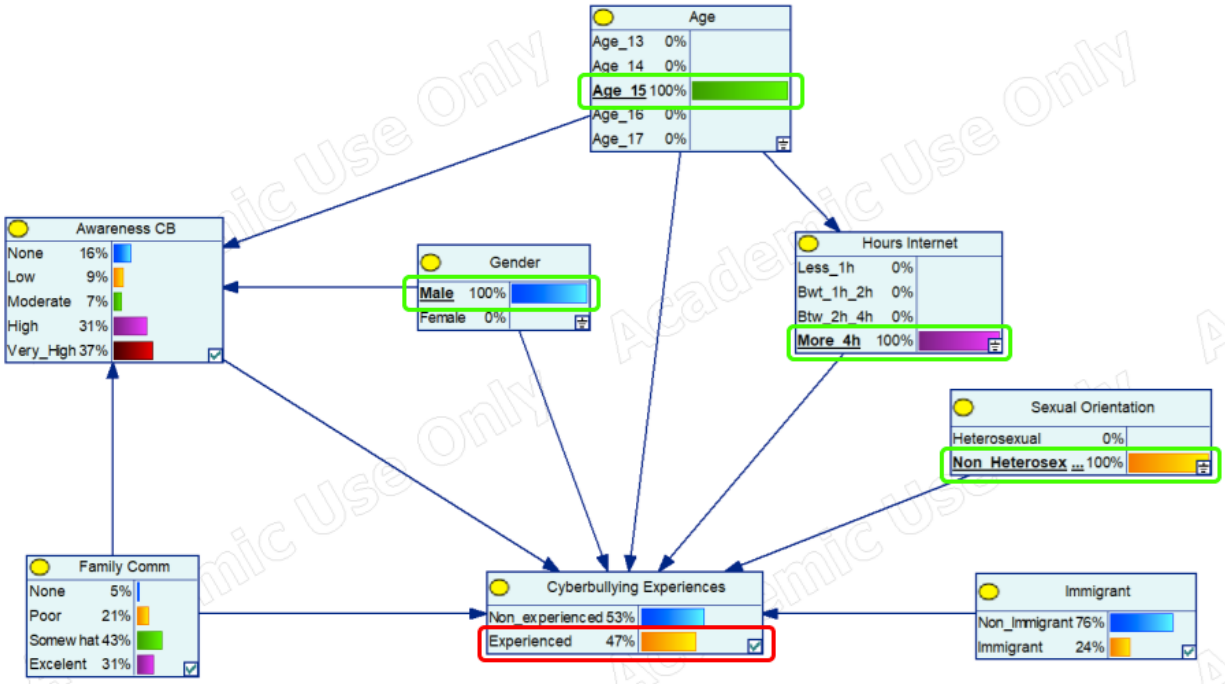


Figure 6. Example of Bayesian Network trained with the survey data, with fixed demographic/personal parameters. Green boxes indicate that the parameter has been fixed to that value. The red box indicates the probability of the output obtained (in this case, having suffered some situation related to cyberbullying).

As mentioned previously, the **probabilities** obtained from the BN will be used by the simulator to obtain in a probabilistic (non-deterministic) way the **risk profile** to which an agent belongs (i.e., its Alpha-α values) given specific demographic/personal characteristics. Equations 2 shows how the simulator uses the BN probabilities to obtain the risk profile of each agent.

$$\alpha_i \sim \mathcal{N}(\mu_{profile}, \sigma)$$
$$\mu_{profile} \sim Categorical([\mu_{risky}, \mu_{safe}], [P_{risky}, P_{safe}])$$

$$P_{risky} = f(Age, Gender, Sex.Orient...)$$
$$P_{safe} = 1 - P_{risky}$$
$$\text{being } f \text{ a Bayesian Network}$$

Equations 2. Computation of each agent's risk profile (i.e. value of α) according to whether they belong to a risky or safe profile. The profile assignment is driven by the probabilities obtained from the Bayesian Network.

The presented method allows us to generate synthetic agents with specific demographic or personal characteristics and obtain a probability of belonging to a risk profile based on real prevalence data, thus creating a **probabilistic** method of **generating synthetic (game) data given certain demographic or personal data, consistently with reality.**

4.4 State Perception

This is an experimental module that aims to model the **imperfect information retrieval** from players. At the time of writing, the RAYUELA game is not in a final version. Therefore, tests have been carried out with a different game, of similar mechanics and objective to the one of RAYUELA. The employed game also relies on a decision-driven narrative, and is focused on the prevention of online grooming. This game was developed at the Universidad Pontificia Comillas (Madrid, Spain) and Figure 7 shows some screenshots taken during the gameplay. In Figure 8 we can observe the decision tree used in the game, representing the days in which this story takes place and the decisions presented in each of them. The ending obtained depends on the decisions made by the player.

Figure 7. Screenshots from the serious game employed to test the "State perception" module, developed at the Universidad Pontificia Comillas (Madrid, Spain).

