Towards Modelling and Verification of Social Explainable AI

Damian Kurpiewski,[1,2,a] Wojciech Jamroga,[1,3,b], and Teofil Sidoruk[1,4,c]
[1] *Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland*
[2] *Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, Toruń, Poland*
[3] *Interdisciplinary Centre for Security, Reliability, and Trust, SnT, University of Luxembourg*
[4] *Faculty of Mathematics and Information Science, Warsaw University of Technology, Warsaw, Poland*
{*d.kurpiewski, w.jamroga, t.sidoruk*}*@ipipan.waw.pl*

Keywords: multi-agent systems, formal verification, Social Explainable AI, strategic ability, model checking

Abstract: Social Explainable AI (SAI) is a new direction in artificial intelligence that emphasises decentralisation, transparency, social context, and focus on the human users. SAI research is still at an early stage. Consequently, it concentrates on delivering the intended functionalities, but largely ignores the possibility of unwelcome behaviours due to malicious or erroneous activity. We propose that, in order to capture the breadth of relevant aspects, one can use models and logics of strategic ability, that have been developed in multi-agent systems. Using the STV model checker, we take the first step towards the formal modelling and verification of SAI environments, in particular of their resistance to various types of attacks by compromised AI modules.

1 INTRODUCTION

Elements of artificial intelligence have become ubiquitous in daily life, being involved in social media, car navigation, recommender algorithms for music and films, and so on. They also provide back-end solutions to many business processes, resulting in a huge societal and economical impact. The idea of *Social Explainable AI (SAI)* represents an interesting new direction in artificial intelligence, which emphasises decentralisation, human-centricity, and explainability [33, 8]. This is in line with the trend to move away from classical, centralised machine learning, not only for purely technical reasons such as scalability constraints, but also to meet the growing ethical expectations regarding transparency and trustworthiness of data storage and computation [10, 28]. The aim is also to put the human again in the spotlight, rather than concentrate on the technological infrastructure [7, 34, 12].

SAI is a new concept, and a subject of ongoing research. It still remains to be seen if it delivers AI solutions that are effective, transparent, and mindful of the user. To this end, it should be extensively studied not only in the context of its intended properties, but also the possible side effects of interaction that involves AI components and human users in a complex environment. In particular, we should carefully analyse the possibilities of adversarial misuse and abuse of the interaction, e.g., by means of impersonation or man-in-the-middle attacks [9, 13]. In those scenarios, one or more nodes of the interaction network are taken over by a malicious party that tries to disrupt communication, corrupt data, and/or spread false information. Clearly, the design of Social AI must be resistant to such abuse; otherwise it will be sooner or later exploited. While the topic of adversarial attacks on machine learning algorithms has recently become popular [14, 20, 22], the research on SAI has so far focused only on its expected functionalities. This is probably because SAI communities are bound to be conceptually, computationally, and socially complex. A comprehensive study of their possible *unintended* behaviors is a highly challenging task.

Here, we propose that *formal methods for multi-agent systems* [35, 31] provide a good framework for multifaceted analysis of Social Explainable AI. Moreover, we put forward a new methodology for such studies, based on the following hypotheses:

1. It is essential to formalise and evaluate multi-agent properties of SAI environments. In particular, we must look at the properties of interaction between SAI components that go beyond joint, fully orchestrated action towards a common predefined goal. This may include various relevant functionality and safety requirements. In particular, we should assess the impact of adversarial

[a] https://orcid.org/0000-0002-9427-2909
[b] https://orcid.org/0000-0001-6340-8845
[c] https://orcid.org/0000-0002-4393-3447

play on these requirements.

2. Many of those properties are underpinned by *strategic ability* of agents and their groups to achieve their goals [29, 2, 5]. In particular, many functionality properties refer to the ability of legitimate users to complete their selected tasks. Conversely, safety and security requirements can be often phrased in terms of the inability of the "bad guys" to disrupt the behavior of the system.
3. Model checking [6] provides a well-defined formal framework for the analysis. Moreover, existing model checking tools for multi-agent systems, such as MCMAS [26] and STV [25] can be used to formally model, visualise, and analyse SAI designs with respect to the relevant properties.
4. Conversely, SAI can be used as a testbed for cutting-edge methods of model checking and their implementations.

In the rest of this paper, we make the first step towards formal modelling, specification, and verification of SAI. We model SAI by means of *asynchronous multi-agent systems (AMAS)* [17], and formalise their properties using formulas of temporal-strategic logic **ATL*** [2, 30]. For instance, one can specify that a malicious AI component can ensure that the remaining components will never be able to build a global model of desired quality, even if they all work together against the rogue component. Alternatively, strategies of the "good" modules can be considered, in order to check whether a certain threshold of non-compromised agents is sufficient to prevent a specific type of attack. Finally, we use the STV model checker [25] to verify the formalised properties against the constructed models. The verification is done by means of the technique of *fixpoint approximation*, proposed and studied in [18].

Note that this study does *not* aim at focused in-depth verification of a specific machine learning procedure, like in [36, 3, 21, 1]. Our goal is to represent and analyse a broad spectrum of interactions, possibly at the price of abstraction that leaves many details out of the formal model.

