Consolidating Incentivization in Distributed Neural Network Training via Decentralized Autonomous Organization

Spyridon Nikolaidis*[1] and Ioannis Refanidis[1]
sp.nikola@uom.edu.gr , yrefanid@uom.edu.gr

[1] Department of Applied Informatics, University of Macedonia,
Thessaloniki 54636, GREECE

Abstract. Big data has reignited research interest in Machine Learning. Massive quantities of data are being generated regularly as a consequence of the development in the internet, social networks, and online sensors. Particularly Deep Neural Networks benefited greatly from this unprecedented data availability. Large models with millions of parameters are becoming common, and big data has been proven to be essential for their effective training. The scientific community has come up with a number of methods to create more accurate models, but most of these methods require high-performance infrastructure. There is also the issue of privacy, since anyone using leased processing power from a remote data center is putting their data in the hands of a third party. Studies on decentralized and non-binding methods among individuals with commodity hardware are scarce, though. Our work on LEARNAE seeks to respond to this challenge by creating a totally distributed and fault-tolerant framework of Artificial Neural Network training. In our recent work we demonstrated a method for incentivizing peers to participate to collaborative process, even if they are not interested in the neural network produced. For this, LEARNAE included a subsystem that rewards participants proportionately to their contribution using digital assets. In this article we add another important piece to the puzzle: A decentralized mechanism to mitigate the effect of bad-actors, such as nodes that attempt to exploit LEARNAE's network power without following the established rewarding rules. This is achieved by a novel reward mechanism, which takes into account the overall contribution of each node to the entire swarm. The network collaboratively builds a *contribution profile* for every participant, and the final rewards are dictated by these profiles. Taking for granted that the majority of the peers are benevolent, the whole process is tamper-proof, since it is implemented on blockchain and thus is protected by distributed consensus. All codebase is structured as a Decentralized Autonomous Organization, which allows LEARNAE to embed new features like digital asset locking, proposal submitting, and voting.

Keywords: Decentralized Neural Network Training, Distributed Ledger Technology, Smart Contracts, Decentralized Autonomous Organizations

1 Introduction

Deep Neural Networks (DNN) are getting increasingly complex. It is almost likely that if academics or enthusiasts attempt to implement new concepts on DNN, they will encounter a stumbling barrier where more development will require the availability of a massive quantity of training data and

computing power. Corporate data centers and large organizations are the only locations where these two resources may be found. As a result, individual practitioners are often barred from engaging in this kind of research. Also, there have been many cases in the past when such data-centers failed to guarantee data privacy. To address these important problems, we proposed LEARNAE [1][2][3], a novel ecosystem of interacting distributed technologies, that enable individual Machine Learning researchers to collaborate in a fully decentralized and democratized environment. The proposed concept removes the need for high-priced equipment and by using permissionless networks it allows individuals to maintain control over their sensitive data. There are many decentralized techniques for training Artificial Neural Networks. A closer study shows, however, that the concept of decentralization can be applied at a variety of scales: from pure peer-to-peer topologies to systems that utilize peers with enhanced role.

LEARNAE employs innovative Distributed Ledger Technology (DLT) as a data diffusion technique. A decentralized filesystem, IPFS [4], and a network architecture targeted at the Internet of Things, IOTA [5], are two novel technologies utilized in the current implementation of the coordinating algorithm. The use of gossip protocols to propagate all information offers excellent resilience, leaving no single point of failure. LEARNAE uses data parallelism [6][7], in which each worker keeps a local copy of the whole model and processes it with its own set of training data. After training, the generated models must be combined. So, during each averaging phase, all local model parameters are averaged with their corresponding ones from a remote model [8]. The model's accuracy is improved as a result of the additional stochasticity and, if privacy is not a concern, nodes may exchange training data using the same decentralized methodology. The collaborative training scheme was developed to cope with loosely connected topologies. There is no need for synchronization, and all data may stay accessible for the peers to consume at their own pace.

Our previous work focused on building the foundation for a new permissionless network in which individuals use commodity hardware to collaboratively develop improved neural models. LEARNAE's node hardware specifications range from lightweight IoT sensors to fully loaded desktop computers. The Internet of Things (IoT) sector has received special attention since the beginning of our research, due to its anticipated significance in the coming years. The incorporation of IoT devices is accomplished in a way that does not jeopardize the overall decentralized approach. One of the most essential elements of our design is resilience, thus eliminating the problem of unavailable resources. This is achieved by utilizing data duplication techniques.

The two core features of the proposed architecture are (a) the ability to cope with loosely connected peers, and (b) the incorporation of peers with diverse specifications. These features are ensured by the purely distributed and asynchronous nature of the scheme. So, there are no hard limits of prerequisites for a participant; having in mind that the level of contribution of each peer will be proportional to its (a) connection reliability, (b) processing power, and (c) training data supply. The proposed architecture can be applied in ML tasks that fit in the Map/Reduce logic, since the averaging phase can be replaced by any other way of reducing data among a pair of nodes.

This article introduces a major paradigm shift regarding the incentivization mechanism. All peers broadcast the help they get from others, and by doing so the network can build up a shared *contribution profile* for every peer. Before the collaborative training session, peers have to lock an amount of digital assets. The created fund is returned to the peers proportionally, according to their contribution profile.

The rest of the article is structured as follows: Section 2 presents the underlying Distributed Ledger Technology; Section 3 presents the related work; Section 4 briefly presents our previous work, whereas Section 5 proposes a novel way to mitigate the effects of bad actors. Section 6 presents the architecture of LEARNAE's Decentralized Autonomous Organization (DAO), Section 7 presents the results of the conducted experiments and, finally, Section 8 concludes the article and poses future research directions.

2 Underlying DLT Cornerstones

The concepts presented in this section serve as the foundation for the LEARNAE ecosystem. It makes no difference what platform is chosen, since the coordinating algorithm is platform-agnostic and can utilize any data-sharing method. Three new DLT projects were used for the current implementation to keep it decentralized: IPFS [4], IOTA [5], and Ethereum [18].

2.1 IPFS

IPFS is a file system that combines Git [9] with BitSwap[1], a method for incentivizing data replication based on BitTorrent [10]. It is a permissionless decentralized filesystem designed for peer-to-peer topologies.

[1] https://docs.ipfs.io/concepts/bitswap

Each node gets a unique hash string, which is used as identification when interacting with other nodes. The node also produces cryptographic hashes solely based on the contents of the submitted files. If the file is larger than a specific size, it is divided into chunks, each of which receives its own hash and is stored separately. Because IPFS does not address files based on their location, all files may be accessed just by knowing their unique hash. This is distinct from the majority of other filesystems. *Distributed Hash Table* (DHT) addressing is based on each node having its own copy of a ledger inspired by [11][12][13], which specifies the locations from which data chunks may be retrieved.

IPFS employs a simple, yet very effective incentivization system. Each peer maintains a list of chunks saved locally as well as a list of chunks that it seeks. It also maintains track of the number of confirmed bytes transmitted and received by each neighbor. This score operates as a credit system, showing how much each individual has contributed to the swarm's progress. Using gossip-style communication, the nodes try to acquire the data they need, while also attempting to improve their reputation with their neighbors, since doing so increases the likelihood that they will find chunks that they will require in the future.

IPFS is decentralized, so it does not need any authoritative entities. Because there is no server-based storage, each peer provides a fixed amount of local space. It is a node's duty to get and store files needed by other peers. The integrated garbage-collection system controls how long a file is available on a given node. Files deemed important by a peer may be *pinned* and therefore maintained on local storage indefinitely. Taking all of the aforementioned aspects into account, IPFS is considered a viable way for transmitting training-related information, as well as providing load balancing and resilience through data replication.

A collaborative training session requires sharing a large amount of relevant data. IPFS features a near-real-time publish-subscribe communication mechanism named PubSub. Nodes may create groups by joining *topics* in PubSub, where all messages are sent via gossip protocols. LEARNAE takes advantage of this feature to propagate crucial training information.

2.2 IOTA

The IOTA initiative's goal is to build a completely decentralized network that connects all IoT devices. According to projections[2], the number of connected IoT devices will reach 50 billion by

[2] https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology

2030. It is obvious that this will be followed by a substantial rise in the amount of data transmitted by such devices. In many cases, the devices are low-energy sensors that gather data from their environment.

To solve the scalability issues that most blockchains face, IOTA is based on a Directed Acyclic Graph (DAG), termed *Tangle*. Transactions are the Tangle's building blocks. In order for a node to attach a new transaction, it must first confirm two previous ones.

The Tangle supports feeless zero-value transactions that serve as data streams, called Masked Authenticated Messages (MAM). In previous work, we proposed a method for incorporating these streams into the LEARNAE scheme, to be utilized as a completely distributed way of collecting data from lightweight IoT devices. MAM streams are built as a single-linked list. A message stream is made up of numerous messages, each of which contains a reference to the next one. Anyone who knows the address of a certain message may only see the stream that follows that message, a feature known as *forward secrecy*.

2.3 Ethereum

Ethereum is a Blockchain [14][15], thus the first and most widely used type of DLT, consisting of a linked list of "blocks". Despite the fact that this description conforms to many other traditional data structures, blockchain provided a plethora of new concepts. Nowadays, there are numerous types, each with its own distinct features.