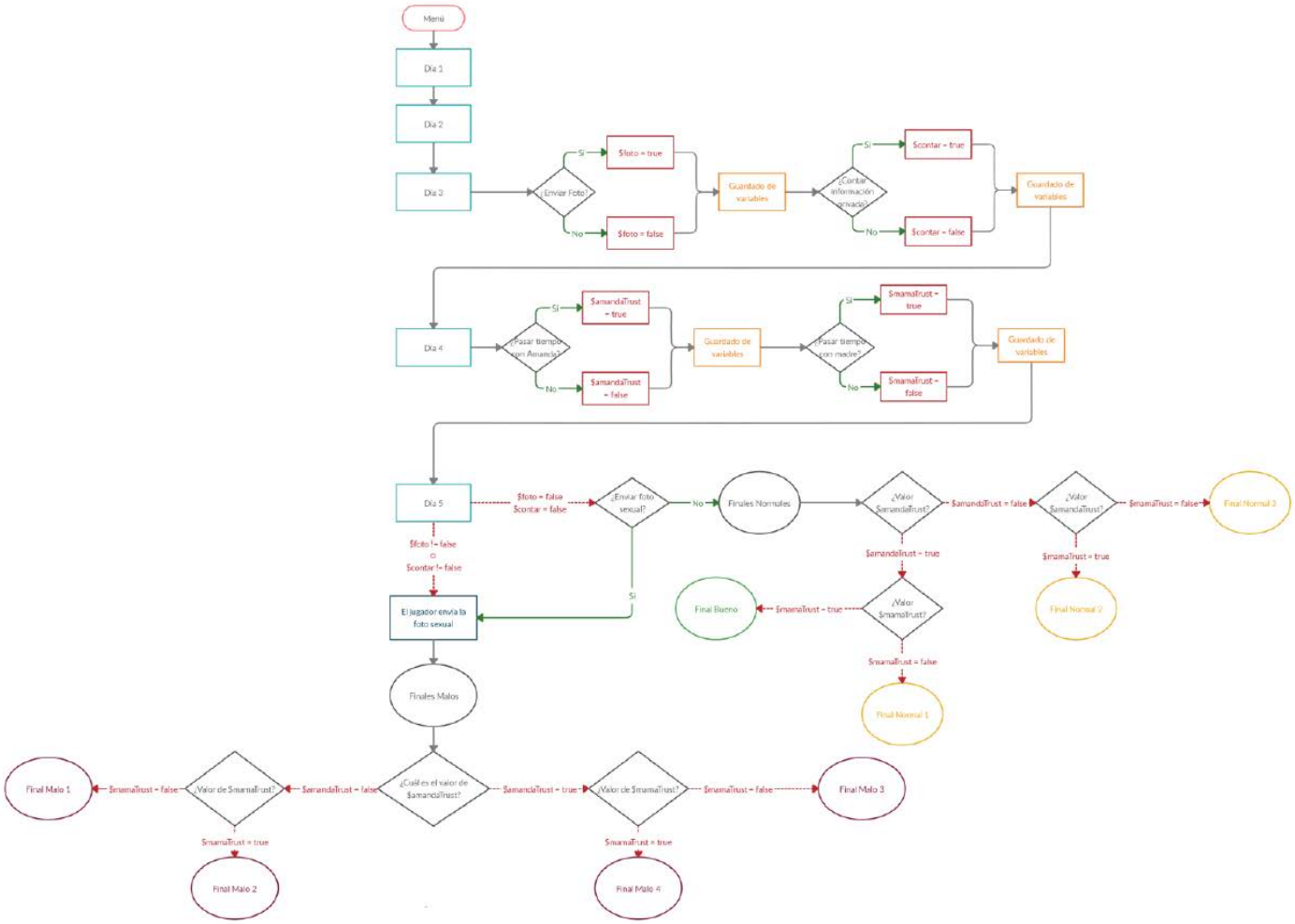


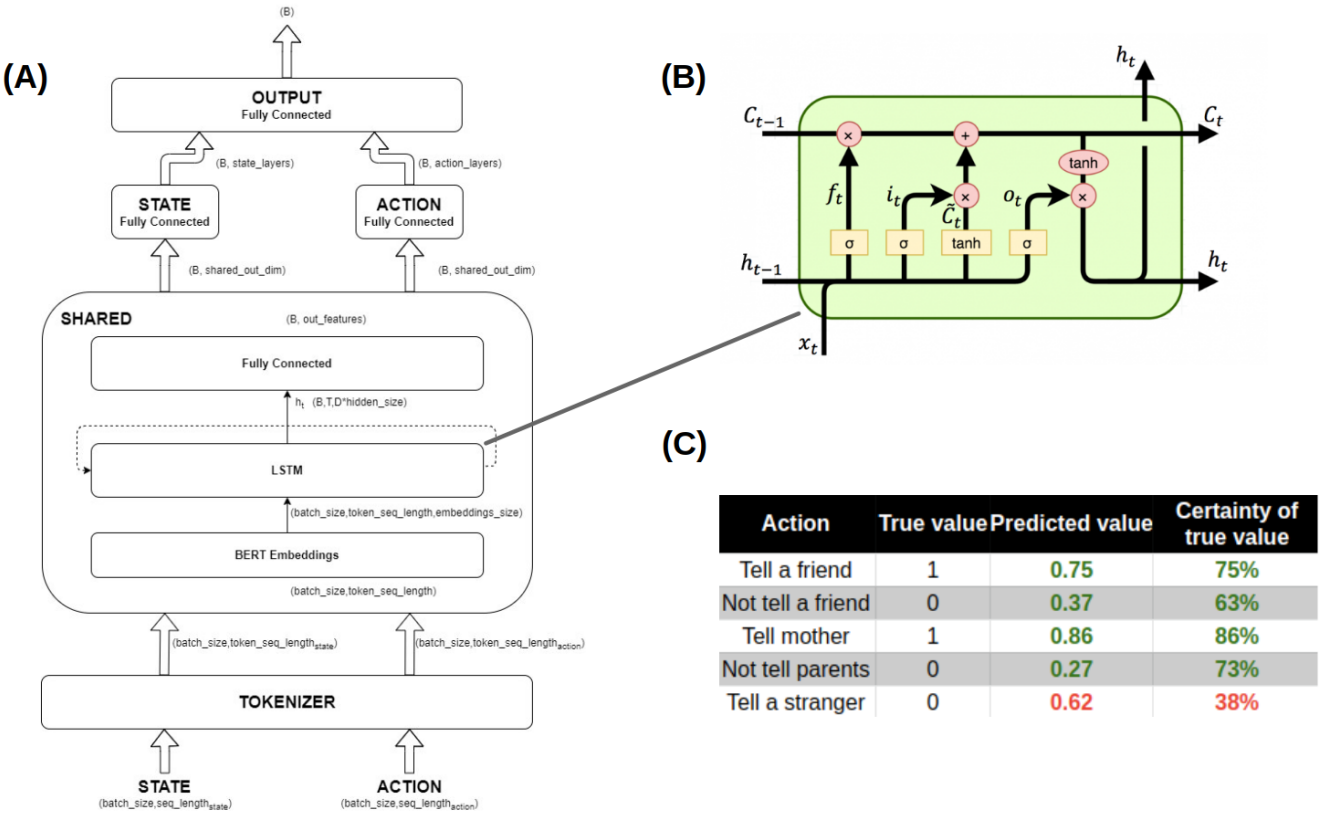
Figure 8. Decision tree from the serious game employed to test the "State perception" module, developed at the Universidad Pontificia Comillas (Madrid, Spain).

