# Decentralised Data Markets

Aida M. Manzano Kharman, Christian Jursitzky, Quan Zhao, Pietro Ferraro, Robert Shorten, Pierre Pinson

CONTENTS

## I. Introduction

### A. Preamble

In recent years there has been a shift in many industries towards data-driven business models [50]. Namely, with the advancement of the field of data analytics, and the increased ease in which data can be collected, it is now possible to use both these disruptive trends to develop insights in various situations, and to monetise these insights for financial gain. Traditionally, users have made collected data available to large platform providers, in exchange for services (for example, web browsing). However, the fairness and even ethics of these business modes continue to be questioned, with more and more stake-holders arguing that such platforms should recompense citizens in a more direct manner for data that they control [52] [32], [3], [4]. Apple has recently responded to such calls by introducing changes to their ecosystem to enable users to retain ownership of data collected on their devices. At the time of writing, it was recently reported that these new privacy changes have caused the profits of Meta, Snap, Twitter and Pinterest plummet (losing a combined value of $278 billion since the update went into effect in late April 2021[1]). The privacy change introduced by Apple allows users to mandate apps not to track their data for targeted advertising. This small change has been received well amongst Apple users, with a reported 62% of users opting out of the tracking [5]. Clearly this change will have a profound impact on companies relying on selling targeted advertisements to the users of their products. Users can now decide how much data they wish to provide to these platforms and they seem keen to retain data ownership. It seems reasonable to expect that in the future, companies wishing to continue to harvest data from Apple will need to incentivise, in some manners, users or apps to make their data available.

The need for new ownership models to give users sovereignty over data is motivated by two principal concerns. The first one concerns fair recompense to the data harvester by data-driven businesses. While it is true that users receive value from companies, in the form of the services their platforms provide (e.g., Google Maps), it is not obvious that this is a fair exchange of value. The second one arises from the potential for unethical behaviours that are inherent to the currently prevailing business models. Scenarios in which unethical behaviour have emerged arising out of poor data-ownership models are well documented. Examples of these include Google Project Nightingale [2] [3], where sensitive medical data was collected of patients that could not opt out of having their data stored in Google Cloud servers. The scale of this project was the largest of its kind, with millions of patient records collected for processing health care data. Another infamous case study was the Cambridge Analytica (CA) scandal that was released in 2015. Personal data of 87 million users was acquired via 270,000 user giving access to a third party app that gave access to the users' friend network, without these people having explicitly given access to CA to collect such data [4] [30]. CA has a vast portfolio of elections they have worked to influence, with the most notorious one being the 2016 US presidential elections that Donald Trump won [58], [39].

It is important to understand that these cases are not anecdotal. Without the adequate infrastructure to trade ownership, cases like the ones outlined above, with mass for example, privacy breaches, having the potential to become more frequent. Apple's actions are an important step in the direction of giving individuals ownership over their data and potentially alleviating such issues, however, one may correctly ask why users should trust Apple, or any other centralised authority, to preserve their privacy and not trade with their data. Motivated by this background, and by this latter question, we argue for the shift towards a more decentralised data ownership model, where this ownership can be publicly verified and audited. We are interested in developing a data-market design that is hybrid in nature; *hybrid* in the sense that some non-critical components of the market are provided by trusted infrastructure, but where the essential components of the market place, governing ownership, trust, data veracity, etc., are all designed in a decentralised manner. The design of such markets is not new and there have been numerous attempts to design marketplaces to enable the exchange of data for money [51]. This, however, is an extremely challenging endeavour. Data cannot be treated like a conventional commodity due to certain properties it possesses. It is easily replicable; its value is time-dependant and intrinsically combinatorial; and dependent on who has access to the data set. It is also difficult for companies to know the value of the data set a priori, and verifying

[1] https://www.bloomberg.com/news/articles/2022-02-03/meta-set-for-200-billion-wipeout-among-worst-in-market-history

[2] https://www.bbc.co.uk/news/technology-50388464

[3] https://www.theguardian.com/technology/2019/nov/12/google-medical-data-project-nightingale-secret-transfer-us-health-information

[4] https://www.nytimes.com/2018/04/04/technology/mark-zuckerberg-testify-congress.html

its authenticity is challenging [2] [8]. These properties make marketplace models difficult to design and have been an emergent research area.

In what follows, a first step in the design of a marketplace where data can be exchanged is described. Furthermore, this marketplace provides certain guarantees to the buyer and seller alike. More specifically, the goal is to rigorously define and address the challenges related to the tasks of selling and buying data from unknown parties, whilst compensating the sellers fairly for their own data. As mentioned, to prevent monopolisation, a partially decentralised setting will be considered, focusing on the important case of data rich environments, where collected data is readily available and not highly sensitive. Accordingly, this work focuses on a specific use case from the automotive industry that, we hope, might represent a first step towards more general architectures.

### *B. Specific motivation*

We focus on a class of problems where there is an oversupply of data but where there is a lack of adequate ownership methods to underpin a market. One example of such a situation is where agents collaborate as part of coalitions in a crowd sourced environment to make data available to potential buyers. It is applications of this nature that will be the focus in this paper. More specifically, the interest is placed in the context of a city where drivers of vehicles wish to monetise the data harvested from their car's sensors. An architecture is proposed that enables vehicle owners to sell their data in coalitions to buyers interested in purchasing their data.

While this situation is certainly a simplified example of a scenario in which there is a need for data market, it is of great interest for two reasons. Firstly, situations of this nature prevail in many application domains. Scenarios where metrics of interest can be aggregated to generate a data rich image of the state of a given environment are of value to a wide range of stakeholders, which, in the given context, could include anyone from vehicle manufacturers, mobility and transport companies to city councils. Secondly, this situation, while simplifying several aspects of the data-market design, captures many pertinent aspects of more general data-market design: for example, detection of fake data; certification of data-quality; and resistance to adversarial attacks.

The context of automotive data collection is a ripe opportunity to develop a decentralised data market. The past decade has seen traditional vehicles transition from being a purely mechanical device to a cyber-physical one, having both a physical and digital identity. From a practical viewpoint, vehicles are quickly increasing their sensing capabilities, especially given the development of autonomous driving research. Already, there is an excess of useful data collected by the latest generation vehicles, and this data is of high value. According to [37] *"car data and shared mobility could add up to more than $ 1.5 trillion by 2030"*. Such conditions prevail not only in the automotive sector; for example, devices such as smart-phones; smart watches; modern vehicles; electric vehicles; e-bikes and scooters; as well as a host of other IoT devices, are capable of sensing many quantities that are of interest to a diverse range of stakeholders. In each of these applications the need for such marketplaces is already emerging. Companies such a Nissan, Tesla, PSA and others have already invested in demonstration pilots in this direction and are already developing legal frameworks to develop such applications[5] in anticipation of opportunities that may emerge. As mentioned, the main issue in the design of such a data market lies in the lack of an adequate ownership method. Who owns the data generated by the vehicle? The answer is unclear. A study by the Harvard Journal of Law & Technology concludes that most likely, it is the company that manufactured the car who owns the data, even though the consumer owns the smart car itself [62]. According to the authors of the study, this is because the definition of ownership of data is not congruent to other existing definitions of ownerships such as intellectual property (IP), and therefore the closest proxy to owning a data set is having the rights to access, limit access to, use, and destroy data. Most importantly consumers do not have the right to economically exploit the data they produce. Nonetheless, EU GDPR laws expressly state that users will be able to transfer their car data to a third party should they so wish. According to [53] "The data portability principle was expressly created to encourage competition". However, if the data is owned by the automobile company, how can consumers verify who has access to their data? Placing trust assumptions on the car manufacturer should be rigorously justified before such marketplaces emerge. Given the lack of verifiability of a centralised authority, such as a car manufacturing company, we propose exploring decentralised or hybrid alternatives.

The objective in this paper is directly motivated by such situations and by the issues described so far. However, as previously discussed, rather than enabling manufacturers to monetize this data, we are interested in situations where device owners, or coalitions of device owners, own the data collected by their devices, and wish to make this data available for recompense. This is fundamentally different to the situation that prevails today, whereby users make their data freely available to platform providers such as Google, in exchange for

[5]https://www.aidataanalytics.network/data-monetization/articles/tesla-automaker-or-data-company

using their platform. Nevertheless, the recent actions of Apple suggest that this new inverted (and emancipated) business model, whereby providers compete and pay for data of interest, could emerge as an alternative model of data management, and also whereby users are able to control and manage the data that they reveal. Given this background context, we are interested in developing a platform whereby data owners can make available, and securely transfer ownership of data streams, to other participants in the market.

### C. Related Work

*1) Decentralised vs Centralised Data Markets:* Numerous works have proposed decentralised data markets. While many of these proposals use Blockchain architectures for their implementations, many simply utilise the underlying technology and fail to address Blockchain design flaws as they pertain to data markets [42] [29] [57]. For example, Proof-of-Work (PoW) based Blockchains reward miners with the most computational power. Aside from the widely discussed issue of energy wastage, the PoW mechanism is itself an opportunity, hitherto that has not been utilised, for a data-market to generate work that can be useful for the operation of the marketplace. As we shall shortly see, the PoW mechanism can itself be adapted to generate work that is useful for the operation of the marketplace. In addition, Blockchain based systems, also typically use commission based rewards to guide the interaction between users of the network, and Blockchain miners. Such a miner-user interaction mechanism is not suitable in the context of data-markets, effectively prioritising wealthier users' access to the data-market. In addition, miners with greater computational power are more likely to earn the right to append a block, and thus earn the commission. This reward can then be invested in more computational power, leading to a positive feedback loop where more powerful miners become more and more likely to write blocks and earn more commissions. Similarly, the wealthier agents are the ones more likely to receive service for transactions of higher monetary value. This could cause traditional PoW-based Blockchains to centralise over time [9]. Indeed, centralised solutions to data markets already exist, such as [10], which namely focus on implementing methods to share and copy data, and certain rights to it, such as read rights. Considering the aforementioned properties of PoW-based Blockchains, the authors explore other distributed ledger architectures to implement a decentralised data market. [citation-needed]

*2) Trust and Honesty Assumptions:* Another possible categorisation of prior work relates to the trust assumptions made in the system. The work in [43] assumes that upon being shared, the data is reported truthfully and fully. In practise, this assumption rarely holds, and a mitigation for malicious behaviour in shared systems must be considered. This assumption is justified in their work by relying on a third party auditor, which the authors of [57] also utilise. However, introducing an auditor simply shifts the trust assumption to their honest behaviour and forgoes decentralisation.

In [2], it is identified that the buyer may not be honest in their valuation of data. They propose an algorithmic solution that prices data by observing the gain in prediction accuracy that it yields to the buyer. However, this comes at the cost of privacy for the buyer: they must reveal their predictive task. In practise, many companies would not reveal this IP, especially when it is the core of their business model. The work of [38] is an example of a publicly verifiable decentralised market. Their system allows for its users to audit transactions without compromising privacy. Unfortunately, their ledger is designed for the transaction of a finite asset: creating or destroying the asset will fail to pass the auditing checks. For the transaction of money this is appropriate: it should not be possible to create or destroy wealth in the ledger (aside from public issuance and withdrawal transactions). However, for data this does not hold. Users should be able to honestly create assets by acquiring and declaring new data sets they wish to sell. Furthermore, their cryptographic scheme is built to transfer ownership of a single value through Pedersen commitments.

There is a need to have trust assumptions in components of the data market, whether it be centralised or decentralised. However, the authors believe that the users of the data market should agree on what or who to trust. A consensus mechanism is a means for a group of agents to agree on a certain outcome. For users to trust the consensus mechanism, they must have a series of provable guarantees that it was executed correctly. Users should be able to audit and verify the functioning of the consensus mechanism. It is not sufficient for the consensus mechanism to function correctly, it should also prove this to the users. Indeed, it can be observed how, in binding governmental elections, voters have means to verify that the election was honest, and if this is not the case, voters have grounds to contest the outcome of the election.

The authors advocate for placing the trust assumptions in consensus mechanisms that can be verified. In other words, the users of a data market should have a means to agree on what they trust, and they should have a means to verify that this agreement was reached in a correct, honest manner.

In fact, this verification should be decentralised and public. Shifting the trust to a third-party auditing mechanism to carry out the verification can lead to a recursion problem, where one could continuously question why a third, fourth, fifth and so on auditing

party should be trusted, until these can generate a public and verifiable proof of honest behaviour.

