History Data-Driven Distributed Consensus in Networks

Venkatraman Renganathan * **Angela Fontan** ** **Karthik Ganapathy** ***

* *Department of Automatic Control - LTH, Lund University, Lund, Sweden (e-mail: venkatraman.renganathan@control.lth.se).*
** *Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden (e-mail: angfon@kth.se)*
*** *Department of Mechanical Engineering, The University of Texas at Dallas, Richardson, TX, USA (e-mail: karthik.ganapathy@utdallas.edu)*

Abstract: The association of weights in a distributed consensus protocol quantify the trust that an agent has on its neighbors. An important problem in such networked systems is the uncertainty in the association of trust to neighboring agents, coupled with the losses arising from mistakenly associating wrong amount of trusts to different neighboring agents. We introduce a probabilistic way of determining the trustworthiness of an agent through the historical data collected by agents. Specifically, we identify the reliability of neighboring agents using the finite history of their shared data to arrive at a neighbors configuration that represents the confidence estimate of every neighboring agent's trustworthiness. Finally, we propose a History-Data-Driven (HDD) distributed consensus protocol which translates the computed configuration data into weights to be used in the consensus update. The probabilistic approach using the historical data in the context of a distributed consensus setting marks the novel contribution of our paper.

Keywords: History, Memory, Data-driven, Distributed Consensus, Networked System.

1. INTRODUCTION

In this paper we want to study the problem of consensus in a multiagent system in the presence of untrustworthy agents. Many cooperative tasks involving networked agents require them to utilize distributed consensus protocols to coordinate agreement on certain quantities of interest, with applications such as formation control in robotics (Fax and Murray (2004); Ren and Beard (2008)), agreement seeking in opinion dynamics ([notes to myself: add cite!]), or cyber-networks comprising of many interconnected smart entities which relies on distributed consensus protocols for efficient operations (Renganathan et al. (2021)). However, as shown for instance in Renganathan et al. (2021) [notes to myself: add cite in opinion dynamics?], the distributed nature of networks opens up many attack points for malicious attackers rendering them vulnerable. This work considers the situation where well-behaving agents (called "cooperative" in our notation) in a network seek to achieve consensus in the presence of "untrustworthy" agents (called "potentially noncooperative" in our notation) whose identity need not be known a priori.

A related problem is that of resilient consensus, which has been largely studied in the literature, see e.g., Sundaram and Hadjicostis (2008); Pasqualetti et al. (2012); LeBlanc et al. (2013); Saldaña et al. (2017); Dibaji et al. (2018), and resilient consensus protocols, such as the W-MSR protocol by LeBlanc et al. (2013), have been developed in the recent past to guarantee resiliency through some intelligently constructed nonlinear consensus update. The interested readers are referred to an non-exhaustive list papers such as Lamport et al. (2019) and Agmon and Peleg (2004). In general, the update rule of these distributed consensus protocols depend upon the current time step information obtained from all the agents in the network. An exception is the protocol for resilient consensus proposed in Saldaña et al. (2017), named SW-MSR, which extends the classical W-MSR algorithm by introducing a sliding window approach that allows the agents to store the values received from their neighbors at the previous T time steps. While in this literature robustness or connectivity assumptions are imposed on the graph describing the network of agents or on the total number of noncooperative (also called nonreliable or malicious) agents, in the recent work Yemini et al. (2021) the authors use the notion of trust in order to maintain consensus in a networked system in the presence of malicious agents.

Departing from the aforementioned literature, our aim is to design a distributed consensus protocol that enables each agent to estimate the trustworthiness of its neighbors, represented by a (normalized) non-negative value in $[0, 1)$ where 0 indicates that the corresponding agents do not trust each other and a value in $(0, 1)$ indicates the *amount* of trust between the corresponding agents, with the idea

⋆ This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under grant agreement No 834142 (Scalable Control).

that an agreement should be reached only between agents that trust each other. Similar to Yemini et al. (2021), in this work we do not impose any structural/connectivity assumption on the network or any assumption on the total number of potentially non-cooperative agents, but we consider the observed history at the previous T time steps to estimate trust between each agent in the network. The reason is that the traditional paradigm involving memory-less update in a distributed consensus protocol does not offer the debugging capabilities of getting to know *when and where* an intentional attack or a fault happened in the network. That is, distributed consensus algorithms lack the retrospecting ability to analyze the network for anomalies. On the other hand, it is not practical for every agent in a network to have infinite book-keeping abilities to store the shared values of its neighbors information to analyze for any anomalies. However, given a finite memory resource is made available for agents in a network, distributed consensus algorithms can be reinforced with retrospecting abilities to enhance the quality of the decisions that they make and mimic the trust-based decision-making behavior of humans in a distributed setting ([notes to myself: add cite on trust-based human decision-making behavior?]). One of the pioneering papers closely aligned with this research idea appeared in Yu and Vorobeychik (2019b,a), where malicious nodes were identified in an uncertain network with high confidence and removed. We consider extending a similar idea as theirs for designing a distributed consensus protocol using an history data-driven approach and, in particular, we assume that each collaborative agent has a finite memory, meaning that it can store the states of its neighbors only for the past T time steps. In fact, such an inference driven process has inherent complications when the true nature of the neighboring agents are not known exactly to an agent as mistaken trust associations can happen. Such an uncertain phenomenon is closely associated to the consensus problems in networks with random weighting matrices studied in Tahbaz-Salehi and Jadbabaie (2008, 2006). However, we use the available finite historical data to estimate the first two moments of an unknown distribution governing the true nature (called "configuration" in our notation) of an agent's neighbors.

The protocol we propose is inspired [notes to myself: change the word inspired?] by bounded confidence models in opinion dynamics, such as the Hegselmann-Krause model (Hegselmann and Krause (2002)), where each agent updates its state (which, in this context, represents its opinion) based only on the states of agents that are within a certain confidence range of its own, enforcing the idea that only trustworthy agents (here intended as agents with similar opinions) can influence each other. To make the protocol more realistic, we assume that each agent can only observe the states of its neighbors. Moreover, inspired by Lorenz (2009) and Morarescu and Girard (2011), respectively, we assume that the confidence bounds are heterogeneous (i.e., agent-dependent) and decreasing with time. At each time step, each collaborative agent receives the states of its neighbors and, by observing the historical data collected for each of its neighbors, infers [notes to myself: estimates, computes] the trustworthiness of each neighbor. Finally, it updates its state based on its neighbors' states, weighted by the estimated corresponding values of trust.