To implement this module, we make use of **NLP** techniques. The objective of the module will be to **identify from the text of each dilemma/question which option best fits the agent's profile**. For this purpose, a Fine-tuned Bidirectional Encoder Representations from Transformers (BERT) classifiers [22] and Long Short-Term Memory neurons (**LSTM**) [23] have been employed.

In the preliminary tests we have carried out with the aforementioned game, the LSTM model obtains an accuracy of 0.54 on the validation set and 0.63 on the test set. The fine-tuned BERT obtains an accuracy of 0.75 on the validation set and 0.71 on the test set. The upper bound in the accuracy (around 0.75) may be caused both by the lack of representative risky choices and the ambiguity of natural language (see Figure 9). Although there is still some room for improvement, adult humans might get closer to 100%, but children and adolescents might not be far from that number.

Inspection of the table in Figure 9 (C) shows how our model has learned from the game that talking about the experience to your loved ones, such as your mother or a friend, is a good decision and, not doing so, is a potentially dangerous decision. Moreover, the architecture measures the uncertainty of the outcome: family-related topics are more informative than “stranger”-based content. We believe the main reason this happens is that the situations presented in the game used are not complex enough to learn this distinction correctly. One solution for this issue might be using a game with more complex interactions and a greater variety of decisions.

As mentioned at the beginning, this is an **experimental module** that will have to be **re-trained** once the RAYUELA game is finished. Nonetheless, we believe that it is an interesting line of work for the simulator and for the development of the project.



Action	True value	Predicted value	Certainty of true value
Tell a friend	1	0.75	75%
Not tell a friend	0	0.37	63%
Tell mother	1	0.86	86%
Not tell parents	0	0.27	73%
Tell a stranger	0	0.62	38%

Figure 9. (A) Natural-Language Processing model. The model takes the text from the game scene to obtain relevant information and determine the risk of taking different actions from the current state (B) Example of an LSTM neuron showing the recurrent nature of the architecture. (C) Table containing 5 randomly chosen examples from the test set.

5. Synthetic Data Generation

This section presents some examples of the synthetic datasets generated with the architecture described in the previous sections. In addition, we will perform some basic inference tests to evaluate the identifiability of the generated data, i.e., to test whether the synthetic players created with the simulator are (a posteriori) identifiable using statistical inference algorithms.

5.1 Example of Generated Dataset: Probabilistic Agent and Causal Model

In this example, we have generated a dataset using a simple causal model that has only personal factors (i.e., age, gender, immigrant background and sexual orientation) and no interactions between them. Figure 10 shows the causal model used to generate this first version of the dataset. As we have explained in the previous sections, from the causal model we will obtain a probability for each synthetic agent of belonging to one group of players or another (i.e., safe players or risky player) as a function of the personal factors generated.

Belonging to one group of players or another implies that the agent's Alpha-α value (i.e. its risk propensity) will be sampled from different probability distributions. In this example, the probability distribution of safe players is centred on -2 and has a variance of 0.6. And the probability distribution of risky players is centred on 0.5 and has a variance of 1.2 (Equation 3). Figure 11 shows a visualisation of these two probability distributions, noting that there is an overlap between them.

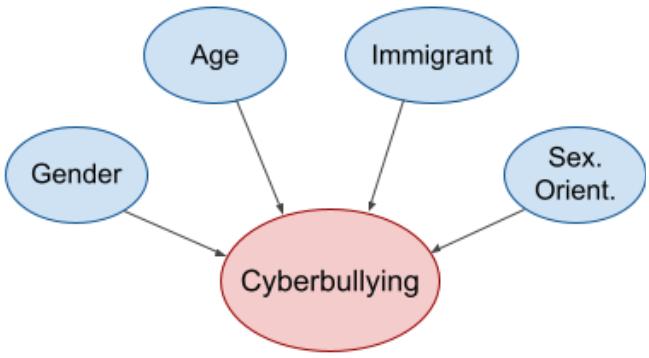


Figure 10. Simple causal model used to generate the first version of the dataset. It only consists of personal factors (i.e., age, gender, immigrant background and sexual orientation).

$$\alpha_{safe} \sim \mathcal{N}(-2, 0.7)$$
$$\alpha_{risky} \sim \mathcal{N}(0.5, 1.2)$$

Equations 3. Sampling of the agents' alpha according to the group to which they belong (safe players or risky players).