*3) Consensus Mechanisms:* Consensus in the context of a distributed mechanism can be mapped to the fault-tolerant state-machine replication problem [44]. The users in the network must come to an agreement as to what is the accepted state of the network. Furthermore, it is unknown which of these users are either faulty or malicious. This scenario is defined as a Byzantine environment, and the consensus mechanism used to address this issue must be Byzantine Fault Tolerant (BFT) [18].

In permissionless networks, probabilistic Byzantine consensus is achieved through the means of certain cryptographic primitives [60]. Commonly this is done by solving a computationally expensive puzzle. This method, known as proof of work (PoW), although commonly used, suffers of a series of drawbacks, earlier described in section I-C1. Aside from its poor scalability and mining races, this method incurs vast energy consumption costs. To put this into perspective, Bitcoin uses more than seven times as much electricity as all of Google's global operations.[6]. By far the biggest share of this energy is used to solve the computational puzzles in its PoW algorithm.

In permissioned networks consensus is reached amongst a smaller subset of users in the network. This is done through BFT consensus mechanisms such as Practical BFT (PBFT) [17] and PAXOS [16]. Often the permissioned users are elected according to how much stake in the network they hold, following a proof of stake (PoS) method. This centralisation enables a higher throughput of transactions at the cost of higher messaging overhead, but ensures immediate consensus finality. They also require precise knowledge of the users' membership [24]. Meanwhile, in permissionless consensus protocols the guarantee of consensus is only probabilistic but does not require high node synchronicity or precise node memberships, and are more robust [59] [60].

When considering consensus mechanisms for permissionless distributed ledgers, there exist a wide range of consensus mechanisms that are a hybrid combination of either PoS and PoW (eg: Snow White), or PoW-BFT (eg: PeerCensus) or PoS-BFT (eg: Tendermint). Each consensus mechanism places greater importance in achieving different properties. For example, Tendermint focuses on deterministic, secure consensus with accountability guarantees and fast throughput [15]. Snow White is a provably secure consensus mechanism that uses a reconfigurable PoS committee [11], and PeerCensus enables strong consistency in Bitcoin transactions, as opposed to eventual consistency [22].

There also exists a class of probabilistic consensus mechanisms, such as FPC [41], Optimal Algorithms for Byzantine Agreements [23], Randomised Byzantine Agreements [56] and Algorand [28]. The authors find this class of consensus mechanisms of particular interest for the context of a data market. Namely, the fact that they are probabilistic makes coercion of agents difficult for a malicious actor. To ensure malicious actors are selected by the consensus algorithm, they must know a priori the randomness used in the mechanism, or coerce a supra-majority of agents in the network. Furthermore, the authors argue that selecting users in a pseudo-random way treats users equally, and is closer to achieving fairness than selecting users with the greatest wealth or greatest computational power. Another consideration of fairness is made in [21], where the mechanism does not rely on a correct leader to terminate, therefore decentralising the mechanism.

In some examples described above, such as in [28], agents with greater wealth in the ledger are more likely to be selected to form part of the voting committee. However, the authors believe that for the context of the work here presented, voting power should be earned and not bought. Indeed, this right should be earned irrespective of wealth or computational power. This opens the question of, how then, should this power be allocated? The authors believe the market should trust the actors that behave honestly and further the correct functioning of the market. Agents should prove their honesty and only then earn the right to be trusted. A collective goal for the data market can be defined, and agents who contribute to this should be adequately rewarded. This goal can be characterised mathematically, and each agent's marginal contribution can be evaluated. In this manner, rights and rewards can be granted proportionally.

Algorand desire to retain the voting power amongst the agents with the most stake is based on the assumption that the more stake an agent has in the system, the more incentive they have to behave honestly. This assumption cannot be made in our data market. Owning more cars does not equate to being more trustworthy, and therefore should not increase an agent's voting power, or their chances of participating in decision making. In fact, owning more vehicles could be an incentive to misbehave in the data market and upload fake data, whether this be to mislead competitors or to force a favourable outcome for themselves as a malicious actor. Purposely reporting fake data in the context of mobility has been described in [45] and [55], where collectives reported fake high congestion levels to divert traffic from their neighbourhoods [7] This attack is known as *data poisoning* and is a known attack of crowd-sourced applications, usually mounted through

[6]https://www.nytimes.com/interactive/2021/09/03/climate/bitcoin-carbon-footprint-electricity.html

[7]https://www.washingtonpost.com/local/traffic-weary-homeowners-and-waze-are-at-war-again-guess-whos-winning/2016/06/05/c466df46-299d-11e6-b989-4e5479715b54_story.html

a Sybil attack. Owning more vehicles makes mounting a data poisoning attack easier. Therefore in this context, having more stake in the market cannot be assumed to be an incentive to behave honestly, and indeed can offer increased ease to behave maliciously.

Furthermore, the work in [28] uses majority rule and their consensus mechanism only has two possible outcomes: accept the transaction or not (timeout or temporary consensus). In the context of the data market, this would not suffice. The consensus mechanism in the work here presented is used to determine which agents are the most trusted to compute the average or median of the data collected of a certain location. In other words, the consensus mechanism is a means to delegate a computation to someone based on how much they are trusted. Agents may be more, or less trusted, with some being preferred over others. These preferences may be stronger or weaker too. Using a majority voting method that only yields two possible options fails to encapsulate this information and is known to exclude minorities. The disadvantages of majority rule systems such as First Past the Post voting are known of and extensively documented [33], [12], [20]. Voting systems that do not achieve proportional representation and retain power within a wealthy minority are not an appropriate consensus mechanism for a context where the authors aim for decentralisation and fairness.

### D. Structure of the Paper

Firstly the authors introduce a series of desirable properties that the market must satisfy. These are outlined in the design criteria section II. Subsequently, a high level overview of the working components of the data market are presented in section IV, as well as describing how each functional component contributes to achieving the desired properties described in the preceding section.
Then the authors proceed to formalising definitions used in each component of the data market, as well as the assumptions made in V. This section describes in detail how each component of the data market works.
Finally, in section VII, the authors describe the set of attacks considered, and in section VIII the robustness of the components of the data market are evaluated.

## II. Design Criteria for the Data Market

Having discussed issues that pertain to and arise from poor data ownership models, the authors present a series of desirable criteria that the data market should achieve. More specifically, the work here proposed, begins to address the following research questions that are associated with data market designs:

How to protect the market against fake data or faulty sensors?
Given an oversupply of data, how to ensure that everybody receives a fair amount of write access to the market?
How to enable verifiable exchange of data ownership?
How to select data points from all those available to add most value to the marketplace?
How to protect the marketplace against adversarial attacks?

Following directly from these open questions, the desirable criteria are defined as:

- *Decentralised Decision Making:* The elements of the marketplace pertaining to trust, ownership and veracity are decentralised and do not rely on placing trust on third parties.

- *Verifiable centralisation:* The infrastructure on which the data market relies that is centralised, can be publicly verified. The reader should note that this ensures trust assumptions are placed on components of the data market that are publicly verifiable.

- *Generalised Fairness:* Access to the data market is governed by the notion of the potential value that a data stream brings to the market (as defined by a given application). This will determine which agents will get priority in monetising their data. Agents with equally valuable data must be treated the same and agents with data of no value should receive no reward. Further notions of fairness are considered and formalised by [48] under the defintion of Shapley Fairness, and described in V.7. These are the definitions of fairness that the authors use for this data market proposal.

- *Resistant to duplication of data:* The datamarket must not allow malicious attackers to earn reward by duplicating their own data. Precisely, a distinction must be made between preventing the monetisation of duplicated data, versus preventing data duplication, given that the latter is an open challenge.

- *Resistant to fake data and faulty sensors:* The data market must be resilient to data poisoning attacks, wherein adversaries collude to provide fake data to influence the network. Congruently, the datamarket must be resilient to poor quality data from honest actors with faulty sensors. Formally, the data for sale on the market must not deviate by more than a desired percent from the ground truth. For the purpose of this work, the ground truth is defined as data measured by centralised infrastructure. Given that this measurement is publicly verifiable (any agent in that location

can verify that this measurement is true), this is considered an acceptable assumption.

- *Resistant to spam attacks:* The data market is not susceptible to spam attacks, and malicious actors are not in the condition of flooding and congesting the network with fake or poor quality data.

## III. Preliminaries

The Architecture in Figure 1 makes reference to several technology components that may be new to the reader. To ease exposition these are now briefly described.

### A. Distributed Ledger Technology

A distributed ledger technology (DLT) will be used to store the transactions of wealth in exchange for access (or any other given right) to a dataset.

A DLT is a decentralized database of transactions where these transactions are timestamped and accessible to the members of the DLT. They are useful to allow agents to track ownership, and are decentralized. Compared to a centralised storage system, this provides a geographically distributed, consensus-based, and verifiable system which is immutable after data has been written and confirmed. It is also a more resilient alternative against failure points than a centralised infrastructure.

There are many types of DLT structures, but they all aim to provide a fast, reliable, and safe way of transferring value and data. Namely, DLTs strive to satisfy the following properties: have only one version of the ledger state, are scaleable, make double spending impossible, and have fast transaction times.

One example of a DLT is the IOTA Tangle, shown in figure 1. In this DLT, all participants contribute to approving transactions, and the transactions are low to zero fee and near-instant. Further, decentralisation is promoted through the alignment of incentives of all actors [46].

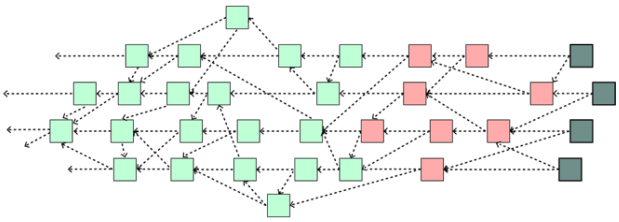

Figure 1. IOTA Tangle. credit: IOTA Foundation

### B. Access control mechanism

Because DLTs are decentralised, they need a method to regulate who can write information to the ledger and who cannot. An access control mechanism is necessary to protect the distributed ledger from spam attacks. One way is by using Proof-of-Work (PoW) as it is done in the Blockchain, where computationally intense puzzles need to be solved to be able to write to the ledger. In this case, users with more computational power earn the right to access the ledger. An alternative is Proof-of-Stake where nodes can stake tokens to gain the access rights proportional to their amount of staked tokens [25].

Another alternative is the Mana based system used in the IOTA Tangle which, in combination with PoW to make Denial-of-Service (DoS) attacks expensive. Consequently, the PoW in the Tangle tends to be computationally much less expensive than the one used in PoW-based Blockchains and thus more resource efficient [63].

### C. Consensus Mechanisms

A consensus mechanism is a means for a collective to come to an agreement on a given statement. In section I-C3 some examples of consensus mechanisms are discussed that are appropriate for Byzantine environments. Some of these utilise a voting mechanism to enable said consensus, and the authors now discuss an alternative voting mechanism that satisfies a different set of properties. It is important to note that there exist numerous methods of aggregating preferences, which are well studied in the field of social choice [47], and voting mechanisms provide different means to enable this aggregation [65].

The taxonomy of voting systems is diverse. They can be either be considered probabilistic or deterministic; proportional or plurality rule or ordinal as opposed to cardinal.

Depending on the set of practical constraints or preferred properties of the implementation context, the authors encourage selecting an appropriate voting mechanism that best satisfies the desired criteria for a given application. Subsequently, the authors discuss Maximum Entropy Voting, and why it has desirable properties for the context of this data market.

*1) Maximum Entropy Voting:* Within the classes of voting schemes, Maximum Entropy Voting (MEV) belongs to the family of probabilistic, proportional and ordinal systems. Arrow famously defined in [7] an impossibility theorem that applies to ordinal voting systems. In it, he states that no ordinal voting systems can satisfy all three of the following properties:

**Definition III.1** (Non-Dictatorial)**.** *There exists no single voter with power to determine the outcome of the voting scheme.*

**Definition III.2** (Independence of Irrelevant Alternatives)**.** *The output of the voting system for candidate A and candidate B should depend only on how the voters ordered*

*candidate A and candidate B, and not on how they ordered other candidates.*

**Definition III.3** (Pareto property)**.** *If all voters prefer candidate A to candidate B, then the voting system should output candidate A over candidate B.*

Whilst this impossibility theorem only applies to ordinal voting systems and not cardinal ones, it has been shown by Gibbard's theorem that every deterministic voting system (including the cardinal ones) is either dictatorial or susceptible to tactical voting [26].