Essentially, a blockchain is a distributed ledger that stores snapshots of its database on multiple nodes. There is no need for a central authority to coordinate, and all peers contribute to the integrity of the system. The consensus is achieved via the use of well-established distributed algorithms [16], and when there is a dispute, the majority decides which version of the truth will be accepted. A new block is published on a regular basis, including information on changes to the blockchain that must be applied. Every block contains a cryptographic hash of the one before it to guarantee coherence. In this way, malicious block altering would be easily detected, since it would generate hash inconsistencies. Immutability is a key feature of blockchains; users may only add information and cannot alter any of the previous data. The only way to update the blockchain's status is to append the necessary changes.

Almost all blockchains include a native token [17], which is a digital asset that can be used in transactions. Every participant has a one-of-a-kind combination of a private key and a public address.

The blockchain consensus uses asymmetric cryptography methods to ensure that all token balances are consistent.

Ethereum was selected in this case mainly for the maturity of its codebase and the wide adoption it enjoys. We conducted our experiments on two Ethereum testing networks, Goerli[3] and Rinkeby[4], rather than Ethereum's main network, since these networks emulate all features of the MainNet, without requiring real financial cost.

The primary aim of the first-generation blockchains was to preserve token balances in a fully distributed and permissionless way. As a second-generation blockchain, the Ethereum Network provided additional functionality. Its node software includes a virtual computer, Ethereum Virtual Machine (EVM) [18], that can execute code snippets known as *Smart Contracts* [19][20]. Following code execution, the network's consensus mechanism evaluates the results. Smart Contracts may also store limited data and perform token transfers. Everyone has full access to the source code of a published Smart Contract [21].

3 Related work

According on how they deal with the two fundamental aspects of centralization and synchronicity, related research may be categorized into four broad areas: (a) centralized synchronous, (b) centralized asynchronous, (c) decentralized synchronous, and (d) decentralized asynchronous. Centralized solutions require a management entity, such as a parameter server or a node with elevated privileges. This frequently results in a communication bottleneck that necessitates high-speed networking, as well as a single point of failure. Synchronous techniques impose some type of time-based coordination amongst peers; there are solutions in which all nodes must function in strict parallel phases, and others in which they must share a clock. Locks and stale updates caused by sluggish workers afflict these solutions in many scenarios.

Regarding centralized synchronous systems, Sandblaster L-BFGS [22] and Parallel minibatch SGD [23] both use a parameter server and require a high-speed infrastructure. Parameter Server [24] introduces fault management by calculating redundancy and offers techniques to alleviate the effect of sluggish nodes. FireCaffe [25] is built on the Caffe [26] framework and needs high-speed networking. It also employs a unique MapReduce protocol. CaffeOnSpark [27] is based on the Spark framework and uses the DistBelief methodology. For a portion of the model, each peer also acts as a

[3] https://goerli.net
[4] https://www.rinkeby.io

parameter server. BigDL [28] mostly follows CaffeOnSpark's concepts, with the exception of the parameter exchange technique. It necessitates distinct training and data exchange cycles, resulting in a synchronous operation design.

In centralized asynchronous domain, a prominent solution is DistBelief [29], which employs a cluster of parameter servers and their peers. Each server-worker group is in charge of a different part of the model. The peers must download an updated copy of the joint model after each cycle. Project Adam [30] groups the parameter servers into a cluster and tries to reduce network traffic by transferring some of the processing from the workers to the servers. Elastic Averaging SGD (EASGD) [31] runs the optimizers on the nodes, which interact with a parameter server independently every N work cycles. TensorFlow [32] is the successor of DistBelief, with the addition of autonomous computation graph optimization, making distributed model parallelism much more feasible.

Regarding decentralized synchronous proposals, SparkNet [33] follows the approach of FireCaffe, while it attempts to adapt to low-bandwidth networks. It implements synchronous decentralized training. Thus, each worker runs a separate optimizer in isolation for N cycles. The resulting models are then reduced through averaging. Before the next computation cycle begins, the averaged model is broadcasted to all workers and replaces their local ones. In Decentralized Parallel Stochastic Gradient Descent (DPSGD) [34] the nodes are synced using a common clock and exchange parameters after every training cycle.

In decentralized asynchronous group, the EASGD algorithm is implemented in GoSGD [35], with the peers grouped in a mesh topology. Every Nth cycle, a randomized process selects the pairs of employees who will trade data [36]. On top of the Spark framework, DeepSpark [37] seeks to implement EASGD on commodity hardware. Asynchronous Decentralized Parallel Stochastic Gradient Descent (AD-PSGD) [38] uses a ring-based network architecture, with each worker choosing a neighbor for averaging after each iteration, and both workers replacing their local models with the averaged model.

4 LEARNAE Background

4.1 General Structure

Fig. 1 depicts the workflow of a LEARNAE Full Node. When a collaborative session begins, the peers who have training data divide it into dataslices of a predetermined size. These files are uploaded to IPFS individually, and their hashes are aggregated into a hashlist. This list is then added to the

network's shared DHT and its hash is broadcasted to the network (purple area). When a peer consumes a dataslice, it broadcasts a message to the others, notifying them that the specific dataslice has been used. LEARNAE members can establish a restriction on how many times a particular dataslice may be utilized for training throughout the network, by setting a parameter called *overuse threshold*.

When a peer receives a message about an available dataslice, it retrieves its pieces from several neighbors and adds them to the queue of locally available dataslices (red area); unless the slice has already surpassed the specified overuse threshold, in which case the peer ignores it.

Peers randomly rotate between training and averaging in every work cycle, increasing overall stochasticity. After each training phase, the peers upload the developed local model to IPFS and make it accessible by broadcasting its metadata (green area). When a peer selects averaging, it looks through the models on its local list. If it discovers a model with a claimed accuracy higher than the local's one, it obtains the remote model and averages it with the local one. If the averaged model improves local accuracy, the peer adopts it, adds it to the shared DHT, and broadcasts its metadata to the network. The broadcasted message contains information such as accuracy and model maturity, e.g., how many work cycles preceded its creation (blue area). As a final optimization step, when all averaging efforts have been completed and the network has attained maximum convergence, each peer downloads the remote model with the highest accuracy. This model is tested against the local dataset, and if it performs better, the peer adopts it as is. Detailed analysis of this process can be found in our previous work.

4.2 IoT section

The IOTA network is used to incorporate data from lightweight IoT devices. The Tangle is queried for new messages on a regular basis by participants who can train models. If they discover new data, they store it locally. Because of the Tangle's IoT-oriented design, it works best when the sent packages are not too big. The stored sensor data are maintained in a buffer until they exceed the predetermined dataslice size, at which point they are added to the list of dataslices ready for training; subsequently, the buffer is purged.

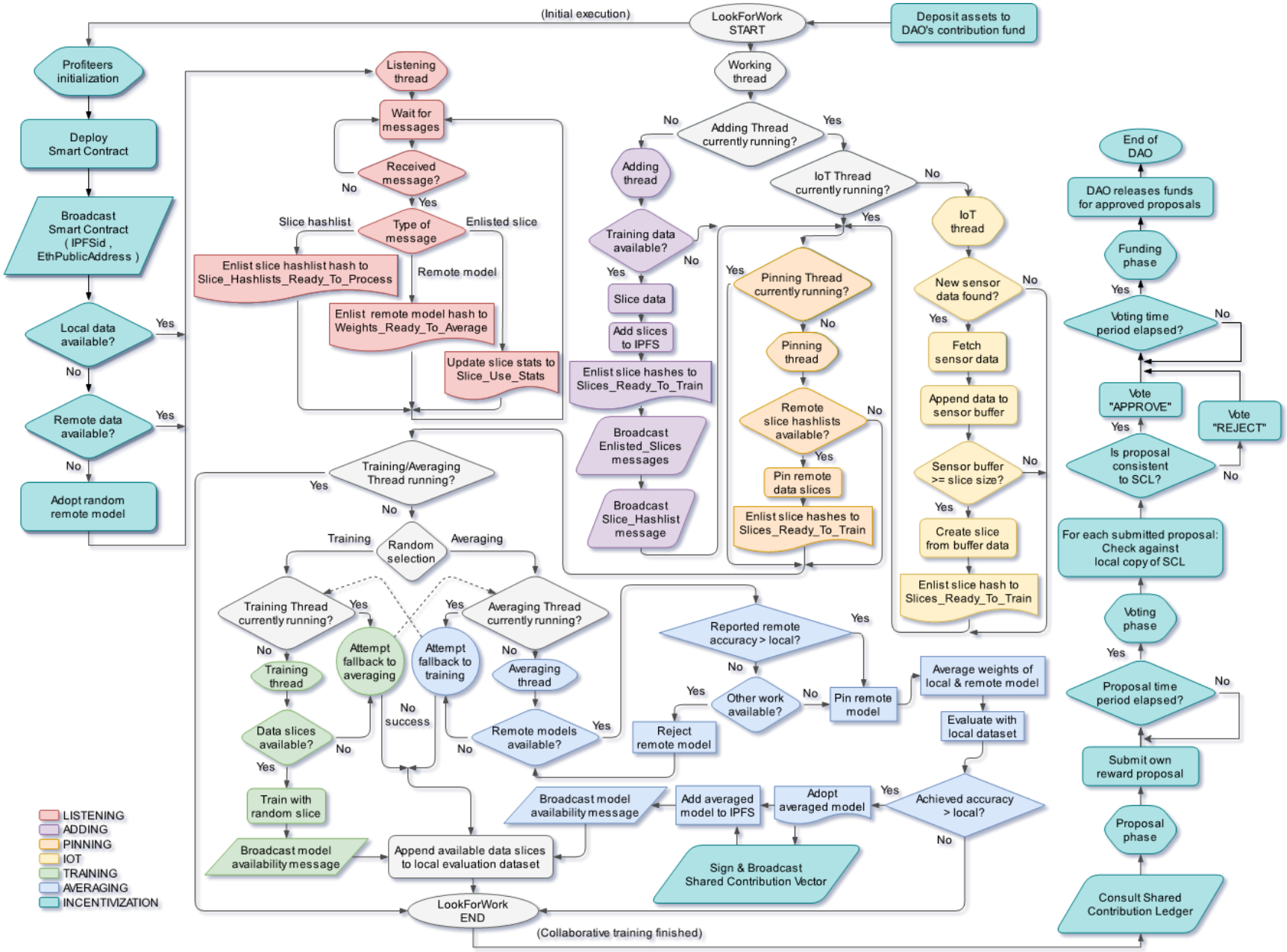

Fig. 1: Workflow of a LEARNAE full node (new incentivization section in teal color)

In addition to the contents, MAM messages may include a *tag* field. LEARNAE uses this tag as the connection between all sensors in a particular training session. For each sensor, a distinct MAM stream is generated. The initial message is labeled with a pre-agreed-upon codename (which is the same as the IPFS PubSub topic name) and includes the sensor ID as well as the date of its creation. This head message refers to the first data message, and so on.