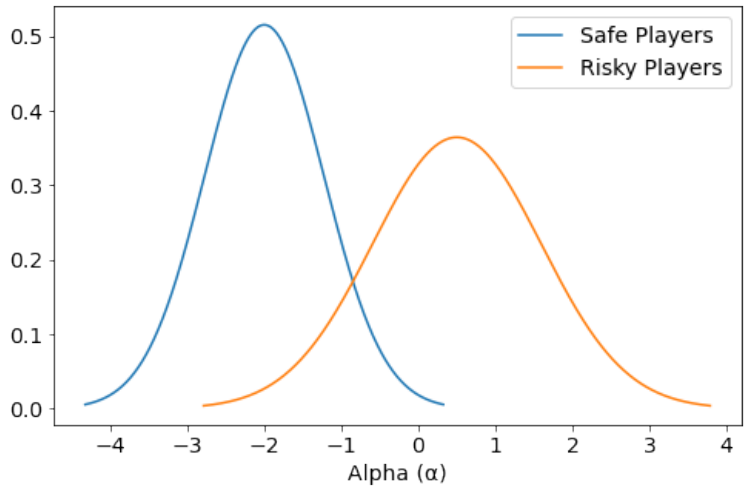


Figure 11. Visualisation of the two probability distributions from which the agents' alpha parameters are sampled.

Afterwards, we generated 500 synthetic agents to participate in a set of 15 sequential dichotomous questions (two possible answers). Figure 12 shows a sample of the generated dataset, where the data for each agent is stored in each row, including the synthetic demographic data (i.e., gender, age, sexual orientation and immigrant background). In the columns of the dataset where each agent's answers are stored, the 1s mean that, in that question, the agent chose the option implying the highest risk propensity. And the opposite with the 0s, the agent has chosen the option implying the lowest risk propensity.

Figure 13 shows a histogram of all the Alpha-α values of the generated agents. We can see that the peaks in this histogram coincide, as expected, with those of the distributions in Figure 11. In Figure 14 we can observe the count of how many players have chosen a specific number of risky responses, by displaying it in a histogram.

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_10	Q_11	Q_12	Q_13	Q_14	Q_15	alpha_risk	risk_label	gender	age	sex_orient	immigrant
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	-2.383001	safe	Male	Age_13	Heterosexual	Non_Immigrant
1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	1	0.675876	risky	Male	Age_15	Non_Heterosexual	Immigrant
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	-2.130556	safe	Male	Age_17	Heterosexual	Non_Immigrant
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	-2.079930	safe	Male	Age_16	Heterosexual	Non_Immigrant
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-3.283155	safe	Female	Age_17	Heterosexual	Immigrant
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
495	1	1	1	1	1	0	0	1	1	1	1	1	0	1	0	1.047410	risky	Male	Age_16	Heterosexual	Non_Immigrant
496	0	1	1	0	1	1	0	1	0	0	0	0	1	1	1	0.403498	risky	Male	Age_14	Heterosexual	Non_Immigrant
497	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	-1.906761	safe	Female	Age_17	Non_Heterosexual	Immigrant
498	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	-1.885603	safe	Female	Age_14	Heterosexual	Non_Immigrant
499	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	-1.730662	safe	Female	Age_13	Heterosexual	Non_Immigrant

500 rows × 21 columns

Figure 12. Sample of the generated dataset: Probabilistic Agent and Causal model (Total = 500 agents).

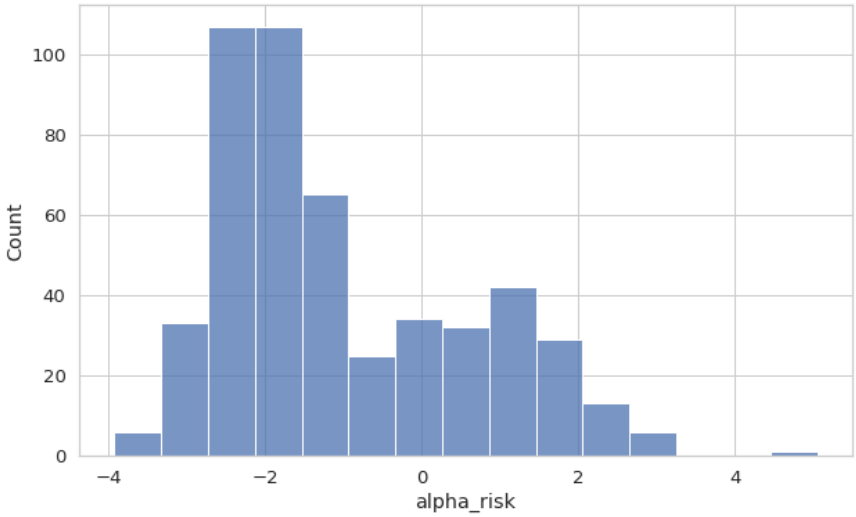


Figure 13. Histogram of all the Alpha-α values of the generated agents.