Gibbard then later shows in [27] that, in the spectrum of the desirable properties that Arrow defines, the voting mechanism that achieves all three properties to the greatest degree is the Random Dictator (RD). With this in mind, the reader can now understand the novelty that the work in [35] presents. Here, MEV is presented as a probabilistic system that first, determines the set of voting outcomes that proportionally represent the electorate's preference, whilst selecting the outcome within this set that minimises surprise.

Lets proceed to elaborate: if one were to pick a voting system that is probabilistic and satisfies Arrow's properties to the greatest degree, the adequate system to choose would be RD. However, whilst computationally an inexpensive method to run, suffers from a series of drawbacks. The one of greatest concern for the context of this work is the following: imagine a ballot is sampled that happens to contain a vote for an extreme candidate (or in this case, for a malicious actor). The entire choices of an individual that votes for extremes now dictate the entire electorate's leaders. In this scenario, a malicious agent would likely only vote for equally malicious agents, although the number of malicious agents is still assumed to be a minority.

Could one reduce the amount of information taken from that sampled ballot? MEV proposes a way to sample ballots that while still representing the electorate's views, minimise the amount of information taken from their preferences. In essence, this is selecting a ballot that reflects the least surprising outcome for the electorate, whilst ensuring that it is still within the set of most representative choices.

Furthermore, MEV still satisfies relaxed versions of the Independence of Irrelevant Alternatives and Pareto properties, whilst not being dictatorial. It also enjoys the benefits of proportional voting schemes as well as being less susceptible to tactical voting [35]. As a result of this, it can be argued that it is difficult to predict the exact outcome of a vote and therefore it is secure against timed attacks because it is costly to have high confidence of success.

## IV. Architecture of the Data Market

Figure 2 depicts the functional components of the proposed data market. As can be observed in the figure, some components of the marketplace are decentralised, and some of the enabling infrastructure is provided by an external, possibly centralised, provider.

In a given city, it is assumed that a number of agents on vehicles are collecting data about the state of the location they are in. They wish to monetise this information, but first, it must be verified that they are real agents. They present a valid proof of their identity and location, as well as demonstrating that they information is timely and relevant.

The agents that successfully generate the aforementioned receive a validity token that allows to form spatial coalitions with other agents in their proximity. They then vote on a group of agents in their coalition that will be entrusted to calculate the agreed upon data of their corresponding location. The chosen agents do this by aggregating the coalition's data following a specified algorithm.

This procedure happens simultaneously in numerous locations of the city. At a given point in time, all the datasets that have been computed by a committee in a spatial coalition then enter the access control mechanism. One can consider the data a queue and the access control mechanism the server. Here, they are ranked in order of priority by determining which data provides the greatest increase in value to the data market. The coalitions with the most valuable data perform the least amount of work.

Coalitions wishing to sell their data must complete a useful proof of work that is inversely proportional to their added value to the market. This PoW entails calculating the added value of new data in the queue. Once this work is completed, the data can be sold. The buyers are allocated to the sellers through a given bidding mechanism $\mathrm{BM}$, and the reward of this sale is distributed amongst the sellers using the reward distribution function $\mathrm{RD}$. Successful transactions are recorded on the distributed ledger, and the data corresponding to the successful transactions are removed from the data market.

Next, the authors describe the high level functioning of each of the components shown in figure 2, and how each contribute do achieving the desired properties described in section II.

### A. Verification

Agents are verified by a centralised authority that ensures they provide a valid position, identity and dataset. This component ensures that spam attacks are

[8]In order of appearance: Icons made by Freepik, Pixel perfect, juicy_fish, srip, Talha Dogar and Triangle Squad from www.flaticon.com

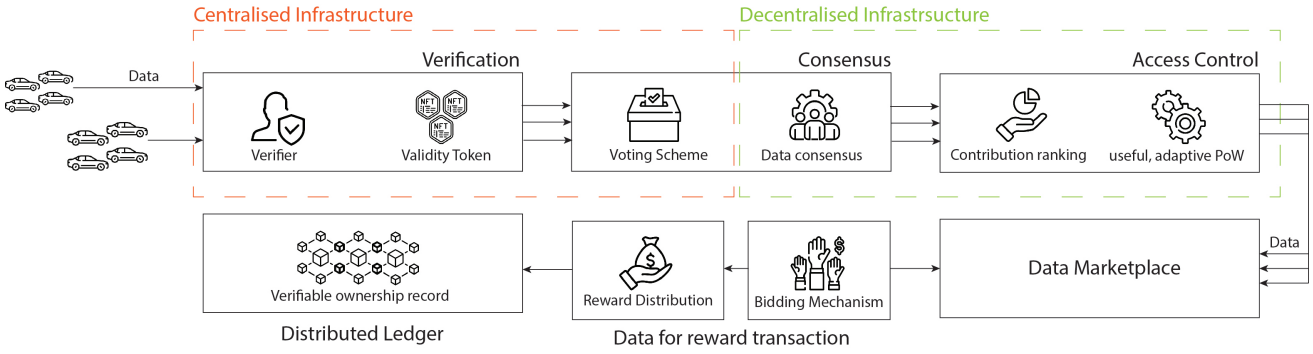


Figure 2. Data Market Architecture. Credit for the images is given in [8]

expensive, as well as enabling verifiable centralisation. All agents in the market can verify the validity of a proof of position and identity because this information is public.

### B. Consensus

*1) Voting Scheme:* In a decentralised environment, agents must agree on what data is worthy of being trusted and sold on the data market. Agents express their preferences for who they trust to compute the most accepted value of a data point in a given location. This is carried out through a voting scheme.

*2) Data Consensus:* Once a group of trusted agents is elected, they must then come to a consensus as to what the accepted data value of a location is. This is computed by the group following an algorithm that aggregates the coalition's data.

These components enable the property of decentralised decision making, allowing coalitions to govern themselves and dictate who to trust for the decision making process. Furthermore, they makes uploading fake data to the market costly, as malicious agents must coerce sufficient agents in the voting system, to ensure enough coerced actors will be elected to compute the value of a dataset that they wish to upload.

### C. Access Control

*1) Contribution Ranking:* Once datapoints are agreed upon for the given locations, it is necessary to regulate which ones should receive priority when being sold. The priority is given to the data that increases the combined worth of the data market. This can be measured by using the Shapley value, that in this case is used to measure the marginal contribution of dataset towards increasing the value of the market with respect to a given objective function. Precise formalisation is presented in definition V.7.
This component provides the property of generalised fairness of the market, wherein agents with more valuable data should do less work to sell their data.

*2) Useful Adaptive Proof of Work:* Coalitions must perform a proof of work that is proportional to how valuable to the market their data is deemed. Hence why the work performed is adaptive, and furthermore, it is useful to the functioning of the market because the work is in fact calculating the worth of the new incoming data into the market. This feature ensures that spam attacks are costly and that the market is resistant to duplication of profit by duplicating data. This is because for every dataset a coalition wishes to sell, they have to complete a PoW. If their dataset is duplicated, it will be deemed less valuable to the market and thus the malicious coalition will have to perform more work.

### D. Data Marketplace

Here is where the collected and agreed upon data of specific locations is posted to be sold. The datasets themselves are not public, but rather a metadata label of the dataset, who it is owned by (the spatial coalition that crowd-sourced it) and the location it is associated with. Sellers can access and browse the market and place bids for specific datasets in exchange for monetary compensation. Sellers may wish to purchase access to the entire dataset, to a specific insight or to other defined rights, such as the rights to re distribute or perform further analytics on said dataset. Each right has a corresponding price.

### E. Bidding Mechanism

The mechanism matches buyers to the sellers of data. Here, the price per the right purchased is determined. At this stage, a spatial coalition formed of multiple agents is considered to be one seller. Successful bids will be recorded in an immutable ownership record that is public, such that all participants of the market can see which agents have rightful access to a dataset.

### F. Reward Distribution

Once a bid is successful, then the reward of the sale is distributed amongst the participants of the spatial coalition that generated the sold dataset. This is a crucial aspect of the market to ensure that all agents participating in the crowd-sourcing of the dataset receive adequate compensation for it.

### G. Distributed ledger

Successful transactions are recorded on a distributed ledger to provide a decentralised, immutable record of ownership. This ledger will represent which agents have access to who's data, and what access rights they are allowed.

## V. Building Blocks of the Data Market

### A. Context

We present a case study with cars driving in a given city. We focus on Air Quality Index (AQI) as our metric of relevance, which can be measured through a car's air quality sensor. Many car manufacturers include an air quality sensor in the car's air conditioning mechanism. To illustrate the function of the data proposed data market, we divide the city into a grid with constant sized quadrants. As the cars drive across the city they measure AQI values at different quadrants of the city. Only agents with a valid license plate are granted access to collect and sell data on the marketplace. This acts as a defence mechanism against sybil attacks.

### B. Assumptions

1) For each potential data point that can be made available in the marketplace, there is an over-supply of measurements.
2) Competing sellers are interested in aggregating (crowd-sourcing) data points from the market to fulfil a specific purpose.
3) Competing buyers only purchase data from the regulated market, and that each data point in the market has a unique identifier so that replicated data made available on secondary markets can be easily detected by data purchasers.
4) There is an existing mechanism that can verify the geographical location of an agent with a certain degree of confidence, and thus the provenance of the aforementioned agent's data collected. Several works have been carried out that corroborate that this is a reasonable assumption to make [34], [64] [14] [61] [13].
5) Following from 4 a Proof of Position algorithm is defined in V.13. Furthermore it is assumed that agents cannot be in more than one location at the same time. When an agent declares a measurement taken from a given location, we can verify this datapoint, the agent's ID and their declared position using V.13.
6) Following from assumption 1 and 2, the cases when a buyer wishes to purchase data from a geographical location where there is no data available are not accounted for. Note that there is the possibility to relax this constrain in future works through estimating for data-poor areas using neighbouring data, for example, with the KNN algorithm framework presented in [31].

### C. Definitions

**Definition V.1** (Datapoint)**.** *A datapoint is defined as $x_i \in X$ where $x_i$ denotes the data point of agent $i$.*

**Definition V.2** (Location quadrant)**.** *The set of all possible car locations is defined as $\mathcal{L}$. The location quadrant $q$, is an instance of $\mathcal{L}$, where $q \in \mathcal{L}$.*

**Definition V.3** (Buyer)**.** *A buyer is defined as $m$, where $m \in M$ and $M$ is the set of agents looking to purchase ownership (or any other given rights) of the datasets that are available for sale on the marketplace.*

**Definition V.4** (Agent)**.** *An agent is defined as $a_{i,s} \in A$ where $A$ is the set of all agents competing to complete the marketplace algorithm to become sellers. The index $i \in N$ where $N$ is the total number of agents on the algorithmic marketplace at a given time interval $t \in T$. The index $s$ denotes the stage in the access control mechanism that agent $a_{i,s}$ is in, where $s \in \{1, 2\}$. For example, agent $a_{5,2}$ is the agent number 5, currently in stage 2 of the access control mechanism in the data marketplace.*

*For brevity, in sections where an agent is not in the access control mechanism, the authors omit the use of the second index.*

**Definition V.5** (Spatial Coalition)**.** *A spatial coalition is defined as a group of agents in the same location quadrant $q$, where these agents aggregate their individual data to provide an agreed upon dataset of that quadrant, $D_q$. The coalition is denoted as $C_q$ where $q$ is the location quadrant where the coalition is formed.*

**Definition V.6** (Value Function)**.** *$v(S) = y$ where $y$ is the number of location quadrants in the city with registered data measurements in them. $S$ is a coalition of agents with corresponding datapoints. The function is strictly non-negative and satisfies $v(\emptyset) = 0$.*

**Definition V.7** (Shapley Value)**.** *The Shapley Value is defined in [48] as a way to distribute reward amongst a coalition of n-person games. Each player $i$ in the game receives a value $\psi_i$ that corresponds to their reward. The Shapley Value satisfies the notions of Shapley fairness which are:*

*1) Balance:*

$$\sum_{a_i=1}^{A} \psi_m(a_i) = 1$$

*2) Efficiency: The sum of the Shapley value of all the agents is equal to the value of the grand coalition of agents $[A]$:*

$$\sum_{a_i=1}^{A} \psi_{a_i}(v) = v(A)$$

*3) Symmetry: If agents $a_i$ and $a_j$ are equivalent in the coalition of agents $S$ such that both agents are providing data of the same value where $v(S \cup \{a_i\}) + v(S \cup \{a_j\})$ for every subset $S$ of $A$ which contains neither $a_i$ nor $a_j$, then $\psi_{a_i}(v) = \psi_{a_j}(v)$*