Embedding an IoT sensor in a LEARNAE network starts with the inquiring peer asking the Tangle whether there are any MAM messages with the specified tag. If it discovers a new message (e.g., with a previously unknown address), it adds it to the list of known sensors. Each time the peer scans for new IoT data, it starts checking the linked list's tip of each known sensor. If it finds new data messages, it appends them to the local buffer and stores a pointer to that last message. Detailed analysis of this process can be found in our previous work.

4.3 Incentivization through direct micropayments

In our previous implementation we used a straightforward tipping mechanism: When a peer improves its local model by averaging it with a remote model, the incentivization algorithm delivers an Eth micropayment to the owner of the remote model. The amount of this payment is determined by a Reward Function depending on the degree of improvement in local model accuracy. To evaluate the whole process in our experiments, we utilized a simple proportional formula, as shown in Eq.1.

$$Payment = RewardFactor\,(AchievedAccuracy - CurrentAccuracy) \quad (1)$$

We acknowledged that this method is unable to deal with bad actors who refuse to pay the proper rewards. In this article we solve the issue by introducing a novel incentivization scheme, based on the concept of Decentralized Autonomous Organizations.

4.4 Distributed proof of identification

In order for the incentivization to work, peers must be able to verify their identities. A certificate authority would make this straightforward. However, in LEARNAE everything must be fully decentralized. We use Ethereum's Smart Contracts to build a Distributed Proof of Identity (DPoID).

To participate in a LEARNAE session, each participant must first deploy a Smart Contract to the Ethereum network. This contract has two data fields: *PoID* (Proof of Identification) and

Timestamp. The creator-peer passes one parameter (PoID) to the network, which is comprised by (a) its IPFSid, (b) its public Ethereum address of its digital wallet, and (c) its PublicKey. Internally, the constructor assigns the current date and time to the Timestamp field.

After deploying the Smart Contract, the peer broadcasts the contract's address to its neighbors, and all other nodes in the network execute the *getPoid()* and *getTimestamp()* methods to get the required data, which are kept locally in every node's storage.

A malicious actor could attempt to highjack payments by broadcasting a Smart Contract that contains its own Ethereum public wallet address but with the IPFSid of another –more active– peer. This effort would be detected and dismissed by its neighbors, due to the fact that the same IPFSid would be associated with two distinct Ethereum public addresses. In such cases, The Smart Contract with the later timestamp is immediately rejected as fraudulent. Detailed analysis of this process can be found in our previous work.

5 The concept of Decentralized Governance

5.1 Background Theory

A novel technological development has recently entered the domain of public organizations, thanks to algorithmic systems [39]. The thing which defines these systems is Machine Learning, which provides new ways to discover knowledge [40]. Machine Learning applications use huge datasets and statistical methods to infer connections that are sometimes hidden or not obvious. These algorithms have the potential to offer significant new insights, therefore they are regarded as a very strong tool for governance purposes [41][42].

Decision-making based on algorithmic analysis is grounded on a completely different logic than traditional bureaucracy. Even though uncertainty is inherent in many decision-making processes, algorithmic systems allow governance structures to use data analysis to quantify the uncertainty and state the information as probabilities, to better rationalize the process. This method could offer, for example, a powerful optimization tool to classify cases based on whether they should be the subject of further investigation. Thus, algorithmic systems are a major shift in how an organization is structured and they have the potential to fundamentally affect how governance is achieved. The term *Algocracy* introduces the concept of exerting governance using algorithms. It is the evolution step after *Machine Bureaucracy* and *Infocracy*.

Machine Bureaucracy is an administrative body that is defined by a distinct set of guidelines for getting things done [43]. Work procedures in this context are quite organized and rule-based. A single, standardized way of doing things dominates the Machine Bureaucracy, since every task must be done in a very rigid manner. Hierarchization and formalization are both high, as is centralization, with the buildup of decision-making capacity at the top of the organization. The underlying technological infrastructure is the pivotal element of the organization. Because it is in charge of standardization, it can thus be said that it is accountable for that implementation.

Infocracy is a form of organization, much like Machine Bureaucracy. The standardization of work is programmed into the technology utilized by the organization, eliminating the need for participants to learn rules that would improve uniformity [44]. Work standardization is extremely prevalent, but all processes are set in motion by information systems following preset laws and regulations [45][46]. This solidifies the technological structure's important role inside the organization. Work is supported by information systems, which have created a fine-grained division of roles with plenty of vertical control. Decisions at a lower organizational level may be taken at some extend, sometimes with moderate decentralization. In addition, hierarchical and formalized organizational structures are decreased, as compared to Machine Bureaucracy.

In Algocratic systems, non-routine labor may be done by using sophisticated technology. Algorithms for data mining, pattern recognition, and prediction are all built using Machine Learning [47][48]. Inherent uncertainty is measured and minimized by the algorithmic system's data analysis [49].

Algocracy suggests that sophisticated algorithms have the ability to push technology into decision-making domains, by converting human judgement into standardized processes [50]. The algorithm is to be used in both a unidirectional and bidirectional fashion, contrary to Machine Bureaucracy and Infocracy, where work control is unidirectionally administered via the organization and information infrastructure, respectively.

It is noteworthy that many of these algorithms depend on their developer, e.g., their judgments, perceptions, and opinions, which may have an impact on the learning process. Thus, the programmers may predetermine and guide the Machine Learning algorithm, and in that way influence the decision-making results given by the system.

5.2 Decentralized Autonomous Organizations

A Decentralized Autonomous Organization (DAO) [51][52] is defined as a structure controlled by encoded laws. Its members exert control through the transparency of these computerized governance rules, and do not rely on central control. Transactions and regulations for a DAO are stored on a blockchain [53][54][55]. The consensus mechanism of the blockchain ensures the proper function of DAO, a critical feature since so far there is no concrete legal framework that covers this novel kind of organization [56]. The benefits gained by the use of blockchain technology include fault-tolerant distributed database, cryptography-based identification and permissionless timestamping. When someone uses this method, they no longer have to rely on a trustworthy third party in their transactions, making things easier and more straightforward.

The costs of a blockchain-based transaction, and the associated data reporting, may be significantly lower than traditional methods, due to the elimination of the need for multiple and independent bureaucratic records and third-party fees that are typically charged in conventional mechanisms. Thus, blockchain data might, at least in principle, replace public papers like deeds, titles and contracts, so long as regulations allow it. Once a DAO launches, it could be structured to operate autonomously, with Smart Contracts managed by a Turing-complete platform that would maintain full-scale administrative support [57][58]. Decentralized self-government organizations aspire to be open platforms where people control their relationships, their identity and their personal data [59].

In the Ethereum blockchain the *Solidity* programming language is what is used to create DAO code. This code can be executed by creating on-chain Ether transactions. Ether, the digital fuel that runs the Ethereum network, is the base for all applications that leverage its blockchain. In order to start functioning, a DAO needs Ether in its account, and thus, its priority is to obtain it. During the creation phase the code is released, and the system allows Ether to be transferred to the DAO's Smart Contract address.

In order to compensate senders of Ether, a DAO generates tokens and assigns them to the senders of the Ether. The tokens provide to the participants the ability to vote and be part-owners. The token creation rate is dependent on the transferred Ether. There are no transfer fees for moving tokens around after the Genesis phase has concluded.

When deployed, the settings for the *Minimum DAO Creation Objective* and *Creation Phase Time-period* are given as arguments to the code. In case the total of DAO Creation Objective does not meet the minimum before the end of the creation phase, all of the Ether will be refunded. After the Creation

Phase has ended, the total Ether raised is denoted by Ξ_{raised} and the total amount of tokens created is denoted by T_{total}. Essentially, DAO is a structure that holds Ether and other Ethereum-based tokens, and transfers them according to the organization's code.

An individual who owns a DAO token may ask for DAO funds (denoted $\Xi_{transfer}$) by proposing a contract. If the proposal is accepted by the majority of the voting power, the DAO sends the requested Ether to a smart contract representing the proposed project. The process of selecting a contract can be enhanced with advanced features, such as collaboration with other DAOs and fetching data from external sources called *Oracles*.