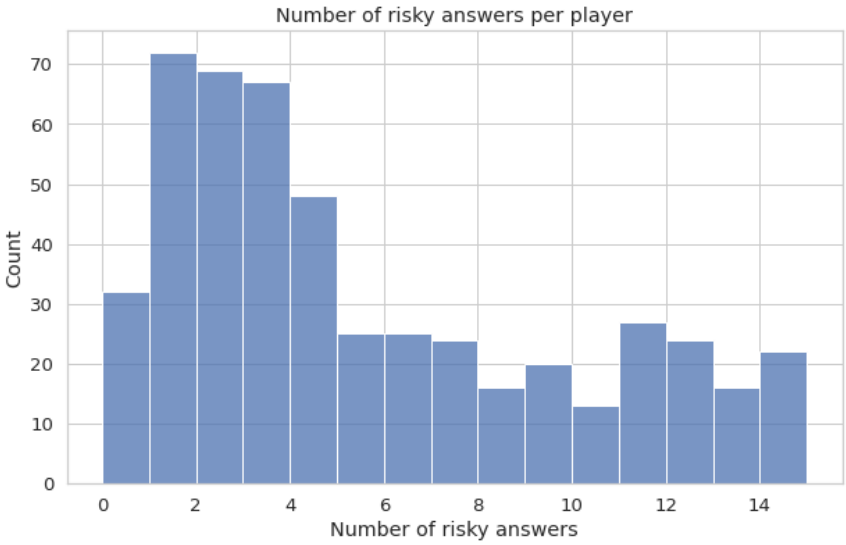


Figure 14. Histogram of how many players have chosen a specific number of risky responses.

5.1.1 Identifiability Test: Bayesian Inference

To evaluate the usefulness and identifiability of the synthetic data generated, in this section, we proceed to use a statistical inference method to reconstruct (a posteriori) the Alpha-α and Beta-β values used. In particular, as our model is entirely probabilistic, the Bayesian structure of the model in companion with the generated data allows estimating the posterior distributions of the parameters used in Equation 1. Namely, from the binary observed responses of each player, we can automatically infer the parameters that plausibly could have generated such evidence. For the implementation, we have used the open source library Stan [24], which is a state-of-the-art platform for statistical modelling and high-performance statistical computation. For these experiments, we have solely used the response data from each synthetic agent, to ensure that this information is robust on its own without the need for the synthetic demographic information.

In Bayesian inference, unlike in the Machine Learning or Deep Learning fields, we do not get a singular number as a result of the prediction. In the Bayesian paradigm, posterior probability distributions are obtained as a result. This obtained probability distribution represents the epistemic uncertainty about the inferred statistical parameter conditional on the collection of observed data. A common metric for prediction quality is the credibility interval, also known as the Highest Density Interval (HDI), which is analogous to confidence intervals in frequentist statistics. For example, if we desire to have a 95% confidence interval, this interval in the posterior probability distribution will be the smallest range that contains 95% of the samples.

Based on this idea, in the experiment conducted, we have considered that **the prediction is correct** when the true value of the parameter falls within the 95% confidence interval (also called HDI) of the posterior probability distribution. According to this criterion, we have obtained a **93.2% accuracy** rate (i.e., 466 parameters correctly identified and 34 failed) for the player parameters Alpha-α. Figure 15 shows an example of a successful prediction of an Alpha-α parameter, and Figure 16 shows an example of a failed prediction.

Concerning the inference of the Beta-β parameters of the questions, following the same criteria, we have obtained an **80% accuracy** (i.e., 12 parameters correctly identified and 3 failed). Figure 17 shows an example of a successful prediction of a Beta-β parameter, and Figure 18 shows an example of a failed prediction. With these results, we can conclude that the synthetic data generation process has been successful a priori, since we have been able to train inference models capable of discriminating between the different profiles.

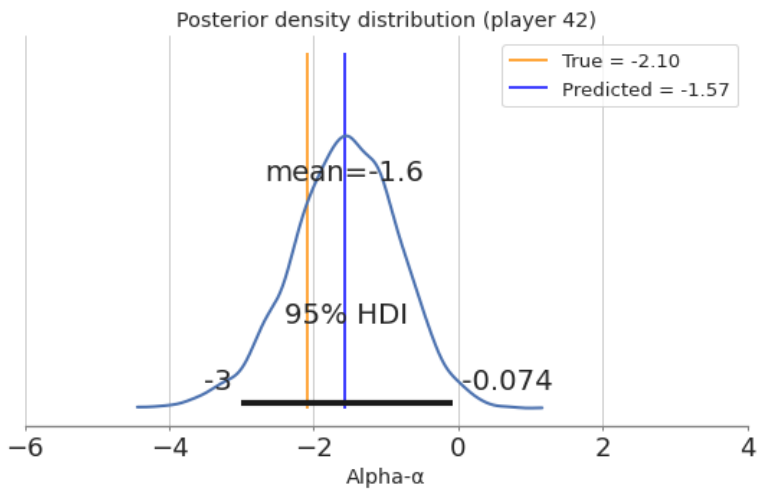


Figure 15. Example of successful prediction of an Alpha-α parameter. The true value falls within the 95% HDI in the posterior probability distribution.

