Recognition of Behavioural Intention in Repeated Games using Machine Learning

Alessandro Di Stefano[1], Chrisina Jayne[1], Claudio Angione[1], The Anh Han[1]

[1]School of Computing, Engineering and Digital Technologies, Teesside University, Middlesbrough, UK

Abstract

Intention recognition entails the process of becoming aware of another agent's intention by inferring it through its actions and their effects on the environment. It allows agents to prevail when interacting with others in both cooperative and hostile environments. One of the main challenges in intention recognition is generating and collecting large amounts of data, and then being able to infer and recognise strategies. To this aim, in the context of repeated interactions, we generate diverse datasets, characterised by various noise levels and complexities. We propose an approach using different popular machine learning methods to classify strategies represented by sequences of actions in the presence of noise. Experiments have been conducted by varying the noise level and the number of generated strategies in the input data. Results show that the adopted methods are able to recognise strategies with high accuracy. Our findings and approach open up a novel research direction, consisting of combining machine learning and game theory in generating large and complex datasets and making inferences. This can allow us to explore and quantify human behaviours based on data-driven and generative models.

Introduction

Intention recognition or mind reading can be found in many kinds of interactions and communications, not only in Human, but also many other species (Tomasello, 2008; Rand et al., 2015; Falk et al., 2008; Meltzoff, 2007; Santos et al., 2015; Pereira and Han, 2009). The knowledge about intention of others in a situation could enable one to plan in advance, either to secure a successful cooperation or to deal with potential hostile behaviours (van Hees and Roy, 2008; Roy, 2009; Pereira et al., 2022). Given the advantage of accurately identifying the intentions of others in strategic interaction, intention recognition has been incorporated in various computational models studying humans' collective behaviours, such as cooperation and trust (Han et al., 2012, 2011a,b; Fujimoto and Kaneko, 2019; Nakamura and Ohtsuki, 2016; Montero-Porras et al., 2022; Falk et al., 2008; Andras et al., 2018).

Intention recognition is defined, in general terms, as the process of becoming aware of the intention of another agent and, more technically, as the problem of inferring an agent's intention through its actions and their effects on the environment (Kautz and Allen, 1986; Charniak and Goldman, 1993; Heinze, 2003; Han, 2013). For the recognition task, several issues can be raised given the distinction between the model an agent creates about himself/herself and the one used to describe others, often addressed in the context of the Theory of Mind study (Iacoboni et al., 2005; Rizzolatti and Craighero, 2004; Nakahara and Miyashita, 2005). The problem of intention recognition has been paid much attention in AI, Philosophy and Psychology for several decades (Kautz and Allen, 1986; Charniak and Goldman, 1993; Bratman, 1987, 1999; Geib and Goldman, 2009; Han, 2013; Sukthankar et al., 2014). Recently, at the intersection of AI and agent-based research, several machine learning techniques have been explored and evaluated in the context of analysing or surrogating agent-based models (Angione et al., 2022; Sert et al., 2020), with applications to population and health-based research (Sivakumar et al., 2022; Silverman et al., 2021).

A benchmark often used in studying intention recognition methods is based on the context of iterated prisoner's dilemma (IPD) game, where an agent's behavioural strategy (considered as the agent's intention) is to be inferred based on the agent's and its opponents' past actions during the course of a repeated game (Han et al., 2012, 2011b; Fujimoto and Kaneko, 2019; Nakamura and Ohtsuki, 2016; Montero-Porras et al., 2022). Game theory has proven suitable and powerful for modelling strategic behaviours in multi-agent settings (Han, 2022; Bloembergen et al., 2015; Parsons and Wooldridge, 2002; Nisan et al., 2007; Shoham and Leyton-Brown, 2008; Abate et al., 2021; Alalawi et al., 2019), which thus provides a suitable framework to study intention recognition.

In this paper, we approach the problem of intention recognition (i.e., strategy inference) in game theoretical tasks as a classification problem, where the intention is modelled as a behavioural strategy, and exploit recent advances in Machine Learning (ML) to address it. We generate a wide range of datasets of abstract plays from IPD to study behaviour prediction and intention recognition.

We show that the datasets provide various challenges for ML classification algorithms. Our results highlight which ML algorithms perform well against different challenges (e.g. noise, missing data, multi-strategy classification).

Given the proven suitability of game theory for modelling human strategic interactions and decision-making processes (Camerer, 2011), we argue that this approach allows us to conveniently and cost-efficiently generate (large-scale) data for investigating human behaviour, e.g. to provide proof of concept analyses and guide data collection. Note a large body of research on using ML for learning mathematical functions from abstract data (Jordan and Mitchell, 2015). We argue that game theoretical tasks, as a suitable framework for studying strategic interactions in real-world environments, provide a scenario where machine learning approaches can allow generating data and enhance both the accuracy of behavioural recognition and its robustness also in presence of noise.

Fig. 1 illustrates our approach and the modelling pipeline. Starting from the sequence of actions corresponding to the different strategies, namely, ALLC, ALLD, TFT, GTFT and WSLS (see next section for details), in presence of noise, we use machine learning models to make inferences and recognise intentions for varying the number of strategies and the noise level.

The remainder of the paper is organised as follows. The next section describes the data generation process in the IPD and the data preparation before applying the ML methods. Then, in the Machine Learning methods section, we present the models and algorithms used for classification. The Results section presents and discusses the performance of the ML models for various noise levels and varying the number of strategies. Finally, the Conclusion and Outlook section discusses findings and a few perspectives and directions of future research.

Data Generation for ML

This section describes how datasets are generated in the context of the iterated prisoner's dilemma game, for training and evaluated ML classification algorithms.

Iterated Prisoner's Dilemma and strategies

Interactions are modelled as symmetric two-player games defined by the payoff matrix

$$\begin{array}{c c} & \begin{array}{cc} C & D \end{array} \\ \begin{array}{c} C \\ D \end{array} & \begin{pmatrix} R,R & S,T \\ T,S & P,P \end{pmatrix}. \end{array}$$

A player who chooses to cooperate (C) with someone who defects (D) receives the sucker's payoff S, whereas the defecting player gains the temptation to defect, T. Mutual cooperation (resp., defection) yields the reward R (resp., punishment P) for both players. Depending on the ordering of these four payoffs, different social dilemmas arise (Macy and Flache, 2002; Santos et al., 2006; Sigmund, 2010). Namely, in this work we are concerned with the Prisoner's Dilemma (PD), being characterised by $T > R > P > S$.

In a single round, it is always best to defect, but cooperation may be rewarded if the game is repeated. In this case, it is additionally required that a mutual cooperation is preferred over an equal probability of unilateral cooperation and defection, i.e. $2R > T + S$; otherwise alternating between cooperation and defection would lead to a higher payoff than mutual cooperation.