The number of votes someone has is proportional to the number of tokens they own. Each proposal has an allotted amount of time for discussion and a vote. Once a proposal has been approved, token holders will be able to execute a DAO contract function which verifies that (i) the majority of votes were in favor of the proposal, (ii) the minimum quorum percentage was met, and (iii) the proposal has been approved. The DAO will fund the proposal if it has been approved, otherwise the proposal will be closed.

A token holder has a say if they have at least one of the tokens in the network. The minimum number of tokens a person must have to be able to influence a decision is denoted by q_{min}. An example of how some of the most popular DAO calculate q_{min} is shown in Eq.2

$$q_{min} = \frac{T_{total}}{d} + \frac{\Xi_{transfer} \cdot T_{total}}{3 \cdot (\Xi_{DAO} + R_{DAO})} \qquad (2)$$

where d is the $minQuorumDivisor$, Ξ_{DAO} is the amount of Ether owned by a DAO and R_{DAO} is the amount of reward tokens owned by this DAO. The sum $(\Xi_{DAO} + R_{DAO})$ is equal to the amount of Ether used to create DAO tokens plus the rewards received or, said another way, the total amount of Ether a DAO has ever received.

The above formula means that, initially and for $d = 5$, a quorum of 20% of all tokens is required for any proposal to pass. In the event $\Xi_{transfer}$ equals the amount of Ether a DAO has ever received, then a quorum of 53.33% is required.

To prevent proposal spam, a deposit is needed in order to have a proposal reviewed; the deposit will be returned if the majority approves the plan. The DAO will retain the deposit if a quorum is not met. The required deposit amount may be modified by the DAO in a later proposal.

The DAO as a decentralized entity cannot be manipulated by any outside influences. Because it is open-source, the organization and all of its code are visible and therefore impossible to corrupt, since all program functions are managed on the blockchain.

Stakeholders must have complete consensus on every choice they need to make, such if one member wishes to pull out their money. Bugs and other problems requiring democracy in the decision-making process are also a concern, and all can be addressed by the DAO's rules.

From a technological standpoint, a DAO is made up of one or more Smart Contracts which are executed on the Ethereum blockchain using its distributed consensus mechanism. Ethereum offers a blockchain with a built-in Turing-complete programming language, where users may design and implement applications on their own terms. Smart Contract transaction costs are paid using Ethereum's currency Ether. Fig. 2 shows the theoretical layers of a Decentralized Autonomous Organization.

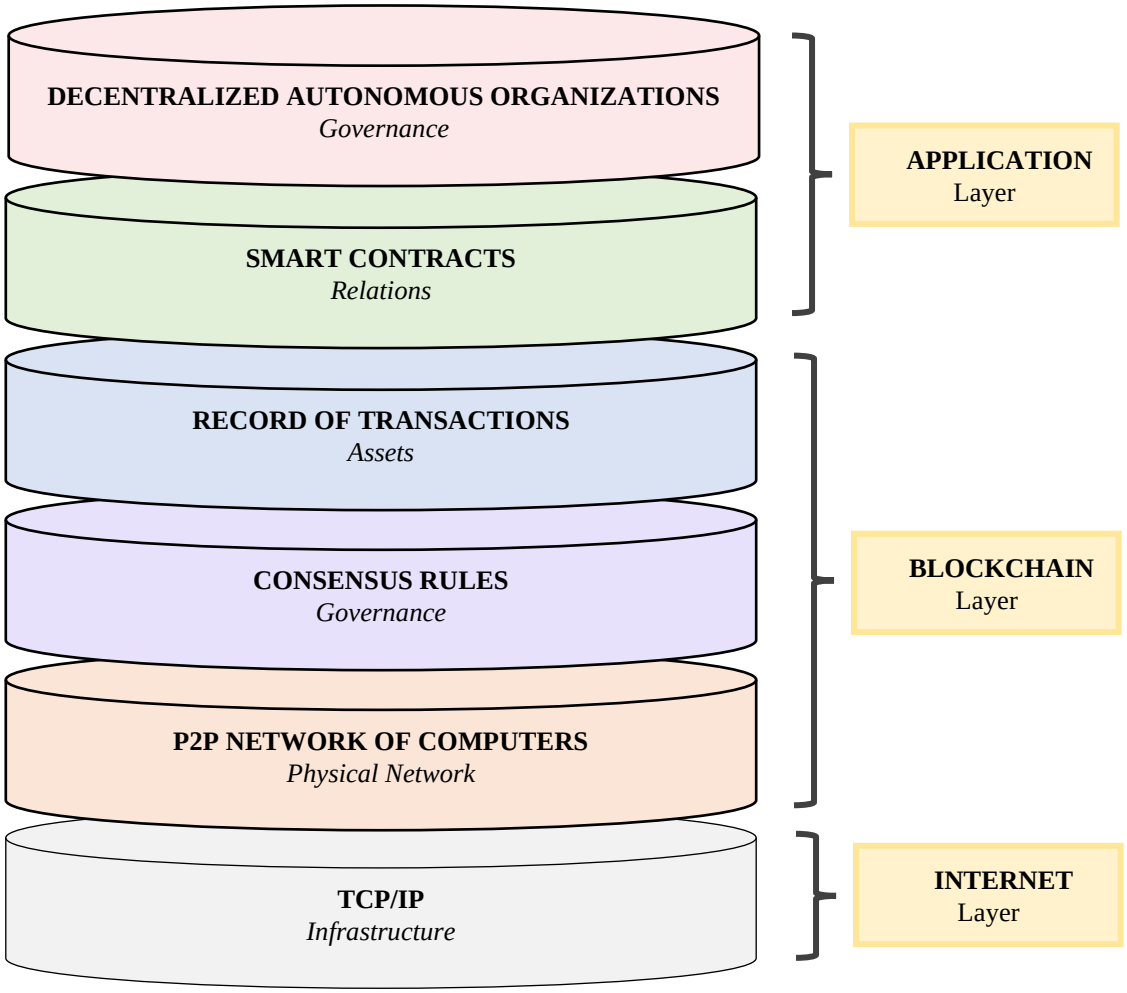

Fig. 2: Architectural layers of DAO

The data structure of a single proposal depends on the characteristics of each use case. For a typical DAO, a proposal could have the following parameters:

Table 1: Typical parameters of a DAO proposal

Parameter	Description
$recipient$	The address where the amount of assets will go to if the proposal is accepted.
$amount$	The amount of assets to transfer to recipient if the proposal is accepted.
$description$	A plain text description of the proposal.
$votingDeadline$	A Unix timestamp, denoting the end of the voting period.
$open$	A Boolean which is false if the votes have already been counted, true otherwise.
$proposalPassed$	A Boolean which is true if a quorum has been achieved with the majority approving the proposal.
$proposalHash$	A hash to check validity of a proposal. Equal to $sha3(recipient, amount, transactionData)$.
$proposalDeposit$	The deposit the creator of a proposal has send to submit a proposal. It is taken from the $msg.value$ of a $newProposal$ call;
yea	Number of tokens in favor of the proposal.
nay	Number of tokens opposed to the proposal.
$votedYes$	Simple mapping to check if a token holder has voted for it.
$votedNo$	Simple mapping to check if a token holder has voted against it.
$creator$	The address of the token holder that created the proposal.

6 LEARNAE's Decentralized Autonomous Organization

The first stage of a LEARNAE training session is the establishment of its Decentralized Autonomous Organization. During this period all participants have to deposit a predefined amount of digital assets. These assets are immediately locked and constitute the DAO's Contribution Fund. Each deposit contains the participant's Ethereum public address. This information, in conjunction with the data contained in the DPoID Smart Contracts, are used to link a deposit to an IPFS ID. Since voting is not weighted, the DAO sends in return a single LEARNAE token to each member. Owning this token grants the right to vote and submit proposals.

When the period for fund raising expires, the network starts the collaborative training. Every time a peer improves its local model by averaging with a remote one, the peer updates, signs with its private key, and broadcasts its own *Shared Contribution Vector* (SCV), a record that contains information on how much help a peer received, in its effort to improve local neural model (Fig. 3).