Statement Of Contributions: To the best of our knowledge, we propose the first probability theory based history data-driven distributed consensus protocol for networks. Specifically, our main contributions are as follows:

(1) We model the true nature of neighboring agents of an agent in a network as a random vector (which we term as the *configuration* of neighbors) and we learn the parameters governing its true but unknown distribution from the collected historical data.
(2) We translate the trustworthiness that resulted from the neighbor configuration into weights and propose a new History-Data-Driven (HDD) distributed consensus protocol for networks.
(3) We demonstrate using numerical simulation to show that our proposed design effectively models the neighbor configuration from their historical data and arrives at a trust-based consensus.

The rest of the paper is organized as follows. The preliminaries of the consensus protocol and the definition of a neighbor configuration are established in section 2. In section 3, the empirical estimation of the configuration parameters from the past historical data is discussed. The proposed HDD distributed consensus protocol is presented in section 4 along with its proposed variations. Our proposed algorithm is then demonstrated in section 5. Finally, the paper is closed in section 6 with a summary and research directions for the future.

NOTATION & PRELIMINARIES

We note the set of real numbers, integers, non-negative real numbers and non-negative integers by $\mathbb{R}, \mathbb{Z}, \mathbb{R}_{\geq 0}, \mathbb{Z}_{\geq 0}$ respectively. The operator $\setminus$ denotes the set subtraction. The cardinality of the set M is denoted by $|M|$ and its i^{th} element is denoted by $\{M\}_i$. The i^{th} element of a vector x is denoted by $[x]_i$ or simply x_i and the Euclidean norm of x is denoted by $\|x\|_2$ or simply $\|x\|$. A vector in $\mathbb{R}^n$ with all its elements being ones is denoted by $\mathbf{1}_n$. The j^{th} column of a matrix A is denoted by A_j. The element present in i^{th} row and j^{th} column of matrix A is denoted by A_{ij}. An uniform distribution between $a, b \in \mathbb{R}, a < b$ is denoted by $U[a, b]$.

2. PROBLEM FORMULATION

2.1 Consensus Dynamics of a Network

Consider a network having N agents whose connectivity is modeled via an undirected and connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of agents with $|\mathcal{V}| = N$. A set of time-invariant communication links amongst the agents is represented using $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. We associate with each agent $i \in \mathcal{V}$, a state $x_i(k) \in \mathbb{R}$ at time $t \in \mathbb{Z}_{\geq 0}$. Let the set of *inclusive neighbors* be defined as $\mathcal{J}_i = \mathcal{N}_i \cup \{i\}$, where $\mathcal{N}_i = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$ is the neighbor set of agent i, whose states are available to agent i via communication links. The degree of i is denoted as $d_i = |\mathcal{N}_i|$, and every agent is assumed to have access to its own state at any time t. At any time t, each agent updates its own state based on its current state and the states of its neighboring agents according to a prescribed memory-less update rule of the form

$$x_i(t+1) = f_i(x_j(t)),\ j \in \mathcal{J}_i,\ i \in \mathcal{V}. \tag{1}$$

Typical distributed consensus protocols of the form (1) involve associating a weight corresponding to every inclusive neighbors $j \in \mathcal{J}_i$ and using it in the consensus update. In this research, we consider the weighted averaging type of update protocols, namely $x(t+1) = W(t)x(t)$ with $x(t) = [x_1(t) \dots x_N(t)]^T$, that is

$$x_i(t+1) = \sum_{j \in \mathcal{J}_i} w_{ij}(t)x_j(t), \quad i \in \mathcal{V}, \tag{2}$$

where $W(t)$ is an element-wise non-negative weighting matrix with its entries $w_{ij}(t)$ modeling the trustworthiness associated by agent i on its inclusive neighboring agents $j \in \mathcal{J}_i$. Olshevsky and Tsitsiklis (2009) showed that, under the assumptions that the graph $\mathcal{G}$ is strongly connected and the weights are chosen according to a convex combination ($W(t)$ being a doubly-stochastic matrix), an asymptotic consensus value is guaranteed by (2).

Definition 1. An agent $i \in \mathcal{V}$ is said to be cooperative if it updates its state based on (2). It is said to be (potentially) non-cooperative [1], otherwise.

Definition 2. An agent $i \in \mathcal{V}$ is said to be in a trust-based consensus with the set of trusted neighbors $j \in \overline{\mathcal{N}}_i \subseteq \mathcal{N}_i$ if $\forall \epsilon > 0, \exists T_c > 0$ such that $\|x_i(k) - x_j(k)\| < \epsilon, \forall k \geq T_c$.

2.2 Availability of Historical Data

The above definition of trust-based consensus not only allows for clustering behaviour between subset of network agents but also allows for a total consensus between all agents in a network. Given such a setting where some agents *might* be non-cooperative, when a memory-less distributed protocol like (2) with any convex combination type weighting is used by the cooperative agents $i \in \mathcal{V}$, the resulting asymptotic consensus can be manipulated by some smart non-cooperative agents. Under this setting, one of the best available mechanisms for an agent with uncertain nature of its neighbors is to observe their shared data for a certain period of time and arrive at a confidence bounds for their trustworthiness and subsequently use it in their update to arrive at a consensus. To facilitate a tractable problem formulation and to make the resulting consensus algorithm suitable for dynamic implementation, we consider a finite historical data of length $T \in \mathbb{N}$ updated in a rolling horizon fashion [2]. At all time steps t, every agent $i \in \mathcal{V}$ has access to the history of its own values and to its neighboring agents' values $x_j(t), j \in \mathcal{N}_i$ for the past T time-steps. That is, with $\kappa_{t,T} = \{t-l\}_{l=0}^{T-1}$, we have

$$\mathfrak{X}_{i,j}^T = \{x_j(k) \mid j \in \mathcal{N}_i,\, k \in \kappa_{t,T}\}, \tag{3}$$

$$\mathfrak{X}_{i,i}^T = \{x_i(k) \mid k \in \kappa_{t,T}\}, \tag{4}$$

$$\mathfrak{X}_{i,\mathcal{N}_i}^T = \left\{\mathfrak{X}_{i,j}^T \mid j \in \mathcal{N}_i\right\}. \tag{5}$$

[1] For instance, an agent $i \in \mathcal{V}$ can act *non-cooperatively* by applying random update function $f_i'(\cdot)$ other than (2) at all time-steps.

[2] Future work will seek to understand the behaviour with growing history over time.