*4) Additivity: If we define a coalition of agents to be $k = \{a_i, a_j\}$ then $\psi_k = \psi_{a_i} + \psi_{a_j}$*

*5) Null agent: An agent $a_i$ is null if $v(S \cup \{a_i\}) = v(S)$. If this is the case then $\psi_{a_i} = 0$.*

*Therefore formal definition of the Shapley value of an agent $a_i$ that is in a set of $A$ players is*

$$\psi_(a_i) = \sum_{S \subseteq A \setminus \{a_i\}} \frac{|S|!(|A| - S - 1)!}{|A|!}(v(S \cup \{a_i\}) - v(S))$$

*The Shapley Value is the unique allocation $\psi$ that satisfies all the properties of Shapley fairness, described above.*

**Definition V.8** (Smart Contract)**.** *A smart contract is a program that will automatically execute a protocol once certain conditions are met. It does not require intermediaries and allow for the automation of certain tasks [19] [54]. In our context, a smart contract will be executed by agent $a_{i,s+1}$ to compute the Shapley value of agent $a_{j,s}$'s dataset.*

*The outputs will be the Shapley value of agent $a_{j,s}$'s dataset and a new smart contract for agent $a_{i,s+1}$. Calculating the new smart contract generated serves as the proof of agent $a_{j,s}$'s useful work.*

*Every agent's smart contract will also contain a record of the buyer IDs and the permission that they have purchased from the agent. These could include permission to read the dataset, to compute analytics or to re-sell the dataset.*

**Definition V.9** (Bidding Mechanism)**.** *Following 6, there is a set of buyers $M_q$ for each $q \in \mathcal{L}$ wishing to purchase data from that quadrant. A Bidding Mechanism is defined, $BM$, as a function that returns a buyer $m$ that will purchase the dataset $D$ corresponding to $q$, such that $m \in M_q$. Consequently, for all $q \in \mathcal{L}$: $m \leftarrow BM(M_q)$.*

**Definition V.10** (Reward Distribution Function)**.** *The reward associated with the datapoint of a specific quadrant is defined as $v(C_q)$. In other words, the value that the spatial coalition $C_q$ provides with their agreed upon datapoint $D_q$, of the location quadrant $q$. Each agent in $C_q$ receives a coefficient $\alpha = \frac{1}{|D_q - d_i|}$, where $d_i$ is the agent's individual datapoint. Consequently, the value $v(C_q)$ is split amongst all the agents in $C_q$ as follows: for each agent, they receive $||\frac{v(C_q)}{|C_q|} \times \alpha||$*

**Definition V.11** (Commitment)**.** *A commitment to a datapoint $d_i$, location quadrant $q$ and ID $i$ of an agent $a_{i,0}$ is defined as $c \leftarrow \mathrm{Commitment}(a_{i,0}, d_i, q)$*

**Definition V.12** (PoID)**.** *Let* PoID *be an algorithm that verifies the valid identity of an agent $a_{i,0}$, with ID $i$. In the context presented, this identification will be the license plate. The algorithm will return a boolean $\alpha$ that will be $True$ if the agent has presented a valid license plate and $False$ otherwise.*

*Then* PoID *is defined as the following algorithm:*
$\alpha \leftarrow \mathrm{PoID}(i, c)$
*This algorithm is executed by a central authority that can verify the validity of an agent's identity.*

**Definition V.13** (PoP)**.** *Let* PoP *be an algorithm that is called by an agent $a_{i,0}$, with ID $i$. The algorithm takes as inputs the agent's committment $c$, and their location quadrant $q$. We define* PoP *as the following algorithm:*
$\beta \leftarrow \mathrm{a}_{\mathrm{i,0}}^{\mathrm{PoP}}(q, c)$
*where the output will be a boolean $\beta$ that will be* True *if the position* q *matches the agent's true location and* False *otherwise. This algorithm is executed by a central authority that can verify the validity of an agent's position.*

**Definition V.14** (TimeCheck)**.** *The function* TimeCheck *takes in three arguments, the timestamp of the datapoint provided, $t$, the current time at which the function is executed, $timeNow$, and an acceptable range of elapsed time, $r$. The output of the function is $\gamma$. If $t - timeNow < r$, $\gamma$ takes value $True$ and $False$ otherwise.*
$\gamma \leftarrow \mathrm{TimeCheck}(t, timeNow, r)$

**Definition V.15** (Verify)**.** *Let* Verify *be an algorithm that checks that outputs of* PoID *and* PoP*. It will return a token $Token$ that will take the value $True$ iff $\alpha$, $\beta$ and $\gamma$ are all $True$, and $False$ otherwise.*
$Token \leftarrow \mathrm{Verify}(\alpha, \beta, \gamma)$
*This algorithm can be executed by any agent in the data market, given that the proofs of* PoID, PoP *and* TimeCheck *are publicly verifiable.*

**Definition V.16** (Reputation)**.** *An agent $a_i$ assigns a score of trustworthiness to an agent $a_j$. This score is denoted as $r_{i \to j}$.*

**Definition V.17** (Election Scheme)**.** *The authors use a generalised definition for voting schemes, following from the work in [36] and [49]*

*An* Election Scheme *is a tuple of probabilistic polynomial-time algorithms*
$(\mathrm{Setup}, \mathrm{Vote}, \mathrm{Partial-Tally}, \mathrm{Recover})$ *such that:*

Setup *denoted* $(pk, sk) \leftarrow \text{Setup}(k)$ *is run by the administrator. The algorithm takes security parameter* $k$ *as an input, and returns public key* $pk$ *and private key* $sk$.

Vote *denoted* $b \leftarrow \text{Vote}(pk, v, k)$ *is run by the voters. The algorithm takes public key* $pk$*, the voter's vote* $v$ *and security parameter* $k$ *as inputs and returns a ballot* $b$*, or an error* $(\perp)$.

Partial – Tally *denoted* $e \leftarrow \text{Partial} - \text{Tally}(sk, \mathfrak{bb}, k)$ *is run by the administrator. The algorithm takes secret key* $sk$*, bulletin board* $\mathfrak{bb}$*, and security parameter* $k$ *as inputs and returns evidence* $e$ *of a computed partial tally.*

Recover *denoted* $\mathfrak{v} \leftarrow \text{Recover}(\mathfrak{bb}, e, pk)$ *is run by the administrator. The algorithm takes bulletin board* $\mathfrak{bb}$*, evidence* $e$ *and public key* $pk$ *as inputs and returns the election outcome* $\mathfrak{v}$.

**Definition V.18** (Ballot Secrecy)**.** *The authors utilise the definition for* Ballot Secrecy *presented in the work [49]. Here, an adversary is tasked with constructing a bulletin board for ballots containing votes in one of two possible lists. The choice of the list is determined by a coin flip. Given the election outcome, the adversary must determine which of the two lists of votes they constructed a bulletin board for. This definition is formalised as a game where if the adversary wins with a significant probability, the property of* Ballot Secrecy *does not hold. An* Election Scheme *is said to satisfy* Ballot Secrecy *if for a probabilistic polynomial-time adversary, their probability of success is negligible. This definition accounts for an adversary that can control ballot collection.*

## VI. The Data Market

### A. The Verification Algorithm

**Algorithm 1:** Verification: Verifying Algorithm $(a_{i,0},\ d_i,\ q,\ t,\ r)$

$c \leftarrow \text{Commitment}(a_{i,0}, d_i, q, t)$;
$\alpha \leftarrow \text{PoID}(i, c)$;
$\beta \leftarrow \text{PoP}(q, c)$;
$\gamma \leftarrow \text{TimeCheck}(timeNow, t, r)$;
$Token \leftarrow \text{Verify}(\alpha, \beta, \gamma)$;
**return** $Token \leftarrow \{True, False\}$;

The validity of the data submission must be verified. This is done before the data reaches the data marketplace, to avoid retroactive correction of poor quality data. This is done through the VerifyingAlgorithm. Firstly, an agent provides an immutable commitment of their datapoint, location quadrant, timestamp and unique identifier. Next, the agent submits their unique identifier to a centralised authority that verifies that this is a valid and real identity. In practise, for this context, this identifier will be the agent's vehicle license plate. Subsequently, the agent generates a valid proof of position. Following from assumption 2, an agent can only provide one valid outcome from algorithm V.13 at a given time instance $t$. Then, the datapoint is checked to ensure it is not obsolete through TimeCheck. Finally, the outputs of all previous functions are verified to ensure the agent has produced a valid proof. If and only iff all of these are $True$, the agent is issued with a unique valid token, that allows them to participate in the consensus mechanism.

### B. Voting Scheme: Reputation-based Maximum Entropy Voting

In what follows the authors present an adaptation of the Maximum Entropy Voting scheme that takes into consideration the reputation of agents in the system. Both components are introduced and will work as a single functional building block in the the data market design.

*1) Reputation:* In the general sense, reputation can be seen as a performance or trustworthiness metric that is assigned to an entity or group. For the purpose of this work, reputation should be earned through proof of honesty and good behaviour. In this case, agents that can demonstrate they have produced an honest computation should receive adequate recompense.

The authors propose presenting proof of running a voting election as a suitable means to earn reputation. An agent running the voting scheme is therefore considered to be an administrator, and acts as a trusted authority.

In the case of Maximum Entropy Voting, the administrator running the election will return a voting outcome. To prove that this outcome does indeed satisfy the optimisation formulation defined in **??**[citation-needed], an administrator can show that their sampled outcome satisfies the Karush–Kuhn–Tucker (KKT) conditions.

"The necessary conditions are sufficient for optimality if the objective function of a maximization problem is a concave function, the inequality constraints $g_j$ are continuously differentiable convex functions and the equality constraints $h_i$ are affine functions. Similarly, if the objective function of a minimization problem is a convex function, the necessary conditions are also sufficient for optimality.".

End to end (E2E) verifiable voting requires that all voters can verify the following three properties: their vote was

- cast as intended
- recorded as cast
- tallied as cast

An example of an E2E voting scheme is Helios [1]. This scheme uses homomorphic encryption, enabling the administrator to demonstrate the correctness of the aggregation of votes operation.

The operations required in the aggregation of votes for MEV can be done using homomorphic encryption, and an E2E voting scheme such as Helios could be used to carry out this step. This aggregation is then used to solve the optimisation problem and yield a final vote outcome. Once the optimisation is solved, the administrator can release the aggregation of votes and prove that:

1) The aggregation operation is correct.
2) The solution of the optimisation problem is indeed optimal, given it satisfies the KKT conditions.
3) The voters can then also verify the properties in VI-B1 hold.

Upon presenting the verifiable proofs mentioned above, agents behaving as administrators should receive reputation from other agents in the network.

The authors note that the Helios voting scheme has been proven not to satisfy Ballot Secrecy in [49] and [36]. Proposing and testing an E2E verifiable voting scheme that satisfies definitions of Ballot Secrecy, receipt-freeness and coercion resistance is beyond the scope of this work, although of interest for future work.

*2) Reputation-based Maximum Entropy Voting:*

**Definition VI.1** (Vote)**.** *The vote of agent $a_i \in A$, is defined as a pairwise preference matrix in $S(i)\mathbb{R}^{N\times N}$. Each entry is indexed by any two agents in $A$ and its value is derived from datapoint $x_i$ V.1 and reputation $r_{i\to j}$ V.16 , as in* (1)*.*

**Definition VI.2** (Aggregation of Votes)**.** *The aggregation of all agents' votes $S(A)$, is defined as the average of $S(i), i \in A$, as in* (2)*.*

**Definition VI.3** (Agent Ordering)**.** *An agent ordering, denoted as $t$, is defined as a permutation of agents in [35], i.e., arranging all agents in order. Further, concerning computation complexity, we suggest $t$ being a combination of agents, i.e., selecting a subset of agents as the preferred group, such that the order of agents does not matter.*

**Definition VI.4** (Ordering Set)**.** *The ordering set $\mathcal{T}$ is the set of all possible agent orderings, such that $t$ is an element of $\mathcal{T}$. See Table 3 for the example of an ordering set of combinations, with 3 agents.*

**Definition VI.5** (Probability Measure of Ordering Set)**.** *The (discrete) probability measure, $\pi : \mathcal{T} \to \mathbb{R}_{\geq 0}$ gives a probability of each ordering $t \in \mathcal{T}$ being selected as the outcome ordering $t^*$. The measure $\pi$ of maximal entropy while adhere to RP III.3, i.e., the optimal solution of Problem 5 is denoted as $\pi^*$.*

The MEV could be summarised as the algorithm 2. Firstly, each agent's data point $x_i$ and reputation $r_{i\to j}$ are extracted as a pairwise preference matrix $S(i)$, i.e., a vote of agent $a_i$. Then, an average of all agents' pairwise preference matrix $S(A)$ is calculated, which is seen as the aggregation of all agents' votes. Further, we are able to find a probability measure $\pi^*$ that has the maximal entropy while adhere to PR property in III.3. Finally, we sample an outcome ordering $t^*$ from $\pi^*$, using a 'prize-wheel' sampling.