In the context of IPD, a crucial factor is noise/error. That is, a player's intended move is wrongly performed with a given execution error, referred here the noise level, denoted by ϵ. For example, a C move is implemented as a D and vice versa.

Given the nature of the IPD, some strategies for playing the game successfully have emerged in the literature, both as they closely capture human behaviours in experimental studies and are shown to be robust in several settings (e.g. evolutionary simulations) (Axelrod, 1984; Sigmund, 2010; Nowak and Sigmund, 1992a). We now describe some of the most important and popular strategies in the context of IPD for our ML classification study.

1. **ALLC**: unconditional cooperators who always cooperate, in every round of the IPD.

2. **ALLD**: unconditional cooperators who always defect, in every round of the IPD.

3. **TFT**: Tit-for-tat strategy starts by cooperating (playing C) in the first round, then does whatever the opponent did in the previous round. It won both Axelrod's tournaments (Axelrod, 1984; Axelrod and Hamilton, 1981).

4. **GTFT**: Generous tit-for-tat strategy is similar to TFT. The only difference is that it cooperates if the opponent cooperated in the previous round, but sometimes cooperates even if the opponent defected (with a fixed probability $p > 0$) (Nowak and Sigmund, 1992b). GTFT is more successful than TFT when noise is present (i.e. $\epsilon > 0$).

5. **WSLS**: Win-stay-lose-shift strategy starts by cooperating and repeats the previous move whenever it did well, but changes otherwise. *WSLS* corrects mistakes better than *GTFT*. It is the winning strategy chosen by evolution, as shown in several models (Nowak and Sigmund, 1993).

Data Generation for ML algorithms

In the following, we describe how to create datasets consisting of sequences of actions by a certain strategy in IPD for training and testing the described ML algorithms, for a given set of strategies, which is in line with (Han et al., 2012). We start by making an assumption that all strategies to be

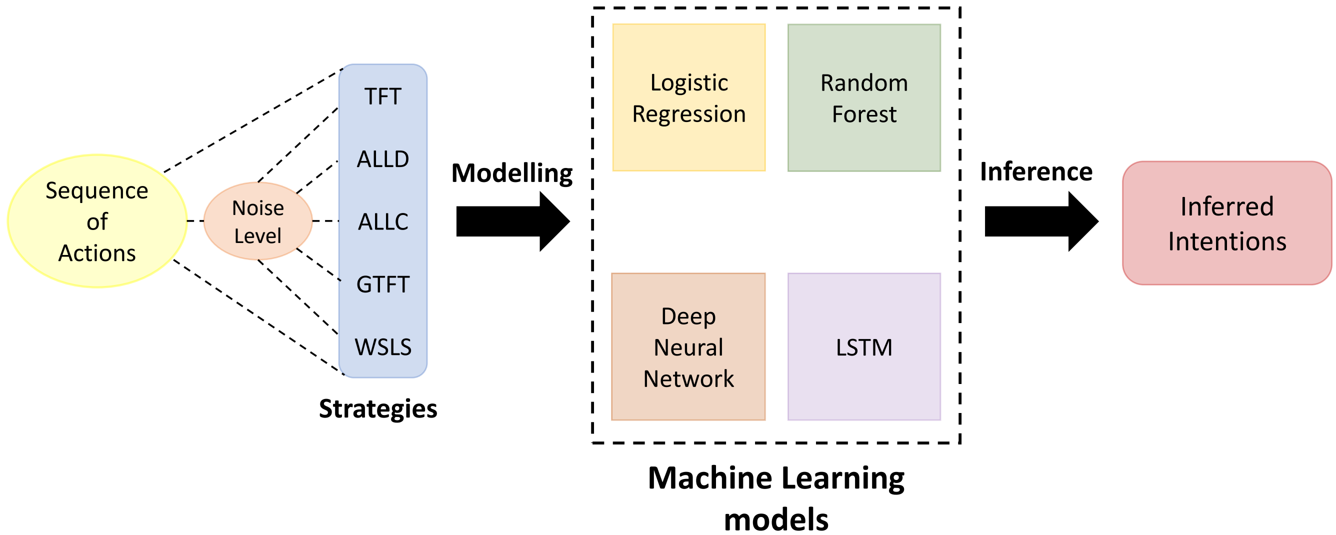

Figure 1: Modelling pipeline - starting from the sequence of actions corresponding to the different strategies (ALLC, ALLD, TFT, GTFT, WSLS), in presence of noise, we use machine learning models used to make inferences and recognise intentions for datasets with varying the number of strategies and the noise level.

recognised have the memory size of at most M ($M \geq 0$)—i.e. they do not take into account in the decision-making any past rounds that are at a time distance greater than M. Although we focus on strategies with a max memory size of $M = 1$, this general framework allows us to easily expand the dataset to more complex strategies with large M, see e.g. (Hilbe et al., 2017; Han, 2013; Devaine et al., 2014; Pereira et al., 2017).

For clarity of representation, abusing notations, R, S, T and P are henceforth also referred to as (elementary) game states, in a single round of interaction. Additionally, E (standing for *empty*) is used to refer to a game state having had no interaction. The most basic elements in a dataset are the (dataset) actions, having the following representation. Then, a data entry/row in a dataset is a sequence of actions that a player uses, for a given strategy (X), written as follows: $X : A1, A2, ...$

An action A_i in the sequence is given in the form $s_1...s_M\xi$, where $s_i \in \{E, R, T, S, P\}$, $1 \leq i \leq M$, are the states of the M last interactions, and $\xi \in \{C, D\}$ is the current move.

We denote by Σ_M the set of all possible types of action. Clearly, $|\Sigma_M| = 2 \times 5^M$. For example,

$$\Sigma_1 = \{EC, RC, TC, SC, PC, ED, RD, TD, SD, PD\}$$

This way of encoding actions enables us to save the game states without having to save the co-player's moves, thus simplifying the representation of dataset entries.

As an example, let us consider *TFT* and the following sequence of its interactions with a player X, in the presence of noise (i.e. $\epsilon > 0$)

round :	0	1	2	3	4	5
TFT :	–	C	C	D	D	D
X :	–	C	D	D	C	D
TFT-states :	E	R	S	P	T	P

The corresponding entry for *TFT* from this interaction in the dataset is thus *TFT: EC RC SD PD TD*. At 0-th round, there is no interaction, thus the game state is E. *TFT* starts by cooperating (1-st round), hence the first action of the plan session is EC. Since player X also cooperates in the 1-st round, the game state at this round is R. *TFT* reciprocates in the 2-nd round by cooperating, hence the second action of the plan session is RC. Similarly for the third and the fourth actions. Now, at the 5-th round, *TFT* should cooperate since X cooperated in 4-th round, but because of noise, it makes an error to defect. Therefore, the 5-th action is TD.