The ideas, reported here, are still work in progress, and the results should be treated as preliminary.

2 SOCIAL EXPLAINABLE AI

The framework of *Social Explainable AI* or *SAI* [33, 8, 12] aims to address several drawbacks inherent to the currently dominant AI paradigm. In particular, state of the art machine learning (ML)-based AI systems are typically centralised. The sheer scale of Big Data collections, as well as the complexity of deep neural networks that process them, mean that effectively these AI systems act as opaque black boxes, non-interpretable even for experts. This naturally raises issues of privacy and trustworthiness, further exacerbated by the fact that storing an ever-increasing amount of sensitive data in a single, central location might eventually become unfeasible, also for non-technical reasons such as local regulations regarding data ownership.

In contrast, SAI envisions novel ML-based AI systems with a focus on the following aspects:

- Individuation: a "Personal AI Valet" (PAIV) associated with each individual, acting as their proxy in a complex ecosystem of interacting PAIVs;
- Personalisation: processing data by PAIVs via explainable AI models tailored to the specific characteristics of individuals;
- Purposeful interaction: PAIVs build global AI models or make collective decisions starting from the local models by interacting with one another;
- Human-centricity: AI algorithms and PAIV interactions driven by quantifiable models of the individual and social behaviour of their human users;
- Explainability by design: extending ML techniques through quantifiable human behavioural models and network science analysis.

The current attempts at building SAI use *gossip learning* as the ML regime for PAIVs [32, 15, 16]. An experimental simulation tool to assess the effectiveness of the process and functionality of the resulting AI components is available in [27]. In this paper, we take a different path, and focus on the multi-agent interaction in the learning process. We model the network of PAIVs as an *asynchronous multi-agent system (AMAS)*, and formalise its properties as formulas of *alternating-time temporal logic (***ATL****)*. The formal framework is introduced in Section 3. In Section 4, we present preliminary multi-agent models of SAI, and show several attacks that can be modelled that way. In Section 5, we formalise several properties and conduct model checking experiments.

3 WHAT AGENTS CAN ACHIEVE

In this section, we introduce the formalism of Asynchronous Multi-agent Systems (AMAS) [17, 19], as well as the syntax and semantics of Alternating-time Temporal Logic **ATL*** [2, 30], which allows for specifying relevant properties of SAI models, in particular the *strategic ability* of agents to enforce a goal.

3.1 Asynchronous MAS

AMAS can be thought of as networks of automata, where each component corresponds to a single agent.

Definition 1 (AMAS [19])**.** *An* asynchronous multi-agent system *(AMAS) consists of n agents $\mathcal{A} = \{1,\ldots,n\}$, each associated with a 7-tuple $A_i = (L_i, \iota_i, Evt_i, R_i, T_i, PV_i, V_i)$, where:*

- *$L_i = \{l_i^1,\ldots,l_i^{n_i}\} \neq \emptyset$ is a finite set of* local states*;*
- *$\iota_i \in L_i$ is an* initial local state*;*
- *$Evt_i = \{e_i^1,\ldots,e_i^{m_i}\} \neq \emptyset$ a finite set of* events*;*
- *$R_i : L_i \to 2^{Evt_i} \setminus \{\emptyset\}$ is a* repertoire of choices*, assigning available subsets of events to local states;*
- *$T_i : L_i \times Evt_i \rightharpoonup L_i$ is a (partial)* local transition function *that indicates the result of executing event e in state l from the perspective of agent i. $T_i(l_i,e)$ is defined iff $e \in \bigcup R_i(l_i)$;*
- *PV_i is a set of the agent's* local propositions*, with PV_j, PV_k (for $j \neq k \in \mathcal{A}$) assumed to be disjoint;*
- *$V_i : L_i \to \mathcal{P}(PV_i)$ is a* valuation function.

Furthermore, we denote:

- *by $Evt = \bigcup_{i\in\mathcal{A}} Evt_i$, the set of all events;*
- *by $L = \bigcup_{i\in\mathcal{A}} L_i$, the set of all local states;*
- *by $Agent(e) = \{i \in \mathcal{A} \mid e \in Evt_i\}$, the set of all agents which have event e in their repertoires;*
- *by $PV = \bigcup_{i\in\mathcal{A}} PV_i$ the set of all local propositions.*

The *model* of an AMAS provides its execution semantics with asynchronous interleaving of private events and synchronisation on shared ones.