**Algorithm 2:** MEV Algorithm ($x_i$, $r_{i\to j}, M$)

$S(i) \leftarrow$ (1) with $x_i, r_{i\to j}$ ;
$S(A) \leftarrow$ (2) with $S(i)$;
$\mathcal{T} \leftarrow$ Constructed with $M$ as in Figure 3;
(auxiliary variables) $S(t) \leftarrow$ (3) with $\mathcal{T}$;
$\pi^* \leftarrow$ Problem 5 with $\mathcal{T}, S(t), S(A)$;
$t^* \leftarrow$ Sampling from measure $\pi^*$ as in Figure 4;
**return** $t^* \leftarrow \mathrm{MEV}(x_i, r_{i\to j}, M)$;

Since MEV happens before the access control mechanism, all agents are in the stage zero. For brevity, agents $a_{i,0} \in A$ will be denoted as $a_i$. Given a set of agents of cardinality $|A| = N$, each agent $a_i$ has a data point $x_i \in \mathcal{X}$ and a reputation $r_{i\to j}$ for all agents $a_k \in A$.

The data point $x_i$ could be defined as temperature or air quality measurements of an agent which they want to submit and sell. The reputation $r_{i\to j} \in \mathbb{R}^+$ is a non-negative value that represents the individualised reputation of agent $a_j$ from the perspective of agent $a_i$. Intuitively, an agent would prefer agents with positive engagements, a good record of recent behaviour, and who is similar to itself. In this case study, every agent would have an impression of how trustworthy every agent in the network is and base it on their previous interaction and the overall track record of an agent in the system, or base on the fact that they know each other and are in coalition.

To draw the analogy to a social network of a community. One would consider people trustworthy who are its friends and have had positive interactions with. Additionally, if given proof that a person has done much good so far, one would also consider this person trustworthy. Finally, for now, assume that this reputation $r_{i\to j}$ is given and its influence on the outcome will be discussed in later sections, namely **??** and **??**.

To combine maximum entropy voting and reputation, a key step is to move from reputation $r_{i\to j}$ to a pairwise preference matrix $S(i) \in \mathbb{R}^{N\times N}$. The entry of a pairwise preference matrix is indexed by every two agents of $A$, and its values is defined as follows:

$$S(i)_{j,k} = \begin{cases} 1 & \text{if } a_i \text{ prefers } a_j \text{ and } j \neq k \\ 0.5 & \text{if } a_i \text{ prefers both equally and } j \neq k \\ 0 & \text{if } a_i \text{ prefers } a_k \text{ or } j = k \end{cases}, \tag{1}$$

for $a_j, a_k \in A$ and $a_j$ is preferred to $a_k$ if and only if $\frac{1+|x_i|\cdot r_{i\to j}}{1+|x_i - x_j|} > \frac{1+|x_i|\cdot r_{i\to k}}{1+|x_i - x_k|}$ and both agents are equally preferred if the two values are equalised, such that the reputation are scaled by their absolute differences from agent $a_i$.

Likewise, we could find a pairwise preference matrix $S(i)$ for each agent $a_i$. The average of pairwise preference matrices over all agents are denoted as the preference matrix $S(A)$, as follows. $S(A)$ represents the pairwise preference of all agents in $A$, whose entry $S(A)_{j,k}$, indeed, displays the proportion of agents that prefers agent $a_j$ over agent $a_k$.

$$S(A) := \frac{1}{N} \sum_{a_i \in A} S(i). \tag{2}$$

The original MEV [35] runs an optimisation over all candidate orderings, which strongly defines the computational complexity of the problem because the number of orderings is the factorial of the number of candidates. As a variant of MEV, we consider agent combinations, in stead of permutations for the ordering set $\mathcal{T}$, such that $A$ is divided into a preferred group $\mathcal{P}$ of cardinality $M$ and non-preferred group $\mathcal{NP}$, where $M$ is the number of winners needed. Hence, the cardinality of the ordering set decreases from $N!$ to $\frac{M!}{M!(N-M)!}$. For reasonable small $M$, this leads to to a dramatic reduction of the computational complexity.

For each ordering $t \in \mathcal{T}$, we could define its pairwise preference matrix $S(t)$, whose entry is defined as the same way as in (1):

$$S(t)_{j,k} = \begin{cases} 1 & \text{if } a_j \text{ is placed over } a_k \\ 0.5 & \text{if both are in the same group and } j \neq k \\ 0 & \text{if } a_k \text{ is placed over } a_j \text{ or } j = k \end{cases}, \tag{3}$$

for $a_j, a_k \in A$. Let us define an **unknown** probability measure $\pi : \mathcal{T} \to \mathbb{R}_{\geq 0}$. $\pi(t), t \in \mathcal{T}$ gives the probability of $t$ being chosen as the outcome ordering. Then, we construct a theoretical preference matrix $S(\pi)$ as follows:

$$S(\pi) := \sum_{t \in \mathcal{T}} \pi(t) \cdot S(t). \tag{4}$$

The entry $S(\pi)_{j,k}$ states that under probability measure $\pi$, the probability of the outcome ordering placing $a_j$ over $a_k$. Recall the definition of RP in Section III.3 or [35], it simply requests $S(\pi) = S(A)$.

Entropy of $\pi$ implies the uncertainty of choosing elements in $\mathcal{T}$. The uniform distribution has the maximum amount of entropy. Associated with $\pi$, the entropy is defined as $-\sum_{t \in \mathcal{T}} \pi(t) \log \pi(t)$. Hence, the original formulation of maximum entropy voting adhere to RP is as (5). In this formulation, when maximising the entropy, we ensure the solution $\pi^*$ to be the most moderate probability measure with obeying RP in III.3.

$$\begin{aligned} \pi^* = \max_{\pi} & -\sum_{t \in \mathcal{T}} \pi(t) \log \pi(t) \\ \text{s.t.} & \sum_{t \in \mathcal{T}} \pi(t) \cdot S(t) = S(A) \\ & \sum_{t \in \mathcal{T}} \pi(t) = 1 \\ & \pi(t) \geq 0 \quad \forall t \in \mathcal{T} \end{aligned} \tag{5}$$

*3) A Motivating Example:* Consider $A = \{a_i, a_j, a_k\}$ and only one winner is needed ($M = 1$), all possible combinations are in Figure 3, while the number of permutations would be $3!$.

| $\mathcal{T}$ | $t_1$ | $t_2$ | $t_3$ |
|---|---|---|---|
| Preferred $\mathcal{P}$ | $(a_i)$ | $(a_j)$ | $(a_k)$ |
| Non-Preferred $\mathcal{NP}$ | $(a_j, a_k)$ | $(a_i, a_k)$ | $(a_i, a_j)$ |

Figure 3. The lower-carnality ordering set when $A = \{a_i, a_j, a_k\}$ and $M = 1$. Agents in the same brackets are given the same rank in an ordering.

As an example, the pairwise preference matrix $S(t_1)$ is displayed in (6), following (3).

$$S(t_1) = \begin{array}{c|ccc} & a_i & a_j & a_k \\ \hline a_i & 0 & 1 & 1 \\ a_j & 0 & 0 & 1/2 \\ a_k & 0 & 1/2 & 0 \end{array} \tag{6}$$

Suppose an optimal measure $\pi^*$ is extracted from Problem 5. Assuming $\pi^*(t_1) = 0.3$, $\pi^*(t_2) = 0.4$ and $\pi^*(t_3) = 0.3$, to sample an outcome ordering $t^*$ from $\pi^*$, consider a prize wheel as in Figure 4. The wheel includes $|\mathcal{T}|$ wedges where each wedge represents one ordering $t$ and takes the share of $\pi^*(t)$. To obtain an outcome ordering, simply spin the wheel and $t^*$ is the wedge where the red arrow stops, i.e., $t_1$ in this figure.

### C. Data Consensus

The core idea is that in an oversubscribed environment crowd-sourcing can be used to estimate a central measure which ideally reflects the ground truth as close as possible. The assumption is that every agent measures the same source and should therefore have the same results within margins of measurement precision. The only reasons why there can be deviations is either the used sensor is faulty or the agent is intentionally submitting incorrect results. Therefore, by comparing agents' measurements against each other the aim is to sort out faulty and incorrect results.

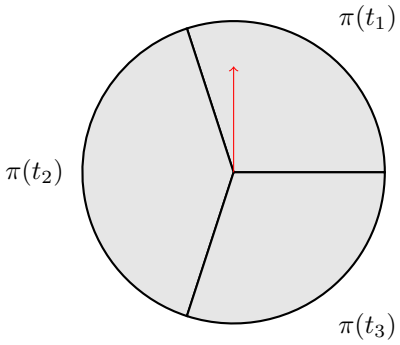


Figure 4. A prize wheel for sampling an outcome ordering $t \in \mathcal{T}$ from a probability measure $\pi$.

There are different ways to approach this and in order to characterise them two concepts will be introduced, namely k-anonymity and the breakdown point.

*1) K-Anonymity:* One definition is "*A data release is said to satisfy k-anonymity if every tuple released cannot be related to fewer than k respondents, where k is a positive integer set by the data holder, possibly as the result of a negotiation with other parties.*" [**?**][9] In other words, if a central measure includes multiple measurements and is assigned a k-anonymity with k=2, it is not possible to identify a single measurement without a second party revealing their measurement.

### *D. Breakdown Point*

The breakdown point characterises the robustness of an estimator and can be defined as follows. *The breakdown point is the smallest fraction of bad observations that may cause an estimator to take on arbitrarily large aberrant values* [**?**]. It is usually dependent on the sample size, n, and can be written as a function of n.

*1) Mean as Central Measure:* The mean $\bar{x}$, arithmetic mean, in its simplest form is defined in equation (7).

$$\bar{x} = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) = \frac{x_1 + x_2 + \cdots + x_n}{n} \tag{7}$$

where $x_i$ are the individual measurements and n the sample size.

The mean can be calculated in a decentralised and privacy preserving manner [40]. In theory it can additionally be calculated under homomorphic encryption. [**?**] The k-anonymity of the mean results then in k = n-1. The breakdown point of the mean is $\frac{1}{n}$ or in other words, a single measurement can cause the mean to take on arbitrarily high or low values. This can be mitigated with domain knowledge, i.e. restraining the range for valid measurements, for example when considering temperature in London to [-30°C - 50°C]. However, even with this counter measure in place, a larger coalition of malevolent agents is still able to influence the mean significantly. This in combination with the face that malevolent agents can be expected in a data market context, it may be that the mean is not sufficiently robust for most use cases.

*2) Median as Central Measure:* The median is a is the value separating the higher half from the lower half of a data sample. It can be defined for a numerically ordered, finite sample of size n, as follows.

$$\text{median}(x) = \begin{cases} x_{(n+1)/2} & \text{if n is odd} \\ \frac{1}{2} \cdot (x_{(n/2)} + x_{(n/2)+1}) & \text{if n is even} \end{cases} \tag{8}$$

This definition is invalid for an unordered sample of measurements. In order to compute the median for such a sample, the measurements need to be sorted numerically first, at least partly. Given a multi agent setting, this can be done in a distributed way by using an selection algorithm that finds the $k^{th}$ smallest element(s). In detail, when a comparison between two values is required by the algorithm, this can be executed in a P2P manner where two agents exchange their measurements and report which is bigger. However, This cannot be done without revealing the measurements, and thus the privacy of the data cannot be protected when using the median to estimate a centrality measure. The breakdown point of the median characterises it as one of the most robust estimators and is for the worst case given with $\frac{1}{2}$.

*3) Mean Median Algorithm as Central Measure:* In an adversarial environment, the high robustness of the median is desirable, however, often protection of privacy is also of concern. Therefore, the Mean Median Algorithm was designed to have an algorithm that estimates a central measure in a robust and privacy-preserving way. It must be said that it is a compromise and this algorithm is not as robust as the median and privacy-preserving as the mean when compared individually.