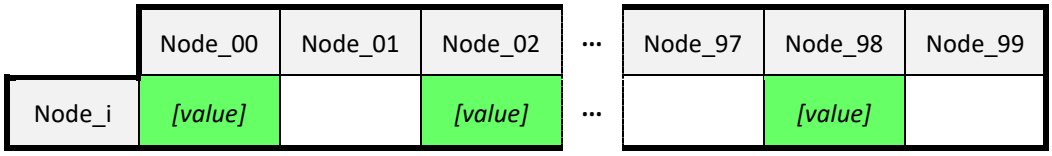

Fig. 3: Shared Contribution Vector of Node #i *(assuming 100 participants)*

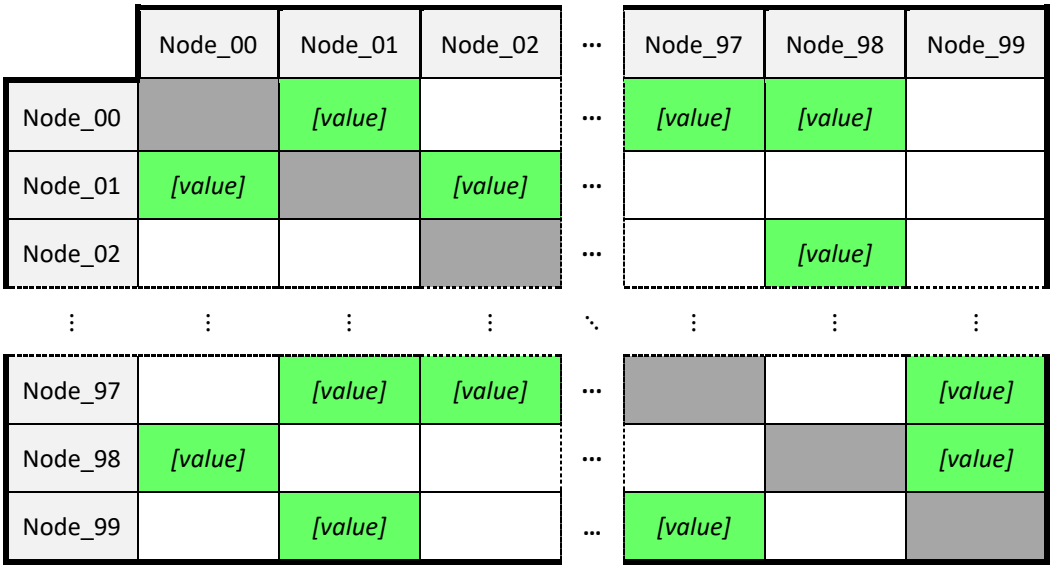

Fig. 4: Shared Contribution Ledger *(assuming 100 participants)*

Peers gather broadcasted SCV messages and use them to construct their local copy of Shared Contribution Ledger (SCL). SCL is an array comprised by all known SCV records and is used to determine a general view of the contribution level throughout the network (Fig. 4). Thus, the value in row X and column Y of the SCL represents the help offered to peer X by peer Y.

When training phase concludes, participants submit their proposals regarding the amount of digital assets they claim as reward for their contribution. Peers review the submitted proposals by comparing them to their local copy of SCL. They vote in favor of every consistent proposal and against of all others. When voting period ends, the DAO automatically releases the whole amount of available funds, distributed proportionally to all approved proposals (Fig. 5). The procedure takes place on the blockchain and requires no central coordination.

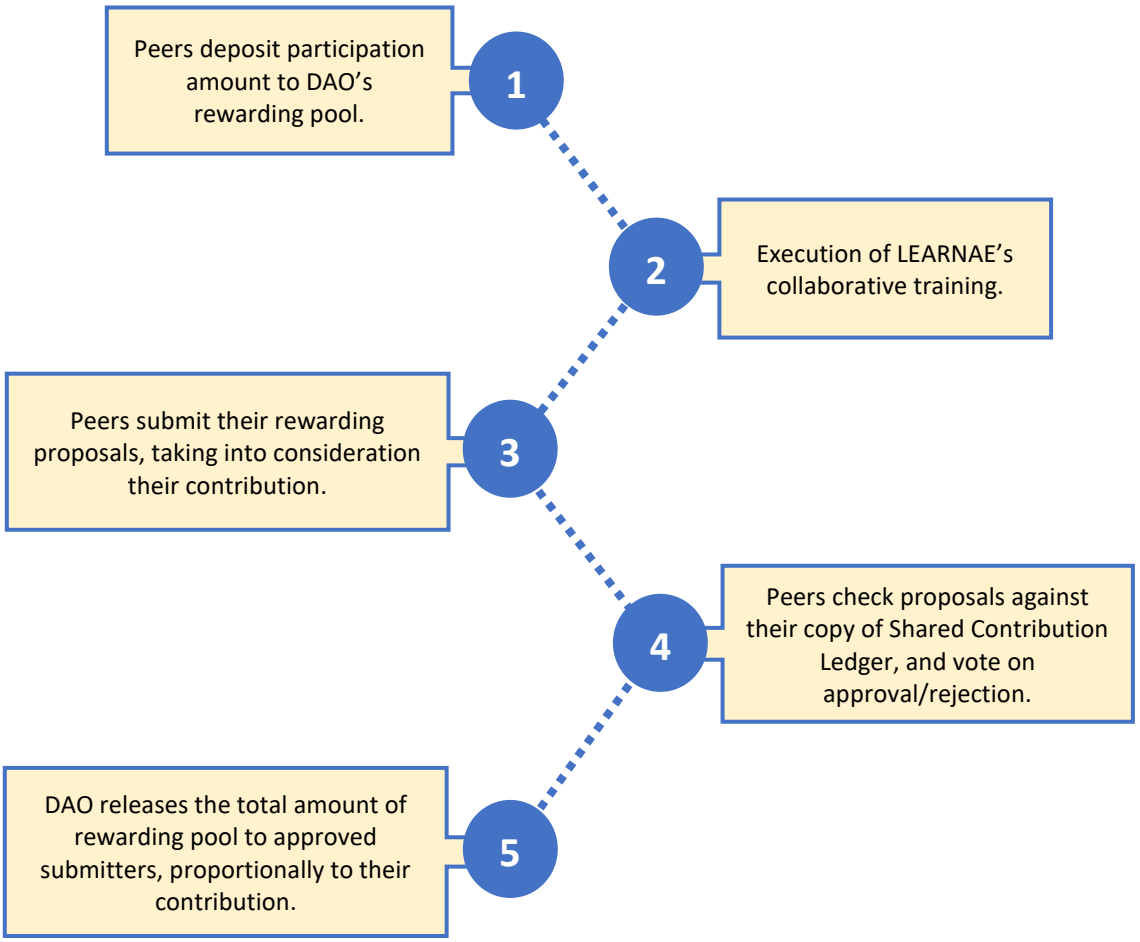

Fig. 5: Stages of LEARNAE's DAO

The new DAO-based incentivization mechanism is a major paradigm shift compared to the tipping method of our previous implementation. Although tipping was a more direct way for a peer to reward the neighbors that actually offered help to that peer, it could not ensure the compliance of the participants. Thus, there could be cases where malicious actors might tamper the procedure to avoid paying. The new implementation is more community-oriented: Peers reward neighbors according to their overall assistance to the swarm. All participants have to declare their engagement to the process, by depositing digital assets to DAO's Contribution Fund. Depending on their assistance during the training phase, peers can end up with less or more assets in their balance, compared to their initial deposit amount. Which one it will be, it depends on their "give help" / "get help" equilibrium.

7 Experimental results

To provide a proof-of-concept for the proposed incentivization architecture, we construct and evaluate the results of an algorithm simulating the collaborative neural network training. In these experiments the LEARNAE swarm consists of 100 participants. The session is comprised of 50 cycles; during each cycle peers attempt to average their model with a remote one. The result of this attempt may be successful or not. The level of change to the local model's accuracy is expressed as a random number (range: *Uniform*[-10..10]), where negative values indicate unsuccessful attempts. To simulate the discrepancy in node's hardware performance, every peer is uniformly assigned a random Contribution Factor (range: *Uniform*[1..9]). The final level of change occurred by an averaging attempt is calculated as:

$$
\begin{aligned}
&ModelChange \;=\; Uniform[\,-10\,,10\,] \\
&if\;(\,ModelChange \;>\; 0\,)\;then \\
&\quad ModelChange = NeighborContributionFactor \times ModelChange
\end{aligned}
$$

The value of $ModelChange$ is also used as the amount of the reward given for this successful averaging. During the initial phase, all participants have to deposit 1000 credit units to DAO's Contribution Fund.

To assess the efficiency of our incentivization proposal on mitigating the impact of malicious nodes, we conducted experiments for both cases, (a) no malicious actors, and (b) 10% of participants

being malicious. In the following experiments, the period during which DAO adapts the peer rewards in order to comply with the new incentivization scheme, is denoted as *Consolidation Phase*.

7.1 Network with no malicious nodes

Fig. 6 demonstrates the balance of each peer during a session with no malicious actors. This means that when a peer achieves a successful averaging, willingly broadcasts the reporting message to the swarm, to let everyone know that a neighbor's help resulted to a specific level of model improvement. The message that is broadcasted is essentially the updated version of its own SCV, signed with the node's private key. The lines indicate how a peer's balance is progressing from this peer's point of view.

During Consolidation Phase, the DAO applies the corrections introduced with our new incentivization algorithm. Thus, every peer rewards its neighbors for their overall contribution to the LEARNAE swarm. As expected, this paradigm shift results to different final balances compared to simple tipping method. Fig. 7 depicts these differences. Out of 100 nodes, 47 ended up with increased balance at an average value of $+241.84$ credit units; 53 nodes ended up with decreased balance at an average value of -214.46 credit units.

7.2 Network with 10% malicious nodes

Fig. 8 demonstrates the progress of every peer's balance during a collaborative session with 10 malicious nodes. For these experiments we define a "malicious" actor as one who refuses to broadcast its updated SCV, in an attempt to avoid paying rewards to others. The malicious peers are denoted with thick dashed lines. As shown in the chart, if we just use our previous tipping method, the balance of the malicious nodes is only increasing, since, although they avoid paying others, they still get rewarded for their contribution.

The DAO's Consolidation Phase is eliminating this malicious effort; dishonest peers ultimately get rewarded not based on their tampered version of truth, but rather according to the evaluation they got from the entire network. The correction is visually expressed by the uniform distribution of malicious balances after the Consolidation Phase.

Fig. 9 depicts the balance differences between the two rewarding schemes. Out of 100 nodes, 51 ended up with increased balance at an average value of $+272.59$ credit units; 49 nodes ended up with decreased balance at an average value of -283.71 credit units.