Definition 2 (Model [19])**.** *The* model *of an AMAS is a 5-tuple $M = (\mathcal{A}, S, \iota, T, V)$, where:*

- *$\mathcal{A}$ is the set of* agents*;*
- *$S \subseteq L_1 \times \ldots \times L_n$ is the set of* global states*, including all states reachable from ι by T (see below);*
- *$\iota = (\iota_1,\ldots,\iota_n) \in S$ is the* initial global state*;*
- *$T : S \times Evt \cup \{\varepsilon\} \rightharpoonup S$ is the (partial)* global transition function*, defined by $T(s_1,e) = s_2$ iff $T_i(s_1^i,e) = s_2^i$ for all $i \in Agent(e)$ and $s_1^i = s_2^i$ for all $i \in \mathcal{A} \setminus Agent(e)$, where $s_j^i \in L_i$ is agent i's local component of s_j. Moreover, $T(s,\varepsilon) = s$ iff there are events $e_1,\ldots,e_n$ st. $T_i(s^i,e_i)$ is defined but none of $e_1,\ldots,e_n$ is selected by all its owners;*
- *$V : S \to 2^{PV}$ is the* global valuation function*, defined as $V(l_1,\ldots,l_n) = \bigcup_{i\in\mathcal{A}} V_i(l_i)$.*

3.2 Strategic Ability

Linear and branching-time temporal logics, such as **LTL** and **CTL*** [11], have long been used in formal verification. They enable to express properties about *how* the state of the system will (or should) evolve over time. However, in systems that involve autonomous agents, whether representing human users or AI components it is usually of interest *who* can direct its evolution a particular way.

ATL* [2] extends temporal logics with *strategic modalities* that allow for reasoning about such properties. The operator $\langle\!\langle A\rangle\!\rangle\gamma$ says that agents in group (coalition) A have a *strategy* to enforce property γ. That is, as long as agents in A select events according to the strategy, γ will hold no matter what the other agents do. **ATL*** has been one of the most important and popular agent logics in the last 25 years.

Definition 3 (Syntax of **ATL***)**.** *The language of* **ATL*** *is defined by the grammar:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi\wedge\varphi \mid \langle\!\langle A\rangle\!\rangle\gamma,$$
$$\gamma ::= \varphi \mid \neg\gamma \mid \gamma\wedge\gamma \mid \mathrm{X}\gamma \mid \gamma\mathrm{U}\gamma,$$

where $\mathsf{p} \in PV$ and $A \subseteq \mathcal{A}$. The definitions of Boolean connectives and temporal operators X *("next") and* U *("strong until") are standard; remaining operators* R *("release"),* G *("always"), and* F *("sometime") can be derived as usual.*

Various types of strategies can be defined, based on the state information and memory of past states available to agents [30]. In this work, we focus on *imperfect information, imperfect recall strategies*.

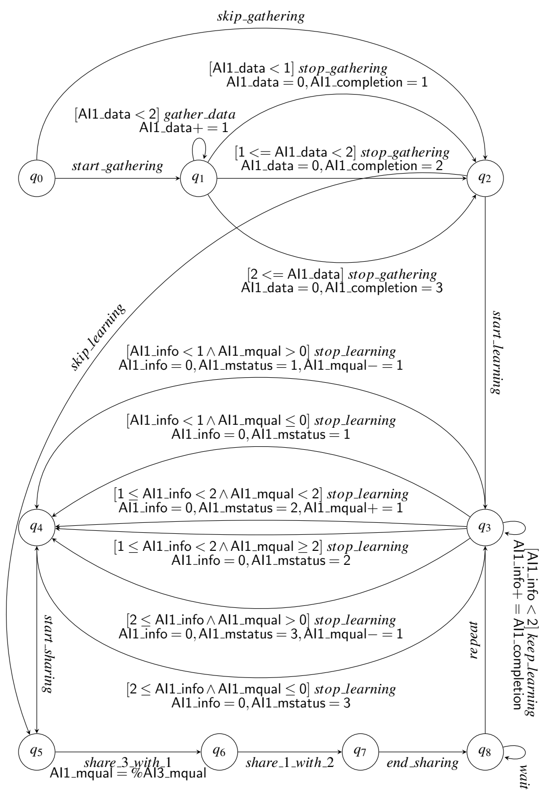
Definition 4 (Strategy)**.** *A* memoryless imperfect information strategy *for agent $i \in \mathcal{A}$ is a function $\sigma_i : L_i \to 2^{Evt_i} \setminus \emptyset$ such that $\sigma_i(l) \in R_i(l)$ for each local state $l \in L_i$. A* joint strategy *σ_A of coalition $A \subseteq \mathcal{A}$ is a tuple of strategies σ_i, one for each agent $i \in A$.*

The outcome set of a strategy collects all the execution paths consistent with the strategy. Formally, $out_M(s,\sigma_A)$ collects all the infinite sequences of states, starting from s, that may occur when the coalition follows strategy σ_A while the opponents choose freely from their repertoires. We use the so called *opponent-reactive outcome*, where the opponents are assumed to respond with matching synchronization events if such responses are available. The interested reader is referred to [19, 24] for the discussion and technical details.

Definition 5 (Asynchronous semantics of **ATL*** [17])**.** *The asynchronous semantics of the strategic modality in* **ATL*** *is defined by the following clause:*

$M,s \models \langle\!\langle A\rangle\!\rangle\gamma$ iff there is a strategy σ_A such that $out_M(s,\sigma_A) \neq \emptyset$ and, for each path $\pi \in out_M(s,\sigma_A)$, we have $M,\pi \models \gamma$.