For the purpose of evaluating different ML classification algorithms for inferring strategies in IPD interactions, we generate datasets for an increasingly larger set of strategies, and for different levels of noise. For a classification problem, the number of strategies corresponds to the number of classes, to one of which an input is classified. Namely, we generate a total of 12 datasets, for three set of strategies $\{ALLC, ALLD, TFT\}$, $\{ALLC, ALLD, TFT, WSLS\}$ and $\{ALLC, ALLD, TFT, WSLS, GTFT\}$, and for four different levels of noise: $\epsilon = 0,\ 0.05,\ 0.1,\ 0.2$, corresponding to situations with no noise to medium levels of noise and rather high level of noise.

As such, these 12 datasets provide us with different challenges for ML classification. We are interested in how different algorithms can handle the different challenges successfully, providing insights for using ML for understanding human intentions in the real-world setting.

Data Preparation

In order to apply ML to classify the sequences of data corresponding to each strategy, these have to be represented as numeric data. This has been done by replacing each letter *C, D, T, R, P, S, E* with a numeric value. The strategies $\{ALLC, ALLD, TFT, WSLS, GTFT\}$ encoded as categorical variables. The generated datasets consist of sequences of variable lengths. To apply traditional ML algorithms, the sequences are converted into a flat vector with a length equal to the maximum length of the sequences by padding with zeros and normalised. In the case of classification with sequence type ML, this step is not required and sequences with different lengths can be used.

Machine Learning methods

Classification algorithms

A wide range of supervised ML classification algorithms can be applied to map a data instance $\mathbf{x}$ to a particular class label y. The data instances as described in the previous section are vectors/sequences representing the actions and payoffs of the players, while the class label represent the strategies in the context of IPD. In this work we utilise widely adopted ML algorithms for classification: Artificial Neural Networks (ANN), Random Forest (RF), Logistic Regression (LR) and Long Short Term Memory ANN (LSTM). The datasets used for the experiments consist of the generated datasets for the set of strategies and different levels of noise. The experiments utilise train test splits based on stratified 5-folds cross-validation.

Artificial Neural Networks ANN is based on the concept of the human brain's neural network and its neurons. An ANN model typically consists of artificial neurons organised in layers known as an input layer, hidden and output layers. In a typical ANN with an input layer, one hidden layer and one output layer, the neurons have connections from input to hidden and from hidden to output layer. The connections are associated with so-called synaptic weights/parameters. The ANN processes inputs through a series of functions in a forward fashion and produces an output. The weights are determined by optimizing an error function that minimizes the error between the predicted and the desired outputs. Fig. 2 illustrates the architecture of the ANN model utilised in this paper with an input layer of 20 inputs, 2 hidden layers with 10 and 8 neurons, respectively and an output layer with 3 to 5 neurons corresponding to the classes representing the strategies of the players $\{ALLC, ALLD, TFT\}$, $\{ALLC, ALLD, TFT, WSLS\}$ and $\{ALLC, ALLD, TFT, WSLS, GTFT\}$. The input layer of 20 neurons corresponds to the flattened sequence

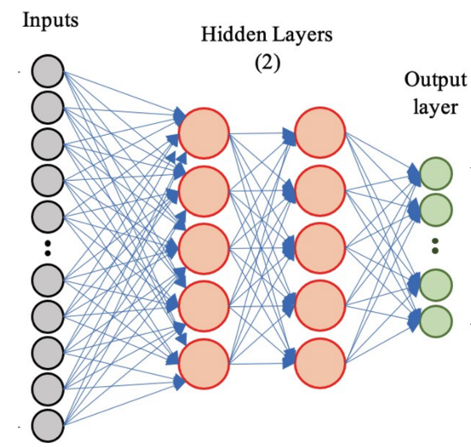

Figure 2: ANN model architecture. The ANN model utilised in this paper consists of an input layer of 20 inputs, w hidden layers with 10 and 8 neurons, respectively and an output layer with 3 to 5 neurons corresponding to the classes representing the strategies of the players

of encoded actions and payoffs of a player. The number of neurons and hidden layers have been selected by experimentation. The activation function utilised in the hidden layers is the ReLU (rectified linear unit) activation function $f(x) = max(0, x)$. SoftMax activation function is applied to the output layer $\sigma(\vec{z}) = \frac{\exp^{z_i}}{\sum_{j=1}^{11} e^{z_j}}$. We utilise Dropout and Batch Normalization to reduce overfitting (Garbin et al., 2020). The Adam optimizer proposed in (Kingma and Ba, 2015) is used as optimization algorithm.

Random Forests RF is an ensemble classification and regression technique introduced by Breiman (2001) that has shown to be an effective classification technique. The ensemble of decision trees is designed to train more than one classifier, and then aggregate the predictions of all models and perform predictions by majority voting. A good ensemble needs models to be diverse enough and independent of each other to ensure good performance. RF generates a diversified ensemble using Bootstrap aggregating (bagging). Bagging is a sampling method that samples data from the training set with replacement. With such an approach, an instance in the dataset can be sampled more than for the same model. At the same time, other instances may not appear at all during the training process. It is estimated that following this approach, more than 63% of unique instances from the training set will be used during the training process, while almost 37% of the instances will not be sampled at all, and will be used to estimate the "out-of-bag" error. In addition, and to ensure more diversified ensemble RF and at each node split, only a subset of features is drawn randomly to assess the quality of each feature. The inputs and outputs of the RF model are the same as in the ANN model described earlier.