Thus, the main purpose of this work is *(i) to design a protocol that allows each cooperative agent $i \in \mathcal{V}$ to estimate the trustworthiness of its neighbors, given the history $\mathfrak{X}_{i,\mathcal{N}_i}^T$ and $\mathfrak{X}_{i,i}^T$ for the past T time steps; (ii) to study the role of the estimated trustworthiness in solving the trust-based consensus problem for the cooperative agents, despite the presence of non-cooperative agents in the network.*

3. SET MEMBERSHIP BASED EMPIRICAL ESTIMATION OF CONFIGURATION

In this section, we describe how to use data-driven techniques to probabilistically learn the reliability of neighbors, given the T time step historical data.

3.1 Learning the Neighbor Configuration

The *configuration* of neighbors $\mathcal{N}_i$ at time t denoted by $\pi_t^i \in [0,1]^{d_i}$ encodes the degree of trustworthiness of every neighbor $j \in \mathcal{N}_i$. A neighbor $j \in \mathcal{N}_i$ is said to be completely trustworthy or not trustworthy if $\left[\pi_t^i\right]_j$ equals to 1 or 0, respectively, and any value in $[0,1]$ defines its degree of trustworthiness. Similarly, we define $\bar{\pi}_t^i = \mathbf{1}_{d_i} - \pi_t^i$ to be the configuration that represents the degree of non-cooperativeness of the neighbors at time t. Note that π_t^i is a random vector where the j^{th} entry of π_t^i corresponding to the neighbor $j \in \mathcal{N}_i$ is supported on a compact interval $[0,1]$. Further, $\pi_t^i \sim P_t^i$ with P_t^i denoting the true but *unknown* distribution of the π_t^i supported on a compact set $[0,1]^{d_i}$. Let $\mu_t^i \in \mathbb{R}^{d_i}$ and $\Sigma_t^i \in \mathbb{R}^{d_i \times d_i}$ denote the true mean and covariance respectively associated with P_t^i. Though in reality P_t^i is not readily available, it can be estimated from data. That is, using the T time step history data, it is possible to form an empirical distribution $\hat{P}_t^i$. Let us denote the mean and the covariance of $\hat{P}_t^i$ by $\hat{\mu}_t^i$ and $\hat{\Sigma}_t^i$, respectively. Here, $\left[\hat{\mu}_t^i\right]_j$ is agent i's estimated trustworthiness at time t about the neighboring agent $j \in \mathcal{N}_i$ given its past T time step historical data.

We propose a set membership based approach to estimate the parameters of the empirical configuration distribution $\hat{P}_t^i$ given the historical data $\mathfrak{X}_{i,i}^T$ and $\mathfrak{X}_{i,\mathcal{N}_i}^T$. We base the following discussion on the presumption that a neighbor $j \in \mathcal{N}_i$ is believed to be more trustworthy by agent i, if j's values were in the desired vicinity of agent i's value either throughout the considered past or at least preferably in the recent past. Such a presumption is reasonable, since the main focus of this section is about reasonable estimation of the parameters of $\hat{P}_t^i$.

3.2 The ϵ-Neighborhood Based Set Membership

To define a set membership based estimation, we require a set of confidence neighborhoods for all the past T time steps. Thus, for all $k \in \kappa_{t,T}$, the confidence neighborhood around the $x_i(k)$ is defined as,

$$\mathcal{B}_{x_i(k)}(\epsilon_{i,k}) = \{y \in \mathbb{R} \mid \|y - x_i(k)\|_2 \leq \epsilon_{i,k}\}, \tag{6}$$

where $\epsilon_{i,k} > 0$ is an confidence bound defined by agent i at time k. To value the recent past more than the distant past and to enforce consensus at the time t, agent $i \in \mathcal{V}$ is free to choose a decreasing sequence of confidence bounds $\epsilon_{i,k}$, $\forall k \in \kappa_{t,T}$ as follows,

$$\epsilon_{i,t-(T-1)} > \cdots > \epsilon_{i,t-2} > \epsilon_{i,t-1} > \epsilon_{i,t} > 0. \quad (7)$$

Using the confidence neighborhoods $\mathcal{B}_{x_i(k)}(\epsilon_{i,k})$, and the information sets $\mathfrak{X}^T_{i,i}, \mathfrak{X}^T_{i,\mathcal{N}_i}$, we define the set membership counter for all time steps $k \in \kappa_{t,T}$ as follows,

$$\mathfrak{N}^i_k = \left\{ j \in \mathcal{N}_i \mid x_j(k) \in \mathcal{B}_{x_i(k)}(\epsilon_{i,k}) \right\} \quad k \in \kappa_{t,T}. \quad (8)$$

Here, $\mathfrak{N}^i_k \subseteq \mathcal{N}_i$ accounts for the neighbors $j \in \mathcal{N}_i$ who share their values in the vicinity of the agent i in each time steps of the past history. The following lemma emphasizes the fact that if the network reaches a consensus in the cooperative agents, then the set membership counter for every agent will be a maximal set.

[Comment: in the paper Morarescu and Girard (2011), the authors show that the HK model with homogeneous confidence bounds which are decaying in time converge to an eq. point and in the Section II.A and III.B they basically prove the following lemma (see Proposition 3). The proof of the following lemma could be written like that.]

Lemma 1. Consider a strongly connected network $\mathcal{G}(\mathcal{V}, \mathcal{E})$, with the set of agents $\mathcal{V}$ and their connections denoted by the set of edges $\mathcal{E}$. Let $\mathcal{B}_{x_i(k)}(\epsilon_{i,k})$ denote the confidence neighborhood of agent $i \in \mathcal{V}$ at time k defined using the confidence bounds satisfying (7). If a trust-based consensus is reached between all cooperative agents in $\mathcal{G}$, then $\forall \epsilon_{i,k} > 0, \exists T_c \in \mathbb{R}_+$ $(T_c < +\infty)$ such that, $\forall t \geq T_c$ and for all cooperative agents $i \in \mathcal{V}$, $\mathfrak{N}^i_t$ is equal to $\mathcal{N}_i$ minus all potentially non-cooperative neighbors in $\mathcal{N}_i$.