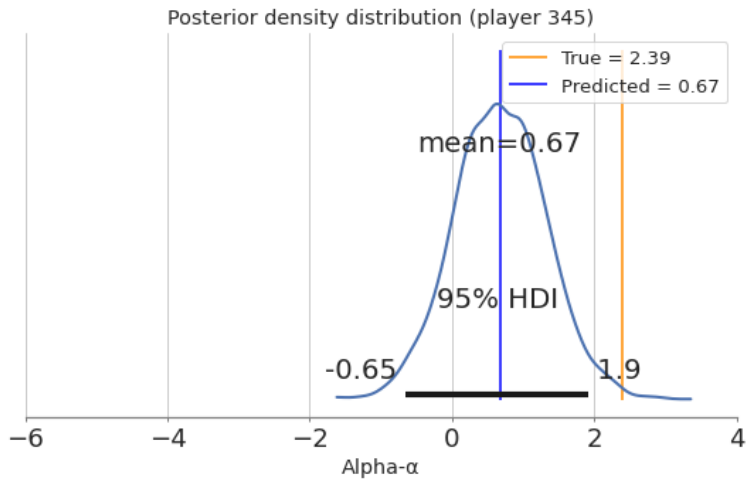


Figure 16. Example of failed prediction of an Alpha-α parameter. The true value does not fall within the 95% HDI in the posterior probability distribution.

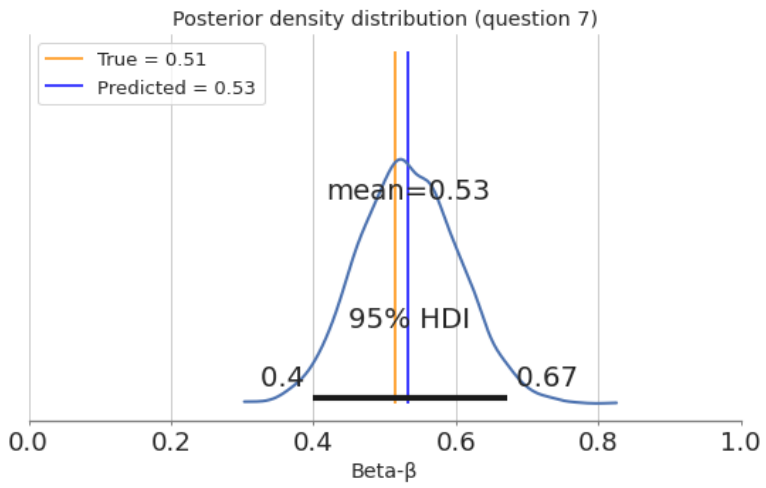


Figure 17. Example of successful prediction of a Beta-β parameter. The true value falls within the 95% HDI in the posterior probability distribution.

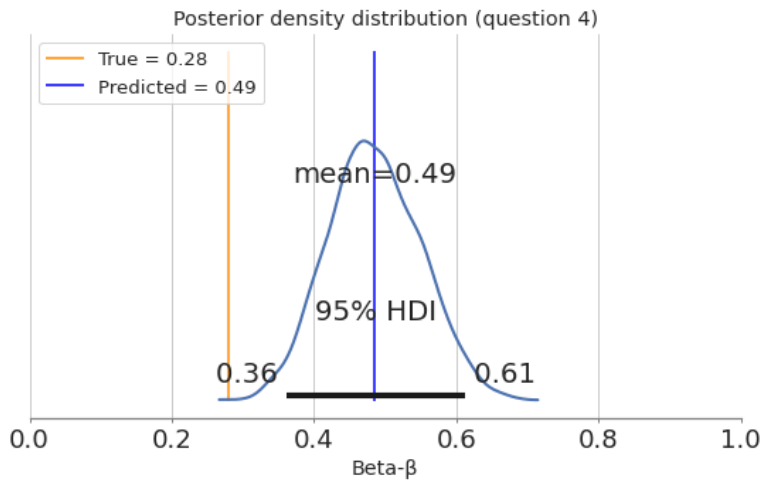


Figure 18. Example of failed prediction of a Beta-β parameter. The true value does not fall within the 95% HDI in the posterior probability distribution.

5.2 Example of Generated Dataset: Memory Agent

In this example, we have generated a dataset exploring the simulator's functionality of generating agents with memory, leaving aside the generation of synthetic demographic data at this stage. For this purpose, five types of profiles have been constructed, three without memory (random, safe, and risky) and two with memory (STR and RTS), as already explained in section 4.2.

We then generated 500 synthetic agents to participate in a set of 17 dichotomous questions (two possible answers) from a decision tree. This means that agents will answer different questions depending on the choices they make, so not all agents will answer all questions. Figure 19 shows a sample of the generated dataset, where the data for each agent is stored in each row. In the columns of the dataset where each agent's answers are stored, the 1s mean that, in that question, the agent chose the option implying the highest risk propensity. And the opposite with the 0s, the agent has chosen the option implying the lowest risk propensity. Values of -1 indicate the branches of the decision tree that the agent has not reached, i.e., invalid.

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_10	Q_11	Q_12	Q_14	Q_15	Q_17	Profile
0	0.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	1.0	-1.0	-1.0	-1.0	Safe
1	0.0	0.0	1.0	1.0	1.0	1.0	0.0	-1.0	0.0	0.0	1.0	0.0	1.0	-1.0	1.0	Risky
2	1.0	0.0	0.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	-1.0	-1.0	-1.0	Random
3	1.0	1.0	1.0	1.0	0.0	-1.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0	-1.0	-1.0	RTS
4	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	-1.0	-1.0	-1.0	Safe
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
495	0.0	1.0	1.0	1.0	1.0	0.0	0.0	-1.0	0.0	1.0	0.0	0.0	0.0	-1.0	-1.0	STR
496	1.0	0.0	0.0	0.0	0.0	-1.0	0.0	-1.0	1.0	0.0	0.0	0.0	-1.0	-1.0	1.0	Random
497	1.0	1.0	0.0	1.0	1.0	0.0	1.0	1.0	0.0	1.0	1.0	0.0	1.0	-1.0	-1.0	Random
498	0.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	-1.0	-1.0	0.0	Random
499	0.0	1.0	0.0	1.0	1.0	1.0	0.0	-1.0	0.0	1.0	1.0	0.0	0.0	-1.0	-1.0	STR

500 rows × 16 columns

Figure 19. Sample of the generated dataset: Memory Agent (Total = 500 agents). Values of -1 indicate the branches of the decision tree that the agent has not reached, i.e. invalid.