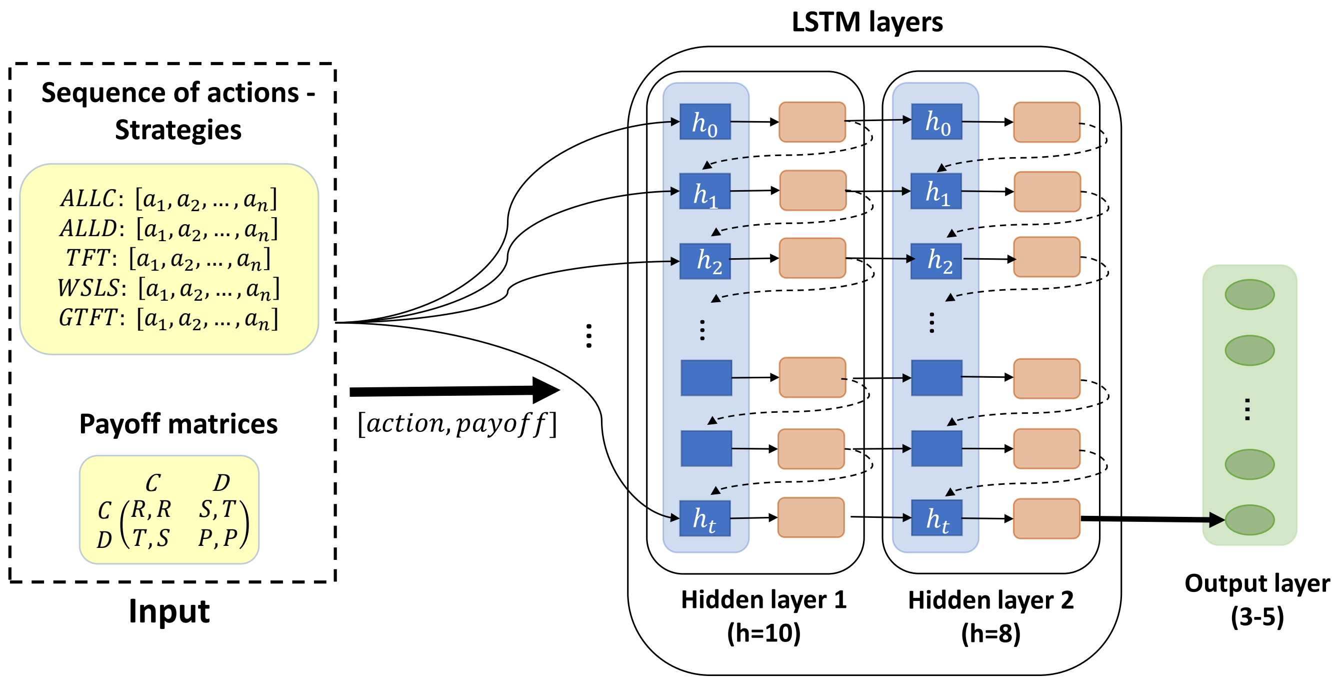

Figure 3: Deep LSTM model architecture. The input layer consists of the sequence of actions of the players. The LSTM layers consist of LSTM cells each having hidden units and two hidden layers with 10 and 8 neurons, respectively. The output layer consists of three to five classes corresponding to the set of strategies.

Logistic Regression LR is used to model the probability of an event happening based on a linear combination of input variables and a binary outcome variable. A *logit* transformation is applied on the odds, i.e. the probability of success divided by the probability of failure. Given input variables, $x_1, \ldots, x_n$ the logistic regression model aims to estimate the coefficients $a_0, a_1, \ldots, a_n$ using maximum likelihood. The probability p is expressed as follows:

$$p = \frac{\exp^{a_0 + \sum_{i=1}^{n} a_i x_i}}{1 + \exp^{a_0 + \sum_{i=1}^{n} a_i x_i}}.$$

The quantity $p/(1-p)$ is called the odds, and can take on any value between 0 and ∞. Using the equation xx we obtain the log odds, or logit, expressed as a linear combination of the input variables, $x_1, \ldots, x_n$ as follows:

$$\ln \frac{p}{(1-p)} = a_0 + \sum_{i=1}^{n} a_i x_i.$$

Typically logistic regression is used for binary classification. However, it can be extended to multi-class cases by using multinomial logistic regression. The inputs and outputs of the LR model are the same as in the ANN model described earlier.

Long Short-Term Memory Long Short-Term Memory Hochreiter and Schmidhuber (1997) models are a type of so-called recurrent artificial neural network capable of learning from sequences. LSTM uses cells and gates (input, output and forget and state candidate) which regulate the direction of information through the model while making use of time backpropagation to modify the weight values. The cell remembers values over time, input and forget gates control what information to retain or discard, while the output gate controls the state of the output information based on the previous and current states. In this work, the input for the LSTM model is the sequence of actions of the players and the output consists of the classes representing the strategies of the players. The architecture of the LSTM model consists of an input layer sequence of [action, game state], two hidden layers with 10 and 8 neurons respectively and an output layer of the classes corresponding to the set of strategies. The Deep LSTM model architecture is shown in Fig. 3. Since not all sequences have the same length, masking is utilised to inform the model that some part of the data is padding and should be ignored. The hidden layers use the ReLU activation function, the output layer uses the SoftMax activation function, and the Adam optimizer is used as the optimisation algorithm (Kingma and Ba, 2015).

Results

We evaluate performance of the four classification algorithms to the 12 generated datasets, which were described above. We first study how different algorithms perform in

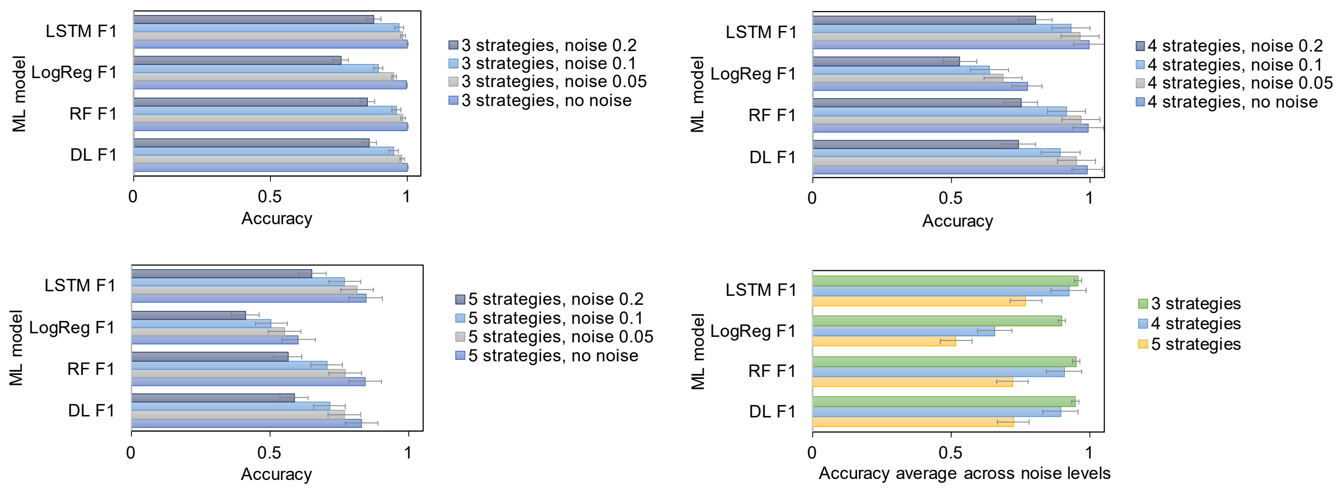

Figure 4: Overview of the average F1 accuracy results obtained by each method for various noise levels. Error bars represent the standard error.

general across the noise levels and number of strategies. We then check how difficult it is to recognise the corresponding strategies in the datasets.