Proof. The proof is based on the fact that the consensus value between all cooperative agents will lie in the intersection of their confidence neighborhoods for $t \geq T_c$. Assume that the network is in consensus, that is, $\forall \varepsilon > 0$ there exist $L_c \in \mathbb{R}, T_c \in \mathbb{R}_+$ for all cooperative agents $i \in \mathcal{V}$ such that

$$|x_i(t) - L_c| < \varepsilon, \quad \forall t \geq T_c. \quad (9)$$

Given any confidence bounds $\epsilon_{i,k} > 0$ for $k \in \kappa_{t,T}$, $t \geq T_c + T - 1$, and $i \in \mathcal{V}$, suppose by contradiction that there exists a $k \in \kappa_{t,T}$ and a cooperative agent i such that $\mathfrak{N}^i_k \subset \mathcal{N}^c_i$, where $\mathcal{N}^c_i$ is the set of neighbors of i minus all potentially non-cooperative agents in $\mathcal{N}_i$. This means that that there exists a neighbor $j \in \mathcal{N}^c_i \setminus \mathfrak{N}^i_k$ such that $|x_j(k) - x_i(k)| > \epsilon_{i,k}$. However, with a "small enough" $\varepsilon > 0$, such as $\varepsilon < \min\limits_t \frac{\epsilon_{i,t}}{2}$ [See Review]), we obtain that $|x_j(k) - x_i(k)| < 2\varepsilon < \epsilon_{i,k}$ (where the first inequality follows from (9)), which leads to a contradiction.

3.3 Estimation of Configuration Distribution Parameters

Define a frequency counter $\mathcal{C}^i_j$ that records the time indices k, where the neighbor $j \in \mathcal{N}_i$ had its value $x_j(k) \in \mathcal{B}_{x_i(k)}(\epsilon_{i,k})$. For all $j \in \mathcal{N}_i$ with $k \in \kappa_{t,T}$,

$$\mathcal{C}^i_j = \left\{ k \in \kappa_{t,T} \mid j \in \mathfrak{N}^i_k \right\}. \quad (10)$$

We define the discounted importance vector $\mathfrak{d}^i_j \in \mathbb{R}^T$ that qualitatively captures how a neighboring agent $j \in \mathcal{N}_i$ behaved with respect to the agent i, by valuing the recent past more than the distant past using a discount factor $\nu_{i,t} \in (0,1)$ as

$$\left[\mathfrak{d}^i_j\right]_k = \begin{cases} \nu^{t-k}_{i,t}, & \text{if } k \in \mathcal{C}^i_j, \\ 0, & \text{if } k \notin \mathcal{C}^i_j. \end{cases} \quad (11)$$

Finally, the estimated mean $\hat{\mu}^i_t$ and estimated covariance $\hat{\Sigma}^i_t$ at time t are computed as

$$\left[\hat{\mu}^i_t\right]_j = \frac{1}{T} \sum_{k \in \kappa_{t,T}} \left[\mathfrak{d}^i_j\right]_k, \quad j \in \mathcal{N}_i, \quad (12)$$

$$\hat{\Sigma}^i_t = \frac{1}{T-1} \sum_{k \in \kappa_{t,T}} \left(\mathcal{D}^i_k - \hat{\mu}^i_t\right) \left(\mathcal{D}^i_k - \hat{\mu}^i_t\right)^\top, \quad (13)$$

where the normalized distance matrix $\mathcal{D}^i \in \mathbb{R}^{d_i \times T}$ is given by

$$\mathcal{D}^i_{jk} = \left\| \frac{x_i(k) - x_j(k)}{\|x_i(k) - \bar{x}^i(k)\|} \right\|, \quad \forall k \in \kappa_{t,T} \quad (14)$$

with $\bar{x}^i(k) = \frac{1}{|\mathcal{J}_i|} \sum_{j \in \mathcal{J}_i} x_j(k)$ being the average of all the inclusive neighbors at each time step k. The estimated [3] trustworthy configuration is then $\hat{\pi}^i_t = \hat{\mu}^i_t$, which implies that the estimated non-cooperative configuration is then $\hat{\bar{\pi}}^i_t = \mathbf{1}_{d_i} - \hat{\pi}^i_t$. In the next section, we elucidate how an agent $i \in \mathcal{V}$ can use the inferred trustworthiness of all its neighboring agents $j \in \mathcal{N}_i$ derived from the estimated parameter $\hat{\mu}^i_t$.

4. DESIGN OF AN HISTORICAL DATA-DRIVEN DISTRIBUTED CONSENSUS PROTOCOL

In this section, we use the obtained trustworthiness information of the neighbors of an agent to design a distributed consensus protocol.

4.1 An HDD Distributed Consensus Protocol

For every reliable agent $i \in \mathcal{V}$, $\hat{\mu}^i_t$ denotes the estimated trustworthiness of its neighbors given their past historical data. Further, every reliable agent i has to trust itself completely at all time steps. Therefore, at each time step t, we form the augmented trust vector $z^\dagger_i(t) \in \mathbb{R}^{|\mathcal{J}_i|}$ as

$$z^\dagger_i(t) = \begin{bmatrix} \hat{\mu}^i_t \\ 1 \end{bmatrix}. \quad (15)$$

Then, every reliable agent $i \in \mathcal{V}$ updates its states using the following proposed History-Data-Driven (HDD) distributed consensus protocol, namely

$$x_i(t+1) = \sum_{j \in \mathcal{J}_i} \underbrace{\frac{[z^\dagger_i(t)]_j}{\left\| z^\dagger_i(t) \right\|_1}}_{:= w_{ij}(t)} x_j(t) \quad (16)$$

The weights $w_{ij}(t)$ given by (16) translates the trustworthiness information of neighboring agents into weights for the distributed consensus update rule.

[3] This style of inferring the trustworthiness has the potential of being vulnerable with smarter adversaries as they can manipulate the estimated parameters $\hat{\mu}^i_t$ and $\hat{\Sigma}^i_t$ to render them off from their respective true values μ^i_t and Σ^i_t. Future research will seek to address this using the distributionally robust stochastic program (DRSP) model described in described in Delage and Ye (2010).

Remark 1. The HDD protocol is actually a nonlinear consensus update, such as the W-MSR protocol, as the weight $w_{ij}(t)$ computed by agent i for its neighbor $j \in \mathcal{N}_i$ with an informed choice of the parameters $T, \epsilon_{i,k}$ and $\nu_{i,t}$ may turn out to be zero based on the inference using the historical data, meaning that at time t, agent i neglects neighbor j's contribution.