5.2.1 Identifiability Test: Machine Learning Classification

To assess the usefulness and identifiability of the synthetic data generated, in this section we proceed to use a statistical inference method to reconstruct the profile that has generated each gameplay, therefore it is a supervised classification problem. These results are preliminary and non-exhaustive, as this sort of Machine Learning analysis is relegated to task 6.2.

We have used the *Scikit-learn*[3] library to train and test a range of supervised Machine Learning models on this classification task: decision tree classifier (DT), logistic regression (LR), AdaBoost classifier (Ada), random forest classifier (RF), k-nearest neighbours classifier (KNN), and multi-layer perceptron classifier (MLP). In the following analysis, it is important to remember that the dataset generated is balanced, having the same number of samples for each profile, which entails that the accuracy is a reliable test metric. Figure 20 summarizes the test metrics obtained by these models. The AdaBoost

[3] *Scikit-learn*: Machine Learning in Python https://scikit-learn.org/stable/

model obtains the worst results with 0.647 accuracy and 0.63 F1-score, while the best results are obtained by the decision tree and MLP models with **0.8 accuracy** and **0.8 F1-score**.

It can also be observed that the results vary depending on the class. Of particular interest are the results obtained for the random class, around 0.5 accuracy and f1-score. An explanation for this is that, since this profile acts completely random, it has no decision-making pattern. Therefore, each of the runs generated by this profile may be similar to those generated by other profiles, leading the models to classify those runs incorrectly. This is not a problem in the data, but an inherent problem of random profile.

Except for the random profile, all classes obtain similar scores with around 0.8 - 0.9 accuracy and F1-scores. Moreover, the best per-class score is 0.93 accuracy and 0.88 F1-score (STR class, DT, and RF models). Except for AdaBoost, the worst per-class score is 0.79 accuracy and 0.78 F1-score (Risky class, KNN model). With these results, we can conclude that the generation of synthetic data has successfully trained models capable of using the decisions taken to differentiate the profiles.

Profile	Metric	DT[a]	LR[b]	Ada[c]	RF[d]	KNN[e]	MLP[f]
Random	F1	0.583	0.506	0.488	0.581	0.548	0.577
	Accuracy	0.527	0.444	0.426	0.522	0.517	0.51
Risky	F1	0.819	0.805	0.466	0.819	0.784	0.82
	Accuracy	0.81	0.815	0.333	0.81	0.789	0.817
Safe	F1	0.842	0.83	0.765	0.842	0.82	0.843
	Accuracy	0.825	0.811	0.681	0.826	0.819	0.833
RTS	F1	0.862	0.809	0.671	0.861	0.841	0.861
	Accuracy	0.928	0.858	0.912	0.929	0.86	0.921
STR	F1	0.881	0.821	0.762	0.88	0.861	0.881
	Accuracy	0.926	0.88	0.886	0.928	0.888	0.935
Global	F1	0.797	0.753	0.63	0.796	0.77	0.796
	Accuracy	0.802	0.761	0.647	0.802	0.774	0.803

[a]Decision tree [b]Logistic regression [c]AdaBoost [d]Random forest [e]K-nearest neighbors [f]Multi-layer Perceptron using adam optimizer

Figure 20. Profile classification results (test metrics) using supervised Machine Learning algorithms.

6. Conclusions

In this work, we have proposed and implemented an agent-based simulator with the primary objective of addressing the data scarcity problem in the context of the RAYUELA project. To this end, we have first gathered together the most relevant design considerations that we had to consider when creating the simulator. Next, we proposed a modular approach capable of incorporating external information (e.g., expert knowledge, prevalence data), enabling an efficient use of the research carried out in previous phases of the project (WP1 and WP2).

In the implementation, we have introduced a novel probabilistic agent-based model inspired on ideas from the Item Response Theory paradigm, which is a widely studied evaluation framework in the scientific community. We have also proposed a model that tries to simulate a memory in the agents, making them more consistent with some real players. In addition, we have explored an experimental module based on AI techniques applied to language (NLP), which is intended to simulate the imperfect information retrieval from players when understanding the state of the game and the questions/dilemmas presented.

In order to test the identifiability and usefulness of the generated synthetic data, we have performed a series of inference tests a posteriori based on Bayesian and Machine Learning methods. The satisfactory results obtained lead us to conclude that, a priori, the data generated can be used to calibrate algorithms that will be used later on with real player data and to increase the amount of available data in future stages of the project.

We have intended to implement the simulator as modular and editable as possible, to make this a living project that can be easily modified in the future if required. This will be especially important when real player data becomes available.

Some key contributions of this work have been incorporated into two scientific articles titled “A child’s play: an agent-based simulator to protect minors online” [25] and “A computational framework for understanding risk factors in cybercrime” [26]. These papers were accepted and presented at the 8th International Conference on Computational Social Science (IC2S2), one of the most prestigious international conferences within the field of computational social science, held at the University of Chicago from July 19 to July 22 2022.

The code developed in this work is open and freely available in the GitHub repository of the RAYUELA project[4].