Performance of the algorithms

Fig. 4 illustrates how the distinct machine learning algorithms used for intention recognition are able to recognise the different strategies. We show the average F1 accuracy results obtained by each method for various noise levels and the number of strategies (namely, 3, 4, or 5).

Logistic regression (LogReg) performance decreases when increasing the number of strategies and the noise level. These results are in line with the inherent limitations of logistic regression, which does not support multi-classification natively (although its extension, i.e., multinomial logistic regression allows for multi-class classification), and is susceptible to noisy data (Lei et al., 2019). Moreover, the logistic regression's capacity to fit data is inadequate, and it is not able to capture the data redundancy effects on classification. To address the logistic regression's susceptibility to noisy data, a few solutions have been recently proposed, such as sparse logistic regression (SLR), which is also able to give sparse solutions by adding an L1 Regularization term (Wu et al., 2019). In (Algamal and Lee, 2019), it has been shown how the adaptive lasso logistic regression (ALLR) algorithm is able to obtain stable feature selection results. Another solution is the elastic-net logistic regression (ENLR) (Zou and Hastie, 2005), which adds L1 and L2 regularization terms, and the function of group feature selection.

Random Forest (RF) and the Deep Neural Network (DL) show similar performance in presence of noise and varying the number of strategies. Differently from logistic regression, these methods are more robust to the increase in the number of strategies. However, we can observe a decrease in the performance when considering 5 strategies, and with the higher noise level. Overall, the higher noise level negatively affects the accuracy when considering more strategies. Results are coherent with research literature on random forests, since random forests are known for being robust to label noise (Boateng et al., 2020; Rolnick et al., 2017). In neural networks, noise is sometimes injected to the inputs, or to the weights, or hidden units (e.g., dropout) (Noh et al., 2017), gradients and even activation function, during training to make the training process more robust and reduce generalization error (Song et al., 2022). Nevertheless, when training a neural network without any noise, the classifier will not be able to handle the presence of noise and this will negatively affect performances.

LSTM outperforms DL and RF algorithms due to its inherent capability of learning order dependence in sequence prediction problems. LSTMs are able to handle long-term dependencies, since these are able to remember information for extended periods of time. Moreover, LSTMs are very efficient at modelling complex sequential data, as these learn high-level representations that capture the structure of the data. Comparing LSTM with DL and RF, we can see how the accuracy is higher in presence of noise when increasing the number of strategies. In other words, although a high noise level reduces the accuracy of LSTM, performance shows a higher accuracy with respect to RF and DL models. The LSTM's higher robustness to the noise is coherent with the recent literature discussing its remarkable de-noising capability (Yeo, 2019).

Recognition of strategies

In Fig. 5 we evaluate which intentions or strategies we are able to recognise. As expected, the unconditional strategies, i.e., ALLC and ALLD, are the easiest to be recog-

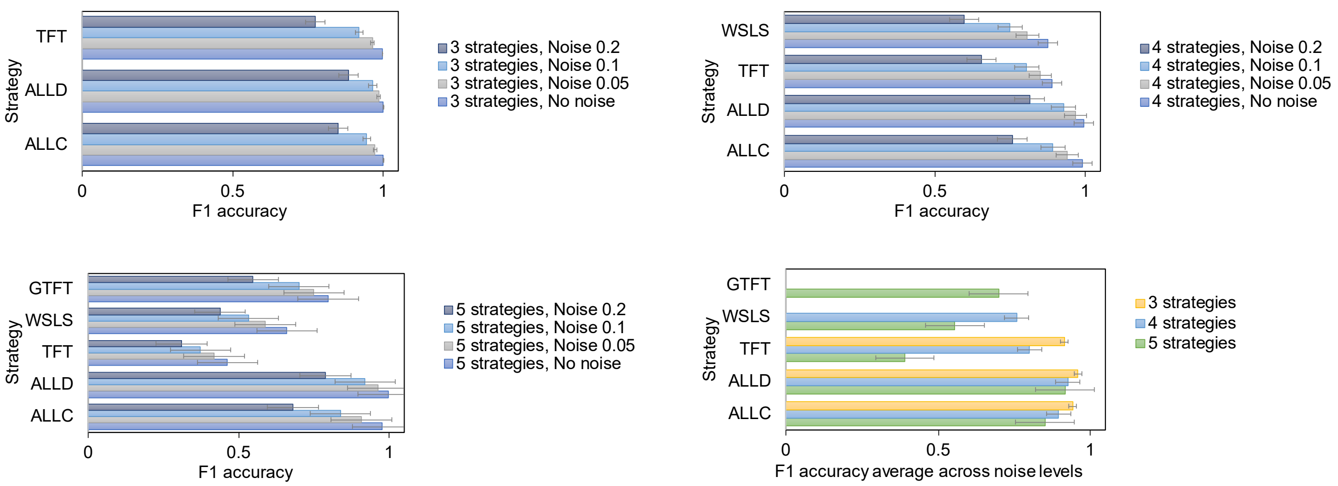

Figure 5: Overview of the average F1 accuracy results obtained for each strategy for various noise levels. Error bars represent the standard error.

nized, even in presence of noise. Even increasing the number of strategies, all the proposed methods are able to identify when players tend to always cooperate or defect. Moreover, ALLD is easier to recognise than ALLC since it is the only one with a defective tendency (while other conditional strategies are cooperative by definition).

Conditional strategies, i.e. TFT, WSLS and GTFT, are more challenging to be recognised, specifically when more than one unconditional strategy is present. Nevertheless, when considering four strategies, all the methods are able to discriminate the TFT and WSLS. This is due to the clear distinction between the two strategies, in particular when the state of the previous round is mutual defection P (TFT would defect and WSLS would cooperate after that). Imhof et al. (2007) have underlined the two main drawbacks related to a TFT strategy. Firstly, TFT is not able to correct mistakes due to erroneous moves. Secondly, a population of TFT players is not evolutionary stable since, once the abundance of ALLC has fluctuated above a certain threshold, ALLD players can invade the population. This allows the ML model to easily recognise the sequence and distinguish TFT from unconditional strategies. While, when also the GTFT is included, TFT becomes the most difficult to be identified, due to its similarity with TFT. Finally, increasing the noise level proportionally reduces the F1 accuracy results.