To illustrate the algorithm, the earlier in section **??** introduced notation will be used. The first step is to randomly assign every agent to a group in such a way that there are r groups with at least s agents each. The way the parameters r and s are chosen determine the anonymity and robustness properties of the algorithm and will be discussed in the simulation section **??**. The next step is to calculate the mean within each group. The resulting mean is at least of k-anonymity with k = s-1. As there are r groups, there are r means of which the median is chosen. This gives a breakdown point given in equation 9.

$$\text{Breakdown point of } \text{meanmedian}(x) = \frac{r}{2n} \tag{9}$$

[9] latanyasweeney.org; k-anonymity

The relationship between s, r and the number of agents n is given with the inequality (10)

$$n \geq s \cdot r \tag{10}$$

After having chosen a central measure the next step is to assess every agent's measurement based on that central measure. Here there are two approaches we will discuss.

- Relative Difference
- Similarity

The first approach is more suitable for the one-dimensional case and we define it as follows.

$$\nu(C) = \frac{CM - \frac{1}{|C|}\sum_i^C m_i}{|CM|} \tag{11}$$

where $C$ is the set of agent that form a coalition.

The second approach is more suitable for the multi-dimensional case and we define it as follows.

$$\nu(C) = cos(CM, \frac{1}{|C|}\sum_i^C m_i) \tag{12}$$

where $cos(.,.)$ is the cosine similarity.

Using these $\nu(C)$ can then be used to estimate Shapley Values for each agent to account for their contribution to the CM.

### *E. Stage 4: Access Control Mechanism to the Data Market*

The previous stages 1, 2 and 3 run simultaneously in numerous rounds, as vehicles sense data and form coalitions to provide an agreed up value for a given location quadrant. By stage 4, there is an excess of datapoints for different locations and these datapoints are on a queue to enter and be sold on the datamarket.

This section outlines the access control mechanism to sell this oversupply of datapoints on the market, and in what order these should be prioritised to enter the datamarket. This access control mechanism can be considered to have two intermiediary steps: firstly, all datapoints are assigned a priority; and secondly, proportionally to this priority, the coalition owning that datapoint must perform an adaptive, useful proof of work.

*1) Contribution ranking: Shapley Value:* At a given time $t$, a new set of datapoints will be submitted to a queue, to ultimately enter the datamarket. Let this set be $\mathbf{D_t} = \{D_{q1}, D_{q2}, D_{q3}...\}$ where each item of the set is the datapoint computed by a coalition $C_q$, of a given location quadrant, where $\{q1, q2, q3...\} \in \mathcal{L}$. For each element in $\mathbf{D_t}$, the Shapley value $\psi(C_q)$ is calculated. Note that each element in $\mathbf{D_t}$ is a datapoint that corresponds to a spatial coalition $C_q$. The grand coalition in this case is considered to be the union of all coalitions that have datapoints already for sale on the datamarket, denoted as $\mathbf{S}$. Each datapoint in $\mathbf{D_t}$ is assessed using the Shapley value, which determines what datapoints would increase the overall value of the datamarket, with respect to the defined value function, should they be added to the grand coalition $\mathbf{S}$. In other words, the datapoints that receive a higher Shapley value, would contribute more towards increasing the combined value of the data already for sale on the market. In this fashion, the Shapley value is used as a metric to rank the most valuable datapoints with respect to a value function.

*2) Useful, adaptive proof of work:* Subsequently, once the datapoints in $\mathbf{D_t}$ have each received a Shapley value, they are then assigned a proof of work they must complete. This proof of work is inversely proportional to the Shapley value. The more valuable a datapoint is deemed for the datamarket, the less proof of work the coalition owning it should complete, to enter the market. This assigned proof of work, in fact, is computing the Shapley Value of the next set of datapoints, $\mathbf{D_{t+1}}$.

## VII. Adversarial Attacks

In an environment like decentralised networks or data markets one must take into account the possibility of attacks on the system. The authors proceed to describe their nature and how these are mitigated by the functional components of the data market architecture.

**Definition VII.1** (Sybil Attack)**.** *Sybil Attacks are a type of attack in which an attacker creates a large number of pseudonymous identities which they use to exert power in and influence the network.*

Sybil attacks are mitigated in the verification stage, as agents must present a valid proof of identity. This proof is granted to them through a centralised authority but all other agents can verify that it exists and therefore that it must be valid. Generating multiple identities is made expensive in this proposed architecture, because agents must provide a valid license plate to enter the market and collect data. Unless the attacker purchases a real vehicle with a valid license plate, they cannot succeed in creating another identity, and therefore sell data in the market.

**Definition VII.2** (Wormhole Attack)**.** *A Wormhole Attack involves a user maliciously reporting they are in a location that is not the one they are truly in.*

An attack can be mounted by a series of malicious actors claiming to measure data from a location they are not truly in, and wishing to monetise this fraudulent data. To mitigate against this attack, agents must present a valid proof of position in the verification stage (defined in V.13. This proof is assumed to be correct and sound, and by definition, agents are only able to present one valid proof.

**Definition VII.3** (Data Poisoning)**.** *Data Poisoning is an attack where malicious agents collude to report fake data in order to influence the agreed upon state of a system.*

Malicious agents wishing to report fake data must influence enough agents in their spacial coalition to ensure that sufficient agents in the data consensus stage will compute a fake data point. Probabilistic voting schemes make the cost of this coercion significantly high. Furthermore, to sell the uploaded data point, the agent must perform a useful proof of work that is proportional to how valuable the data point is deemed. The more useful the data point the less work the agent must carry out to sell it. Selling spam data will therefore be very time consuming for an attacker.

## VIII. Evaluation

### A. Simulation Setup

*1) Median Algorithm:*

- Decentralised
- Robust up to 50%

The Median

*2) Mean Median Algorithm:*

- Decentralised
- Privacy Preserving
- Robust up to 50%

### B. Max Entropy Voting

- centralised
- adapted to our problem

The authors have already introduced Max Entropy Voting in the section III-C1 and will continue here to adapt the technique for the data market. This is necessary because of the computational complexity the original problem poses with $N!$ where $N$ is the number of candidates that are voted on. For the case of data markets this is computationally infeasible because it is necessary to process on a high frequency double digits numbers of candidates.

One way to reduce the complexity of the problem is to consider a different set of combinations in which the candidates are ranked instead of all possible permutations like in the original problem. The key idea is to look at it from perspective of sampling with replacement, i.e. the set of candidates is seen as pool from which a subset of preferred agents is drawn without replacement. By considering only the combinations in which a subset can be drawn without demanding a ranking within the groups reduces the complexity to $\frac{N!}{K!(N-K)!}$, where N is the number of candidates and K the number of drawn agents. This adapted way allows the application of the max entropy voting strategy to a reasonable large set of candidates as long as the number of drawn agents is relatively small.

A practical challenge that poses this change is that the constraint (**??**) cannot be satisfied anymore and therefore the whole linear problem becomes infeasible. To solve this problem an amendment of the objective function (**??**) and the aforementioned constraint as follows.

$$\max_{\pi, S_{free}} -\alpha||S_{free}||_2 - \sum_t \pi_t \log \pi_t \tag{13}$$

$$\sum_t^T \pi_t S(t) + S_{free} = Z_{j,k} \tag{14}$$

where $S_{free}$ is a matrix of free, unbound variables, $||.||_2$ denotes the Frobenius norm, and $\alpha$ a constant which in practice is assigned a high value to force the linear program to the solution closest to the one of the original problem.

### C. Conclusion

future work: if the running of MEV can be verified by an easily verifiable but hard to generate statement, we could 'SNARKify' this and provably demonstrate in a non interacitve succint argument of knowledge method that certain nodes have run a valid MEV. This way, the running of the election can also be done privately to preserve ballot secrecy (public elections are verifiable but not private so not free, everyone can see how you voted so you are susceptible to coercion). If this statement is generated successfully, everyone can verify its validity, and then these nodes should earn reputation points in exchange.

We can even use compact certificates of collective knowledge to ensure that a group of agents agree that an MEV election was run correctly. in fact, for compact certificates of collective knowledge, any NP statement will suffice. Using comp. certs means that the more weighty attestors signing, the smaller the compact certificate, so the system incentivises be design to be decentralised. To optimise this, the attestors should sign on the fact that a unique MEV outcome that is correct and valid, as opposed to the validity of the signatures of the high number of attestors.

For MEV particularly, what statements suffice to prove that MEV was run correctly? ie: what is the NP statement that we could use in the comp. certs? (Pietro- that the solution provided is indeed the optimum point of the optimisation problem. In that case, would it suffice to prove that the point satisfies the KKT conditions? the objective function for a max. problem must be concave, the inequality constraints are continuously differentiable

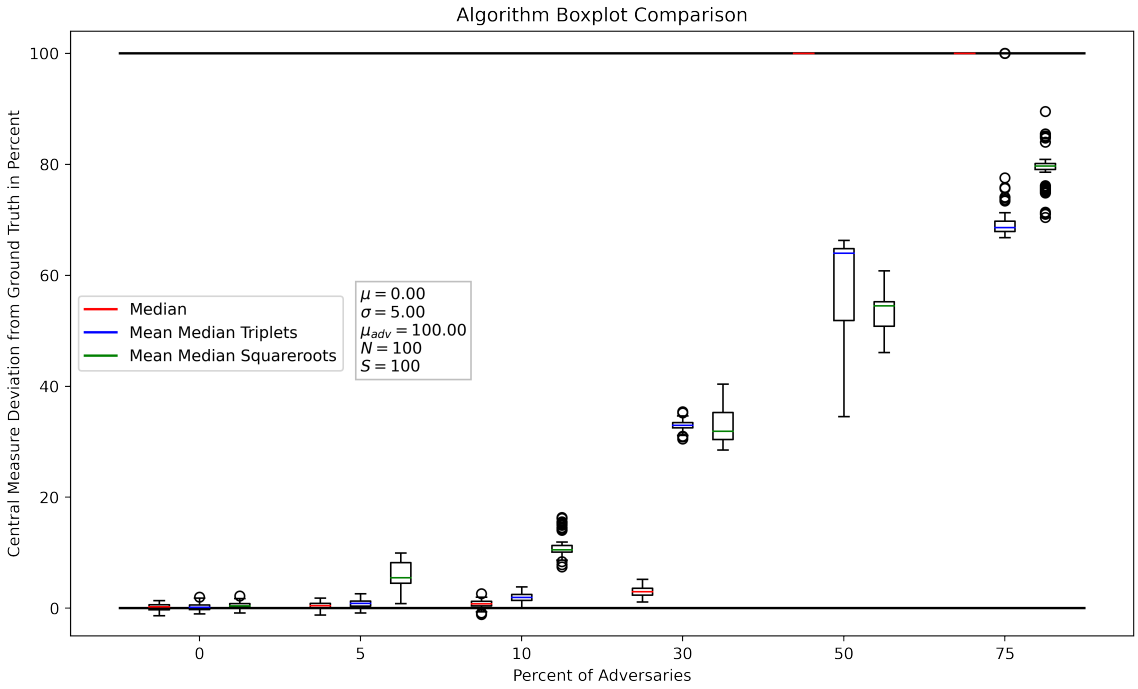


Figure 5. Breakdown analysis of data consensus algorithms, with a coordinated data poisoning attack

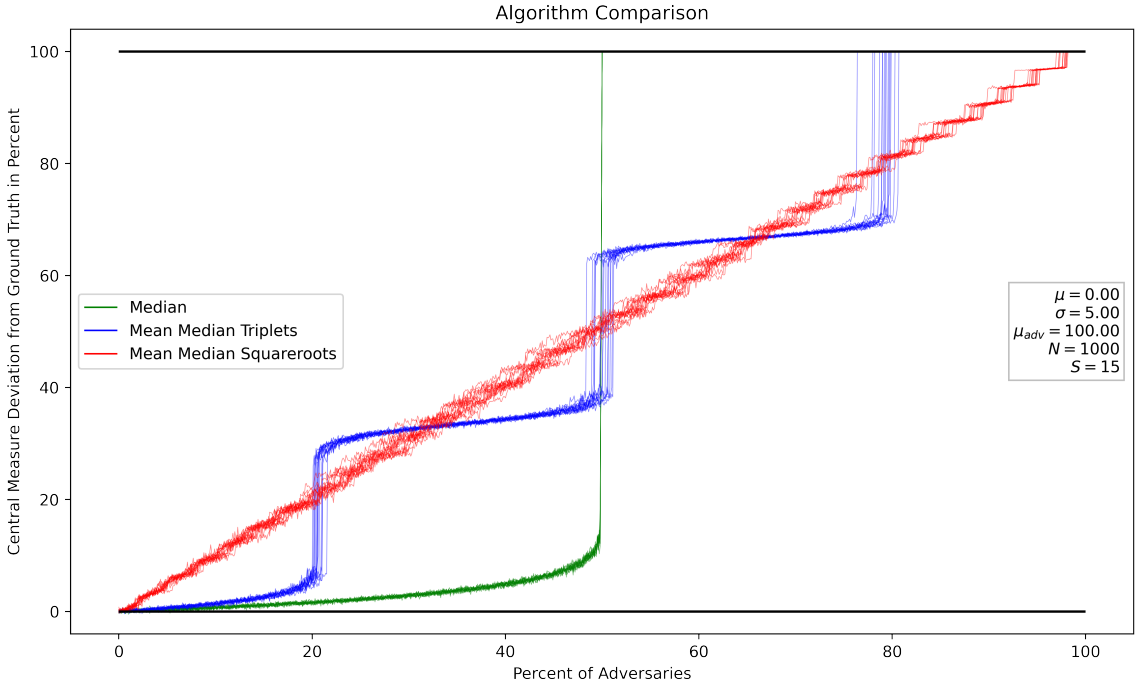


Figure 6. Characterisation of data consensus algorithms' behaviour under different degrees of coordinated data poisoning attacks