The remaining clauses for temporal operators and Boolean connectives are standard, see [11].


skip_gathering
[AI1_data < 1] stop_gathering
AI1_data = 0, AI1_completion = 1
[AI1_data < 2] gather_data
AI1_data+ = 1
[1 <= AI1_data < 2] stop_gathering
AI1_data = 0, AI1_completion = 2
start_gathering
q0
q1
q2
[2 <= AI1_data] stop_gathering
AI1_data = 0, AI1_completion = 3
skip_learning
start_learning
[AI1_info < 1 ∧ AI1_mqual > 0] stop_learning
AI1_info = 0, AI1_mstatus = 1, AI1_mqual− = 1
[AI1_info < 1 ∧ AI1_mqual ≤ 0] stop_learning
AI1_info = 0, AI1_mstatus = 1
[1 ≤ AI1_info < 2 ∧ AI1_mqual < 2] stop_learning
AI1_info = 0, AI1_mstatus = 2, AI1_mqual+ = 1
q4
q3
[1 ≤ AI1_info < 2 ∧ AI1_mqual ≥ 2] stop_learning
AI1_info = 0, AI1_mstatus = 2
[AI1_info < 2] keep_learning
AI1_info+ = AI1_completion
[2 ≤ AI1_info ∧ AI1_mqual > 0] stop_learning
AI1_info = 0, AI1_mstatus = 3, AI1_mqual− = 1
start_sharing
repeat
[2 ≤ AI1_info ∧ AI1_mqual ≤ 0] stop_learning
AI1_info = 0, AI1_mstatus = 3
q5
q6
q7
q8
share_3_with_1
AI1_mqual = %AI3_mqual
share_1_with_2
end_sharing
wait

```
Agent AI1:
init: start
%% ---Phase1: Gathering data---
start_gathering_data: start -> gather
gather_data: gather -[AI1_data<2]> gather
        [AI1_data+=1]
%% 1: Incomplete data
stop_gathering_data: gather -[AI1_data < 1]> data_ready
        [AI1_data=0, AI1_data_completion=1]
%% 2: Complete data
stop_gathering_data: gather -[1 <= AI1_data < 2]> data_ready
        [AI1_data=0, AI1_data_completion=2]
%% 3: Too much data
stop_gathering_data: gather -[2 <= AI1_data]> data_ready
        [AI1_data=0, AI1_data_completion=3]
skip_gathering_data: start -> data_ready
%% ---Phase2: Learning (building local model)---
start_learning: data_ready -> learn
keep_learning: learn -[AI1_information < 2]> learn
        [AI1_information+=AI1_data_completion]
%% 1: Incomplete model
stop_learning: learn
        -[AI1_information < 1 and AI1_model_quality > 0]> educated
        [AI1_information=0, AI1_model_status=1, AI1_model_quality-=1]
stop_learning: learn
        -[AI1_information < 1 and AI1_model_quality <= 0]> educated
        [AI1_information=0, AI1_model_status=1]
%% 2: Complete model
stop_learning: learn
        -[1 <= AI1_information < 2 and AI1_model_quality < 2]> educated
        [AI1_information=0, AI1_model_status=2, AI1_model_quality+=1]
stop_learning: learn
        -[1 <= AI1_information < 2 and AI1_model_quality >= 2]> educated
        [AI1_information=0, AI1_model_status=2]
%% 3: Overtrained model
stop_learning: learn
        -[2 <= AI1_information and AI1_model_quality > 0]> educated
        [AI1_information=0, AI1_model_status=3, AI1_model_quality-=1]
stop_learning: learn
        -[2 <= AI1_information and AI1_model_quality <= 0]> educated
        [AI1_information=0, AI1_model_status=3]
skip_learning: data_ready -> sharing
%% ---Phase3: Sharing local models---
start_sharing: educated -> sharing
%% Share local model and get average quality of both models
%% receive left
shared share_3_with_1: sharing -> sharing2
        [AI1_model_quality=%AI3_model_quality]
%% send right
shared share_1_with_2: sharing2 -> sharing3
end_sharing: sharing3 -> end
%% ---Phase4: End---
wait: end -> end
repeat: end -> learn
```

Figure 1: Honest AI agent: AMAS (left) and its specification in STV (right)

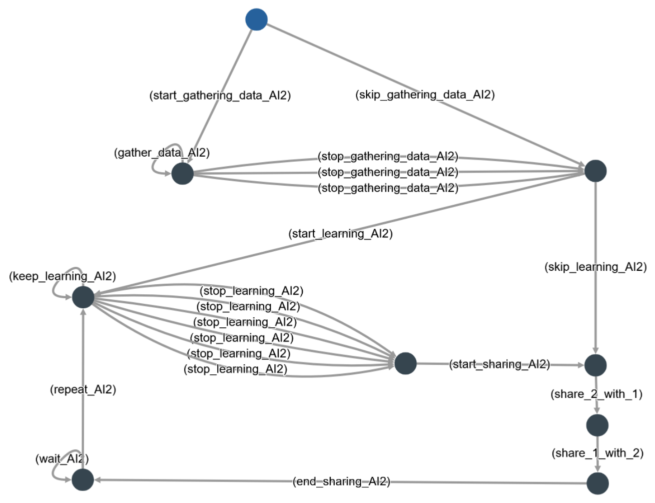


Figure 2: Visualization of honest AI in STV

4 MODELS

The first step towards the verification of the interaction between agents in Social Explainable AI is a thorough and detailed analysis of the underlying protocol. We begin by looking into the actions performed and the messages exchanged by the machines that take part in the learning phase. Then, we can start designing multi-agent models. Usually, such systems are too complex to be modelled as they are. In that case, we create an abstract view of the system.