Conclusions and Outlook

In the context of strategic games, we can ask a fundamental question: given a dataset consisting of sequences generated from a (well-defined) strategy, when is it possible to identify it (with high accuracy) as the true underlying strategy among all possible well-defined strategies, for someone who does not know the true strategy beforehand? There has been little attention to this issue. We argue that this behavioural strategy is a very distinct domain for this learning, given the complex dynamics and evolving nature of human (or animal or even any living organism in general) behaviours. Thus, game theoretical tasks provide a suitable mathematical abstraction to generate data. Our work is thus the first step in the characterisation of the rich phenomenology arising from the interplay between data size and noise, and parameters and strategies of identification from data.

There has been increasing attention to the study of how to learn closed-form mathematical models from data, see e.g. (Fajardo-Fontiveros et al., 2023) That is, given a data set generated from a closed-form model, when is it possible to identify it as the true generating model among all possible closed-form mathematical models? Similar learning problems have been studied for probabilistic graphical models and network models (Guimerà and Sales-Pardo, 2009; Vallès-Català et al., 2018).

In this work, we are able to deal with a huge amount of data (input sequences of strategies for every player) and recognise human intents also in presence of noise through ML/DL methods. This allows us to overcome a typical issue present in game theory, represented by the difficulty to generate and collect data and be able to infer and recognise strategies. Thus, through ML/DL we are able to address this issue and make our model data-driven.

Future directions. In this work, we considered intention recognition within the framework of repeated interactions. In the context of direct reciprocity (Trivers, 1971), intention recognition is being performed using information about past direct interactions. Naturally, the same principles could be extended to cope with indirect information, as in indirect reciprocity (Nowak and Sigmund, 2005; Pacheco et al., 2006; Perret et al., 2021; Krellner and Han, 2021), or in complex networks where individuals differ in their local neighbourhood and the number of connections (Boccaletti et al., 2006; Albert and Barabási, 2002; Newman, 2003; Cimpeanu et al., 2023; Ogbo et al., 2022). In real-world complex systems, interactions between nodes are not only dyadic but occur on multiple layers (Di Stefano et al., 2015; Battiston et al., 2021; Di Stefano et al., 2020). Recently, there has been an increased interest in understanding how cooperation or coordination can emerge or evolve in higher-order structures or hypergraphs (Burgio et al., 2020; Majhi et al., 2022; Bianconi, 2021). Hypergraphs provide a mathematical representation of a system, allowing from pairs to larger groups (Battiston et al., 2021; Bianconi, 2021). Intention recognition on hypergraphs calls for more accurate descriptions of networks, going beyond pairwise models towards higher-dimensional generalization of dyadic interactions (Burgio et al., 2020). A few works have recently addressed this challenge and several graph neural networks-based methods have been proposed for efficient graph representation learning through simplicial complexes able to characterise the higher-order interaction (Yang et al., 2022; Ebli et al., 2020; Roddenberry et al., 2021). Simplicial neural networks can be used for trajectory prediction based on sequential multidimensional data. Incorporating attention mechanisms in simplicial neural networks, and dynamically weight the interactions between neighbouring simplicies allow assigning a different importance, adapt and generalise to unseen simplicial structures (Goh et al., 2022). There are promising research directions for intention recognition when dealing with higher-order sequential multidimensional data.

Performing intention or plan recognition of a group or team of agents (instead of a single agent) has also been considered a major challenge (Pereira and Han, 2009; Sukthankar et al., 2014). Using multi-player games, such as the public goods game (Sigmund, 2010), we can readily generate suitable datasets for addressing this challenge. It would also be interesting to study behavioural strategies in the presence of an institution, which is studied for example in the context of institutional incentives modelling (Powers et al., 2018). We will generate datasets for capturing these extended settings and develop suitable ML methodologies for strategy classification.

In our analysis, all ML algorithms considered show a weaker performance for a higher level of noise. We will explore approaches to mitigate noise in data before performing ML algorithms (Zhu and Wu, 2004; Gupta and Gupta, 2019). Moreover, in this work, we focus on implementation noise. We can study other kinds of noise in the future, e.g. observation noises (Nowak and Sigmund, 2005; Perret et al., 2021). In other recent works, it has been underlined how there is a need to improve causality from observational data, and this can be even more challenging in the presence of noise (Huang et al., 2020). Several methods have been proposed, and typical approaches are based on the Granger causality (Shojaie and Fox, 2022). The main issue is that the underlying assumption is the presence of linear dynamics, while many real-world systems are inherently nonlinear (Tank et al., 2021). To this aim, the idea has been to incorporate neural networks (able to represent complex nonlinear interactions between inputs and outputs) in the definition of Granger causality (Tank et al., 2021; Yin and Barucca, 2022; Marcinkevičs and Vogt, 2020). These constitute viable approaches to address the presence of noise and produce interpretable models in recurrent neural networks (Marcinkevičs and Vogt, 2020).

References

Abate, A., Gutierrez, J., Hammond, L., Harrenstein, P., Kwiatkowska, M., Najib, M., Perelli, G., Steeples, T., and Wooldridge, M. (2021). Rational verification: game-theoretic verification of multi-agent systems. *Applied Intelligence*, 51(9):6569–6584.

Alalawi, Z., Zeng, Y., Elragig, A., and Han, T. A. (2019). Pathways to good healthcare services and patient satisfaction: An evolutionary game theoretical approach. In *ALIFE 2019: The 2019 Conference on Artificial Life*, pages 135–142. MIT Press.

Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47.

Algamal, Z. Y. and Lee, M. H. (2019). A two-stage sparse logistic regression for optimal gene selection in high-dimensional microarray data classification. *Advances in data analysis and classification*, 13(3):753–771.

Andras, P., Esterle, L., Guckert, M., Han, T. A., Lewis, P. R., Milanovic, K., Payne, T., Perret, C., Pitt, J., Powers, S. T., et al. (2018). Trusting intelligent machines: Deepening trust within socio-technical systems. *IEEE Technology and Society Magazine*, 37(4):76–83.

Angione, C., Silverman, E., and Yaneske, E. (2022). Using machine learning as a surrogate model for agent-based simulations. *PloS one*, 17(2):e0263150.

Axelrod, R. (1984). *The Evolution of Cooperation*. Basic Books, ISBN 0-465-02122-2.

Axelrod, R. and Hamilton, W. (1981). The evolution of cooperation. *Science*, 211:1390–1396.

Battiston, F., Amico, E., Barrat, A., Bianconi, G., Ferraz de Arruda, G., Franceschiello, B., Iacopini, I., Kéfi, S., Latora, V., Moreno, Y., et al. (2021). The physics of higher-order interactions in complex systems. *Nature Physics*, 17(10):1093–1098.

Bianconi, G. (2021). *Higher-order networks*. Cambridge University Press.

Bloembergen, D., Tuyls, K., Hennes, D., and Kaisers, M. (2015). Evolutionary dynamics of multi-agent learning: A survey. *Journal of Artificial Intelligence Research*, 53:659–697.