4.2 Effects of Parameter Variations

The design parameters of our algorithm are $T \in \mathbb{R}_+$ (T finite), $\{\nu_{i,t}\}_{i\in\mathcal{V},t\in\mathbb{R}_+}$, and $\{\epsilon_{i,k}\}_{i\in\mathcal{V},k\in\kappa_{t,T}}$. The parameter $\nu_{i,t} \in (0,1)$ can be regarded as the forgetting factor for an agent $i \in \mathcal{V}$ at time t and thus influences how much an agent is willing to remember its neighbors' past interactions from the time t given the history length T (defined respecting the available memory constraints). For instance, $\nu_{i,t}$ closer to 1 indicates that the agent emphasizes its recent past interactions with its neighbors more and hence its forgetfulness decreases rather slowly over time. On the other hand, $\nu_{i,t}$ closer to 0 indicates that the agent forgets quickly. The next parameter confidence bound $\epsilon_{i,k}$ at time k is the agent i's freedom to choose its desired vicinity area around its values to values its neighbors appropriately. A cautious agent $i \in \mathcal{V}$ will tend to have a small $\epsilon_{i,k}$ even at its distant past, and similarly a relaxed agent may tend to choose a generous $\epsilon_{i,k}, \forall k \in \kappa_{t,T}$. Given that $\mathfrak{N}^i_k$ for an agent $i \in \mathcal{V}$ directly depends upon the $\{\epsilon_{i,k}\}_{k\in\kappa_{t,T}}$ and which in turn defines the $\hat{\mu}^i_t$, it is clear that the resulting consensus is directly impacted by the choice of the confidence bound that an agent chooses according to its behavioural aspects. Though the efficacy of the proposed algorithm is limited upon the memory constraint defining the parameter T, its freedom in the design of other design parameters makes it both an interesting and powerful algorithm.

Remark 2. The collected historical data can be used to predict the neighbors value using machine learning techniques and if the neighbor shares a value closer to the predicted value, agent i can allocate an higher trust to value their contribution more and thereby define *data-driven predictive consensus* protocol. Another variation to HDD protocol would be to remove neighbors whose trustworthiness fall below a specified trust-threshold and subsequently using only the remaining neighbors values. Future work will seek to investigate adaptive designs of discount factor $\nu_{i,t}$ and confidence bounds $\epsilon_{i,k}, \forall k \in \kappa_{t,T}$.

5. A NUMERICAL EXAMPLE

In this section, we elaborate the simulation results that we performed to demonstrate our proposed HDD distributed consensus protocol. We considered a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ with $|\mathcal{V}| = N = 13$ agents connected in such a way that the edge set $\mathcal{E}$ resulted in a strongly connected graph. Further, the first 10 agents were assumed to be cooperative and are denoted by $N_c = \{1,\ldots,10\}$ and the rest $N_{nc} = N - N_c$ being non-cooperative. The cooperative nodes $i \in N_c$ were randomly connected with probability $p = 0.4$ (here, p denotes the probability of an edge between two nodes), while each non-cooperative node $i \in N_{nc}$ was connected to all cooperative nodes. To start with, all agents were given a random history of data generated for a considered history of length $T = 10$ time steps. The HDD protocol was demonstrated for a total of $T_t = 200$ time steps with the new state trajectories of neighboring agents being used to update the historical data in a rolling horizon fashion. Each agent $i \in \mathcal{V}$ was given a random confidence bounds for all $k \in \kappa_{t,T}$ where each confidence bound $\epsilon_{i,k}$ was drawn from $U[\underline{\epsilon},\overline{\epsilon}]$. While the lower limit was constantly set to $\underline{\epsilon} = 0.01$, the upper limit $\overline{\epsilon}$ was varied from 0.5 to 1.5 to observe different behaviours. Further, every agent $i \in \mathcal{V}$ was given the same discount factor $\nu_{i,t} \in (0,1)$ at a given time instant t for the sake of simplicity, although $\nu_{i,k}$ varied along the history $k \in \kappa_{t,T}$. For consistency, we generated random initial states for all agents at time $t = 0$ and fixed it for all the following simulations, where the HDD protocol was executed by varying one of the following parameters $T, \nu_{i,t}, \epsilon_{i,k} \forall k \in \kappa_{t,T}$ and $\forall i \in \mathcal{V}$ while keeping the rest fixed.

The results of our simulation are shown in Figure 1. On all the sub-figures of Figure 1, the discount factor variations with $\nu_{i,t} = \{0.05, 0.50, 0.95\}$ are shown. With low values of the discount factor, we observed clustering behaviour between agents and with higher values of discount factor, the normal consensus convergence is observed. This is due to the fact that higher $\nu_{i,t}$ enabled the agents to remember the past interactions of its neighbors to a greater extent. The effect of varying the confidence bounds are shown in sub-figures 1b, 1a and 1c where the decreasing sequence of confidence bounds were sampled from $U[\underline{\epsilon},\overline{\epsilon}]$ with $\underline{\epsilon} = 0.01$ and $\overline{\epsilon} = \{0.5, 1.0, 1.5\}$ respectively. Higher values of confidence bounds encouraged the network to arrive at a consensus faster, while lower values of confidence bounds resulted in slower convergence rate and as a result, sometimes clustering behaviour between agents emerged.

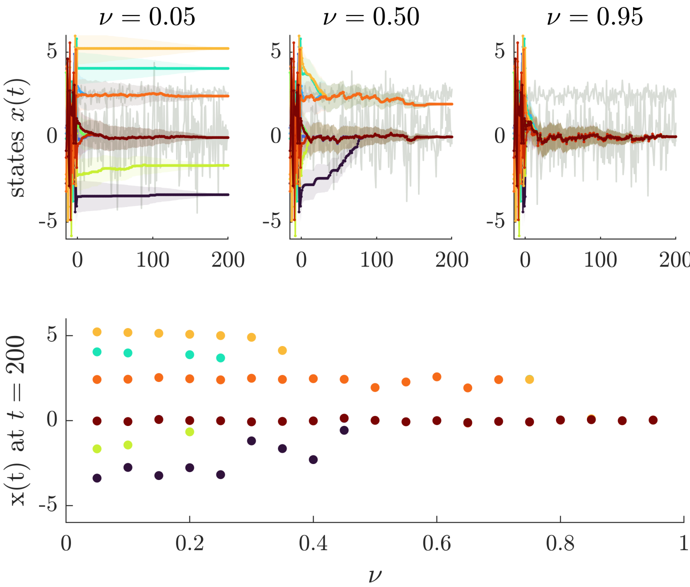
6. CONCLUSION & FUTURE OUTLOOKS

We proposed a new historical data-driven distributed consensus for uncertain networks. Our proposed approach formulates the uncertainty about the trustworthiness of neighbors of an agent as a random vector termed as neighbor configuration and learns the parameters defining its unknown but true distribution via history data-driven approach. Subsequently, the trustworthiness of all neighbors of an agent is inferred leading to the proposed HDD distributed consensus protocol. Our simulation results demonstrated the effectiveness of our proposed idea. As a future work, we seek to investigate the moment uncertainty along with the losses due to mistakenly associating wrong trust to neighbors given their historical data using distributionally robust optimization techniques. Other promising directions are to design a data-driven resilient consensus algorithm and an history data-driven predictive consensus algorithm as detailed in the subsection **??**.