[4] GitHub repository of the RAYUELA project (Simulator): https://github.com/rayuelaproject/Simulator_Synthetic-Data

References

[1] Georgios N Yannakakis and Julian Togelius. “Artificial intelligence and games,” volume 2. Springer, 2018.

[2] Danial Hooshyar, Moslem Yousefi, and Heuiseok Lim. “Data-driven approaches to game player modeling: A systematic literature review.” ACM Comput. Surv., 50(6), jan 2018.

[3] Alexander Streicher and Michael Aydinbas. “Bayesian cognitive state modeling for adaptive serious games.” In Robert A. Sottilare and Jessica Schwarz, editors, Adaptive Instructional Systems, pages 14–25, Cham, 2022. Springer International Publishing.

[4] Richard Kim, Max Kleiman-Weiner, Andrés Abeliuk, Edmond Awad, Sohan Dsouza, Joshua B. Tenenbaum, and Iyad Rahwan. “A computational model of commonsense moral decision making.” In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES ’18, page 197–203, New York, NY, USA, 2018. Association for Computing Machinery.

[5] Giel van Lankveld, Pieter Spronck, Jaap van den Herik, and Arnoud Arntz. “Games as personality profiling tools.” In 2011 IEEE Conference on Computational Intelligence and Games (CIG’11), pages 197–202, 2011.

[6] Nick Yee, Nicolas Ducheneaut, Les Nelson, and Peter Likarish. “Introverted elves & conscientious gnomes: The expression of personality in world of warcraft.” In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’11, page 753–762, New York, NY, USA, 2011. Association for Computing Machinery.

[7] Zahid Halim, Muhammad Atif, Ahmar Rashid, and Cedric A. Edwin. “Profiling players using real-world datasets: Clustering the data and correlating the results with the big-five personality traits.” IEEE Transactions on Affective Computing, 10(4):568–584, 2019.

[8] Mouna Denden, Ahmed Tlili, Fathi Essalmi, and Mohamed Jemni. “Implicit modeling of learners’ personalities in a game-based learning environment using their gaming behaviors.” Smart Learning Environments, 5(1):29, November 2018

[9] John-Luke McCord, Jason L. Harman, and Justin Purl. “Game-like personality testing: An emerging mode of personality assessment.” Personality and Individual Differences, 143:95–102, 2019.

[10] R. Lebaredian, “Synthetic Data will Drive Next Wave of Business Applications," NVIDIA GTC Silicon Valley, 2019.

[11] K. Duemig, “Accelerating time-to-market with fabricated test data,” IBM Big Data & Analytics Hub, 2017.

[12] K. Duemig, “Protect your customer data with relevant test data,” IBM Big Data & Analytics Hub, 2017.

[13] T. Kohlberger and Y. Liu, “Generating Diverse Synthetic Medical Image Data for Training Machine Learning Models,” Google AI Blog, 2020.

[14] “Modernization of Statistical Disclosure Limitation at US Census Bureau,” tech. rep., US Census Bureau. Section: Government, 2020.

[15] J. Banks; J. Carson; B. Nelson; D. Nicol, "Discrete-Event System Simulation." Prentice Hall, 2001. ISBN 978-0-13-088702-3.

[16] Shorten, C., Khoshgoftaar, T.M. "A survey on Image Data Augmentation for Deep Learning." Journal of Big Data, 2019. https://doi.org/10.1186/s40537-019-0197-0

[17] Niazi, M., Hussain, A. "Agent-based computing from multi-agent systems to agent-based models: a visual survey." Scientometrics, 2011. https://doi.org/10.1007/s11192-011-0468-9

[18] Silver, D., Huang, A., Maddison, C. et al. Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–489 (2016). https://doi.org/10.1038/nature16961

[19] Wang, B., Sun, T., & Zheng, X. S. (2019). Beyond Winning and Losing: Modeling Human Motivations and Behaviors with Vector-Valued Inverse Reinforcement Learning. Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 15(1), 195-201. Retrieved from https://ojs.aaai.org/index.php/AIIDE/article/view/5244

[20] Lin, B., Cecchi, G., Bouneouf, D., Reinen, J., Rish, I. A Story of Two Streams: Reinforcement Learning Models from Human Behavior and Neuropsychiatry. Proceedings of the Nineteenth International Conference on Autonomous Agents and Multi-Agent Systems, 744-752 (2020). https://ifaamas.org/Proceedings/aamas2020/pdfs/p744.pdf

[21] Embretson, S.E., & Reise, S.P. (2000). Item Response Theory for Psychologists (1st ed.). Psychology Press. https://doi.org/10.4324/9781410605269

[22] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, 4171–4186 (2019). https://aclanthology.org/N19-1423/

[23] Hochreiter, S., Schmidhuber, J. Long short-term memory. Neural Computation 9(8) 1735-1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735

[24] Stan Development Team, "Stan Modeling Language Users Guide and Reference Manual, Version 2.30." 2022. https://mc-stan.org

[25] J. Pérez, V. Balmaseda, A. L. Urbistondo, E. Awad, M. Castro and G. López, “A Child’s Play: An Agent-based Simulator to Protect Minors Online”. 8th International Conference on Computational Social Science (IC2S2), 2022.

[26] J. Pérez, E. Awad, M. Castro, G. López, N. Bueno-Guerra, M. Reneses Botija, M. Riberas Gutiérrez, A. Gómez-Dorado, “A computational framework for understanding risk factors in cybercrime”, 8th International Conference on Computational Social Science (IC2S2), 2022.