Boateng, E. Y., Otoo, J., and Abaye, D. A. (2020). Basic tenets of classification algorithms k-nearest-neighbor, support vector machine, random forest and neural network: a review. *Journal of Data Analysis and Information Processing*, 8(4):341–357.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U. (2006). Complex networks: Structure and dynamics. *Physics reports*, 424(4-5):175–308.

Bratman, M. E. (1987). *Intention, Plans, and Practical Reason*. The David Hume Series, CSLI.

Bratman, M. E. (1999). *Faces of Intention: Selected Essays on Intention and Agency*. Cambridge University Press.

Breiman, L. (2001). Random forests. *Mach. Learn.*, 45(1):5–32.

Burgio, G., Matamalas, J. T., Gómez, S., and Arenas, A. (2020). Evolution of cooperation in the presence of higher-order interactions: From networks to hypergraphs. *Entropy*, 22(7):744.

Camerer, C. F. (2011). *Behavioral game theory: Experiments in strategic interaction*. Princeton university press.

Charniak, E. and Goldman, R. (1993). A bayesian model of plan recognition. *Artificial Intelligence*, 64(1):53–79.

Cimpeanu, T., Di Stefano, A., Perret, C., and Han, T. A. (2023). Social diversity reduces the complexity and cost of fostering fairness. *Chaos, Solitons & Fractals*, 167:113051.

Devaine, M., Hollard, G., and Daunizeau, J. (2014). Theory of mind: did evolution fool us? *PloS One*, 9(2):e87619.

Di Stefano, A., Scatà, M., Attanasio, B., La Corte, A., Liò, P., and Das, S. K. (2020). A novel methodology for designing policies in mobile crowdsensing systems. *Pervasive and Mobile Computing*, 67:101230.

Di Stefano, A., Scatà, M., La Corte, A., Liò, P., Catania, E., Guardo, E., and Pagano, S. (2015). Quantifying the role of homophily in human cooperation using multiplex evolutionary game theory. *PloS one*, 10(10):e0140646.

Ebli, S., Defferrard, M., and Spreemann, G. (2020). Simplicial neural networks. *arXiv preprint arXiv:2010.03633*.

Fajardo-Fontiveros, O., Reichardt, I., De Los Ríos, H. R., Duch, J., Sales-Pardo, M., and Guimerà, R. (2023). Fundamental limits to learning closed-form mathematical models from data. *Nature Communications*, 14(1):1043.

Falk, A., Fehr, E., and Fischbacher, U. (2008). Testing theories of fairness—intentions matter. *Games and Economic Behavior*, 62(1):287–303.

Fujimoto, Y. and Kaneko, K. (2019). Functional dynamic by intention recognition in iterated games. *New Journal of Physics*, 21(2):023025.

Garbin, C., Zhu, X., and Marques, O. (2020). Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimedia Tools and Applications*, 79:12777–12815.

Geib, C. W. and Goldman, R. P. (2009). A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, 173(2009):1101–1132.

Goh, C. W. J., Bodnar, C., and Lio, P. (2022). Simplicial attention networks. *arXiv preprint arXiv:2204.09455*.

Guimerà, R. and Sales-Pardo, M. (2009). Missing and spurious interactions and the reconstruction of complex networks. *Proceedings of the National Academy of Sciences*, 106(52):22073–22078.

Gupta, S. and Gupta, A. (2019). Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science*, 161:466–474. The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia.

Han, T. A. (2013). *Intention Recognition, Commitments and Their Roles in the Evolution of Cooperation: From Artificial Intelligence Techniques to Evolutionary Game Theory Models*, volume 9. Springer SAPERE series.

Han, T. A. (2022). Emergent behaviours in multi-agent systems with evolutionary game theory. *AI Communications*, 35(4):327–337.

Han, T. A., Moniz Pereira, L., and Santos, F. C. (2011a). Intention recognition promotes the emergence of cooperation. *Adaptive Behavior*, 19(4):264–279.

Han, T. A., Pereira, L. M., and Santos, F. C. (2011b). The role of intention recognition in the evolution of cooperative behavior. In *IJCAI 2011-22nd International Joint Conference on Artificial Intelligence*, pages 1684–1689.

Han, T. A., Pereira, L. M., and Santos, F. C. (2012). Corpus-based intention recognition in cooperation dilemmas. *Artificial Life*, 18(4):365–383.

Heinze, C. (2003). *Modeling Intention Recognition for Intelligent Agent Systems*. PhD thesis, The University of Melbourne, Australia.

Hilbe, C., Martinez-Vaquero, L. A., Chatterjee, K., and Nowak, M. A. (2017). Memory-n strategies of direct reciprocity. *Proceedings of the National Academy of Sciences*, 114(18):4715–4720.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Huang, Y., Fu, Z., and Franzke, C. L. (2020). Detecting causality from time series in a machine learning framework. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(6):063116.

Iacoboni, M., Molnar-Szakacs, I., Gallese, V., Buccino, G., Mazziotta, J. C., and Rizzolatti, G. (2005). Grasping the intentions of others with one's own mirror neuron system. *PLoS biology*, 3(3):e79.

Imhof, L. A., Fudenberg, D., and Nowak, M. A. (2007). Tit-for-tat or win-stay, lose-shift? *Journal of theoretical biology*, 247(3):574–580.

Jordan, M. I. and Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260.

Kautz, H. and Allen, J. (1986). Generalized plan recognition. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI-86)*, pages 32–38. AAAI.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Krellner, M. and Han, T. A. (2021). Pleasing enhances indirect reciprocity-based cooperation under private assessment. *Artificial Life*, 27(3–4):246–276.

Lei, D., Zhang, H., Liu, H., Li, Z., and Wu, Y. (2019). Maximal uncorrelated multinomial logistic regression. *IEEE Access*, 7:89924–89935.

Macy, M. W. and Flache, A. (2002). Learning dynamics in social dilemmas. *Proc. Natl. Acad. Sci. USA*, 99:7229–7236.

Majhi, S., Perc, M., and Ghosh, D. (2022). Dynamics on higher-order networks: A review. *Journal of the Royal Society Interface*, 19(188):20220043.

Marcinkevičs, R. and Vogt, J. E. (2020). Interpretability and explainability: A machine learning zoo mini-tour. *arXiv preprint arXiv:2012.01805*.

Meltzoff, A. N. (2007). The 'like me'framework for recognizing and becoming an intentional agent. *Acta psychologica*, 124(1):26–43.

Montero-Porras, E., Grujić, J., Fernández Domingos, E., and Lenaerts, T. (2022). Inferring strategies from observations in long iterated prisoner's dilemma experiments. *Scientific reports*, 12(1):7589.