4.1 Agents

In this work, we focus on the learning phase of the SAI protocol. We model each machine equipped with an AI module as a separate agent. The local model of an AI agent consists of three phases: the data gathering phase, the learning phase and the sharing phase.

Data gathering phase. In this phase, the agent is

```
Agent Mim:
init: start
shared share_1_with_mim: start -> start
    [Mim_model_quality=AI1_model_quality]

shared share_mim_with_1: start -> start

shared share_2_with_mim: start -> start
    [Mim_model_quality=AI2_model_quality]

shared share_mim_with_2: start -> start
```

Figure 3: Specification of the Man in the Middle agent

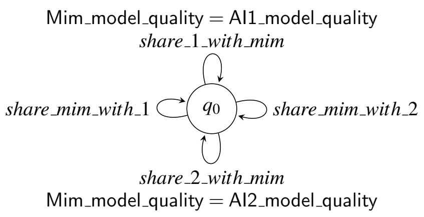

Mim_model_quality = AI1_model_quality
share_1_with_mim
share_mim_with_1
q_0
share_mim_with_2
share_2_with_mim
Mim_model_quality = AI2_model_quality

Figure 4: Graphical representation of Man in the Middle

able to gather the data required for the learning phase. The corresponding action can be performed multiple times, each time increasing the local variable that represents the amount of gathered data. At the end of the phase, the amount of gathered data is analysed and, depending on the exact value, the agent's preparation is marked as incomplete, complete, or excessive. From this, the agent proceeds to the learning phase.

Learning phase. Here, the agent can use the previously gathered data to train its local AI model. The effectiveness of the training depends on the amount of gathered data. Excessive data means that the model can be easily overtrained, while insufficient data may lead to more iterations required to properly train the model. The training action can be performed multiple times each time increasing the local variable related to the quality of the model. At the end of this phase the internal AI model can be overtrained, undertrained or properly trained. After this phase, the agent is required to share its model with other agents.

Sharing phase. Agents share their local AI models with each other following a simple sharing protocol. The protocol is based on packet traversal in the ring topology. Each agent receives the model from the agent with previous ID and sends its model to the agent with next ID, while the last agent shares its model with the first agent to close the ring. In order to avoid any deadlocks, each agent with odd ID first receives the model and then sends its own, and each agent with even ID first sends its own model and then receives the model from the agent before him.

When receiving the model, the agent can either accept it or reject it, and its decision is based on the quality of the model being shared. After accepting the model, the agent merges it with its own and the

```
Agent AI2:
init: start
set_quality_0: start -> set_quality
                   [AI2_model_quality=0]

set_quality_1: start -> set_quality
                   [AI2_model_quality=1]

set_quality_2: start -> set_quality
                   [AI2_model_quality=2]

shared share_2_with_3: set_quality -> sharing
shared share_1_with_2: sharing -> start
```

Figure 5: Specification of the Impersonator agent

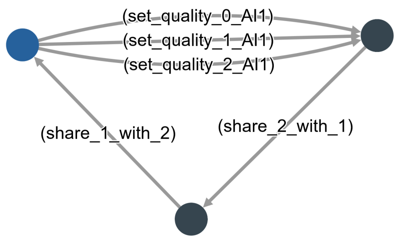


Figure 6: Graphical representation of Impersonator in STV

resulting model quality is the maximum of the two.

After the sharing phase, the agent can go back to the learning phase to further train its model.

To formalize the details of the procedure, we have utilised the open-source experimental model checker STV [25], which was used, e.g., to model and verify the real-world voting protocol Selene [24]. Figure 1 presents a detailed representation of an honest AI component as an AMAS (left) and the STV code specifying its behavior (right). Figure 2 shows the visualization of the component, produced by the tool.

4.2 Attacks

The model described in Section 4.1 reflects the ideal scenario in which each agent is honest and directly follows the protocol. Of course, it is not always the case. A machine can malfunction, and take actions not permitted by the protocol. Also, an agent can be infected by malicious software, and function improperly. This leads to two possible attack scenarios: the man in the middle attack and the impersonator attack.

Man in the middle. Assume the existence of another, dishonest agent, called the *intruder*. This agent does not participate in the data gathering and learning phases, but it is particularly interested in the sharing phase. The intruder can intercept any model that is being sent by one of the honest agents and then pass it to any other agent. The STV code for the man-in-the-middle attacker is presented in Figure 3, and its graphical representation in Figure 4.