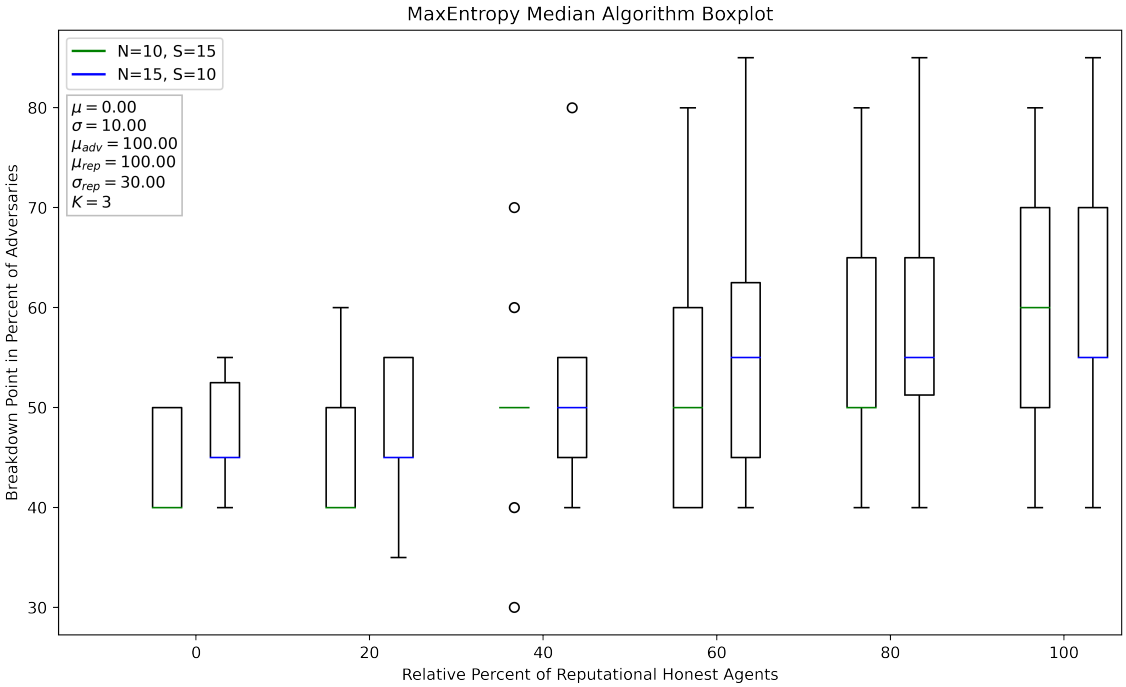


Figure 7. Characterisation of breakdown of MEV combined with Median Algorithm

convex functions and the equality constraints are affine functions.)

At time of writing, Algorand just announced that they are aware of the critique towards their voting algorithm. Passive agents with more wealth have more voting power than the active agents that are enabling the functioning of the network. Algorand have stated they agree with this critique and will be rolling out changes in June 2022 to reward active network users. https://algorand.foundation/news/governance-voting-update-g3

## References


[1] Adida, B. Helios: Web-based open-audit voting. In *USENIX security symposium* (2008), vol. 17, pp. 335–348.

[2] Agarwal, A., Dahleh, M., and Sarkar, T. A marketplace for data: An algorithmic solution. In *Proceedings of the 2019 ACM Conference on Economics and Computation* (2019), pp. 701–726.

[3] Andrews, L. Facebook is using you. *New York Times*.

[4] Aperjis, C., and Huberman, B. A. A market for unbiased private data: Paying individuals according to their privacy attitudes. *Available at SSRN 2046861* (2012).

[5] AppsFlyeA. The impact of ios 14+ & att on the mobile app economy.

[6] ApS, M. *MOSEK Optimizer API for Python 9.3.20*, 2019.

[7] Arrow, K. J. Social choice and individual values. In *Social Choice and Individual Values*. Yale university press, 2012.

[8] Arrow, K. J. *Economic welfare and the allocation of resources for invention*. Princeton University Press, 2015.

[9] Beikverdi, A., and Song, J. Trend of centralization in bitcoin's distributed network. In *2015 IEEE/ACIS 16th international conference on software engineering, artificial intelligence, networking and parallel/distributed computing (SNPD)* (2015), IEEE, pp. 1–6.

[10] Bell, F., Chirumamilla, R., Joshi, B. B., Lindstrom, B., Soni, R., and Videkar, S. Data sharing, data exchanges, and the snowflake data marketplace. In *Snowflake Essentials*. Springer, 2022, pp. 299–328.

[11] Bentov, I., Pass, R., and Shi, E. Snow white: Provably secure proofs of stake. *IACR Cryptol. ePrint Arch. 2016*, 919 (2016).

[12] Blais, A., and Massicotte, L. Electoral systems. *Comparing democracies 2* (1996), 40–69.

[13] Boeira, F., Asplund, M., and Barcellos, M. Decentralized proof of location in vehicular ad hoc networks. *Computer Communications 147* (2019), 98–110.

[14] Bornholdt, L., Reher, J., and Skwarek, V. Proof-of-location: A method for securing sensor-data-communication in a byzantine fault tolerant way. In *Mobile Communication - Technologies and Applications; 24. ITG-Symposium* (2019), pp. 1–6.

[15] Buchman, E. *Tendermint: Byzantine fault tolerance in the age of blockchains*. PhD thesis, University of Guelph, 2016.

[16] Cachin, C. Yet another visit to paxos. *IBM Research, Zurich, Switzerland, Tech. Rep. RZ3754* (2009).

[17] Castro, M., and Liskov, B. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS) 20*, 4 (2002), 398–461.

[18] Castro, M., Liskov, B., et al. Practical byzantine fault tolerance. In *OsDI* (1999), vol. 99, pp. 173–186.

[19] Christidis, K., and Devetsikiotis, M. Blockchains and smart contracts for the internet of things. *IEEE Access 4* (2016), 2292–2303.

[20] Courtney, J. C. Plurality-majority electoral systems: A review. *Electoral Insight 1*, 1 (1999), 7–11.

[21] Crain, T., Gramoli, V., Larrea, M., and Raynal, M. Dbft: Efficient leaderless byzantine consensus and its application to blockchains. In *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)* (2018), pp. 1–8.

[22] DECKER, C., SEIDEL, J., AND WATTENHOFER, R. Bitcoin meets strong consistency. In *Proceedings of the 17th International Conference on Distributed Computing and Networking* (2016), pp. 1–10.

[23] FELDMAN, P., AND MICALI, S. Optimal algorithms for byzantine agreement. In *Proceedings of the twentieth annual ACM symposium on Theory of computing* (1988), pp. 148–161.

[24] FOUNDATION, I. Consensus in the iota tangle — fpc, 2019.

[25] GHAFFARI, F., BERTIN, E., HATIN, J., AND CRESPI, N. Authentication and access control based on distributed ledger technology: A survey. *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)* (2020).

[26] GIBBARD, A. Manipulation of voting schemes: a general result. *Econometrica: journal of the Econometric Society* (1973), 587–601.

[27] GIBBARD, A. Manipulation of schemes that mix voting with chance. *Econometrica: Journal of the Econometric Society* (1977), 665–681.

[28] GILAD, Y., HEMO, R., MICALI, S., VLACHOS, G., AND ZELDOVICH, N. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th symposium on operating systems principles* (2017), pp. 51–68.

[29] HYNES, N., DAO, D., YAN, D., CHENG, R., AND SONG, D. A demonstration of sterling: a privacy-preserving data marketplace. *Proceedings of the VLDB Endowment 11*, 12 (2018), 2086–2089.

[30] ISAAK, J., AND HANNA, M. J. User data privacy: Facebook, cambridge analytica, and privacy protection. *Computer 51*, 8 (2018), 56–59.

[31] JIA, R., DAO, D., WANG, B., HUBIS, F. A., GUREL, N. M., LI, B., ZHANG, C., SPANOS, C., AND SONG, D. Efficient task specific data valuation for nearest neighbor algorithms. *Proceedings of the VLDB Endowment 12*, 11 (2018), 1610–1623.

[32] LAOUTARIS, N. Why online services should pay you for your data? the arguments for a human-centric data economy. *IEEE Internet Computing 23*, 5 (2019), 29–35.

[33] LASLIER, J.-F. And the loser is... plurality voting. In *Electoral systems*. Springer, 2012, pp. 327–351.

[34] LUO, W., AND HENGARTNER, U. Veriplace: A privacy-aware location proof architecture. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems* (New York, NY, USA, 2010), GIS '10, Association for Computing Machinery, p. 23–32.

[35] MACKAY;, R. S. D., AND MCLEAN, I. Probabilistic electoral methods, representative probability, and maximum entropy. *Voting matters* (2009).

[36] MANZANO KHARMAN, A. M., AND SMYTH, B. Is your vote truly secret? ballot secrecy iff ballot independence: Proving necessary conditions and analysing case studies, 2021.

[37] MCKINZIE, AND COMPANY. Monetizing car data: New service business opportunities to create new customer benefits, 2016.

[38] NARULA, N., VASQUEZ, W., AND VIRZA, M. zkledger: Privacy-preserving auditing for distributed ledgers. In *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)* (2018), pp. 65–80.

[39] OPENSECRETS. Expenditures breakdown, donald trump, 2016 cycle, 2016.

[40] OVERKO, R., ORDOPOEZ-HURTADO, R., ZHUK, S., FERRARO, P., CULLEN, A., AND SHORTEN, R. Spatial positioning token (SPToken) for smart mobility. *2019 8th IEEE International Conference on Connected Vehicles and Expo, ICCVE 2019 - Proceedings* (2019).

[41] POPOV, S., AND BUCHANAN, W. J. Fpc-bi: Fast probabilistic consensus within byzantine infrastructures. *Journal of Parallel and Distributed Computing 147* (2021), 77–86.

[42] RAMACHANDRAN, G. S., RADHAKRISHNAN, R., AND KRISHNAMACHARI, B. Towards a decentralized data marketplace for smart cities. In *2018 IEEE International Smart Cities Conference (ISC2)* (2018), IEEE, pp. 1–8.

[43] RASOULI, M., AND JORDAN, M. I. Data sharing markets. *arXiv preprint arXiv:2107.08630* (2021).

[44] RAYNAL, M. Communication and agreement abstractions for fault-tolerant asynchronous distributed systems. *Synthesis Lectures on Distributed Computing Theory 1*, 1 (2010), 1–273.

[45] SANCHEZ, L., ROSAS, E., AND HIDALGO, N. Crowdsourcing under attack: Detecting malicious behaviors in waze. In *IFIP International Conference on Trust Management* (2018), Springer, pp. 91–106.

[46] SCHUEFFEL, P. Alternative Distributed Ledger Technologies Blockchain vs. Tangle vs. Hashgraph - A High-Level Overview and Comparison -. *SSRN Electronic Journal* (2018), 1–8.

[47] SEN, A. Social choice. the new palgrave dictionary of economics, abstract & toc, 2008.

[48] SHAPLEY, L. *Contributions to the Theory of Games (AM-28), Volume II*. Princeton University Press, 1953.

[49] SMYTH, B. Ballot secrecy: Security definition, sufficient conditions, and analysis of Helios. *Journal of Computer Security 29*, 6 (2021), 551–611.