Nakahara, K. and Miyashita, Y. (2005). Understanding intentions: Through the looking glass. *Science*, 308(5722):644–645.

Nakamura, M. and Ohtsuki, H. (2016). Optimal decision rules in repeated games where players infer an opponent's mind via simplified belief calculation. *Games*, 7(3):19.

Newman, M. E. (2003). The structure and function of complex networks. *SIAM review*, 45(2):167–256.

Nisan, N., Roughgarden, T., Tardos, E., and Vazirani, V. V. (2007). *Algorithmic game theory*. Cambridge university press.

Noh, H., You, T., Mun, J., and Han, B. (2017). Regularizing deep neural networks by noise: Its interpretation and optimization. *Advances in Neural Information Processing Systems*, 30.

Nowak, M. A. and Sigmund, K. (1992a). Tit for tat in heterogeneous populations. *Nature*, 355:250–253.

Nowak, M. A. and Sigmund, K. (1992b). Tit for tat in heterogeneous populations. *Nature*, 355:250–253.

Nowak, M. A. and Sigmund, K. (1993). A strategy of win-stay, lose-shift that outperforms tit for tat in prisoner's dilemma. *Nature*, 364:56–58.

Nowak, M. A. and Sigmund, K. (2005). Evolution of indirect reciprocity. *Nature*, 437(1291-1298).

Ogbo, N. B., Cimpeanu, T., Di Stefano, A., and Han, T. A. (2022). Shake on it: The role of commitments and the evolution of coordination in networks of technology firms. In *Artificial Life Conference Proceedings 34*, volume 2022, page 41.

Pacheco, J. M., Santos, F. C., and Chalub, F. A. C. C. (2006). Stern-judging: A simple, successful norm which promotes cooperation under indirect reciprocity. *PLoS Comput. Biol.*, 2(12):e178.

Parsons, S. and Wooldridge, M. (2002). Game theory and decision theory in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 5:243–254.

Pereira, L. M. and Han, T. A. (2009). Intention recognition via causal bayes networks plus plan generation. In *Progress in Artificial Intelligence: 14th Portuguese Conference on Artificial Intelligence, EPIA 2009, Aveiro, Portugal, October 12-15, 2009. Proceedings 14*, pages 138–149. Springer.

Pereira, L. M., Lenaerts, T., Martinez-Vaquero, L. A., and Han, T. A. (2017). Social manifestation of guilt leads to stable cooperation in multi-agent systems. In *AAMAS*, pages 1422–1430.

Pereira, L. M., Lopes, A. B., et al. (2022). Employing ai to better understand our morals. *Entropy*, 24(1):10.

Perret, C., Krellner, M., and Han, T. A. (2021). The evolution of moral rules in a model of indirect reciprocity with private assessment. *Scientific Reports*, 11(1):23581.

Powers, S. T., Ekárt, A., and Lewis, P. R. (2018). Modelling enduring institutions: The complementarity of evolutionary and agent-based approaches. *Cognitive Systems Research*, 52:67–81.

Rand, D. G., Fudenberg, D., and Dreber, A. (2015). It's the thought that counts: The role of intentions in noisy repeated games. *Journal of Economic Behavior & Organization*, 116:481–499.

Rizzolatti, G. and Craighero, L. (2004). The mirror-neuron system. *Annual Review of Neuroscience*, 27:169–192.

Roddenberry, T. M., Glaze, N., and Segarra, S. (2021). Principled simplicial neural networks for trajectory prediction. In *International Conference on Machine Learning*, pages 9020–9029. PMLR.

Rolnick, D., Veit, A., Belongie, S., and Shavit, N. (2017). Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*.

Roy, O. (2009). Intentions and interactive transformations of decision problems. *Synthese*, 169(2):335–349.

Santos, F. C., Lenaerts, T., Pereira, L. M., et al. (2015). Synergy between intention recognition and commitments in cooperation dilemmas. *Scientific reports*, 5(1):1–7.

Santos, F. C., Pacheco, J. M., and Lenaerts, T. (2006). Evolutionary dynamics of social dilemmas in structured heterogeneous populations. *Proc. Natl. Acad. Sci. U.S.A.*, 103:3490–3494.

Sert, E., Bar-Yam, Y., and Morales, A. J. (2020). Segregation dynamics with reinforcement learning and agent based modeling. *Scientific reports*, 10(1):11771.

Shoham, Y. and Leyton-Brown, K. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

Shojaie, A. and Fox, E. B. (2022). Granger causality: A review and recent advances. *Annual Review of Statistics and Its Application*, 9:289–319.

Sigmund, K. (2010). *The Calculus of Selfishness*. Princeton U. Press.

Silverman, E., Gostoli, U., Picascia, S., Almagor, J., McCann, M., Shaw, R., and Angione, C. (2021). Situating agent-based modelling in population health research. *Emerging Themes in Epidemiology*, 18(1):1–15.

Sivakumar, N., Mura, C., and Peirce, S. M. (2022). Combining machine learning and agent-based modeling to study biomedical systems. *arXiv preprint arXiv:2206.01092*.

Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. (2022). Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Sukthankar, G., Geib, C., Bui, H., Pynadath, D., and Goldman, R. P. (2014). *Plan, activity, and intent recognition: Theory and practice*. Newnes.

Tank, A., Covert, I., Foti, N., Shojaie, A., and Fox, E. B. (2021). Neural granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4267–4279.

Tomasello, M. (2008). *Origins of Human Communication*. MIT Press.

Trivers, R. L. (1971). The evolution of reciprocal altruism. *Quaterly Review of Biology*, 46:35–57.

Vallès-Català, T., Peixoto, T. P., Sales-Pardo, M., and Guimerà, R. (2018). Consistencies and inconsistencies between model selection and link prediction in networks. *Physical Review E*, 97(6):062316.

van Hees, M. and Roy, O. (2008). Intentions and plans in decision and game theory. In *Reasons and intentions*, pages 207–226. Ashgate Publishers.

Wu, S., Sanghavi, S., and Dimakis, A. G. (2019). Sparse logistic regression learns all discrete pairwise graphical models. *Advances in Neural Information Processing Systems*, 32.

Yang, R., Sala, F., and Bogdan, P. (2022). Efficient representation learning for higher-order data with simplicial complexes. In *Learning on Graphs Conference*, pages 13–1. PMLR.

Yeo, K. (2019). Short note on the behavior of recurrent neural network for noisy dynamical system. *arXiv preprint arXiv:1904.05158*.

Yin, Z. and Barucca, P. (2022). Deep recurrent modelling of granger causality with latent confounding. *Expert Systems with Applications*, 207:118036.

Zhu, X. and Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *The Artificial Intelligence Review*, 22(3):177.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2):301–320.