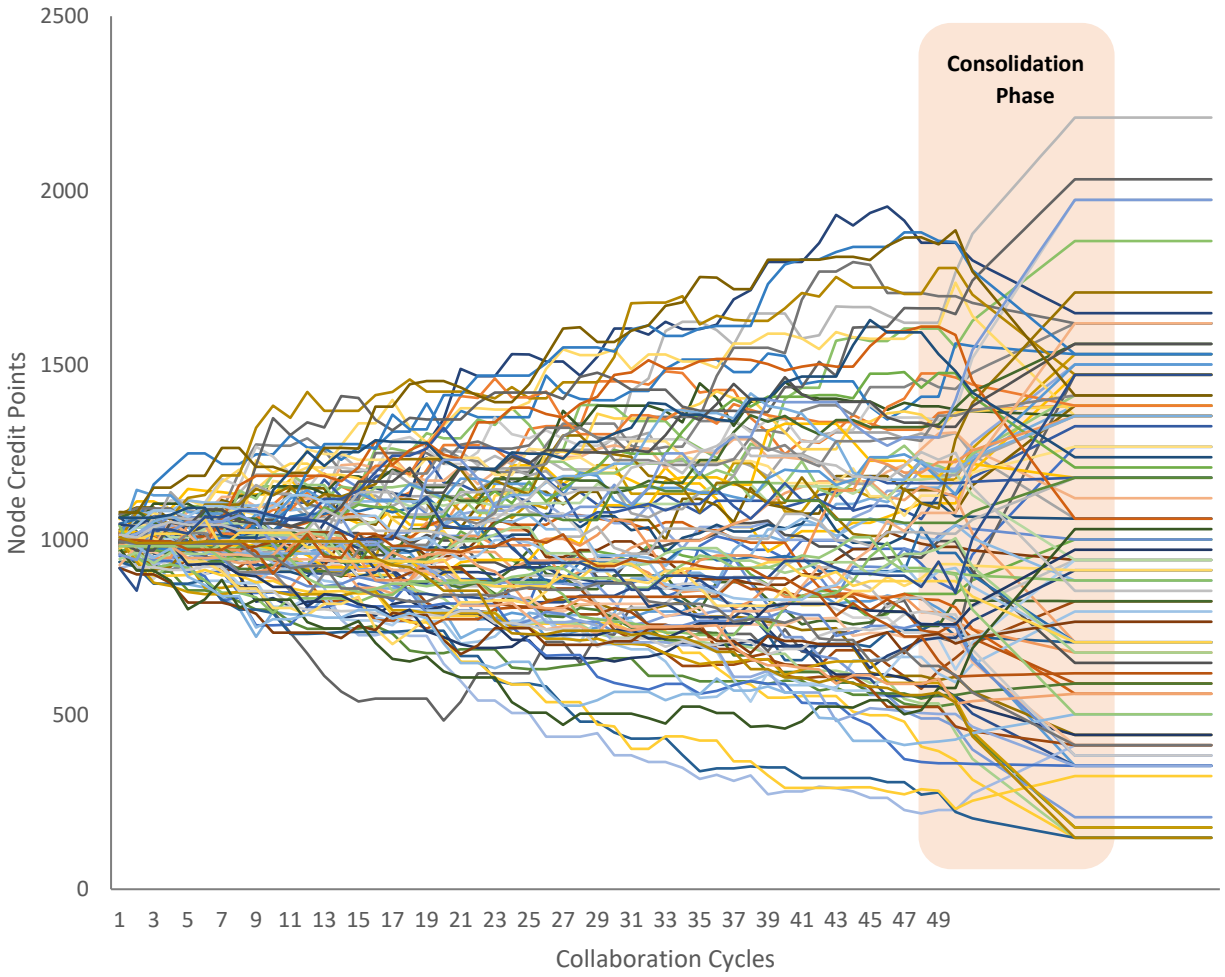

Fig. 6: Progress of credit balances in case of no malicious peers

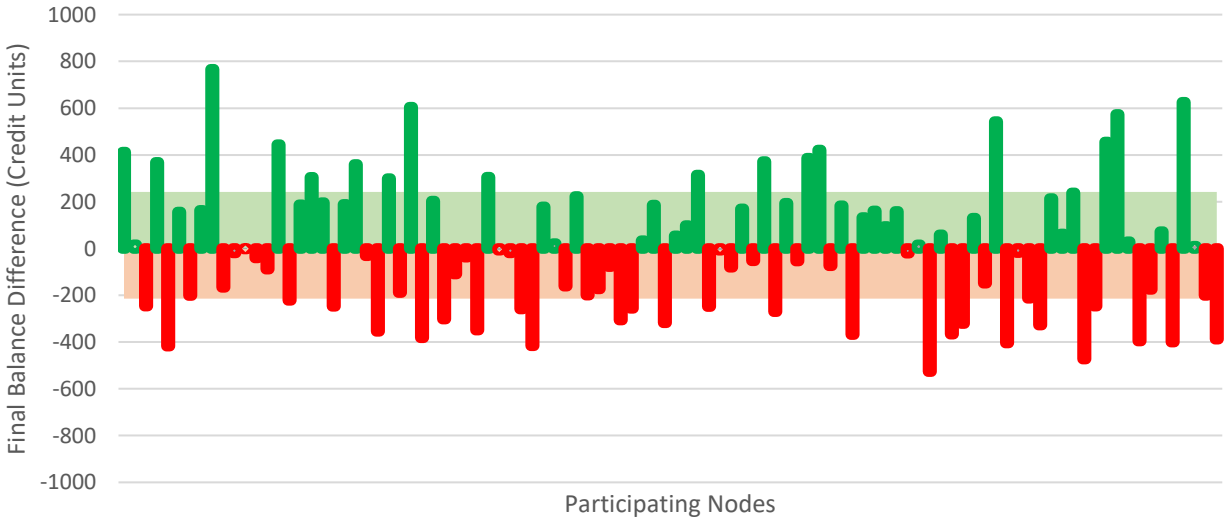

Fig. 7: Correction in credit balances in case of no malicious peers

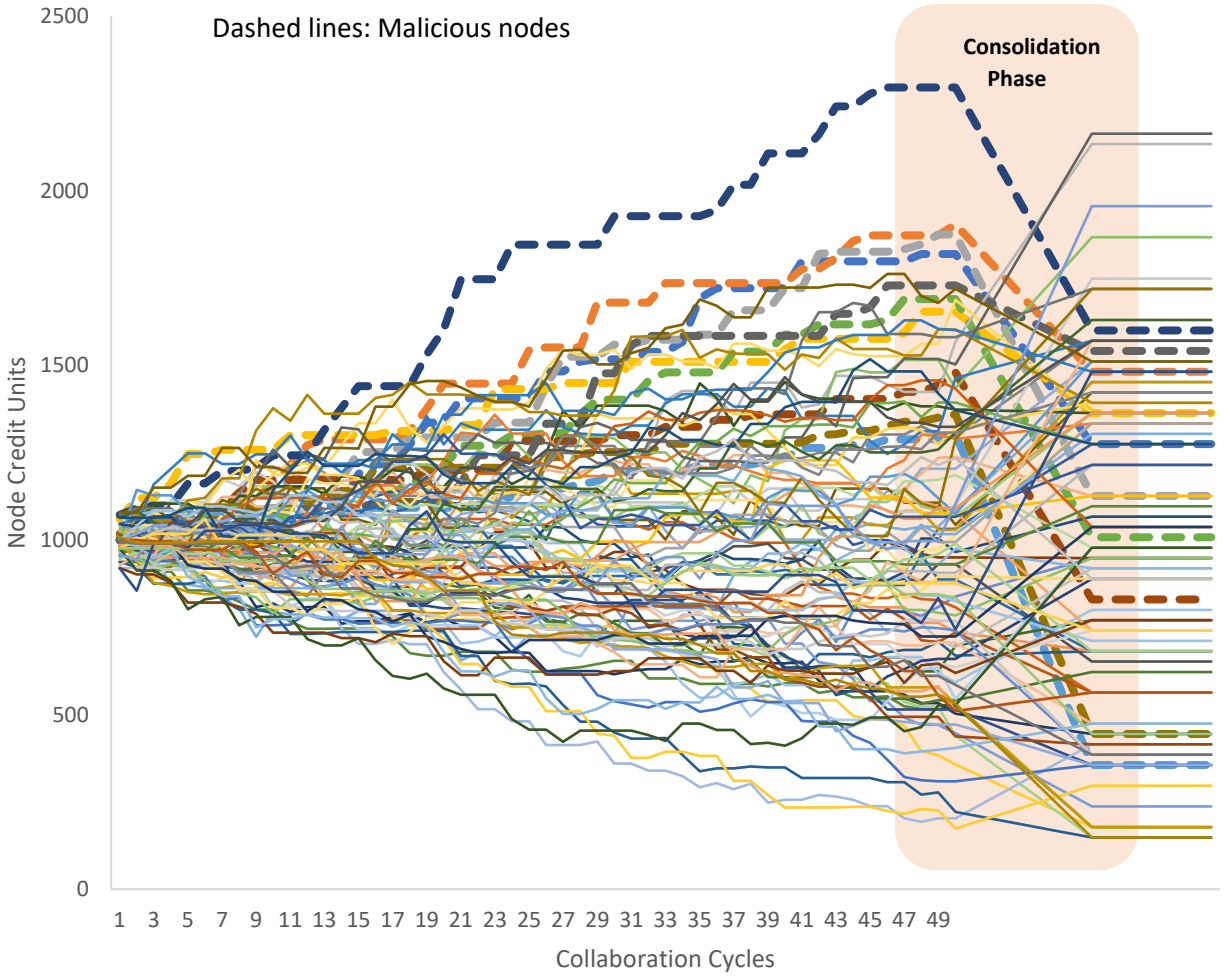

Fig. 8: Progress of credit balances in case of 10% malicious peers

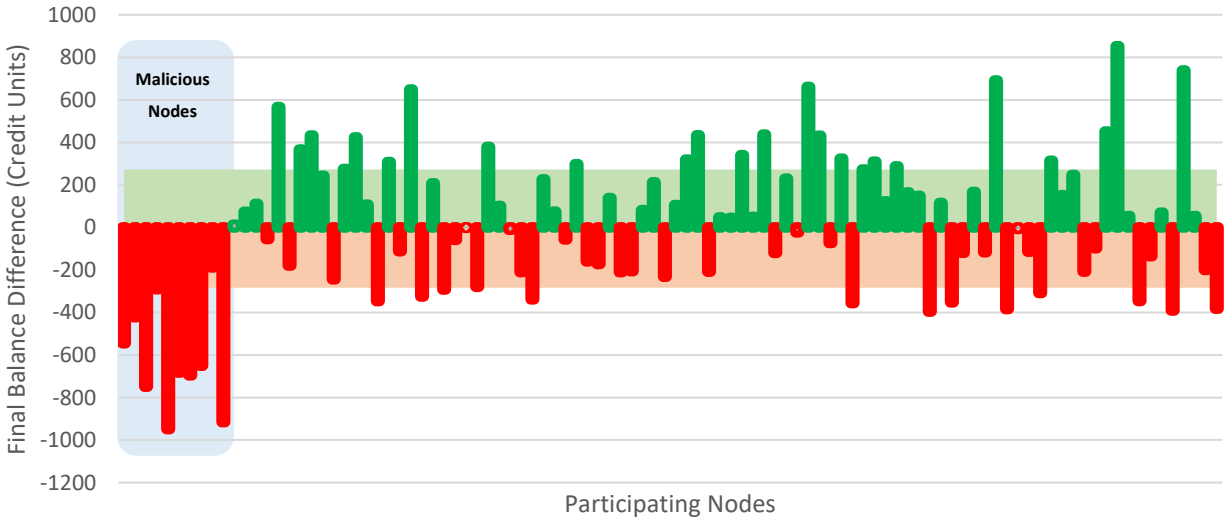

Fig. 9: Correction in credit balances in case of 10% malicious peers

Regarding malicious actors, the achieved correction leads to a rough decline in their final balance. The first 10 nodes being malicious, experience a substantial decrease in their final balance. The average decrease for the malicious nodes is -608.56 credit units, while the average value for all other reductions is -200.42 credit units.

7.3 Discussion

To evaluate our proposal's efficiency on mitigating the effects of malicious actors who attempt to abuse the incentivization mechanism, we conducted two distinct experiments: (i) a session with no malicious actors and (ii) a session where 10% of the participants denied to report the help they enjoyed from their neighbors, in order to evade giving the appropriate rewards. Although most node metrics were randomly generated for the simulations, in both cases we used the same randomization seed; thus, the contribution of all peers was identical in both experiments. This allows us to quantify the correction achieved by the Consolidation Phase. As shown in Fig. 10, the mean discrepancy of final balances between the two experiments was 3.2%. These results suggest that the effect of a significant portion of malicious peers was successfully eliminated at the expense of a minor decline in the final reward amount.

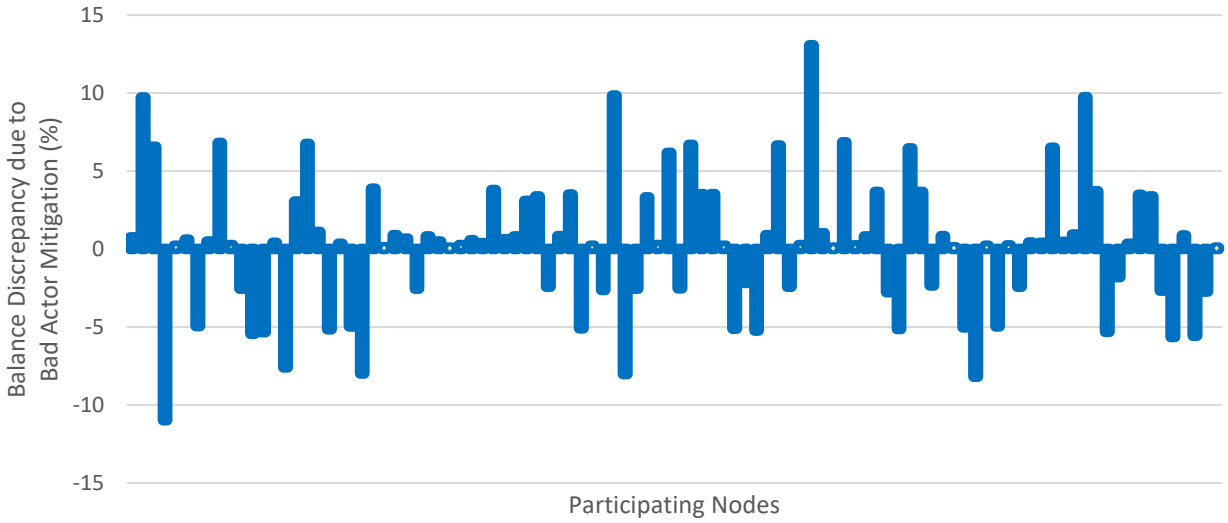

Fig. 10: Credit balance discrepancy due to malicious peers

The motivating use case of LEARNAE is the following: A community of individual researchers/enthusiasts share the same interest in training a specific neural model. They have no

access to expensive sophisticated infrastructure; all they have is a plain Internet connection, and either their commodity-grade computers, or IoT sensors that provide training data.

LEARNAE can provide a way for them to join forces and achieve an improved accuracy, despite potential connectivity issues, node offline times and bad actors. This collaborative process can take place in a strict timeframe or it can run continuously, fed by IoT data streams, ensuring that at any given time the participants will have the optimal update of the neural model.

It is of great importance to outline that, unlike almost all of the other proposals, our approach does not contain the notion of a "global model". During a collaborative training session, every node has its own local model and broadcasts its availability to the whole swarm. So, at any given time, a peer can either train its model using (local or remote) data, or average it with a model broadcasted by another node. The number of available models is increased over time, since the nodes broadcast not only their initial model, but also their updated (and improved, accuracy wise) versions. Every