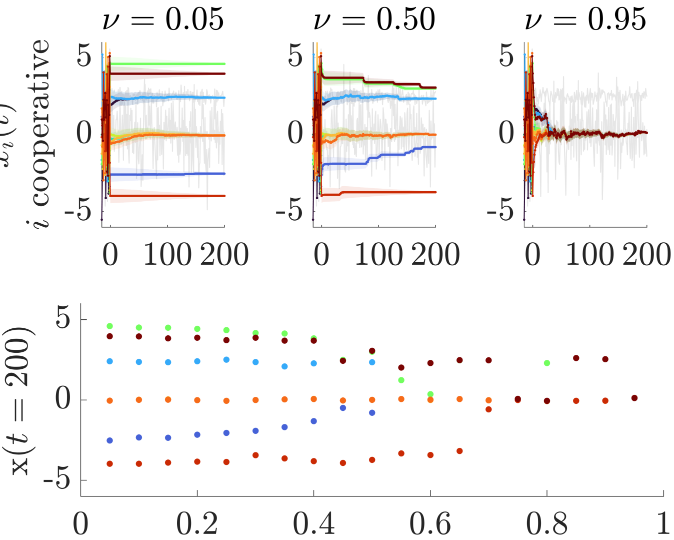
REFERENCES

Agmon, N. and Peleg, D. (2004). Fault-tolerant gathering algorithms for autonomous mobile robots. 1070–1078.

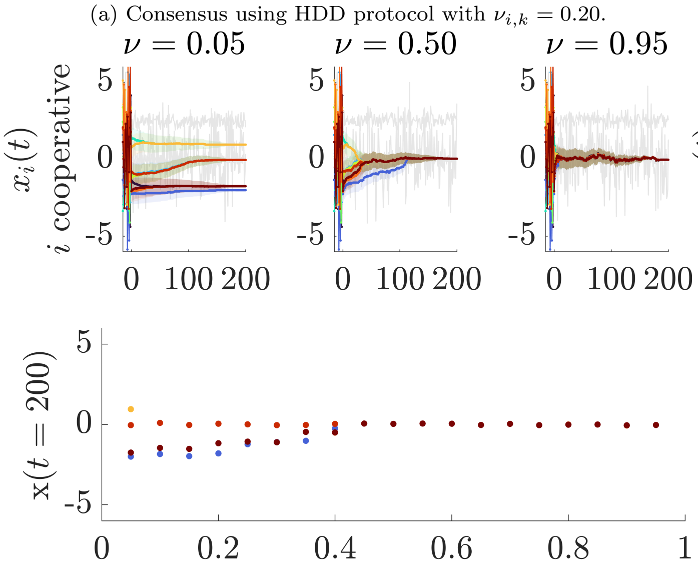
Delage, E. and Ye, Y. (2010). Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3), 595–612.



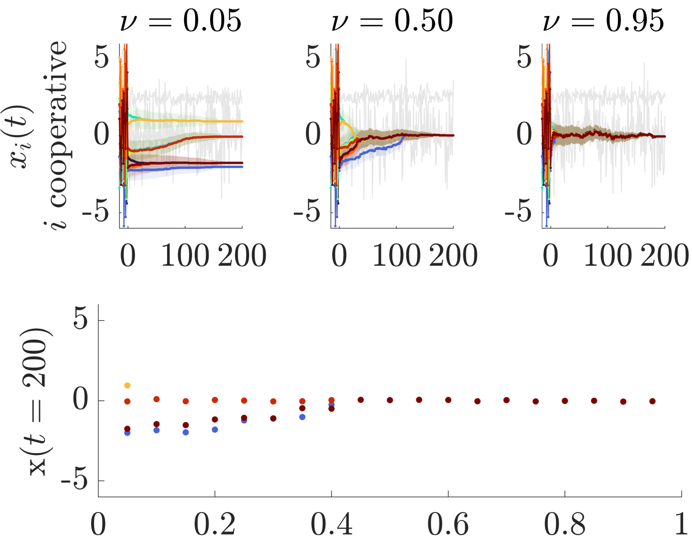
(a) Consensus using HDD protocol with $\nu_{i,k} = 0.20$.



(b) Consensus using HDD protocol with $\nu_{i,k} = 0.95$.



(c) Consensus using HDD protocol with $\epsilon_{i,k} \sim U(0, 5)$.



(d) Consensus using HDD protocol with $\epsilon_{i,k} \sim U(0, 20)$.

Fig. 1. The effects of varying the parameters $\nu_{i,k}, \epsilon_{i,k}$ for $k \in \kappa_{t,T}, i \in \mathcal{V}$ and T defining the HDD protocol are shown here. The color map indicates cooperative agents and non-cooperative agents with random state updates are shown in grey color. For all the sub-figure (a), (b), (c) and (d), the top plot shows the evolution of agents' states $x_i(t)$, $i \in \mathcal{V}$ (with $i \in N_c$ being cooperative and $i \in N_m$ being non-cooperative) for $\nu = 0.05, 0.5, 0.95$, with the decreasing sequence of confidence bounds $\{\epsilon_{i,t}\}$ represented by shaded areas (top) drawn from $U[0, 0.5]$; and the bottom plot shows the states of cooperative agents $i \in N_c$ at time $t = 200$ for increasing values of $\nu \in \{0.05, 0.1, \ldots, 0.95\}$.

Dibaji, S.M., Ishii, H., and Tempo, R. (2018). Resilient randomized quantized consensus. *IEEE Transactions on Automatic Control*, 63(8), 2508–2522.

Fax, J. and Murray, R. (2004). Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9), 1465–1476.

Hegselmann, R. and Krause, U. (2002). Opinion Dynamics and Bounded Confidence Models, Analysis and Simulation. *Journal of Artificial Societies and Social Simulation*, 5(3), 1–2.

Lamport, L., Shostak, R., and Pease, M. (2019). The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, 203–226.

LeBlanc, H.J., Zhang, H., Koutsoukos, X., and Sundaram, S. (2013). Resilient asymptotic consensus in robust networks. *IEEE Journal on Selected Areas in Communications*, 31(4), 766–781.

Lorenz, J. (2009). Heterogeneous bounds of confidence: Meet, discuss and find consensus! *Complexity*, 15(4), 43–52.