[50] STAHL, F., SCHOMM, F., AND VOSSEN, G. The data marketplace survey revisited. Tech. rep., ERCIS Working Paper, 2014.

[51] STAHL, F., SCHOMM, F., AND VOSSEN, G. Data marketplaces: An emerging species. In *DB&IS* (2014), pp. 145–158.

[52] STUCKE, M. E. Should we be concerned about data-opolies? *Geo. L. Tech. Rev. 2* (2017), 275.

[53] STÖRING, M. What eu legislation says about car data legal memorandum on connected vehicles and data, 2017.

[54] SZABO, N. The idea of smart contracts. https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_idea.html, 1997.

[55] TAHMASEBIAN, F., XIONG, L., SOTOODEH, M., AND SUNDERAM, V. Crowdsourcing under data poisoning attacks: A comparative study. In *IFIP Annual Conference on Data and Applications Security and Privacy* (2020), Springer, pp. 310–332.

[56] TOUEG, S. Randomized byzantine agreements. In *Proceedings of the third annual ACM symposium on Principles of distributed computing* (1984), pp. 163–178.

[57] TRAVIZANO, M., SARRAUTE, C., AJZENMAN, G., AND MINNONI, M. Wibson: A decentralized data marketplace. *arXiv preprint arXiv:1812.09966* (2018).

[58] UR REHMAN, I. Facebook-cambridge analytica data harvesting: What you need to know. *Library Philosophy and Practice* (2019), 1–11.

[59] VUKOLIĆ, M. The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In *International workshop on open problems in network security* (2015), Springer, pp. 112–125.

[60] WANG, W., HOANG, D. T., HU, P., XIONG, Z., NIYATO, D., WANG, P., WEN, Y., AND KIM, D. I. A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access 7* (2019), 22328–22370.

[61] WU, W., LIU, E., GONG, X., AND WANG, R. Blockchain based zero-knowledge proof of location in iot. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)* (2020), pp. 1–7.

[62] ZHANG, S. Who owns the data generated by your smart car. *Harv. JL & Tech. 32* (2018), 299.

[63] ZHAO, L., VIGNERI, L., CULLEN, A., SANDERS, W., FERRARO, P., AND SHORTEN, R. Secure Access Control for DAG-based Distributed Ledgers. *IEEE Internet of Things Journal* (2021), 1–15.

[64] ZHU, Z., AND CAO, G. Toward privacy preserving and collusion resistance in a location proof updating system. *IEEE Transactions on Mobile Computing 12*, 1 (2013), 51–64.

[65] ZWICKER W. S., M., HERVE (2016), B., FELIX; CONITZER, V. E., AND ULLE; LANG, J. Introduction to the theory of voting, 2016.

## APPENDIX

### A. Conic Optimisation and Lagrangian Relaxations

Relative entropy programs (REPs) and second-order cone programs (SOCPs) are conic optimisation problems in the relative entropy cones and second-order cones, possibly subject to other linear constraints. They could be solved via interior-point methods.

Let $\pi, \delta, \mathbf{1}$ be $|\mathcal{T}|$-dimensional vectors. The elements of $\pi$ are $\pi(t), t \in \mathcal{T}$, and $\mathbf{1}$ is an all-ones vector $\mathbf{1}$ of compatible size. A relative entropy cone $(\pi, \mathbf{1}, \delta) \in \mathcal{RE}$ is defined as:

$$\mathcal{RE} := \left\{ (\pi, \mathbf{1}, \delta) \in \mathbb{R}_{\geq 0}^{|\mathcal{T}|} \times \mathbb{R}_{\geq 0}^{|\mathcal{T}|} \times \mathbb{R}^{|\mathcal{T}|} | \quad \pi(t) \log(\pi(t)/1) \leq \delta_t, \forall t \in \mathcal{T} \right\}, \tag{15}$$

The objective function in (5) can be reformatted into (15). The relative entropy cone $(\pi, \mathbf{1}, \delta) \in \mathcal{RE}$ induces that $-\sum_{t\in\mathcal{T}} \pi(t) \log \pi(t) \geq -\sum_{t\in\mathcal{T}} \delta_t$ and we can just minimise $\sum_{t\in\mathcal{T}} \delta_t$ to obtain a maximum entropy solution. Hence, the Problem 5 is re-formulated as

$$\begin{aligned} \max_{\pi,\delta} & \sum_{t\in\mathcal{T}} \delta_t \\ \text{s.t.} & \sum_{t\in\mathcal{T}} \pi(t) \cdot S(t) = S(A), \quad \sum_{t\in\mathcal{T}} \pi(t) = 1 \\ & \pi(t) \geq 0 \ \forall t \in \mathcal{T}, \quad (\pi, \mathbf{1}, \delta) \in \mathcal{RE}. \end{aligned} \tag{16}$$

If $\mathcal{T}$ is the set of combinations, the constraint $\sum_{t\in\mathcal{T}} \pi(t) \cdot S(t) = S(A)$ in Problem 5 or 16 cannot always be satisfied. Correspondingly, we lift up this constraint to the objective function, with a multiplier $\lambda > 0$. Let

$$S^{\text{diff}} := \sum_{t\in\mathcal{T}} \pi(t) \cdot S(t) - S(A) \tag{17}$$

According to the definitions of $S(A), S(t)$, $S^{\text{diff}}$ is an $N \times N$ symmetric matrix, with its diagonal being all zeros. On the other hand, $S^{\text{diff}}$ implies the distortion of solution $\pi$ from RP property III.3.

Further, a second-order cone $(S^{\text{diff}}, \eta) \in \mathcal{SO}$ is defined as (18).

$$\mathcal{SO} := \left\{ (S^{\text{diff}}, \eta) \in \mathbb{R}_{\geq 0}^{N\times N} \times \mathbb{R}_{\geq 0} \left| \sqrt{\sum_{i,j\in A, i<j} 2 \left(S_{i,j}^{\text{diff}}\right)^2} \leq \eta \right. \right\}. \tag{18}$$

The Lagrangian relaxation of Problem 16, using second-order cone, reads

$$\begin{aligned} \max_{\pi,\delta,\eta} & \sum_{t\in\mathcal{T}} \delta_t + \lambda\eta \\ \text{s.t.} & \sum_{t\in\mathcal{T}} \pi(t) \cdot S(t) - S(A) = S^{\text{diff}}, \quad \sum_{t\in\mathcal{T}} \pi(t) = 1 \\ & \pi(t) \geq 0 \ \forall t \in \mathcal{T}, \quad (\pi, \mathbf{1}, \delta) \in \mathcal{RE}, \quad (S^{\text{diff}}, \eta) \in \mathcal{SO}. \end{aligned} \tag{19}$$

### B. Data Generation

### C. Measuring Entropy

Given a set of agents $A$ and the number of winners needed $M$, we can build two orderings sets: one of combinations $\mathcal{T}_{\text{com}}$ and the other one of permutations $\mathcal{T}_{\text{per}}$. Suppose an optimal probability measure is obtained from Problem 5 for each ordering set, denoted as $\pi^*_{\text{com}}$ for $\mathcal{T}_{\text{com}}$ and $\pi^*_{\text{per}}$ for $\mathcal{T}_{\text{per}}$, with the same input $S(A)$. See Figure 8 for an example when $A = \{a_i, a_j, a_k\}$ and $M = 1$.

Notice that for each element $t \in \mathcal{T}_{\text{com}}$, we can find $M!(N-M)!$ elements in $\mathcal{T}_{\text{per}}$ that are equivalent to $t$, in terms of the election results. We use $\sim$ to denote this equivalence relation. For instance, each row of Figure 8 displays a equivalent tuple of $t \in \mathcal{T}_{\text{com}}$ and $\tau \in \mathcal{T}_{\text{per}}$. Specifically, $t_1 \in \mathcal{T}_{\text{com}}$ is equivalent to $\tau_1, \tau_2 \in \mathcal{T}_{\text{per}}$, because their results are the same, i.e., only agent $a_i$ gets elected. Then, we have $t_1 \sim \tau_1 \sim \tau_2$.

| Combination Set $\mathcal{T}_{\text{com}}$ | | Permutation Set $\mathcal{T}_{\text{per}}$ | |
|---|---|---|---|
| $t_1$ | $a_i > (a_j, a_k)$ | $\tau_1$ | $a_i > a_j > a_k$ |
| | | $\tau_2$ | $a_i > a_k > a_j$ |
| $t_2$ | $a_j > (a_i, a_k)$ | $\tau_3$ | $a_j > a_i > a_k$ |
| | | $\tau_4$ | $a_j > a_k > a_i$ |
| $t_3$ | $a_k > (a_i, a_j)$ | $\tau_5$ | $a_k > a_i > a_j$ |
| | | $\tau_6$ | $a_k > a_j > a_i$ |

Figure 8. This table displays two ordering sets, i.e., combination set and permutation set, when $A = \{a_i, a_j, a_k\}$ and $M = 1$. All orderings, including combinations and permutations, in the same row are equivalent, in terms of election results.

To compare the entropy of $\pi^*_{\text{com}}$ and $\pi^*_{\text{per}}$, we suggest

$$\begin{aligned} \text{Entropy}(\pi^*_{\text{com}}) &:= \sum_{t \in \mathcal{T}_{\text{com}}} \pi^*_{\text{com}}(t) \log \pi^*_{\text{com}}(t) \\ \text{Entropy}(\pi^*_{\text{per}}) &:= \sum_{t \in \mathcal{T}_{\text{com}}} \left( \sum_{\tau \in \mathcal{T}_{\text{per}}, \tau \sim t} \pi^*_{\text{per}}(\tau) \right) \log \left( \sum_{\tau \in \mathcal{T}_{\text{per}}, \tau \sim t} \pi^*_{\text{per}}(\tau) \right) \end{aligned} \tag{20}$$

### D. Numeric Illustration

With $S(A)$ extracted from data generated in B, we have the following implementations:

- Problem 16 with input $\mathcal{T} = \mathcal{T}_{\text{per}}, S(t), S(A)$, denoted as "Permutation".
- Problem 19 with input $\mathcal{T} = \mathcal{T}_{\text{com}}, \lambda = 2, S(t), S(A)$, denoted as "Combination_Lag".

Both are solved by MOSEK Optimizer API for Python 9.3.20 [6]. Figure 9 displays the results of runtime, entropy in (20) and RP distortion $S^{\text{diff}}$ in (17), of optimal solutions $\pi^*_{\text{per}}$ and $\pi^*_{\text{com}}$, when the number of agents $N$ are $6, \ldots, 14$. Under each $N$, both implementations are conducted 6 times ($6 \times 2$ runs in total), with a new $S(A)$ generated every time. The average entropy, runtime and RP distortion of 6 runs are presented as solid curves for "Permutation" and dashed curves for "Combination_Lag".

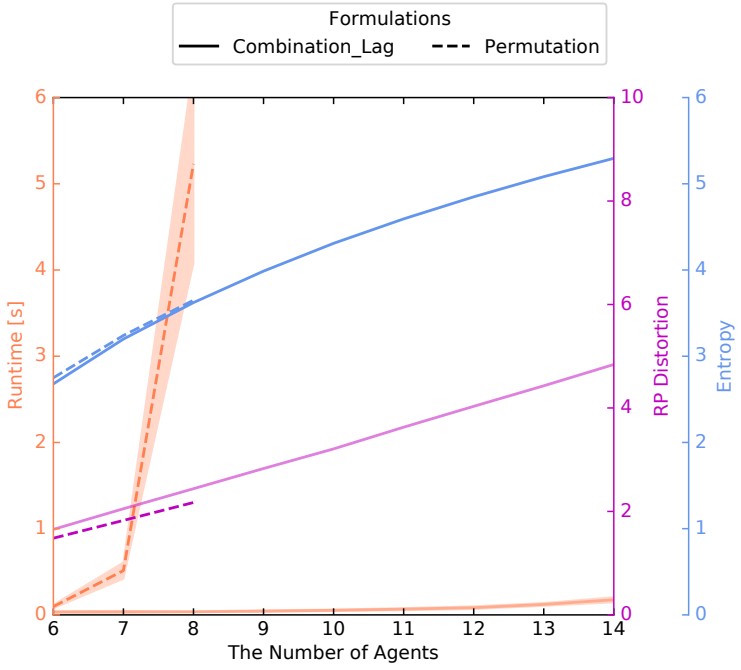


Figure 9. The average results of entropy, runtime and RP distortion, of implementing "Permutation" and "Combination_Lag"for 6 times, with the number of agents $N$ being $6, \ldots, 14$.