Impersonator. In this scenario, one of the AI agents is infected with malicious code that results in unwanted behavior. The agent cannot participate in the

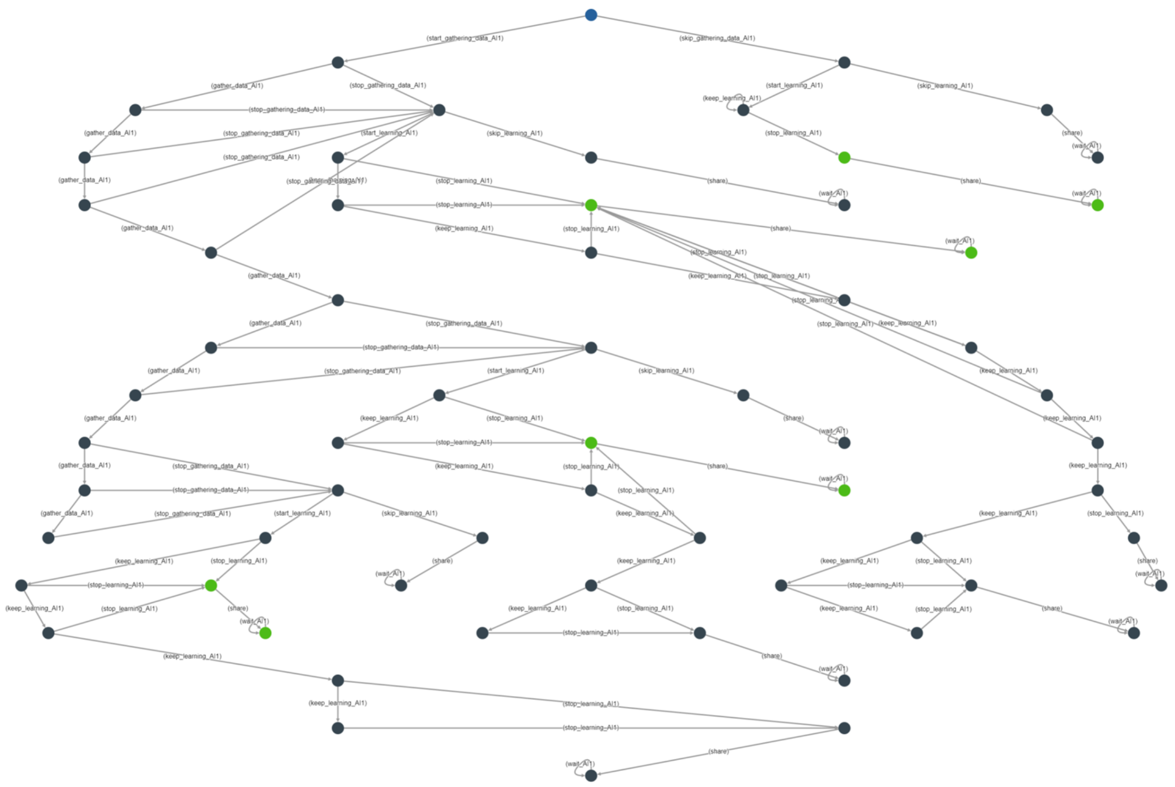


Figure 7: Model of SAI with one honest agent

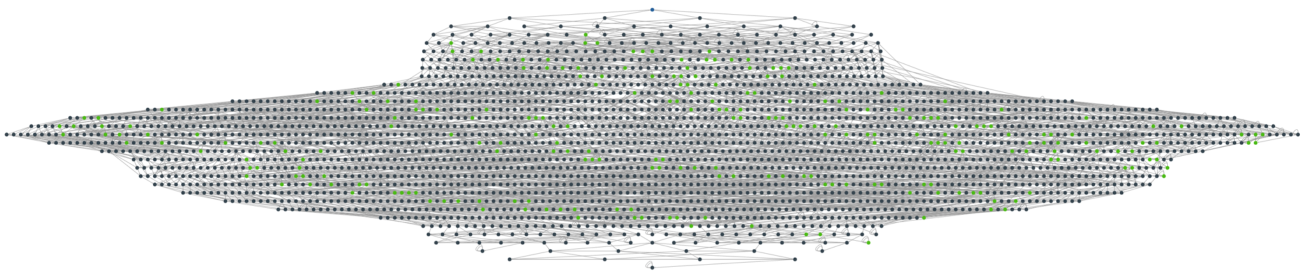


Figure 8: Model of SAI with two honest agents

data gathering or learning phases, but can share its model with others following the sharing protocol. The difference between the honest agent and the impersonator is that the latter can fake the quality of its local AI model, hence tricking the next agent into accepting it. The STV code and its visualization for Impersonator are presented in Figures 5 and 6.

5 EXPERIMENTS

The STV tool can be used to combine the modules presented in Figures 1–6, and generate the global model of interaction. We present the output in Figures 7 (for the system with one honest AI agent) and 8 (for two honest agents). The models provide invaluable insights into the structure of possible interactions. Still, a visual scrutiny is possible only in the simplest cases due to the state-space explosion.

In more complex instances, we can use STV to attempt an automated verification of strategic requirements. Since model checking of strategic ability is hard for scenarios with partial observability (**NP**-hard to undecidable, depending on the precise syntax and semantics of the specification language [4]), exact verification is infeasible. Instead, we use the technique of *fixpoint approximation*, proposed in [18], and implemented in STV. In what follows, we summarise the experimental results obtained that way.