Morarescu, I.C. and Girard, A. (2011). Opinion dynamics with decaying confidence: Application to community detection in graphs. *IEEE Transactions on Automatic Control*, 56(8), 1862–1873.

Olshevsky, A. and Tsitsiklis, J.N. (2009). Convergence speed in distributed consensus and averaging. *SIAM journal on control and optimization*, 48(1), 33–55.

Pasqualetti, F., Bicchi, A., and Bullo, F. (2012). Consensus computation in unreliable networks: A system theoretic approach. *IEEE Transactions on Automatic*

Control, 57(1), 90–104.

Ren, W. and Beard, R.W. (2008). *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer, London.

Renganathan, V., Fathian, K., Safaoui, S., and Summers, T. (2021). Spoof resilient coordination in distributed and robust robotic networks. *IEEE Transactions on Control Systems Technology*, 1–8. doi: 10.1109/TCST.2021.3063924.

Saldaña, D., Prorok, A., Sundaram, S., Campos, M.F.M., and Kumar, V. (2017). Resilient consensus for time-varying networks of dynamic agents. In *2017 American Control Conference (ACC)*, 252–258.

Sundaram, S. and Hadjicostis, C.N. (2008). Distributed function calculation via linear iterations in the presence of malicious agents — part ii: Overcoming malicious behavior. In *2008 American Control Conference*, 1356–1361.

Tahbaz-Salehi, A. and Jadbabaie, A. (2006). On consensus over random networks. In *44th Annual Allerton Conference*. Citeseer.

Tahbaz-Salehi, A. and Jadbabaie, A. (2008). A necessary and sufficient condition for consensus over random networks. *IEEE Transactions on Automatic Control*, 53(3), 791–795.

Yemini, M., Nedic, A., Goldsmith, A.J., and Gil, S. (2021). Characterizing trust and resilience in distributed consensus for cyberphysical systems. *IEEE Transactions on Robotics*, 1–21.

Yu, S. and Vorobeychik, Y. (2019a). Distributionally robust removal of malicious nodes from networks. *arXiv preprint arXiv:1901.11463*.

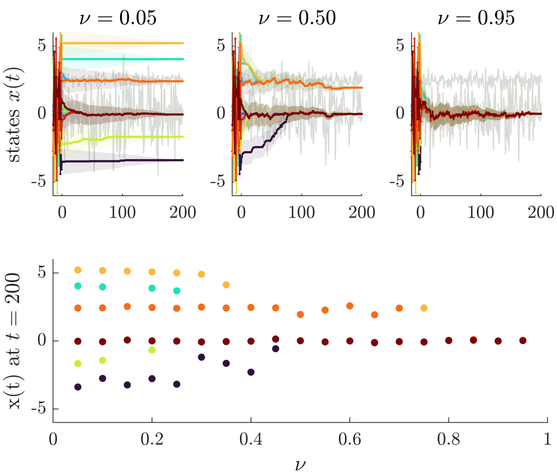
Yu, S. and Vorobeychik, Y. (2019b). Removing malicious nodes from networks. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 314–322.

Appendix A. NAME

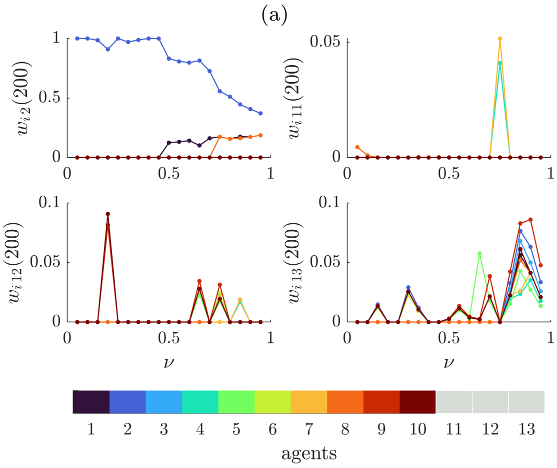
Example 1. This example aims at illustrating the role of the parameters ν (forgetting factor) in the HDD protocol.

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $N = 13$ nodes. Assume that nodes 1 to 10 are cooperative, while nodes $11, 12, 13$ are non-cooperative; moreover, assume that nodes 1 to 10 are randomly connected with probability $p = 0.4$ (i.e., the probability of an edge between two nodes in $\{1, \ldots, 10\}$ is p), while each non-cooperative node $(11, 12, 13)$ is connected to all nodes in $\{1, \ldots, 10\}$. The design parameters of our algorithm are chosen as follows: history of length $T = 10$; homogeneous (time-invariant) forgetting factors $\nu_{i,t} = \nu$ for all $i \in \mathcal{V}$ and $t \geq 0$; homogeneous (time-decreasing) confidence bounds $\epsilon_{i,t} = \epsilon_t$ for all $i \in \mathcal{V}$ where $\{\epsilon_t\}_{t \geq 0}$ are drawn from an uniform distribution in the interval $[0.01, 1]$ and subsequently arranged in a decreasing order.

To observe the effect of the forgetting factor ν in the dynamics, we consider increasing values of ν in $\in \{0.05, 0.1, \ldots, 0.95\}$. As shown in Fig. A.1a, when the forgetting factor is small, we can observe a "clustering" behavior: this is due to the facts that the confidence bounds ϵ_t are small and that the HDD protocol values only the most recent past. For instance, for $\nu < 0.1$ $\nu^{t-k} \approx 0$ for $k = t - T + 1, \ldots, t - 2$, that is, the agents "forget quickly". Instead, when the forgetting factor is close to 1, the states tend to converge to consensus, do to the ability of the agents to remember past events (such as remembering agent j in its ϵ_k-neighborhood for a certain time $k \in \{t - T + 1, \ldots, t\}$). Fig. A.1b shows the elements of the j^{th} $(j = 2, 11, 12, 13)$ column of $W(t)$ when $t = 200$, for increasing values of ν. Remember that each element w_{ij} $(i \in \mathcal{V}, j = 2, 11, 12, 13)$ represents the trust that each agent i has on its neighbor j; then, the intuition is that if there exists a value of ν such that $w_{ij}(200) = 0$ for all $i \in \mathcal{V}$ and $j \in \{11, 12, 13\}$, it means that the cooperative agents correctly decide to not trust the non-cooperative agents. In this case, Fig. A.1b shows that the non-cooperative agents $11, 12$ are "detected" for most values of ν. One could think that $\nu \approx 1$ corresponds to an optimal choice; this is however not the case if a non-cooperative agent adopts a smart behavior (see agent 13 and bottom-right panel of Fig. A.1b), obtaining the trust of all the other agents. Finally, it is interesting to notice that in order to achieve consensus, an agent needs to trust its neighbors's states and lower the certainty in its own state (see the element $w_{22}(200)$ depicted in blue in the top-left panel of Fig. A.1b).