participant can examine the metadata that accompany each remote model and select which ones to try for averaging. One of the most fundamental challenges of our proposal is how to formulate these metadata in order to optimize this selection mechanism. So far, the criterion is the reported accuracy, thus the accuracy achieved by the model on the remote peer. Fine-tuning this feature could minimize the overall data transfer and improve the performance of the averaging process.

8 Future work

There are still issues with high research interest. Our experiments covered the scenario of bad actors who choose not to report the assistance they receive from others, but there are other ways of manipulating a distributed incentivization algorithm [60]. A usual attack vector is the attempt to distort the reputation system, which in this case translates into pushing LEARNAE's Shared Contribution Ledger to desired values. There are numerous ways for a system to deal with such attacks, including isolation of behaviors that do not conform with a globally accepted view, by using statistical analysis and machine learning. These concerns will be the subject of our future work.

9 Conflict of interest

The authors declare that they have no conflict of interest.

10 References

[1] S. Nikolaidis and I. Refanidis, “Learnae: Distributed and Resilient Deep Neural Network Training for Heterogeneous Peer to Peer Topologies,” in Engineering Applications of Neural Networks, Cham, 2019, pp. 286–298. doi: 10.1007/978-3-030-20257-6_24.

[2] S. Nikolaidis and I. Refanidis, “Privacy preserving distributed training of neural networks,” Neural Comput & Applic, vol. 32, no. 23, pp. 17333–17350, Dec. 2020, doi: 10.1007/s00521-020-04880-0.

[3] S. Nikolaidis and I. Refanidis, “Using distributed ledger technology to democratize neural network training,” Appl Intell, Mar. 2021, doi: 10.1007/s10489-021-02340-3.

[4] J. Benet, “IPFS - Content Addressed, Versioned, P2P File System,” arXiv:1407.3561 [cs], Jul. 2014, Accessed: Apr. 01, 2021. [Online]. Available: http://arxiv.org/abs/1407.3561

[5] S. Popov, O. Saa, and P. Finardi, “Equilibria in the Tangle,” Computers & Industrial Engineering, vol. 136, pp. 160–172, Oct. 2019, doi: 10.1016/j.cie.2019.07.025.

[6] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, “Improving deep neural network acoustic models using generalized maxout networks,” in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2014, pp. 215–219. doi: 10.1109/ICASSP.2014.6853589.

[7] Y. Miao, H. Zhang, and F. Metze, “Distributed learning of multilingual dnn feature extractors using gpus,” 2014.

[8] J. Dean et al., “Large Scale Distributed Deep Networks,” in Advances in Neural Information Processing Systems, 2012, vol. 25. Accessed: Aug. 11, 2021. [Online]. Available: https://papers.nips.cc/paper/2012/hash/6aca97005c68f1206823815f66102863-Abstract.html

[9] A. J. Mashtizadeh, A. Bittau, Y. F. Huang, and D. Mazières, “Replication, history, and grafting in the Ori file system,” in Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles, New York, NY, USA, Nov. 2013, pp. 151–166. doi: 10.1145/2517349.2522721.