#Ag	#st	#tr	Gen	Verif ϕ_1	Verif ϕ_2
2	886	2007	< 0.1	< 0.1/False	< 0.1/True
3	79806	273548	28	151/False	202/True
4	6538103	29471247	1284	5061/False	5102/True
5	93581930	623680431	7845	25828/False	25916/True
6	timeout				

Figure 9: Verification results for the Impersonator attack

Models and formulas. The scalable class of models has been described in detail in Section 4. In the model checking experiments, we have used two variants of the system specification, one with a possible impersonation attack, and the other one with the possibility of a man-in-the-middle attack. In each case, we verified the following formulas:

- $\phi_1 \equiv \langle\langle I \rangle\rangle \mathrm{G}\,(shared_p \rightarrow (\bigwedge_{i\in[1,n]} mqual_i \leq k))$
- $\phi_2 \equiv \langle\langle I \rangle\rangle \mathrm{G}\,(shared_p \rightarrow (\bigvee_{i\in[1,n]} mqual_i \leq k))$

Formula ϕ_1 checks whether the Intruder has a strategy to ensure that all honest agents will not achieve quality greater than k. Formula ϕ_2 checks whether the same is possible for at least one agent.

Configuration of the experiments. The experiments have been conducted with the latest version of STV [23]. The test platform was a server equipped with ninety-six 2.40 GHz Intel Xeon Platinum 8260 CPUs, 991 GB RAM, and 64-bit Linux.

Results. We present the verification results in Figures 9 and 10. **#Ag** specifies the scalability factor, namely the number of agents in the system. **#st** and **#tr** report the number of global states and transitions in the resulting model of the system, and **Gen** gives the time of model generation. **Verif** ϕ_1 and **Verif** ϕ_2 present the verification time and its output for formulas ϕ_1 and ϕ_2, respectively. All times are given in seconds. The timeout was set to 8 hours.

Discussion. We were able to verify models of SAI for up to 5 agents. The verification outcome was conclusive in all cases, i.e., the model checker always returned either True or False. This means that we successfully model-checked systems for up to almost a billion transitions, which is a serious achievement for an **NP**-hard verification problem. In all cases, formula ϕ_2 turned out to be true. That is, both impersonation and man-in-the-middle attacks can disrupt the learning process and prevent some agents from obtaining good quality PAIVs. At the same time, ϕ_1 was false in all cases. Thus, the intruder cannot disrupt *all* PAIVs, even with its best attack.

6 CONCLUSIONS

In this paper, we present our work in progress on formal analysis of Social Explainable AI. We propose that formal methods for multi-agent systems provide a good framework for multifaceted analysis of SAI environments. As a proof of concept, we demonstrate simple multi-agent models of SAI, prepared with the model checker STV. Then, we use STV to formalize and verify two variants of resistance to impersonation and man-in-the-middle attacks, with very promising results. Notably, we have been able to successfully model-check models of systems for up to almost a billion transitions – a considerable achievement for an **NP**-hard verification problem.

#Ag	#st	#tr	Gen	Verif ϕ_1	Verif ϕ_2
3	23966	67666	12	21/False	33/True
4	4798302	20257664	875	3810/False	3882/True
5	71529973	503249452	5688	19074/False	20103/True
6	timeout				

Figure 10: Verification results for Man in the Middle

Acknowledgements. The work has been supported by NCBR Poland and FNR Luxembourg under the PolLux/FNR-CORE project STV (POLLUX-VII/1/2019), by NCN Poland under the CHIST-ERA grant CHIST-ERA-19-XAI-010 (2020/02/Y/ST6/00064), and by the CNRS IEA project MoSART.

REFERENCES

[1] Michael E. Akintunde, Elena Botoeva, Panagiotis Kouvaros, and Alessio Lomuscio. Formal verification of neural agents in non-deterministic environments. *Auton. Agents Multi Agent Syst.*, 36(1):6, 2022.

[2] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002.

[3] Ben Batten, Panagiotis Kouvaros, Alessio Lomuscio, and Yang Zheng. Efficient neural network verification via layer-based semidefinite relaxations and linear cuts. In *Proceedings of IJCAI*, pages 2184–2190. ijcai.org, 2021.

[4] N. Bulling, J. Dix, and W. Jamroga. Model checking logics of strategic ability: Complexity. In M. Dastani, K. Hindriks, and J.-J. Meyer, editors, *Specification and Verification of Multi-Agent Systems*, pages 125–159. Springer, 2010.

[5] N. Bulling, V. Goranko, and W. Jamroga. Logics for reasoning about strategic abilities in multi-player games. In J. van Benthem, S. Ghosh, and R. Verbrugge, editors, *Models of Strategic Reasoning. Logics, Games, and Communities*, volume 8972 of *Lecture Notes in Computer Science*, pages 93–136. Springer, 2015.

[6] E.M. Clarke, T.A. Henzinger, H. Veith, and R. Bloem, editors. *Handbook of Model Checking*. Springer, 2018.

[7] Marco Conti and Andrea Passarella. The internet of people: A human and data-centric paradigm for the next generation internet. *Comput. Commun.*, 131:51–65, 2018.

[8] Pierluigi Contucci, Janos Kertesz, and Godwin Osabutey. Human-ai ecosystem with abrupt changes as a function of the composition. *PLOS ONE*, 17(5):1–12, 05 2022.