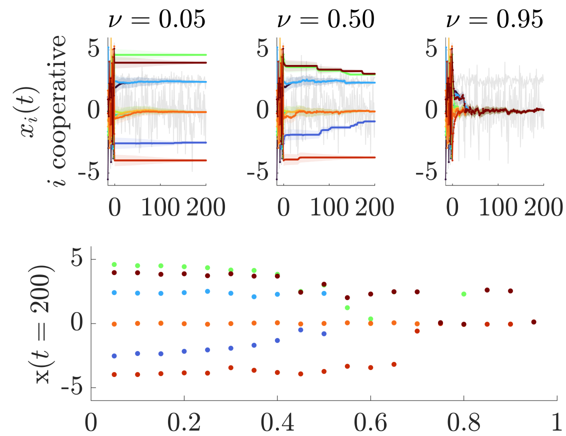


(a)

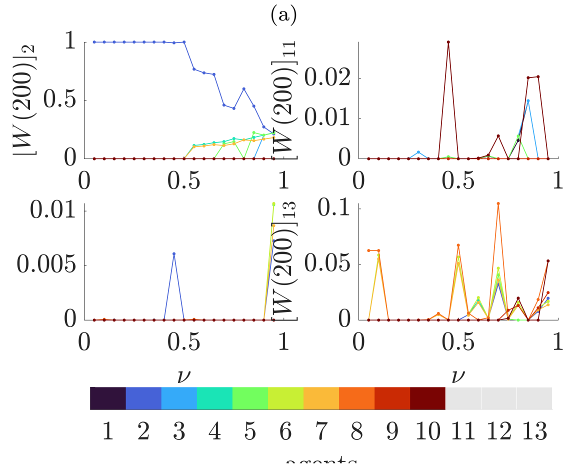


(b)

Fig. A.1. Example 1. (a): Evolution of agents' states $x_i(t)$, $i = 1, \ldots, 13$ for $\nu = 0.05, 0.5, 0.95$, with the confidence bounds $\{\epsilon_{i,t}\}$ represented by shaded areas (top) drawn from $U[0, 1]$; states $x_i(t)$ of cooperative agents $i = 1, \ldots, 10$ at time $t = 200$ for increasing values of $\nu \in \{0.05, 0.1, \ldots, 0.95\}$ (bottom). (b): Elements $w_{ij}(t)$, for $i = 1, \ldots, 10$ (i.e, cooperative agents) and $j = 2, 11, 12, 13$ (top-left top-right, bottom-left, bottom-right, respectively) at time $t = 200$, for increasing values of $\nu \in \{0.05, 0.1, \ldots, 0.95\}$. Legend: Cooperative agents are indicated using the colormap above, while non-cooperative agents are represented in grey.

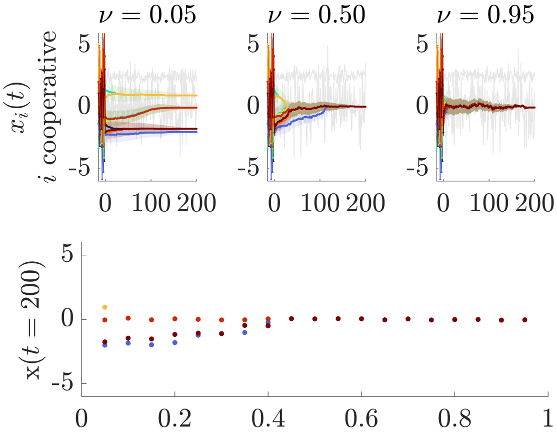


(a)

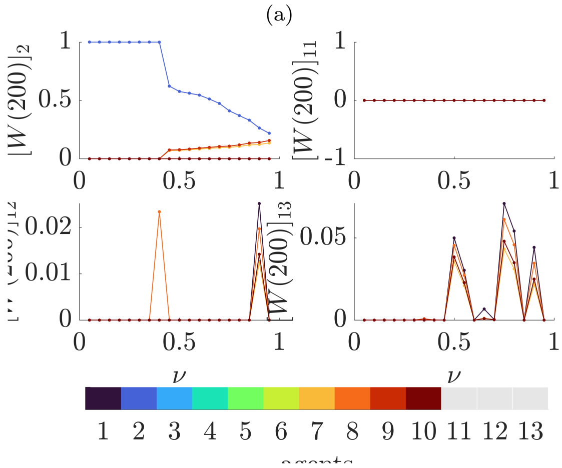


(b)

Fig. A.2. Example 1. (a): Evolution of agents' states $x_i(t)$, $i = 1, \ldots, 13$ for $\nu = 0.05, 0.5, 0.95$, with the confidence bounds $\{\epsilon_{i,t}\}$ represented by shaded areas (top) drawn from $U[0, 0.5]$; states $x_i(t)$ of cooperative agents $i = 1, \ldots, 10$ at time $t = 200$ for increasing values of $\nu \in \{0.05, 0.1, \ldots, 0.95\}$ (bottom). (b): Columns 2 (top-left), 11 (top-right), 12 (bottom-left), 13 (bottom-right) of $W(t)$ at time $t = 200$ for increasing values of $\nu \in \{0.05, 0.1, \ldots, 0.95\}$. Legend: Cooperative agents are indicated using the colormap above, while non-cooperative agents are represented in grey.



(a)



(b)

Fig. A.3. Example 1. (a): Evolution of agents' states $x_i(t)$, $i = 1, \ldots, 13$ for $\nu = 0.05, 0.5, 0.95$, with the confidence bounds $\{\epsilon_{i,t}\}$ represented by shaded areas (top) drawn from $U[0, 1.5]$; states $x_i(t)$ of cooperative agents $i = 1, \ldots, 10$ at time $t = 200$ for increasing values of $\nu \in \{0.05, 0.1, \ldots, 0.95\}$ (bottom). (b): Columns 2 (top-left), 11 (top-right), 12 (bottom-left), 13 (bottom-right) of $W(t)$ at time $t = 200$ for increasing values of $\nu \in \{0.05, 0.1, \ldots, 0.95\}$. Legend: Cooperative agents are indicated using the colormap above, while non-cooperative agents are represented in grey.