[10] B. Cohen, “Incentives build robustness in BitTorrent,” Workshop on Economics of PeertoPeer systems, vol. 6, Jun. 2003.

[11] I. Baumgart and S. Mies, “S/Kademlia: A practicable approach towards secure key-based routing,” in 2007 International Conference on Parallel and Distributed Systems, Dec. 2007, pp. 1–8. doi: 10.1109/ICPADS.2007.4447808.

[12] M. J. Freedman, E. Freudenthal, and D. Mazières, "Democratizing content publication with coral," in Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation - Volume 1, USA, Mar. 2004, p. 18.

[13] L. Wang and J. Kangasharju, "Measuring Large-Scale Distributed Systems: Case of BitTorrent Mainline DHT," presented at the IEEE International Conference on Peer-to-Peer Computing, Sep. 2013. doi: 10.1109/P2P.2013.6688697.

[14] Nofer M, Gomber P, Hinz O, et al (2017) Blockchain. Bus Inf Syst Eng 59, 183–187. https://doi.org/10.1007/s12599-017-0467-3

[15] Pilkington M (2016) Blockchain technology: principles and applications. Research hand-book on digital transformations. Edward Elgar Publishing

[16] Mingxiao D, Ma X, Zhang Z, Wang X, and Chen Q (2017) A review on consensus algo-rithm of blockchain. IEEE international conference on systems, man, and cybernetics (SMC) 2567-2572

[17] Phillip A, Chan JS, Peiris S (2018) A new look at cryptocurrencies. Economics Letters 163 6-9

[18] Hildenbrandt E, Saxena M, Rodrigues N, Zhu X, Daian P, Guth D, Rosu G (2018) Kevm: A complete formal semantics of the ethereum virtual machine. IEEE 31st Computer Security Foundations Symposium (CSF) 204-217

[19] Luu L, Chu D H, Olickel H, Saxena P, Hobor A (2016) Making smart contracts smarter. Proceedings of the 2016 ACM SIGSAC conference on computer and communications security 254-269

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

[21] Dannen C (2017) Introducing Ethereum and solidity (Vol. 318). Berkeley: Apress

[22] Dean J, Corrado GS, Monga R, Chen K, Devin M, Le QV, Mao M, Razato M, Senior A, Tucker P, Yang K, Ng AY (2012) Large Scale Distributed Deep Networks. Advances in Neural Information Processing Systems 1223-1231

[23] Dekel O, Gilad-Bachrach R, Shamir 0, Xiao L (2012) Optimal distributed online prediction using mini-batches. Journal of Machine Learning Research 165-202

[24] Li M, Andersen DG, Park JW, Smola AJ, Ahmed A, Josifovski V, Long J, Shekita EJ, Su BY (2014) Scaling Distributed Machine Learning with the Parameter Server. 11th USENIX Symposium on Operating Systems Design and Implementation 583-598

[25] Iandola FN, Ashraf K, Moskewicz MW, Keutzer K (2015) FireCaffe: near-linear acceleration of deep neural network training on compute clusters. arXiv:1511.00175

[26] Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick RB, Guadarrama S, Darrell T (2014) Caffe: Convolutional Architecture for Fast Feature Embedding. ACM Intl. Conference on Multimedia 675-678

[27] Feng A, Shi J, Jain M (2016) CaffeOnSpark Open Sourced for Distributed Deep Learning on Big Data Clusters

[28] Wang Y, Zhang X, Wong I, Dai J, Zhang Y et al (2017) BigDL Programming Guide

[29] Dean J, Corrado GS, Monga R, Chen K, Devin M, Le QV, Mao M, Razato M, Senior A, Tucker P, Yang K, Ng AY (2012) Large Scale Distributed Deep Networks

[30] Chilimbi T, Suzue Y, Apacible Y, Kalyanaraman K (2014) Project Adam: Building an Efficient and Scalable Deep Learning Training System. 11th USENIX Symposium on Operating Systems Design and Implementation 571-582

[31] Zhang S, Choromanska A, LeCun Y (2015) Deep learning with Elastic Averaging SGD. Advances in Neural Information Processing Systems 685-693

[32] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, Kudlur M, Levenberg J, Monga R, Moore S, Murray DG, Steiner B, Tucker P, Vasudevan V, Warden P, Wicke M, Yu Y, Zheng X (2016) TensorFlow: A System for Large-Scale Machine Learning. 12th USENIX Symposium on Operating Systems Design and Implementation 265-283

[33] Moritz P, Nishihara R, Stoica I, Jordan MI (2016) SparkNet: Training Deep Networks in Spark. Intl. Conference on Learning Representations

[34] Lian X, Zhang C, Zhang H, Hsieh CJ, Zhang W, Liu J (2017) Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. Advances in Neural Information Processing Systems (NIPS)

[35] Blot M, Picard D, Cord M, Thome N (2016) Gossip training for deep learning. arXiv:1611.09726

[36] Boyd S, Ghosh A, Prabhakar B, Shah D (2006) Randomized Gossip Algorithms. IEEE Transactions on Information Theory 52:2508-2530

[37] Kim H, Park J, Jang J, Yoon S (2016) DeepSpark: Spark-Based Deep Learning Supporting Asynchronous Updates and Caffe Compatibility. arXiv:1602.08191

[38] Lian X, Zhang W, Zhang C, Liu J (2018) Asynchronous decentralized parallel stochastic gradient descent International Conference on Machine Learning (ICML)

[39] Peeters, R., & Schuilenburg, M. (2018). Machine justice: Governing security through the bureaucracy of algorithms. Information Polity, 23(3), 267-280. https://doi.org/10.3233/IP-180074

[40] Kitchin R. Big Data, new epistemologies and paradigm shifts. Big Data & Society. April 2014. doi:10.1177/2053951714528481

[41] Veale, Michael and Brass, Irina, Administration by Algorithm? Public Management Meets Public Sector Machine Learning (2019). In: Algorithmic Regulation (Karen Yeung and Martin Lodge eds., Oxford University Press, 2019), Available at SSRN: https://ssrn.com/abstract=3375391

[42] Veale, M; Brass, I; (2019) Administration by Algorithm? Public Management meets Public Sector Machine Learning. In: Yeung, K and Lodge, M, (eds.) Algorithmic Regulation. (pp. 121-149). Oxford University Press: Oxford, UK.

[43] Mintzberg, H. (1980). Structure in 5's: A Synthesis of the Research on Organization Design. Management Science, 26, 322-341.

[44] Zuurmond, A. (1994). De infocratie: Een theoretische en empirische heroriëntatie op Weber's ideaaltype in het informatietijdperk. Den Haag: Phaedrus.

[45] Lessig, Lawrence. (2006). Code: And Other Laws of Cyberspace, Version 2.0.

[46] Aneesh A. Global Labor: Algocratic Modes of Organization. Sociological Theory. 2009;27(4):347-370. doi:10.1111/j.1467-9558.2009.01352.x

[47] Cormen, T. H. (2013). Algorithms unlocked.

[48] Wirtz, J., Patterson, P.G., Kunz, W.H., Gruber, T., Lu, V.N., Paluch, S. and Martins, A. (2018), "Brave new world: service robots in the frontline", Journal of Service Management, Vol. 29 No. 5, pp. 907-931. https://doi.org/10.1108/JOSM-04-2018-0119

[49] Mohabbat Kar, R., Thapa, B. E. P., & Parycek, P. (Hrsg.). (2018). (Un)berechenbar? Algorithmen und Automatisierung in Staat und Gesellschaft. Berlin: Fraunhofer-Institut für Offene Kommunikationssysteme FOKUS, Kompetenzzentrum Öffentliche IT (ÖFIT). https://nbn-resolving.org/urn:nbn:de:0168-ssoar-57518-2

[50] Danaher J, Hogan MJ, Noone C, et al. Algorithmic governance: Developing a research agenda through the power of collective intelligence. Big Data & Society. December 2017. doi:10.1177/2053951717726554

[51] Prusty, Narayan (27 April 2017). Building Blockchain Projects. Birmingham, UK: Packt. p. 9. ISBN 9781787125339.

[52] The Decentralized Autonomous Organization and Governance Issues Regulation of Financial Institutions Journal: Social Science Research Network (SSRN). 5 December 2017.

[53] Vigna, P.; Casey, M. J. (27 January 2015). The Age of Cryptocurrency: How Bitcoin and the Blockchain Are Challenging the Global Economic Order. St. Martin's Press. ISBN 9781250065636.

[54] Hodson, H. (20 November 2013). "Bitcoin moves beyond mere money". New Scientist.

[55] "The DAO of accrue: A new, automated investment fund has attracted stacks of digital money". The Economist. 21 May 2016.

[56] Popper, N. (21 May 2016). "A Venture Fund with Plenty of Virtual Capital, but No Capitalist". New York Times.

[57] Pangburn, D. J. (19 June 2015). "The Humans Who Dream of Companies That Won't Need Us". FastCompany.

[58] Evans, J. (1 August 2015). "Vapor No More: Ethereum Has Launched". TechCrunch.

[59] Deegan, P. (2014). From Bitcoin to Burning Man and Beyond: The Quest for Identity and Autonomy in a Digital Society. Amherst, Massachusetts: Institute for Institutional Innovation. pp. 160–176.

[60] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. 2009. A survey of attack and defense techniques for reputation systems. ACM Comput. Surv. 42, 1, Article 1 (December 2009)