[9] Danny Dolev and Andrew Chi-Chih Yao. On the security of public key protocols. *IEEE Trans. Inf. Theory*, 29(2):198–207, 1983.

[10] Georgios Drainakis, Konstantinos V. Katsaros, Panagiotis Pantazopoulos, Vasilis Sourlas, and Angelos Amditis. Federated vs. centralized machine learning under privacy-elastic users: A comparative analysis. In *Proceedings of NCA*, pages 1–8. IEEE, 2020.

[11] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier, 1990.

[12] Andrew Fuchs, Andrea Passarella, and Marco Conti. Modeling human behavior part I - learning and belief approaches. *CoRR*, abs/2205.06485, 2022.

[13] Dieter Gollmann. *Computer Security (3. ed.)*. Wiley, 2011.

[14] Ian J. Goodfellow, Patrick D. McDaniel, and Nicolas Papernot. Making machine learning robust against adversarial inputs. *Commun. ACM*, 61(7):56–66, 2018.

[15] István Hegedüs, Gábor Danner, and Márk Jelasity. Gossip learning as a decentralized alternative to federated learning. In *Proceedings of IFIP DAIS*, volume 11534 of *Lecture Notes in Computer Science*, pages 74–90. Springer, 2019.

[16] István Hegedüs, Gábor Danner, and Márk Jelasity. Decentralized learning works: An empirical comparison of gossip learning and federated learning. *J. Parallel Distributed Comput.*, 148:109–124, 2021.

[17] W. Jamroga, W. Penczek, T. Sidoruk, P. Dembiński, and A. Mazurkiewicz. Towards partial order reductions for strategic ability. *Journal of Artificial Intelligence Research*, 68:817–850, 2020.

[18] Wojciech Jamroga, Michał Knapik, Damian Kurpiewski, and Łukasz Mikulski. Approximate verification of strategic abilities under imperfect information. *Artificial Intelligence*, 277, 2019.

[19] Wojciech Jamroga, Wojciech Penczek, and Teofil Sidoruk. Strategic abilities of asynchronous agents: Semantic side effects and how to tame them. In *Proceedings of KR 2021*, pages 368–378, 2021.

[20] Mazaher Kianpour and Shao-Fang Wen. Timing attacks on machine learning: State of the art. In *IntelliSys Volume 1*, volume 1037 of *Advances in Intelligent Systems and Computing*, pages 111–125. Springer, 2019.

[21] Panagiotis Kouvaros and Alessio Lomuscio. Towards scalable complete verification of relu neural networks via dependency-based branching. In *Proceedings of IJCAI*, pages 2643–2650. ijcai.org, 2021.

[22] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning-industry perspectives. In *IEEE Security and Privacy Workshops*, pages 69–75. IEEE, 2020.

[23] Damian Kurpiewski. STV – StraTegic Verifier. code repository, 2022. https://github.com/blackbat13/stv.

[24] Damian Kurpiewski, Wojciech Jamroga, Lukasz Masko, Lukasz Mikulski, Witold Pazderski, Wojciech Penczek, and Teofil Sidoruk. Verification of Multi-agent Properties in Electronic Voting: A Case Study. In *Proceedings of AiML 2022*, 2022.

[25] Damian Kurpiewski, Witold Pazderski, Wojciech Jamroga, and Yan Kim. STV+Reductions: Towards practical verification of strategic ability using model reductions. In *Proceedings of AAMAS*, pages 1770–1772. ACM, 2021.

[26] Alessio Lomuscio, Hongyang Qu, and Franco Raimondi. MCMAS: An open-source model checker for the verification of multi-agent systems. *International Journal on Software Tools for Technology Transfer*, 19(1):9–30, 2017.

[27] Valerio Lorenzo, Chiara Boldrini, and Andrea Passarella. SAI simulator for social AI gossiping, 2022. https://zenodo.org/record/5780042.

[28] Abdul-Rasheed Ottun, Pramod C. Mane, Zhigang Yin, Souvik Paul, Mohan Liyanage, Jason Pridmore, Aaron Yi Ding, Rajesh Sharma, Petteri Nurmi, and Huber Flores. Social-aware federated learning: Challenges and opportunities in collaborative data training. *IEEE Internet Computing*, pages 1–7, 2022.

[29] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[30] Pierre-Yves Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2):82–93, 2004.

[31] Y. Shoham and K. Leyton-Brown. *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.

[32] Social AI gossiping. Micro-project in Humane-AI-Net. Project website, 2022. https://www.ai4europe.eu/research/research-bundles/social-ai-gossiping.

[33] Social Explainable AI, CHIST-ERA. Project website, 2021–24. http://www.sai-project.eu/.

[34] Mustafa Toprak, Chiara Boldrini, Andrea Passarella, and Marco Conti. Harnessing the power of ego network layers for link prediction in online social networks. *CoRR*, abs/2109.09190, 2021.

[35] G. Weiss, editor. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. MIT Press: Cambridge, Mass, 1999.

[36] Min Wu, Matthew Wicker, Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. A game-based approximate verification of deep neural networks with provable guarantees. *Theor. Comput. Sci.*, 807:298–329, 2020.